Discovering Model Structure of Dynamical Systems with Combinatorial Bayesian Optimization

Anonymous authors
Paper under double-blind review

Abstract

Deciding on a model structure is a fundamental problem in machine learning. We are interested in building a data-based model for the governing equations of a physical system from a library of discrete components. In addition to optimizing the model for performance, we consider crash and inequality constraints that arise from additional model requirements, such as real-time capability and model complexity regularization. We address this task of model structure selection with a focus on dynamical systems and propose to search over potential model structures efficiently using a constrained combinatorial Bayesian Optimization (BO) algorithm. We propose expressive surrogate models suited for combinatorial domains and a novel acquisition function that can handle both inequality and crash constraints and can be computed in closed form. We provide simulated benchmark problems within the domain of equation discovery of nonlinear dynamical systems. Our method outperforms the state-of-the-art in constrained combinatorial optimization of black-box functions and has a favorable computational overhead compared to other BO methods. As a real-world application example, we apply our method to optimize the configuration of an electric vehicle's digital twin while ensuring its real-time capability for the use in one of the world's largest driving simulators.

1 Introduction

The task of minimizing the discrepancy between the simulated behavior of digital twins and the measured behavior of the observed system is essential in engineering applications and is commonly known as system identification (Ljung, 1998) or model learning (Nguyen-Tuong & Peters, 2011). In general, system identification comprises two interleaved sub-tasks, structure identification and parameter estimation. The former aims to determine the structure of model equations, while the latter focuses on finding the model parameters (Tanevski et al., 2015). While a lot of the machine learning literature focuses on the second task, in this paper, we are interested in the first.

Structure identification methods work by exploring the space of potential models that provide the best-fitting representation of the dynamical system. On the one hand, structure identification can be approached by symbolic-regression methods (Bongard & Lipson, 2007; Schmidt & Lipson, 2009; Brunton et al., 2016), which explore the space of possible arithmetical expressions for building mathematical models and often use regularization techniques to introduce bias towards simpler models (Tanevski et al., 2020). These methods can be very general and require only the definition of the mathematical operations and functions that can compose the description of the system's behavior. On the other hand, knowledge-driven approaches (Bradley et al., 2001) require experts to encode domain-specific knowledge into model fragments, usually following known relationships, physical laws, constraints, or structural properties of the observed system. In both approaches, the idea is to break down the overall model into smaller, more manageable fragments or sub-models. This decomposition allows for a more modular representation of the system dynamics and enables the identification of local relationships or subsystem behaviors, which is especially important when dealing with complex systems. In addition, both methods provide flexibility in defining the structure and form of the model fragments, which can be combined, composed, or connected in a way that reflects the relationships and interactions between the different components of the system.

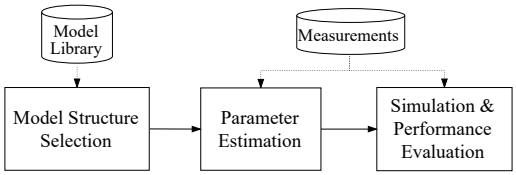


Figure 1: The process of system identification

In Figure 1, we depict an iterative process of system identification. In the structure identification phase, model components are selected from a model library of predefined parameterized templates that align with the system's characteristics. In the next step, regression algorithms are employed to estimate the coefficients or parameters associated with the selected templates of the model fragments. The final model is then evaluated using criteria such as goodness-of-fit, model complexity, or predictive performance. This evaluation helps determine the quality and suitability of the identified model and is used as feedback to search algorithms in order to select better model structures for the next iteration.

In practice, structure identification search algorithms enumerate plausible model structures and select the model templates that result in the best model evaluation (Tanevski et al., 2020). Due to the discrete nature of the problem, this task is often approached as a combinatorial optimization problem. Nevertheless, this is not a trivial task. By design, the interplay between components disallows selecting templates independently, and exhaustively searching over all potential model structures quickly becomes infeasible due to the combinatorial explosion of the search domain. In addition, evaluating a model structure candidate involves fitting coefficients and simulating dynamical systems, which can be computationally demanding and not necessarily differentiable. For this reason, a sample-efficient optimization method allows finding better models in larger search spaces. Bayesian Optimization (BO) methods (Garnett, 2023) are sample efficient and designed for expensive-to-evaluate black-box objective functions.

Moreover, the ability to handle inequality constraints and evaluation failures during the search for the best model structure plays an important role in system identification. Inequality constraints are important because they can be used to limit the computational budget available for the model, help mitigate overfitting, model over-completeness, and can prevent the choice of models that violate physical principles. In addition, the selection of certain templates might lead to an invalid composition and, in the worst case, lead to a model that is numerically unstable during simulation (Chakrabarty et al., 2021). Inconveniently, these failures or 'crashes' prohibit the assessment of the model performance (Bachoc et al., 2020).

For the above-mentioned reasons, we address the task of model structure selection as a combinatorial BO problem and propose a method especially designed for models that have additional requirements that can be translated into inequality and crash constraints in the optimization. Our method efficiently searches over potential model structures with a novel constrained combinatorial BO algorithm that handles these constraints and can efficiently scale up to many design parameters, allowing to find better models faster. Our optimization method employs expressive surrogate models suited for combinatorial spaces and implements an acquisition function that handles both inequality and crash constraints. While our contribution focuses on finding models for dynamical systems, the method we propose is general and can be used for a wide variety of machine learning tasks.

We empirically evaluate the method on established symbolic benchmark problems in the context of equation discovery of nonlinear dynamical systems (Brunton et al., 2016; Mangan et al., 2017). We extend these benchmarks to include inequality constraints in the form of Lasso regularization to combat overfitting, while tackling model numerical instabilities or failures that arise from choosing certain model structures in the form of a crash constraint. As a real-world application example, we further employ the method to optimize a knowledge-driven formulation of a multibody dynamical system of an electric vehicle, which operates in a driving simulator. Because the model simulates the dynamic response to the driver's inputs, it must match the timing and movement of a real car on the road and therefore must run in real time. Thus, besides optimizing the model performance, we consider the real-time capability as an inequality constraint in the model structure search problem.

Our contributions are as follows:

- We propose a general formulation for model structure selection as part of a constrained combinatorial optimization problem that considers inequality and crash constraints as a way to handle additional model requirements during the model search.
- We present a constrained combinatorial Bayesian optimization algorithm (`CBO-FRCHEI`) that can solve the model structure selection problem efficiently and is able to solve combinatorial optimization problems with up to 10^{18} combinations. `CBO-FRCHEI` employs a novel and efficient-to-compute acquisition function and combines recently proposed surrogate models for discrete domains.
- We show empirical evidence that the proposed algorithm outperforms state-of-the-art methods on benchmark problems for equation discovery and further provide a real-world application example where `CBO-FRCHEI` can build a digital twin for a driving simulator from a large library of modules, while ensuring its real-time capability.

This article continues as follows: in Section 2 we provide a general problem formulation for structure selection. The related work is discussed in Section 3. The proposed combinatorial BO method is described in Section 4. In Section 5, we investigate the performance of our method with various system identification problems and compare to other approaches.

The code for the optimizer and the benchmark problems are publicly available at https://github.com/

2 Problem Statement

The structure identification problem at hand requires four main elements. (i) First, we require noisy time measurements of the states and inputs that were collected during the operation of the real system. These are used to measure the model performance as well as to estimate the model parameters. (ii) Next, we require a library of model templates $\mathcal{X}$ and a symbolic-regression or knowledge-driven procedure for composing the selected templates in order to generate the resulting differential equation. The selection of model templates are to be parameterized with a vector of categorical decision variables denoted as $\boldsymbol{x} \in \mathcal{X}$. (iii) Further, we require a method for estimating the free parameters associated with the selected templates. (iv) Finally, we need an evaluation metric $f(\boldsymbol{x})$ for the model performance and optionally inequality $\boldsymbol{g}(\boldsymbol{x}) \leq 0$ and binary equality $h(\boldsymbol{x}) = 1$ constraint functions that can be used for additional model requirements. The inequality constraint can be used, for example, to restrict the computational budget, the model complexity or as a regularization, while the binary equality constraint is used to indicate evaluation failures.

We address this structure identification problem as a constrained combinatorial optimization problem. We introduce the objective function $f : \mathcal{X} \mapsto \mathbb{R}$ that maps the decision variables $\boldsymbol{x}$ defined over a combinatorial domain $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$ with d categorical decision variables with respective cardinalities $k_1, \ldots, k_d$. In practice, this function represents both parameter estimation and model evaluation depicted in Figure 1 and measures the performance of the model structure specified by $\boldsymbol{x} \in \mathcal{X}$. Similarly, we define the M inequality constraints as $g_j : \mathcal{X} \to \mathbb{R}$ with $j \in \{1, \ldots, M\}$, and the binary equality constraint function as $h : \mathcal{X} \to \{0, 1\}$.

We formulate the structure selection problem as the search for the global optimizer $\boldsymbol{x}^*$ that fulfills:

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \quad & f(\boldsymbol{x}) \\ s.t. \quad & g_j(\boldsymbol{x}) \leq 0 \quad \forall j \in \{1, \ldots, M\} \\ & h(\boldsymbol{x}) = 1\,. \end{aligned} \tag{1}$$

The functions f, $\boldsymbol{g}$, and h are all expensive-to-evaluate black-box functions and can only be obtained simultaneously. The function f and $\boldsymbol{g}$ are noisy and can only be assessed when the experiment is successful, i.e.

$$(y, \boldsymbol{c}, l) = \begin{cases} (f(\boldsymbol{x}) + \epsilon_y, & \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\epsilon}_c, & 1) & \text{if } \boldsymbol{x} \text{ is evaluation success} \\ (\varnothing, & \varnothing & 0) & \text{if } \boldsymbol{x} \text{ is evaluation failure} \end{cases} \tag{2}$$

where the noise $\epsilon_y \sim \mathcal{N}(0, \sigma_y^2)$ and $\boldsymbol{\epsilon}_c \sim \mathcal{N}(0, \mathrm{diag}(\boldsymbol{\sigma}_c^2))$ are i.i.d. and normally distributed. All past observations are collected in a dataset $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{c}^{(i)}, l^{(i)})\}_{i \in [N]}$, where N is the number of experiments.

3 Related Work

Several approaches have been proposed to address the challenge of model structure selection within the field of system identification, which has usually been tackled as a combinatorial optimization problem, with approaches ranging from exhaustive search methods to elaborated heuristic algorithms. In this section, we review some of the existing literature and highlight the potential of employing combinatorial Bayesian optimization as a promising approach.

Model Structure Selection. Traditional methods for structure selection involve exhaustive search over the space of potential model structures, such as random search, A* search, and simulated annealing (Bertsimas & Tsitsiklis, 1993) algorithms. Others employed evolutionary algorithms (Schmidt & Lipson, 2009; Tanevski et al., 2020) or causal inference (Baumann et al., 2022). However, the main limitation of these methods is their sample efficiency, as the number of potential model structures grows exponentially with the number of discrete variables. Generally, these algorithms are not suitable for expensive-to-evaluate system identification problems.

Mangan et al. (2017) proposed a sparse symbolic-regression structure selection approach using SINDy (Brunton et al., 2016) as a tool to down-sample parsimonious models from the combinatorially large model space and further used information criteria to rank the remaining models. Reducing the search space allowed them to brute-force and search for the best model among the remaining ones. In essence, SINDy works by recursively performing linear regression and pruning terms with small model coefficients up to a certain threshold. Although very efficient, this method, as well as other backward-elimination algorithms, over-exploits the model space and does not guarantee to find the optimal set of features (Guyon & Elisseeff, 2003). In addition, SINDy cannot handle constraint problems and can only be applied to symbolic-regression and therefore is not applicable to knowledge-driven problems. Another limitation is that the coefficients are pruned based on the regression results obtained by fitting a prediction model, which is not always the best strategy if the model is to be used as a simulation or auto-regressive model. BO methods are more general and allow for arbitrary cost functions, whereas SINDy decides on the structure based on the least squares cost.

Combinatorial Bayesian Optimization. Recently, progress has been made in the field of combinatorial Bayesian optimization (Luong et al., 2019). The two main challenges in this field are the development of (i) better probabilistic regression approaches that can capture the complex interaction between discrete variables in the combinatorial domain and (ii) acquisition function optimizers that efficiently search the combinatorial space using the surrogates. While many of the methods have been defined over a mixed space of continuous and discrete variables, we focus here on the specific characteristics that are useful for discrete domains. `SMAC` (Hutter et al., 2011) leveraged tree-based surrogate models and used random walks to obtain a local optimum of acquisition function. This method can be applied to mixed input spaces but neglect high-order interaction between variables. `BOCS` (Baptista & Poloczek, 2018) encoded categorical variables in a combinatorial one-hot binary domain and used polynomial features within sparse Bayesian linear regression (Carvalho et al., 2010; Makalic & Schmidt, 2015) combined with Thompson sampling to express the interaction between variables and their effect on the objective function. They optimized the acquisition function using simulated annealing and semi-definite programming. Dadkhahi et al. (2022) extended `BOCS` and proposed a more compact but still complete and unique encoding that results in fewer monomials. Hase et al. (2018) and Häse et al. (2021) combined Bayesian neural networks and density kernel estimation as a surrogate for BO with categorical variables. Wan et al. (2021) proposed the `Cosmopolitan` algorithm and used a modified Hamming kernel as part of Gaussian process regression, which is tailored for combinatorial spaces and defines the correlation between two inputs by their Hamming distance in the combinatorial graph. To avoid over-exploration due to high-dimensional combinatorial spaces they adapted the trust region algorithm from `TURBO` (Eriksson et al., 2019) to explore only locally near the best location found so far. Further, Oh et al. (2019) proposed `COMBO`, which used the discrete diffusion kernel built from the graph cartesian product of discrete parameters and was able to model high-order interactions between variables leading to better performance. `COMBO` was improved by Deshwal et al. (2021) (`HyBO`) by using the closed-form of the discrete diffusion kernel proposed in Imre (2002).

Constrained Bayesian optimization is an active area of research for continuous domains (Gardner et al., 2014; Eriksson & Poloczek, 2021; Ungredda & Branke, 2021; Marco et al., 2021) but rarely investigated for

discrete variables. Daulton et al. (2022) proposed Probabilistic Reparameterization (`PR`), the first method for constrained combinatorial BO, which reparameterized the discrete acquisition function optimization problem by introducing discrete probability distributions defined by continuous parameters. This allowed them to optimize the AF using gradient-based methods. We also note that other methods, such as Papalexopoulos et al. (2022), exist for constrained combinatorial optimization but only consider inexpensive and white-box constraint functions, which is not applicable to our problem.

As pointed out by Daulton et al. (2022), there are many computational issues with most of the existing methods that make it difficult to apply them to constrained optimization problems. Our proposed method combines ideas from the literature and extends to constrained problems. In addition, most of the existing methods scale poorly with respect to the number of inputs and the number of data points, since they rely on expensive MCMC methods to infer the posterior distributions, making them impracticable for high dimensional discrete problems that arise from structure model selection. Therefore, we focus on reducing the high computational overhead of the state-of-the-art methods by relying on closed-form models and AF, avoiding expensive approximations.

4 Methods

In the context of structure model selection, Bayesian optimization can be employed to find the optimal model structure that minimizes a chosen evaluation metric. Bayesian optimization is particularly useful when the objective function is expensive-to-evaluate and has no explicit expression. Based on past observations, the BO algorithm constructs a probabilistic surrogate model of the objective and the constraints, typically a Gaussian Process ($\mathcal{GP}$) (Rasmussen et al., 2006). The surrogate model is iteratively updated by incorporating new evaluations, allowing for the sequential exploration of the search space. The selection of the next promising evaluation point is guided by an acquisition function, which bases on the surrogate model's predictions and the uncertainty associated with them to balance the exploration-exploitation trade-off.

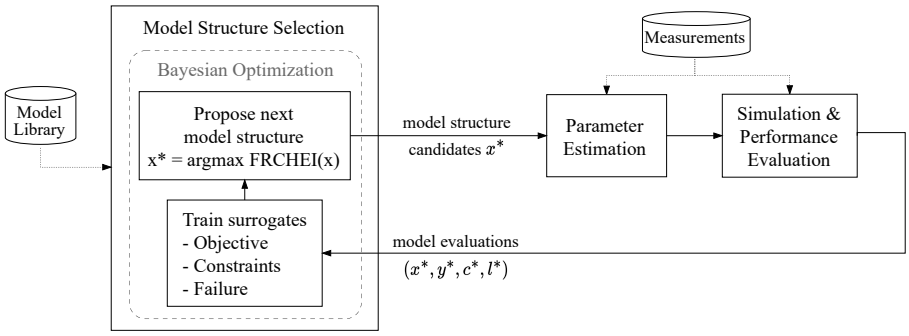


Figure 2: The proposed method for model structure selection using combinatorial Bayesian optimization. Every iteration, the model structure selection algorithm proposes a new model structure $\boldsymbol{x}$ from the model library. Using measurements, the model's parameters are estimated and the composed model performance is evaluated according to objective $y = f(\boldsymbol{x}) + \epsilon_y$, inequality constraints $\boldsymbol{c} = \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\epsilon}_c$ and crash $l = h(\boldsymbol{x})$ function evaluations.

We call our method Combinatorial Bayesian Optimization with Failure-Robust Constrained Hierarchical Expected Improvement (`CBO-FRCHEI`). In the following subsections, we present a detailed description of this method. We define (i) probabilistic surrogate models for regression, which will approximate the black-box objective and constraint functions, (ii) a probabilistic model for classification, which will approximate the black-box binary function for failures and finally (iii) the acquisition function. An overview of the method is depicted in Figure 2, and the algorithm is given in Algorithm 2.

4.1 Probabilistic Surrogate Model for Regression

We consider the setting, where objective function observations are given by $y^{(i)} = f(\boldsymbol{x}^{(i)}) + \epsilon_y^{(i)}$ and we assume the observation noise $\epsilon_y^{(i)} \sim \mathcal{N}(0, \sigma_y^2)$ to be an i.i.d. Gaussian random variable with unknown variance σ_y^2 (cf. Equation 2). After making N observations, the joint distribution of observations is

$$\boldsymbol{y} \mid X, \sigma_y^2 \sim \mathcal{N}(\boldsymbol{f}(X),\ \sigma_y^2 I_N)\,, \tag{3}$$

where $X = [\boldsymbol{x}^{(i)}, \ldots, \boldsymbol{x}^{(N)}]^\top \in \mathcal{X}^N$, $\boldsymbol{f}(X) = [f(\boldsymbol{x}^{(i)}), \ldots, f(\boldsymbol{x}^{(N)})]^\top \in \mathbb{R}^N$, $\boldsymbol{y} = [y^{(i)}, \ldots, y^{(N)}] \in \mathbb{R}^N$ and $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix. We assume the objective and constraints to be conditionally independent given X and use the same setup for each j-th dimension of constraint function evaluations $c_j^{(i)} = g_j(\boldsymbol{x}^{(i)}) + \epsilon_{c_j}^{(i)}$. Similarly, $\boldsymbol{c}_j \mid X, \sigma_c^2 \sim \mathcal{N}(g_j(X),\ \sigma_c^2 I_N)$.

We now present a surrogate model for the expensive function f as a Gaussian process prior over f

$$f \mid \sigma^2 \sim \mathcal{GP}\left(m, k\right)\,, \tag{4}$$

where the mean function $m : \mathcal{X} \to \mathbb{R}$ and the kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are the crucial ingredients in $\mathcal{GP}$ regression since they define the kind of structure that will be captured by the regression model. These components are to be defined in the following subsections.

The posterior predictive distribution of a function value $f(X')$ at test points X' is given analytically (Rasmussen et al., 2006)

$$\begin{aligned} &\boldsymbol{f}(X') \mid X', \boldsymbol{y}, X, \sigma^2 \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_f, \bar{\Sigma}_f) \\ &\text{with} \qquad \begin{aligned} \bar{\boldsymbol{\mu}}_f &= \boldsymbol{m}(X') + K(X', X)(K(X, X) + I_N \sigma_y^2)^{-1}(\boldsymbol{y} - \boldsymbol{m}(X)) \\ \bar{\Sigma}_f &= K(X', X') - K(X', X)(K(X, X) + I_N \sigma_y^2)^{-1} K(X, X') \end{aligned} \end{aligned} \tag{5}$$

where $\boldsymbol{m}(X)_i = m(\boldsymbol{x}_i)$ and $K(X, X')_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}'_j)$.

One of the limitations of Gaussian process regression is the light tails of the predictive distribution, which are not robust to outliers (Duvenaud, 2014) and might not capture well enough the discrepancy of the surrogate model (Baptista & Poloczek, 2018). To allow for better robustness against outliers and characterization of the uncertainty in the prediction while still allowing for a closed-form posterior, we make use of hierarchical priors. A hierarchical Bayesian model is a model in which the prior distribution of some of the model parameters depends on other parameters, which are also assigned a prior (Elster et al., 2015). We place the following hierarchical $\mathcal{GP}$ prior over f

$$f \mid \sigma^2 \sim \mathcal{GP}\left(m, \sigma^2\, k\right) \tag{6}$$

$$\sigma^2 \sim \Gamma^{-1}(\nu, \sigma_m^2)\,, \tag{7}$$

which leads to $p(f(X), \sigma^2)$ being an inverse-gamma-normal distribution. To ease the derivations, we use a more intuitive inverse-gamma parameterization, whose shape is $a = \nu/2$ and scale $b = \sigma_m^2\, \nu/2$ (Taboga, 2017). This parameterization directly reflects the mean and the variance of this distribution since $\mathbb{E}[1/\sigma^2] = 1/\sigma_m^2$ and $Var[1/\sigma^2] = 2/(\nu\sigma_m^4)$. Therefore σ_m^2 can be interpreted as the best guess of the regression precision, while ν expresses the degree of confidence about the precision. We follow Chen et al. (2023) and place hyper-prior distributions over $\nu \sim \Gamma(\zeta_\nu, \xi_\nu)$ and $\sigma_m^2 \sim \Gamma(\zeta_{\sigma_m^2}, \xi_{\sigma_m^2})$.

The predictive prior distribution at X can be calculated analytically by

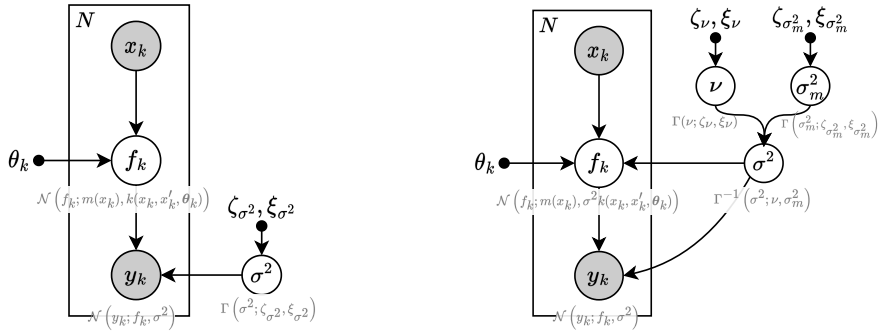
$$\boldsymbol{f}(X) \mid X \sim \int p(\boldsymbol{f}(X) \mid \sigma^2, X)\ p(\sigma^2)\ d\sigma^2 = T(\nu, \boldsymbol{m}(X), \sigma_m^2\, K(X, X))\,, \tag{8}$$

where $T(\nu, \boldsymbol{\mu}, \Sigma)$ is the multivariate t-distribution, with degrees of freedom ν, mean $\boldsymbol{\mu}$ and scale matrix Σ. Furthermore, the predictive posterior distribution at test points X' is also given in closed-form by

$$\boldsymbol{f}(X') \mid X', \boldsymbol{y}, X \sim \int p(\boldsymbol{f}(X') \mid X', \boldsymbol{y}, X, \sigma^2)\, p(\sigma^2 \mid \boldsymbol{y}, X)\, d\sigma^2 = T(\bar{\nu}_f, \bar{\boldsymbol{\mu}}_f, \bar{\Sigma}_f) \tag{9}$$

$$\begin{aligned}
\text{with} \quad & \bar{\nu}_f = \nu + N \\
& \bar{\boldsymbol{\mu}}_f = \boldsymbol{m}(X') + K(X', X') \left(K(X, X) + I\right)^{-1} \left(\boldsymbol{y} - \boldsymbol{m}(X)\right) \\
& \bar{\Sigma}_f = \bar{\sigma}_m^2 \Big(K(X', X') - K(X', X) \left(K(X, X) + I\right)^{-1} K(X, X') \Big) \\
& \bar{\sigma}_m^2 = \Big(\nu\, \sigma_m^2 + \left(\boldsymbol{y} - \boldsymbol{m}(X)\right)^\top \left(K(X, X) + I\right)^{-1} \left(\boldsymbol{y} - \boldsymbol{m}(X)\right) \Big) \,/ \big(\nu + N\big)\,,
\end{aligned}$$

which defines the so-called t-Process ($\mathcal{TP}$) regression framework (Shah et al., 2013; Tracey & Wolpert, 2018). Interestingly, one can show that as N goes to infinity, the $\mathcal{TP}$ prior and posterior tends to the $\mathcal{GP}$ distributions. Moreover, the equations above show that the parameters ν, σ_m^2 defined in the inverse-Gamma distribution of the prediction uncertainty σ^2 directly impose how heavy-tailed the predictive distribution will be. The probabilistic graph models for both $\mathcal{GP}$ and $\mathcal{TP}$ are provided in Figure 3.



(a) Gaussian Process Regression

(b) T Process Regression

Figure 3: Graph model of two probabilistic regression models. The array $\boldsymbol{\theta}_k$ refers to the kernel function k hyperparameters.

4.1.1 Kernel Design

Kernels play a crucial role in $\mathcal{GP}$ and $\mathcal{TP}$ regression because they define the assumptions and properties of the underlying probabilistic model. The kernel defines correlations between candidates and is required for accurate and meaningful predictions. Notably, common kernels found in the literature for continuous spaces rely on a natural order of candidates, which is not given in categorical spaces. To address this limitation, many kernels specialized for categorical spaces have been recently proposed in the literature (Imre, 2002; Wan et al., 2021; Deshwal et al., 2021; Oh et al., 2019; Baptista & Poloczek, 2018; Dadkhahi et al., 2022).

In particular, the automatic relevance determination (ARD) discrete diffusion kernel became recently popular and is the discrete analog to the diffusion kernel over continuous spaces, aka the radial basis function (RBF) (Imre, 2002; Deshwal et al., 2021). This kernel defines diffusion over the entire discrete space represented by a combinatorial graph, where two nodes are connected if two configurations differ in exactly one variable. It is given by

$$k_{\text{diff}}(\boldsymbol{x}, \boldsymbol{x}') = \prod_{i=1}^{d} \left(\frac{1 - e^{-k_i \beta_i}}{1 + (k_i - 1) e^{-k_i \beta_i}} \right)^{\delta(\boldsymbol{x}_i, \boldsymbol{x}_i')}, \tag{10}$$

where k_i is the cardinality of the i-th categorical input variable, $\delta(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is equal zero if $\boldsymbol{x}_i$ is equal $\boldsymbol{x}_j$ and equal one otherwise, and β_i are hyperparameters that control the importance of the i-th discrete variable. Intuitively, this kernel is well suited for globally capturing interactions between configurations. However,

one of the main limitations is that it captures similarity based on exact matches and does not consider the degree of dissimilarity between categories, which may lead to a loss of information when trying to model relationships or interactions between categorical variables.

Alternatively, Baptista & Poloczek (2018) proposed a polynomial kernel that was used in a Bayesian linear regression scheme. They observed that a universal surrogate model for a binary discrete input domain $\bar{\mathcal{X}} = \{0,1\}^{\bar{d}}$ is given by $f(\boldsymbol{x}) = \sum_{\mathcal{S}\in 2^{\bar{\mathcal{X}}}} \boldsymbol{\alpha}_{\mathcal{S}} \prod_{i\in\mathcal{S}} \bar{\boldsymbol{x}}_{\boldsymbol{i}}$, where $2^{\bar{\mathcal{X}}}$ is the power set of the domain, $\boldsymbol{\alpha}_{\mathcal{S}}$ is the coefficient vector, and $\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}$ is the one-hot encoding of the categorical input $\boldsymbol{x}$. In fact, this model describes all the possible configurations and becomes quickly impractical due to the exponential number of monomials. The idea presented is to truncate monomials up to a certain degree

$$f(\boldsymbol{x}) = \boldsymbol{\alpha}_0 + \sum_j \boldsymbol{\alpha}_j \bar{\boldsymbol{x}}_{\boldsymbol{j}} + \sum_{i,j>i} \boldsymbol{\alpha}_{ij} \bar{\boldsymbol{x}}_{\boldsymbol{i}} \bar{\boldsymbol{x}}_{\boldsymbol{j}} + \sum_{i,j>i,k>j} \boldsymbol{\alpha}_{ijk} \bar{\boldsymbol{x}}_{\boldsymbol{i}} \bar{\boldsymbol{x}}_{\boldsymbol{j}} \bar{\boldsymbol{x}}_{\boldsymbol{k}} + \dots , \tag{11}$$

which can be naturally modeled as a kernel

$$k_{poly}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_\alpha^2 \left(1 + \sum_i^{\bar{d}} \bar{\boldsymbol{x}}_{\boldsymbol{i}}\, \bar{\boldsymbol{x}}'_{\boldsymbol{i}} + \sum_{i,j>i}^{\bar{d}} \bar{\boldsymbol{x}}_{\boldsymbol{i}} \bar{\boldsymbol{x}}_{\boldsymbol{j}}\, \bar{\boldsymbol{x}}'_{\boldsymbol{i}} \bar{\boldsymbol{x}}'_{\boldsymbol{j}} + \sum_{i,j>i,k>j}^{\bar{d}} \bar{\boldsymbol{x}}_{\boldsymbol{i}} \bar{\boldsymbol{x}}_{\boldsymbol{j}} \bar{\boldsymbol{x}}_{\boldsymbol{k}}\, \bar{\boldsymbol{x}}'_{\boldsymbol{i}} \bar{\boldsymbol{x}}'_{\boldsymbol{j}} \bar{\boldsymbol{x}}'_{\boldsymbol{k}} + \cdots \right) , \tag{12}$$

where σ_α^2 is the hyperparameter related to the prior variance of the coefficients. Notably, this kernel is more complex than the discrete diffusion kernel, especially when considering higher polynomial degrees, since it can capture higher-order interactions between categories. However, as the degree increases, the number of parameters becomes prohibitively high. By design, it also suffers from the combinatorial explosion of the discrete domain. Note also that this kernel distinguishes not only the degree of dissimilarity between different categories but the interaction between them.

In this paper, we propose to combine the strength of multiple kernels (Duvenaud, 2014) in the following way

$$k_{\text{polydiff}}(\boldsymbol{x}, \boldsymbol{x}') = \lambda \left(k_{\text{poly}}(\boldsymbol{x}, \boldsymbol{x}') \cdot k_{\text{diff}}(\boldsymbol{x}, \boldsymbol{x}') \right) + (1-\lambda) \left(k_{\text{poly}}(\boldsymbol{x}, \boldsymbol{x}') + k_{\text{diff}}(\boldsymbol{x}, \boldsymbol{x}') \right) \tag{13}$$

where $\lambda \sim \text{Beta}(\alpha_\lambda, \beta_\lambda) \in [0,1]$ is a hyper-parameter that controls whether the kernels should be added or multiplied together. The idea is to combine the global approximation properties of the discrete diffusion kernel with the high-order interaction and non-stationary properties of polynomial kernels. In the appendix Section A.4, we provide an ablation study of different kernels applied to our benchmark problems.

4.1.2 Hyperprior-parameter estimation

To optimize the hyper-parameters, we resort to empirical Bayes, as suggested by Chen et al. (2023), to avoid expensive MCMC samplings. This is possible because both $\mathcal{GP}$ and $\mathcal{TP}$ regression methods have tractable marginal likelihoods. We achieve this by maximizing the marginal a-posteriori (MMAP) for the prior hyperparameters

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \; p(\boldsymbol{y} \,|\, X, \boldsymbol{\theta}) \prod_i p(\boldsymbol{\theta}_i) \,, \tag{14}$$

where the marginal likelihood $p(\boldsymbol{y}|X, \boldsymbol{\theta}) = T(\nu, \boldsymbol{m}(X), \sigma_m^2 K(X,X) + \sigma_m^2 I_N)$ and the hyperprior distribution $p(\boldsymbol{\theta}_i)$ is assumed conditionally independent and is specifically designed for each hyperparameter $\boldsymbol{\theta}_i$. Hyper-parameters without defined hyper-priors are treated as non-informative and are set to the Jeffrey's prior $p(\boldsymbol{\theta}_i) \propto 1$. Note that we train all model hyperparameters in the model, which for the example of the $\mathcal{TP}$ model with k_{polydiff} kernel consists of $\boldsymbol{\theta} = \{\nu, \sigma_m^2, \sigma_\alpha, \boldsymbol{\beta}, \lambda\}$.

4.2 Probabilistic Surrogate Model for Classification

When searching for possible structures of dynamical systems, it is to be expected that simulating the dynamical system might fail due to numerical instabilities. Unfortunately, it is usually not possible to delimit the regions that lead to instabilities a-priori. We model the set of feasible candidates as black-box functions and learn the

failure regions from data using a probabilistic $\mathcal{GP}$ classifier. In this way, the Bayesian optimization algorithm can learn to avoid these regions while searching for the optimum. We use respectively the following prior and likelihood functions

$$h \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{15}$$

$$\boldsymbol{l} \mid \boldsymbol{h}(X) \sim \prod_i \text{Bernoulli}(\boldsymbol{l}_i;\ S(h(X_i)))\,, \tag{16}$$

where S is the sigmoid function, and $\boldsymbol{l}$ are previous failure observations.

We approximate the intractable Bernoulli posterior distribution $p(\boldsymbol{h}(X^*) \mid X^*, \boldsymbol{l}, X)$ with Laplacian approximation, which is simple to implement and can approximate the posterior distribution in closed-form. We use the same kernel as in Equation 13. We implement the alternating optimization procedure from Rasmussen et al. (2006) for finding the posterior mode of the Laplacian approximation (Newton-Raphson combined with Strong-Wolfe linear search) and the model hyper-parameters (L-BFGS).

4.3 Acquisition Function

The goal of the acquisition function (AF) is to guide the search for the optimal solution in an optimization problem addressing the tension between exploitation and exploration. For the constrained problem, the AF needs to balance the potential utility of a candidate and the probability of feasibility and success. The next candidate model $\boldsymbol{x}^{(t)}$ to be evaluated at iteration t is the one that maximizes the acquisition function α for the current model:

$$\boldsymbol{x}^{(t)} = \arg\max_{\boldsymbol{x}' \in \mathcal{X}}\ \alpha(\boldsymbol{x}')\,. \tag{17}$$

The AF needs to be optimized in every BO iteration. Generally, this optimization problem is the main computational bottleneck in BO which is why we look for an acquisition function that is cheap to evaluate and given in closed-form. In this work, we combine recent ideas in the field of Bayesian optimization and propose a new acquisition function called Failure-Robust Constrained Hierarchical Expected Improvement (`FRCHEI`), defined as follows

$$\alpha(\boldsymbol{x}') = \text{ FRCHEI}(\boldsymbol{x}') = \ P_{\text{succ}}(\boldsymbol{x}')^{\beta_{succ}\ n/N} \cdot P_{\text{feas}}(\boldsymbol{x}')^{\beta_{feas}\ n/N} \cdot \text{HEI}(\boldsymbol{x}')\,, \tag{18}$$

where

$$P_{\text{succ}}(\boldsymbol{x}') = p(h(\boldsymbol{x}') = 1 \mid \boldsymbol{x}', \boldsymbol{l}, X) \tag{19}$$

$$P_{\text{feas}}(\boldsymbol{x}') = \prod_j^m\ p(g_j(\boldsymbol{x}') \leq 0 \mid \boldsymbol{x}', \boldsymbol{c}_j, X) \tag{20}$$

$$\text{HEI}(\boldsymbol{x}') = \mathbb{E}_{f(\boldsymbol{x}') \sim p(f(\boldsymbol{x}') \mid \boldsymbol{x}', \boldsymbol{y}, X)}\left[\max\{0, y^+ - f(\boldsymbol{x}')\}\right]\,. \tag{21}$$

The idea of this AF is twofold. First, we consider the hierarchical expected improvement (`HEI`) (similar to Chen et al. (2023)). The `HEI` uses the objective function probabilistic surrogate $p(f(\boldsymbol{x}') | \boldsymbol{x}', \boldsymbol{y}, X)$ to quantify the potential improvement in the objective value over the current best feasible solution found so far $y^+ \in \boldsymbol{y}$. Samples with lower predicted mean values and higher uncertainties are more likely to be explored as they offer the potential for a better solution. `HEI` modifies the traditional `EI` by replacing the $\mathcal{GP}$ surrogate by $\mathcal{TP}$ regression, which should improve the predictive performance while preserving the closed-form solution of the AF, which is given by

$$\text{HEI}(\boldsymbol{x}') = \bar{\sigma}_f\big[\underbrace{\tau^+ \Phi_T\left(\tau^+; \bar{\nu}_f, 0, 1\right)}_{\text{exploitation}} + \underbrace{\frac{\bar{\nu}_f + (\tau^+)^2}{\bar{\nu}_f - 1} T(\tau^+; \bar{\nu}_f, 0, 1)}_{\text{exploration}}\big]\,, \tag{22}$$

where $\tau^+ = (y^+ - \bar{\mu}_f)/\bar{\sigma}_f$, and ϕ_T is the CDF and T is the PDF of the t-distribution, respectively. The parameters $\bar{\nu}_f, \bar{\mu}_f, \bar{\sigma}_f^2$ are from the posterior t-distribution at $\boldsymbol{x}'$ and given as in Equation 9. Note also that as

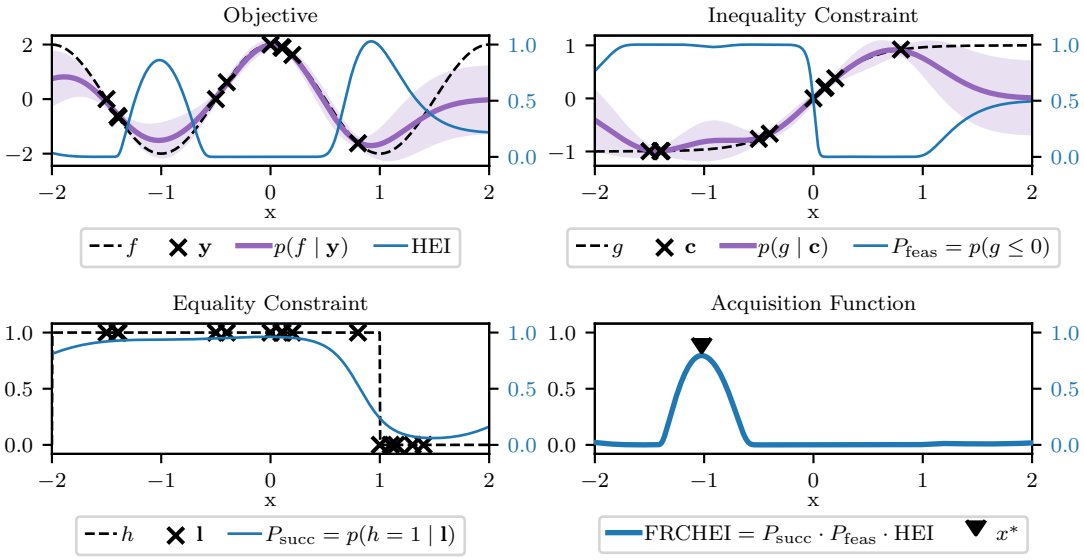


Figure 4: Illustration of the `FRCHEI` acquisition function for a one-dimensional continuous space example. The true unknown functions are depicted as dashed lines, the observations as crosses and the surrogate predictive posteriors mean and confidence region as purple lines and areas, respectively. Note that no observations could be assessed for the objective and constraint functions in failure regions $\{x : h(x) = 0\}$. The next candidate point x^* is obtained by searching for the maximum of the `FRCHEI` function. Clearly, the surrogate for the constraint functions g and h push down regions of the expected improvement `HEI` which are likely to be unfeasible or fail. Without these two surrogates, the next candidate would be sampled on an unfeasible and failure-prone region, where the `HEI` is maximized. Note that the search space for the experiments we consider is discrete and high-dimensional, which can not be well displayed. For illustration purposes, we employed the same method but replaced the combinatorial kernels with a squared exponential.

$\bar{\nu}_f \to \infty$, the `EI` of the t-distributed posterior converges to the `EI` of a normal distributed posterior, which is expected since t-distributions converge to Gaussian distributions as the degrees of freedom parameter $\nu \to \infty$. The derivation of Equation 22 is given in the appendix Section A.1.

Second, we avoid unfeasible regions by multiplying `HEI` with the probability of feasibility P_{feas}, similar to Gardner et al. (2014) and avoid instability regions by multiplying the `HEI` with the probability of success P_{succ}, as proposed in Chakrabarty et al. (2021). Following Hvarfner et al. (2022), we scale these probabilities with $\beta_{feas}\ n/N$ and $\beta_{succ}\ n/N$, where n is the current iteration number, N is the maximum number of iterations and β_{feas}, β_{succ} are hyper-parameters. The idea is to relax the constraints at the beginning of the optimization when we do not have much knowledge about these black-box functions. As the optimization progresses, we increasingly trust these surrogate models. Note that even though unfeasible models are never considered the best experiment, they are still useful to improve the predictive posterior distributions (Gardner et al., 2014). An illustration of this acquisition function for a one-dimensional continuous problem example can be seen in Figure 4.

4.3.1 Acquisition Function Optimization

Since the input space is fully discrete, optimization of the AF can not be done directly using gradient-based methods. Instead, we use simulated annealing (Bertsimas & Tsitsiklis, 1993), which is a standard and performant method for unconstrained optimization over discrete spaces. We use the same simulated annealing approach presented in Dadkhahi et al. (2022), since it is simple, computationally efficient, and can optimize over categorical variables. For completeness, we present the algorithm again in Algorithm 1.

Algorithm 1 Simulated annealing for categorical variables

function `SA`(objective function f, categorical domain $\mathcal{X}$, starting point $\boldsymbol{x}^{(0)}$, annealing scheduler $s(t)$)
  for $t = 1$ to N **do**
    $i \sim$ `unif(d)`
    $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)}$
    $\boldsymbol{x}_i^{(t)} \sim \texttt{Softmax}\left(\{-f(\boldsymbol{x}_i = w, \boldsymbol{x}_{-i})/s(t)\}_{w \in \mathcal{X}_i}\right)$
  return $\boldsymbol{x}^{(t)}$

4.4 Algorithm

We present our final algorithm for black-box constrained combinatorial optimization in Algorithm 2. The algorithm requires an initial dataset $\mathcal{D}^{(1)} = \{(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{c}^{(i)}, l^{(i)})\}_{i=1}^{N}$, which can be obtained by evaluating the model structure at N randomly sampled initial points $x^{(i)}$. It has been shown in recent studies (Wan et al., 2021; Deshwal et al., 2021; Müller et al., 2021; Daulton et al., 2022) that local exploitation is beneficial, especially for high-dimensional discrete spaces. When maximizing the AF, in line 8, we make use of this idea and set the initial starting point for the `SA` optimization method as the best feasible sample evaluated so far.

Algorithm 2 Bayesian Optimization for Model Structure Selection

1: **function** `CBO-FHCHEI`(initial dataset $\mathcal{D}^{(1)}$)
2: **for** $t = 1$ to N **do**
3: **Train surrogates:**
4: $\boldsymbol{\theta}_f \leftarrow \text{MMAP}(\mathcal{D}^{(t)})$ (Equation 14)
5: $\boldsymbol{\theta}_g \leftarrow \text{MMAP}(\mathcal{D}^{(t)})$ (Equation 14)
6: $\boldsymbol{\theta}_l \leftarrow \text{LaplaceMode-MMAP}(\mathcal{D}^{(t)})$ (Section 4.2)
7: **Optimize acquisition function:**
8: $\boldsymbol{x} \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} \text{FRCHEI}(\boldsymbol{x}, \boldsymbol{\theta}_f, \boldsymbol{\theta}_g, \boldsymbol{\theta}_l, \mathcal{D}^{(t)})$ (Algorithm 1 with Equation 18)
9: $y, \boldsymbol{c}, l \leftarrow$ EvaluateModelStructure($\boldsymbol{x}$)
10: Update data $\mathcal{D}^{(t+1)} = \mathcal{D}^{(t)} \cup \{(\boldsymbol{x}, y, \boldsymbol{c}, l)\}$
11: **return** lowest feasible evaluation $\boldsymbol{x}^* = \arg\min_x \{y : (\boldsymbol{x}, y, \boldsymbol{c}, l) \in \mathcal{D}, \boldsymbol{c}_i(\boldsymbol{x}) \leq 0, l(\boldsymbol{x}) = 1\}$
12:
13: **function** EvaluateModelStructure($\boldsymbol{x}$)
14: Estimate model parameters, simulate and evaluate performance $(f, \boldsymbol{g}, h)$.
15: **if** Failure **then**
16: **return** $\{\varnothing, \varnothing, l = 0\}$
17: **else**
18: **return** $\{y, \boldsymbol{c}, l = 1\}$

5 Experiments and Results

In this section, we evaluate the empirical performance of `CBO-FHCHEI` on a set of constrained equation discovery problems for nonlinear dynamical systems as well as a knowledge-driven configuration of a driving simulator. On the benchmark examples, we compare against two simple baselines, random sampling (`RS`) and simulated annealing (`SA`). We also compare against the state-of-the-art in constrained combinatorial optimization probabilistic reparameterization (`PR`) (Daulton et al., 2022). In our experiments, we find:

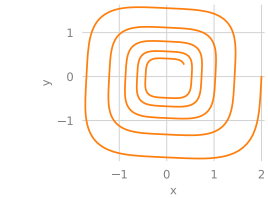
1. On the equation discovery problems, `CBO-FHCHEI` is either competitive with or outperforms the other methods consistently. It always yields good equations independent of the random seed.
2. At a fixed budget of 500 evaluations, `CBO-FHCHEI` is approximately 10× faster than `PR` in terms of wall clock time. This highlights the reduced computational overhead due to the closed-form inference and acquisition function.

3. Both constrained BO methods `CBO-FHCHEI` and `PR` perform especially well when finding feasible solutions is more difficult.
4. `CBO-FHCHEI` is able to tune a complex driving simulator with approximately 10^{14} possible configurations towards a driver's preference with the constraint that the simulation needs to be real-time capable.

While other methods such as `BOCS`, `Cosmopolitan`, `COMBO`, and `HyBO` would be interesting for this investigation, they have not been considered since there are issues that prevent them from naively handling constraints of any kind, and their very high computational demands exceeded an acceptable time budget for the optimization Daulton et al. (2022). These issues made them inappropriate for our use-case.

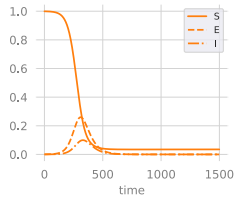
5.1 Equation Discovery for Nonlinear Dynamical Systems

In this section, we investigate system identification for the set of low-dimensional nonlinear dynamical systems shown in Figure 5. We use some of the benchmark problems and a similar learning setup from Brunton et al. (2016); Mangan et al. (2017). We investigate a range of dynamical systems: a simple nonlinear damped oscillator, a disease transmission model (SEIR), the chaotic Lorenz Oscillator, and the mean-field model for the cylinder wake in reduced coordinates. More details about these dynamical systems can be found in the appendix section A.3.



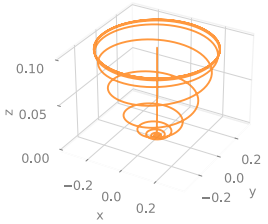
Nonlinear Damped Osc.

$$\dot{x} = -\alpha x^3 + \beta y^3$$
$$\dot{y} = -\beta x^3 - \alpha y^3$$



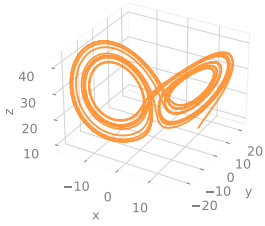
SEIR

$$\dot{S} = \mu - \mu S - \beta\, SI$$
$$\dot{E} = -(\mu + \alpha)E + \beta\, SI$$
$$\dot{I} = \alpha\, E - (\mu + \gamma)I$$



Cylinder Wake

$$\dot{x} = \mu\, x - \omega\, y + A\, xz$$
$$\dot{y} = \omega\, x + \mu\, y + A\, yz$$
$$\dot{z} = -\lambda\, z + \lambda\, x^2 + \lambda\, y^2$$



Lorenz Oscillator

$$\dot{x} = -\sigma\, x + \sigma y$$
$$\dot{y} = \rho\, x - 1\, y - 1\, xz$$
$$\dot{z} = -\beta\, z + 1\, xy$$

Figure 5: Equation discovery benchmark problems

Let $\boldsymbol{z}^{(t)} = \begin{bmatrix} z_1^{(t)} & \cdots & z_d^{(t)} \end{bmatrix} \in \mathbb{R}^d$ and $Z = \begin{bmatrix} \boldsymbol{z}^{(1)} & \cdots & \boldsymbol{z}^{(N)} \end{bmatrix}^\top \in \mathbb{R}^{N \times d}$ be the matrix of N noisy time measurements of all d states, which are available for learning the system. By differentiating the state measurements, we obtain noisy estimates of the state derivatives over time $\dot{\boldsymbol{z}}^{(t)} = \boldsymbol{f}_{\text{ode}}(\boldsymbol{z}^{(t)}) + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma_{sn}^2)$. We seek to find the structure of the governing nonlinear differential equation $\boldsymbol{f}_{\text{ode}}$, which yields the best simulation model.

A common practice in regression problems is to assume that each dimension of $\boldsymbol{f}_{\text{ode}}$ can be represented by a linear combination of features $\Theta(Z) \in \mathbb{R}^{N \times n_f}$, weighted by the coefficient vector $\boldsymbol{\Xi} \in \mathbb{R}^{n_f \times d}$. These n_f features compose a large library of possible nonlinear candidate functions that can represent the real system. For the benchmark problems at hand, we consider that each ODE dimension can be represented by polynomial terms up to degree p, which is indeed true for all the problems. In vector form, we can write

$$\underbrace{\begin{bmatrix} | & & | \\ \dot{\boldsymbol{z}}_1 & \cdots & \dot{\boldsymbol{z}}_d \\ | & & | \end{bmatrix}}_{\dot{Z}} = \underbrace{\begin{bmatrix} 1 & \boldsymbol{z}_1^{(1)} & \cdots & \boldsymbol{z}_d^{(1)} & (z_1^{(1)})^2 & \boldsymbol{z}_1^{(1)}\boldsymbol{z}_2^{(1)} & \cdots & (z_d^{(1)})^p \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \boldsymbol{z}_1^{(N)} & \cdots & \boldsymbol{z}_d^{(N)} & (z_1^{(N)})^2 & \boldsymbol{z}_1^{(N)}\boldsymbol{z}_2^{(N)} & \cdots & (z_d^{(N)})^p \end{bmatrix}}_{\Theta(\boldsymbol{Z})} \cdot \underbrace{\begin{bmatrix} | & & | \\ \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_d \\ | & & | \end{bmatrix}}_{\boldsymbol{\Xi}} + \boldsymbol{\epsilon}\,, \tag{23}$$

which can be solved for the model coefficients Ξ, using a least squares estimator.

Considering a large feature library enhances the flexibility and possibly the accuracy of the model up to the point that the predictive error $\dot{Z} - \Theta(Z)\,\Xi$ approaches zero. However, especially in the presence of noise, complex models have the potential to overfit the data leading to a deterioration of the simulation performance and even instabilities. To address this issue, we want to consider a large feature library but use structure selection to search efficiently and globally for the optimal subset of terms that construct parsimonious models, i.e., the least number of terms that significantly reduce the simulation error (Brunton et al., 2016).

Note that the number of monomials is equal to $n_t = d\,n_f = d\,\sum_{i=0}^{p} \binom{d+i-1}{i}$, and the total number of possible model structure combinations is given by $n_m = 2^{n_t}$, which grows exponentially with the number of monomials, which further grows factorially with the number of dimensions and the polynomial degree. An overview of the search space dimension for the benchmark problems is given in Table 1. Notably, the number of models is in the order of quintillions.

Table 1: Equation discovery benchmark problems

Dynamical System	d	p	n_t	n_m
Nonlinear Damped Oscillator	2	5	42	4.4e+12
SEIR	3	3	60	1.2e+18
Cylinder Wake	3	3	60	1.2e+18
Lorenz Oscillator	3	3	60	1.2e+18

We define the optimization problem in the same form as Equation 1 as follows

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \quad f(\boldsymbol{x}) = \quad \log_{10}\frac{1}{N}\sum_{t}^{N} \|\boldsymbol{z}_{sim}^{(t)}(\boldsymbol{x}) - \boldsymbol{z}^{(t)}\|_1 + \lambda \log_2 \sum \boldsymbol{x} \tag{24}$$

$$s.t. \quad g(\boldsymbol{x}) = \sum_{i}^{d} \|\boldsymbol{\xi}_i(\boldsymbol{x})\|_1 - \delta \leq 0 \tag{25}$$

$$h(\boldsymbol{x}) = 1 \tag{26}$$

$$\dot{Z}_{sim} = \Theta(Z_{sim}) \cdot \Xi(\boldsymbol{x})\,, \tag{27}$$

where $\boldsymbol{x} \in \{0,1\}^{n_t}$ are the $n_t = d\,n_f$ binary decision variables that select which of the n_f features or columns of $\Theta(Z)$ will be present in the model for each of the d dimensions. λ is a hyper-parameter, δ is a constraint threshold, and $\boldsymbol{z}_{sim}(\boldsymbol{x})$ is the auto-regressive simulation trajectory of the system. The coefficients $\Xi(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{\xi}_1 & \dots & \boldsymbol{\xi}_d\end{bmatrix} \in \mathbb{R}^{n_f \times d}$ of the selected system are estimated by solving the least squares problem for each subset of $\boldsymbol{\xi}_i$, selected by $\boldsymbol{x}$. The first right-hand-side term of Equation 24 quantifies the mean absolute error between the measurements and the simulated system. The second term is a parsimony-based criterion that rewards models with fewer monomials. The inequality in Equation 25 describes the Lasso constraint and is used to restrict the model complexity. If the simulation is unstable, then $h(\boldsymbol{x}) = 0$. The final identified simulation model is given by Equation 27.

We run our method `CBO-FRCHEI`, as in Algorithm 2, with batch size equal two to reduce the computation needed for the same evaluation budget. For `RS` we pick a configuration uniformly at random, and `SA` is the same as shown in Algorithm 1, with the difference that we set the objective function to infinity if the constraint is violated or the simulation is unstable. `PR` is used with the recommended settings for discrete binary optimization and batch size equal one. We provide the same randomly sampled 50 evaluations and let the optimizers run for additional 450 model evaluations. We execute each method 10 times and report mean performance and standard error.

Figure 6 shows the best feasible objective function found after a number of evaluations. The proposed method is able to find good model structures after only a few evaluations and consistently gives the best performances for all benchmark problems. Additionally, our method is around 10× faster than `PR`, mainly because our AF can be calculated in closed-form, while `PR` requires expensive MCMC sampling methods.

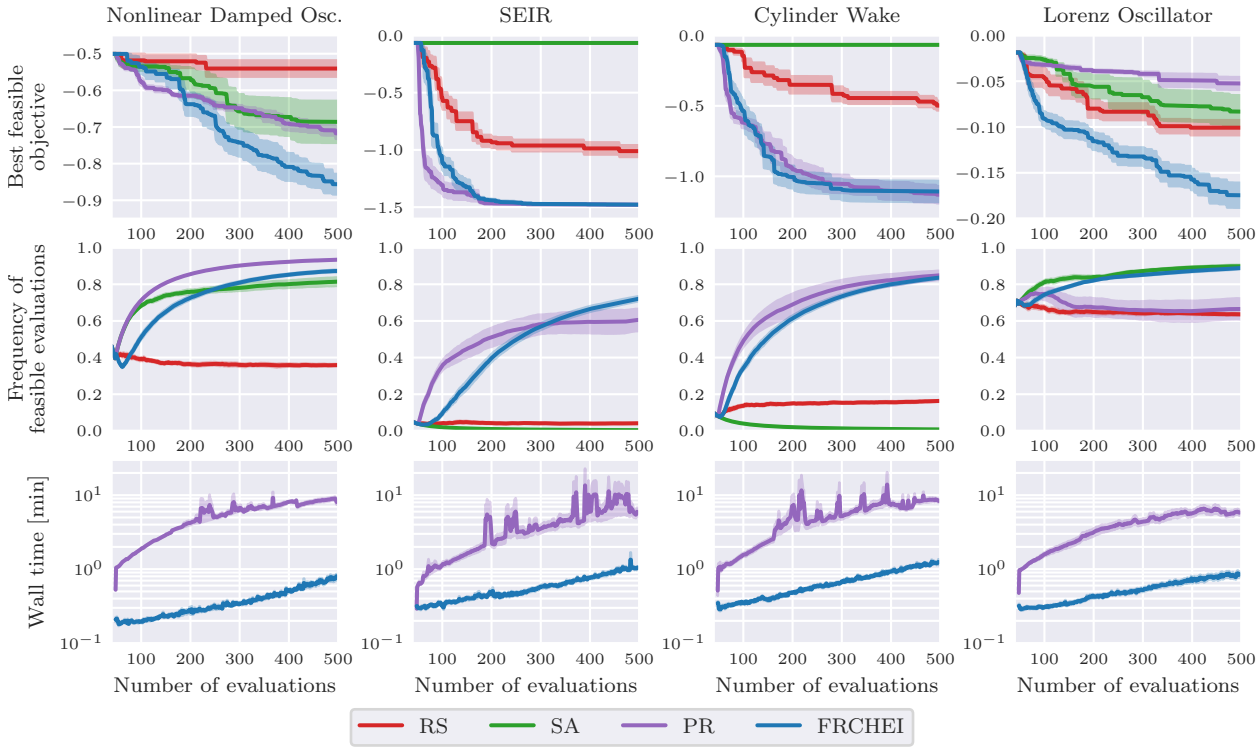
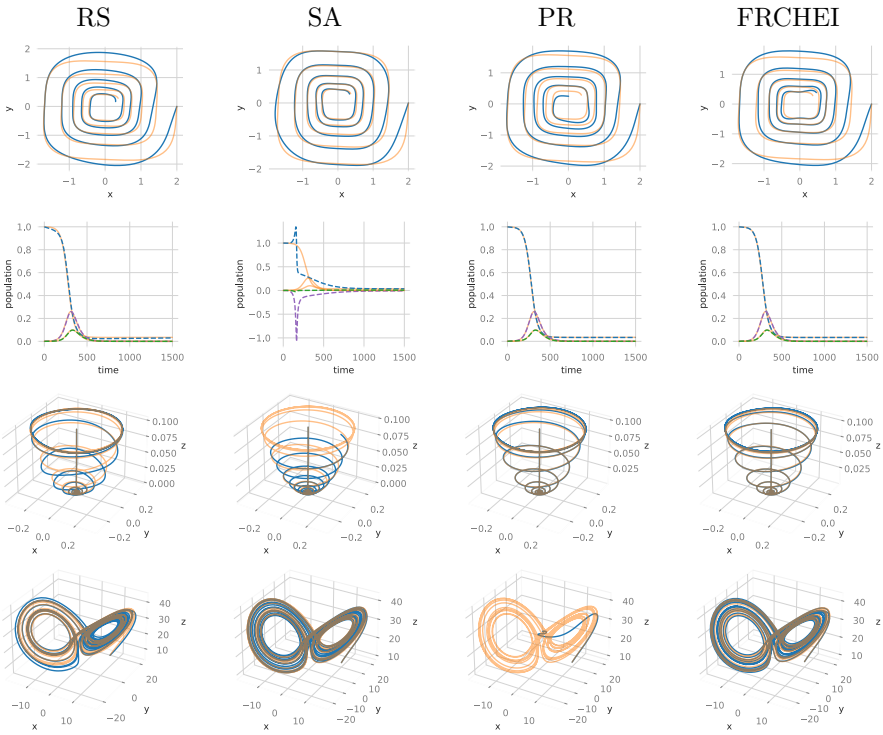


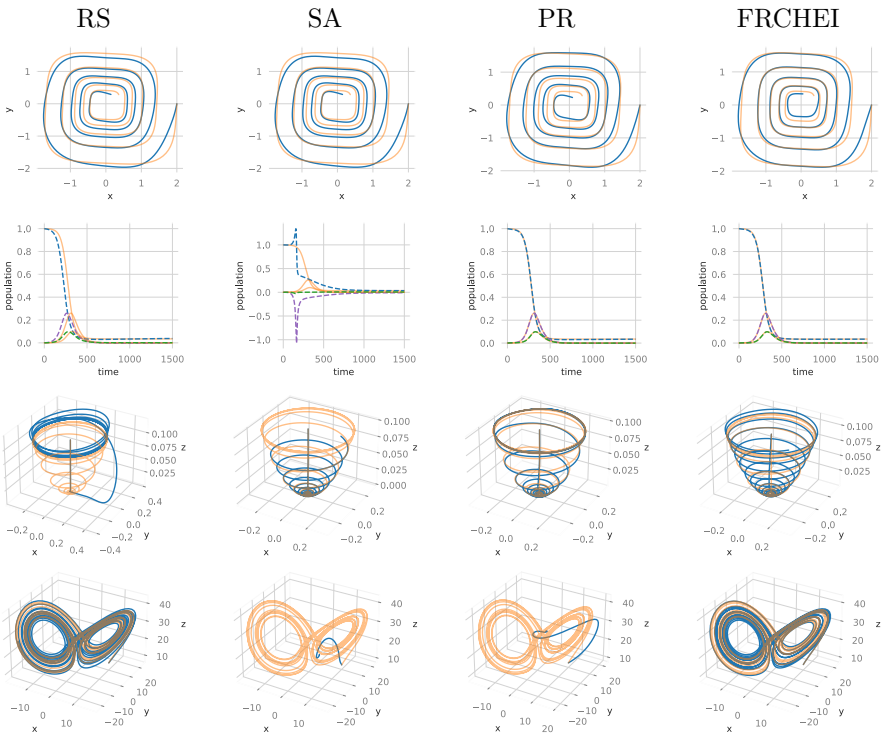
Figure 6: Best feasible objective function evaluation, wall-time per evaluation, and frequency of feasible solutions over the number of iterations for different optimization methods applied to the equation discovery problem.

After 500 evaluations, the wall-clock time of `PR` is around $30 - 42$ hours, while our method achieves similar or better results in $3 - 5$ hours. The `RS` and `SA` methods have a negligible computation time, and a simulation takes around 4 seconds. As we gather more data points, the cubic scaling of $\mathcal{GP}$ regression will dominate the computation times. Note that our method runs with a batch size of two, meaning that we choose two candidates in each iteration, which halves the computational overhead. In appendix Section A.4 we show that a batch size of two is a good compromise between computation time and performance. For the SEIR and cylinder wake systems, `SA` never improves from the initial feasible solution. This is because the algorithm changes only one variable at a time, which seems to lead to either constraint violations or crashes. This highlights the need for global search methods that can escape from bad initial configurations. Both `SA` and `PR` do not perform well for the Lorenz chaotic oscillator and are stuck in local minima, where the states converge fast to a stationary point around the oscillation center (cf. Figure 7a and Figure 7b). In Figure 6 we also show the frequency of feasible points evaluated by all methods. Our proposed AF (Equation 18) encourages early exploration and, as the classifier and constraint functions are learned, evaluates more feasible points and therefore finds better solutions. The frequency of infeasible evaluations of the baselines further motivates the benefits of constraint BO. If the search space contains many infeasible points the surrogate model can learn these regions and concentrate the search on feasible regions. This leads to better overall solutions.

Figures 7a and 7b compare, for each method, the measurements with the simulation trajectories of the resulting optimized model for the best and worst among 10 optimization runs. Notably, our method can yield good models for all runs and benchmark problems, while other methods exhibit a much higher variance. Low variance results are desirable for expensive optimization problems since we usually run them only once. The pure exploratory policy from `RS` also does not perform well, even in the best case. Overall, it is important that the method for structure selection provides a good balance between exploitation and exploration, such that it can visit a wide range of different model structures and also avoid local minima. Our experiments show that the proposed method `CBO-FRCHEI` achieves a favorable trade-off.



(a) Best models among 10 optimization runs.



(b) Worst models among 10 optimization runs.

Figure 7: Comparison between measurements (orange) depicted without noise and simulated trajectories of the best feasible model after optimization (blue).

The appendix Section A.4 provides ablation studies with further insights and analysis of the proposed method. We investigate the performance of `CBO-FRCEI` and `CBO-CHEI`, two variants of the proposed method. The first is similar to `CBO-FRCHEI` but uses a $\mathcal{GP}$ instead of a hierarchical $\mathcal{TP}$ prior for the regression surrogates. The second does not account for failures in the AF. Further, we investigate the effect of using different kernels and show that the proposed combination of popular kernels outperforms others for these benchmark problems. Finally, we investigate the effect of varying the batch size.

5.2 Real-time Multibody Dynamics Model Optimization

In this experiment, we want to find a configuration for a digital twin of an electric vehicle for a driving simulator. For years, driving simulators have played a key role in driving dynamics and advanced driver assistance systems development. They reduce the number of prototypes required and the duration of the development cycle. Many aspects of the car development can be tested under varying, realistic conditions before road testing has even started (BMW AG, 2020). To allow for a realistic reproduction of driving dynamics, there is the need to account for simulators with hardware specifications that can reproduce the dynamics in the operation range of the dynamical system, as well as accurate real-time capable motion cueing and digital-twin vehicle models. From the hardware perspective, companies have invested many millions of euros in state-of-the-art simulators that fulfill those requirements. Altogether, the design of accurate simulation models is one of the main enablers and a key factor in virtual vehicle development. Figure 8 shows the high-fidelity simulator at the BMW AG driving simulation site.

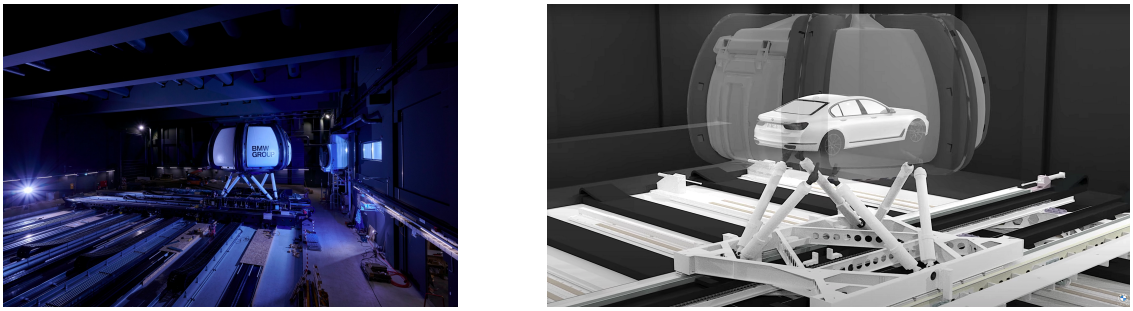


Figure 8: High-fidelity driving simulator with 9 degrees of freedom, motion area of nearly 400 square meters and a peak electrical power required up to 6.5 MW. Source BMW AG (2020).

The digital twin is a multibody dynamics model, which consist of multiple entities (bodies, joints, force and control elements, etc.) (Featherstone, 2014). Every entity has multiple possible implementations, and the goal is to choose a suitable one from a library while retaining the overall model real-time capability. For different use-cases of the driving simulator, different implementations are necessary. This requires the digital twin to be adapted on a regular basis. We aim to automate this process using our proposed BO method. For example, a damper can be implemented as a simple lookup table or as a differential equation. While a complex damper model improves simulation accuracy on a bumpy road, a simple lookup table is sufficient for longitudinal dynamics maneuvers. In addition, entities are interacting. For example, reproducing accurately vertical dynamics requires an accurate tire model that provides the correct inputs to the damper model and, subsequently, the right forces to the chassis. In summary, the major challenge in system identification for the driving simulator boils down to selecting the appropriate templates for each entity in order to optimize model simulation accuracy while simultaneously constraining the computational complexity to enable real-time simulations. From a system identification perspective, the task of selecting among the available templates can be viewed as a knowledge-driven structure selection task.

However, this task is not straightforward. Model designers face an overwhelming number of potential configurations arising from the combinatorial explosion when selecting templates. Furthermore, increasing the complexity of a specific component does not necessarily guarantee overall model accuracy improvement. Similarly, enhancing the computational complexity of a single component does not linearly affect the complexity of the entire vehicle model. For example, replacing a rigid joint with a force element allows for the representation of compliance between joints, consequently increasing the total number of degrees of

freedom and potentially slowing down the simulation. Nevertheless, modern simulation softwares optimize and parallelize computations. Conversely, removing a joint may split the kinematic topology of the multibody system into two separate, parallelizable systems, resulting in faster simulations.

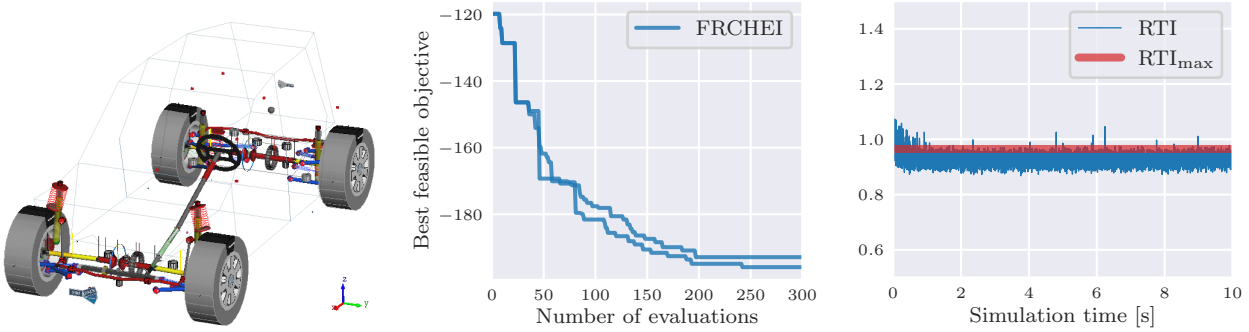
We investigate automatic template selection for knowledge-driven structure selection with our proposed combinatorial BO method `CBO-FRCHEI`. We model the vehicle using the Simpack simulation software (Dassault Systemes, 2023) and parameterize the model with 46 categorical variables that alter the template of different components. The vehicle for one of the configurations is depicted in Figure 9a. All the template coefficients were identified in a preliminary phase, so there is no need for a parameter estimator. The discrete design space $\mathcal{X}$ is defined as follows:

- 5 binary variables switch between rigid and flexible formulation of bodies.
- 6 categorical variables switch the number of modes used in the model-order-reduction of various flexible bodies.
- 2 binary variables switch the tire contact model approach.
- 4 binary variables switch between rigid (joint) and compliant (force element) motor mounts, suspension rods and stabilizer bushings.
- 29 binary variables switching the complexity of lookup tables (linear and nonlinear) representing the stiffness of various compliant bushing elements.

The structure identification as a constrained combinatorial optimization problem is defined as

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}) \\ s.t. \quad & g(\boldsymbol{x}) = \mathrm{RTI}_{\max}(\boldsymbol{x}) - 1 \leq 0 \\ & h(\boldsymbol{x}) = 1\,, \end{aligned} \tag{28}$$

where the function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ defines the model performance. In our experiment, the objective function reflects the preference of expert drivers for one model over another. The real-time-index (RTI), is defined as the ratio of the time required to advance one time step in the simulation divided by the simulation time step size. Real-time models must have an RTI below 1 in 99% after the first 2 seconds of 'warming-up'. Simulation failures arise from certain input combinations that result in numerical instabilities or in an invalid model, for instance, when it induces kinematic loops that can not be handled by an ODE solver.



(a) 3D view of the multibody dynamical model representing the vehicle.

(b) Progress of the 2 optimization runs for the best feasible solution over number of model evaluations.

(c) RTI measurement of the final best vehicle found during optimization.

Figure 9: Optimization results for the multibody dynamics problem

We provide the evaluation of 30 initial random samples for the optimizer, with a total evaluation budget of 300 evaluations and run the optimization twice. Figure 9b shows the simulation results for the problem at hand. It can be seen that the proposed method finds good real-time capable configurations. Figure 9c depicts the `RTI` signal for the first 10 seconds of simulation for the best feasible model found during optimization among the two runs. We leave the detailed objective and subjective evaluation of the optimized simulation model for future work.

6 Conclusion and Outlook

In this paper, we address the task of structure identification for dynamical systems, where the model evaluation is subject to constraints and failures. Constraints can be used to limit the computational budget available for the model and help to mitigate overfitting and over-completeness of models. We propose to search efficiently over potential model structures with combinatorial Bayesian optimization. We proposed `CBO-FRCHEI`, an efficient and competitive combinatorial BO algorithm that handles both constraints and failures and has relatively little computational overhead.

We encode the choice of the structure of the dynamical system using binary and categorical decision variables. Our method handles these discrete inputs by designing kernels specific for discrete inputs. We combine recent ideas in kernel design and show that the proposed kernel outperforms state-of-the-art kernels. Our method handles the black-box constraint and failure functions by learning them with Bayesian regression and classification methods and therefore learns to avoid these regions during search. Finally, we focus on scalability up to a large number of discrete decision variables and make design choices that favour the run time. Our surrogate model and acquisition function is evaluated in closed-form. This allows our method to optimize problems with up to 10^{18} possible combinations.

We provide benchmark problems in the field of symbolic-regression that provide evidence that our method outperforms other methods for system identification of a variety of nonlinear dynamical systems, such as disease models, oscillators and chaotic systems. In addition, we provide a complex real-world application example of knowledge-driven system identification where the choice of templates of a multibody dynamical system of an electric vehicle are optimized for accuracy, real-time capabilities and numerical robustness. As the proposed method is not specific to the presented application, it is in principle applicable to other constrained combinatorial problems. Investigating its potential and performance on problems, such as network structure optimization (Du et al., 2022), wait-and-judge scenario optimization (Campi & Garatti, 2018), neural structure optimization with pure categorical variables, etc. are interesting topics for future research.

Promising future research directions are improved surrogate models that capture higher-order interactions of variables or leverage the structure of dynamical systems. In addition, there is often a lot of additional prior knowledge about the problem at hand that can be incorporated. For example, promising candidates based on expert knowledge or qualitative knowledge about the interaction of terms which can be modeled as graphs.

References

Francois Bachoc, Celine Helbert, and Victor Picheny. Gaussian process optimization with failures: classification and convergence proof. *Journal of Global Optimization*, 78(3):483–506, jul 2020. doi: 10.1007/s10898-020-00920-0.

Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pp. 462–471. PMLR, 2018.

Dominik Baumann, Friedrich Solowjow, Karl Henrik Johansson, and Sebastian Trimpe. Identifying causal structure in dynamical systems. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=X2BodlyLvT`.

Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

BMW AG. BMW Group sets new standards for driving simulation. `https://www.press.bmwgroup.com/global/article/detail/T0320021EN/bmw-group-sets-new-standards-for-driving-simulation-nextgen-2020-offers-exclusive-insights-before-the-new-driving-simulation-centre-starts-work?language=en`, November 2020. Accessed: 2023-08-04.

Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

Elizabeth Bradley, Matthew Easley, and Reinhard Stolle. Reasoning about nonlinear system identification. *Artificial Intelligence*, 133(1-2):139–188, 2001.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113 (15):3932–3937, 2016.

Marco C Campi and Simone Garatti. Wait-and-judge scenario optimization. *Mathematical Programming*, 167:155–189, 2018.

Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

Ankush Chakrabarty, Scott A Bortoff, and Christopher R Laughman. Simulation failure robust bayesian optimization for estimating black-box model parameters. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1533–1538. IEEE, 2021.

Rick Chartrand. Numerical differentiation of noisy, nonsmooth data. *International Scholarly Research Notices*, 2011, 2011.

Zhehui Chen, Simon Mak, and CF Jeff Wu. A hierarchical expected improvement method for bayesian optimization. *Journal of the American Statistical Association*, pp. 1–14, 2023.

Hamid Dadkhahi, Jesus Rios, Karthikeyan Shanmugam, and Payel Das. Fourier representations for black-box optimization over categorical variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10156–10165, 2022.

Dassault Systemes. Simpack: Multibody systems simulation software. `https://www.3ds.com/products-services/simulia/products/simpack/`, June 2023. Accessed: 2023-08-04.

Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *Advances in Neural Information Processing Systems*, 35:12760–12774, 2022.

Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *International Conference on Machine Learning*, pp. 2632–2643. PMLR, 2021.

Wei Du, Gang Li, and Xiaochen He. Network structure optimization for social networks by minimizing the average path length. *Computing*, 104(6):1461–1480, 2022.

David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.

C Elster, K Klauenberg, M Walzel, G Wubbeler, P Harris, M Cox, C Matthews, I Smith, L Wright, A Allard, et al. A guide to bayesian inference for regression problems, deliverable of emrp project new04 "novel mathematical and statistical approaches to uncertainty evaluation". 2015.

David Eriksson and Matthias Poloczek. Scalable constrained bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 730–738. PMLR, 2021.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pp. 937–945, 2014.

Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational intelligence in expensive optimization problems*, pp. 131–162. Springer, 2010.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Florian Hase, Loïc M Roch, Christoph Kreisbeck, and Alán Aspuru-Guzik. Phoenics: a bayesian optimizer for chemistry. *ACS central science*, 4(9):1134–1145, 2018.

Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 8(3), 2021.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pp. 507–523. Springer, 2011.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. π bo: Augmenting acquisition functions with user beliefs for bayesian optimization. *arXiv preprint arXiv:2204.11051*, 2022.

Kondor Risi Imre. Diffusion kernels on graphs and other discrete input spaces. In *Proc. 19th Int. Conf. Machine Learning, 2002*, 2002.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

Phuc Luong, Sunil Gupta, Dang Nguyen, Santu Rana, and Svetha Venkatesh. Bayesian optimization with discrete variables. In *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019, Proceedings 32*, pp. 473–484. Springer, 2019.

Enes Makalic and Daniel F Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182, 2015.

Niall M Mangan, J Nathan Kutz, Steven L Brunton, and Joshua L Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009, 2017.

Alonso Marco, Dominik Baumann, Majid Khadiv, Philipp Hennig, Ludovic Righetti, and Sebastian Trimpe. Robot learning with crash constraints. *IEEE Robotics and Automation Letters*, 6(2):1439–1446, 2021.

Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. Local policy search with bayesian optimization. *Advances in Neural Information Processing Systems*, 34:20708–20720, 2021.

Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive Processing*, 12(4): 319–340, Nov 2011. ISSN 1612-4790. doi: 10.1007/s10339-011-0404-1. URL `https://doi.org/10.1007/s10339-011-0404-1`.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. *Advances in Neural Information Processing Systems*, 32, 2019.

Theodore P Papalexopoulos, Christian Tjandraatmadja, Ross Anderson, Juan Pablo Vielma, and David Belanger. Constrained discrete black-box optimization using mixed-integer programming. In *International Conference on Machine Learning*, pp. 17295–17322. PMLR, 2022.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324 (5923):81–85, 2009.

Amar Shah, Andrew G Wilson, and Zoubin Ghahramani. Bayesian optimization using student-t processes. In *NIPS Workshop on Bayesian Optimization*, 2013.

Marco Taboga. Lectures on probability theory and mathematical statistics, 2017.

Jovan Tanevski, Ljupčo Todorovski, Yannis Kalaidzidis, and Sašo Džeroski. Domain-specific model selection for structural identification of the rab5-rab7 dynamics in endocytosis. *BMC Systems Biology*, 9(1):1–17, 2015.

Jovan Tanevski, Ljupčo Todorovski, and Sašo Džeroski. Combinatorial search for selecting the structure of models of dynamical systems with equation discovery. *Engineering Applications of Artificial Intelligence*, 89:103423, 2020.

Brendan D Tracey and David Wolpert. Upgrading from gaussian processes to student'st processes. In *2018 AIAA Non-Deterministic Approaches Conference*, pp. 1659, 2018.

Juan Ungredda and Juergen Branke. Bayesian optimisation for constrained problems. *arXiv preprint arXiv:2105.13245*, 2021.

Floris Van Breugel, Yuying Liu, Bingni W Brunton, and J Nathan Kutz. Pynumdiff: A python package for numerical differentiation of noisy time-series data. *Journal of Open Source Software*, 7(71):4078, 2022.

Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. *arXiv preprint arXiv:2102.07188*, 2021.

A Appendix

A.1 Constrained Hierarchical Expected Improvement

In this section, we provide the remaining ingredients to derive the closed-form expression for the acquisition function used: the failure-robust constrained hierarchical expected improvement (`FRCHEI`) as defined in equations 18-22.

Let the hierarchical predictive posterior for the objective and for each i-th constraint function at a test point $\boldsymbol{x}'$ be defined as $\tilde{f}(\boldsymbol{x}') = f(\boldsymbol{x}') \mid \boldsymbol{x}', \boldsymbol{y}, X \sim T(\bar{\nu}_f, \bar{\mu}_f, \bar{\sigma}_f^2)$ and $\tilde{g_i}(\boldsymbol{x}') = g_i(\boldsymbol{x}') \mid \boldsymbol{x}', \boldsymbol{y}, X \sim T(\bar{\nu}_{g_i}, \bar{\mu}_{g_i}, \bar{\sigma}_{g_i}^2)$ respectively. The definition of the constrained hierarchical improvement function is inspired by Gardner et al. (2014) and is defined as

$$\mathrm{CHI}(\boldsymbol{x}') = \Delta(\boldsymbol{x}') \max\{0, y^+ - \tilde{f}(\boldsymbol{x}')\}\,, \tag{29}$$

where y^+ is the lowest feasible objective function observed so far and $\Delta(\boldsymbol{x}') \sim \mathrm{Bernoulli}(\gamma(\boldsymbol{x}')) \in \{0, 1\}$ is the feasibility indicator function with parameter

$$\begin{aligned}
P_{\text{feas}}(\boldsymbol{x}') &:= \gamma(\boldsymbol{x}') && (30)\\
&= \mathbb{E}\left[\Delta(\boldsymbol{x}')\right] && (31)\\
&= p(\tilde{g}_1(\boldsymbol{x}') \leq 0, \ldots, \tilde{g}_m(\boldsymbol{x}') \leq 0) && (32)\\
&= \prod_{i=1}^{m} p(\tilde{g}_i(\boldsymbol{x}') \leq 0) && (33)\\
&= \prod_{i=1}^{m} \int_{-\infty}^{0} p(g_i(\boldsymbol{x}') \mid \boldsymbol{x}', \boldsymbol{y}, X)\, dg_i(\boldsymbol{x}') && (34)
\end{aligned}$$

$$= \prod_{i=1}^{m} \Phi_T(0;\, \bar{\nu}_{g_i}, \mu_{g_i}, \sigma^2_{g_i})\,. \tag{35}$$

Note that from Equation 32 to Equation 33 we assumed the simplest case, where are constraints are conditionally independent given $\boldsymbol{x}'$. In addition, Equation 34 is a simple product of univariate cumulative density of the t-distribution Φ_T evaluated at 0, which can be calculated analytically for any $\boldsymbol{x}'$.

The constrained hierarchical expected improvement is obtained by taking the expectation of Equation 29:

$$\begin{aligned}
\mathrm{CHEI}(x') &= \mathbb{E}\left[\mathrm{CHI}(\boldsymbol{x}')\right] && (36)\\
&= \mathbb{E}\left[\Delta(\boldsymbol{x}')\right]\ \mathbb{E}\left[\max\{0, y^+ - \tilde{f}(\boldsymbol{x}')\right] && (37)\\
&= P_{\text{feas}}(\boldsymbol{x}')\ \mathrm{HEI}(\boldsymbol{x}')\,, && (38)
\end{aligned}$$

where $P_{\text{feas}}(\boldsymbol{x}')$ is defined as in Equation 30 and HEI can be obtained with the help of the reparameterization trick (Tracey & Wolpert, 2018) with $\tau \sim T(\bar{\nu}_f, 0, 1)$ and $\tau^+ = (y^+ - \bar{\mu}_y)/\bar{\sigma}_y$:

$$\begin{aligned}
\mathrm{HEI}(\boldsymbol{x}') &= \mathbb{E}_{f(\boldsymbol{x}')\sim p(f(\boldsymbol{x}')|\boldsymbol{x}',\boldsymbol{y},X)}\left[\max\{0, y^+ - f(\boldsymbol{x}')\right] && (39)\\
&= \int_{-\infty}^{y^+} (y^+ - f(\boldsymbol{x}'))\ T(f(\boldsymbol{x}')\ ;\ \bar{\nu}_f, \bar{\mu}_f, \bar{\sigma}^2_f)\ df(\boldsymbol{x}') && (40)\\
&= \int_{-\infty}^{\tau^+} (y^+ - \bar{\mu}_f - \bar{\sigma}_f\tau)\ T(\tau\ ;\ \bar{\nu}_f, \bar{\mu}_f, \bar{\sigma}^2_f)\ d\tau && (41)\\
&= (y^+ - \bar{\mu}_f)\ \Phi_T\left(\tau^+; \bar{\mu}_f, 0, 1\right) - \bar{\sigma}_f \int_{-\infty}^{\tau^+} \tau\ T(\tau\ ;\ \bar{\nu}_f, \bar{\mu}_f, \bar{\sigma}^2_f)\ d\tau && (42)\\
&= \bar{\sigma}_f\ \tau^+\ \Phi_T\left(\tau^+; \bar{\nu}_f, 0, 1\right) + \bar{\sigma}_f \left(\frac{\bar{\nu}_f}{\bar{\nu}_f - 1}\right)\left(1 + \frac{\tau^{+2}}{\bar{\nu}_f}\right)\ T(\tau^+\ ;\ \bar{\nu}_f, 0, 1) && (43)\\
&= \bar{\sigma}_f\ \left[\tau^+\ \Phi_T\left(\tau^+; \bar{\nu}_f, 0, 1\right) + \frac{\bar{\nu}_f + \tau^{+2}}{\bar{\nu}_f - 1}\ T(\tau^+\ ;\ \bar{\nu}_f, 0, 1)\right]. && (44)
\end{aligned}$$

The other minor modifications to turn `CHEI` into `FRCHEI` are defined in Section 4.3.

A.2 Implementation Details

We implement Gaussian and student-t process regression, Gaussian process classification, acquisition function and optimization from scratch using PyTorch (Paszke et al., 2017) and Pyro (Bingham et al., 2019). All the simulations for the symbolic-regression problems have been performed with a fixed-step RK45 solver, where the 5th stage has been used to detect numerical instabilities, which stops the simulation and reports the failure to the main program. We provide a self-implementation of this solver that performs simulations in parallel and is useful to speed up computation when many simulations are to be made with the same integration time but with different model parameters.

A.3 Experiment Details

In this section, we provide the experiment details required to reproduce each benchmark problem in Section 5.1. Details are provided in Table 2. The parameters $\boldsymbol{\theta}_{ode}$ refer to the differential equations for each benchmark problem, as depicted in Figure 5. The available noisy measurements for parameter estimation and model evaluation consists of a single simulation run, starting at an initial state $x(t_0 = 0)$, simulated with a fixed simulation time step size Δ_t up to the stop time t_f. The state measurements are corrupted equally for each dimension with a zero-mean Gaussian noise with standard deviation σ_{sn}. All measurements have been filtered using the Total Variation Regularized Denoising technique (Chartrand, 2011; Van Breugel et al., 2022) with the same regularization parameter $T\gamma = 0.01$ and number of iterations $T_i = 10$.

The benchmark experiments ran on Intel Xeon Platinum 8160 Processors "SkyLake" at 2.1 GHz, on 4 isolated physical cores. The optimization and the simulations of the multibody dynamical problem were performed on

Table 2: Benchmark problem parameters

Dynamical System	$\boldsymbol{\theta}_{ode}$	$x(t_0)$	Δ_t	t_f	σ_{sn}
Nonlinear Damped Osc.	α=0.1, β=1.75, γ=1/2	$[2, 0]$	0.01	35	0.1
SEIR	μ=1e−5, α=1/5, β=1.75, γ=1/2	$[0.9995, 4e{-}4, 1e{-}4]$	0.1	150	0.01
Cylinder Wake	ω=1, μ=0.1, A= − 1, λ=1.	$[0.001, 0, 0.1]$	0.1	100	0.01
Lorenz Oscillator	σ=10, ρ=28, β=8./3	$[10, 10, 10]$	0.01	20	1

a 2x Intel Xeon Gold 6256 3.6Hz computer, running a RedHawk Linux RTOS (by Concurrent Real-Time). The first 12 cores were dedicated to the optimizer, while the remaining 12 cores of the second CPU socket were shielded and dedicated to simulations.

A.4 Ablation Studies

In this section, we provide ablation studies that reinforce the design choices made for the proposed algorithm.

Figure 10 shows simulation trajectories for the best feasible and stable models found over the progress of the optimization for the `CBO-FRCHEI` method. It can be verified that the `CBO-FRCHEI` algorithm quickly and progressively improves the model simulation performance. The model with the optimized model structure represents the measurements well and shows to be a good simulation model, despite the high noise level present in the measurements.

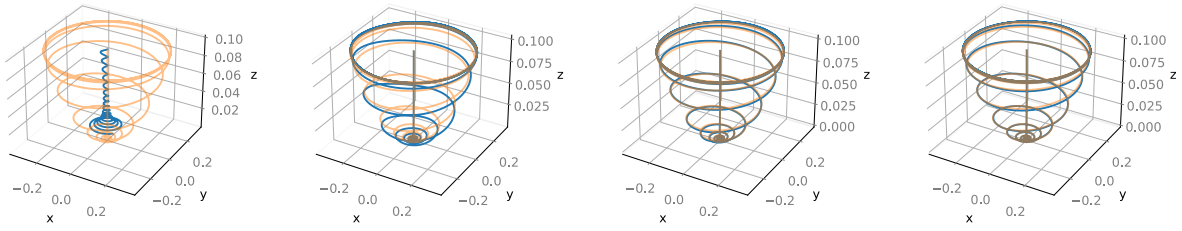


Figure 10: Comparison between measurements (orange) depicted without noise and simulated trajectories (blue) for the best feasible model found in the course of optimization for the Cylinder Wake problem and `CBO-FRCHEI`. From the left to the right are depicted the models at iteration 8, 163, 200 and 436, respectively.

In Figure 11, we compare `CBO-FRCHEI` to `CBO-FRCEI` and `CBO-CHEI`, two variants of the proposed method. The first is similar to `CBO-FRCHEI` but uses a $\mathcal{GP}$ instead of a hierarchical $\mathcal{TP}$ prior for the regression surrogates. The second does not account for failures in the AF. It can be seen that `FRCHEI` outperforms `CHEI` for all benchmarks, which agrees with Chakrabarty et al. (2021) and is an indication that learning failure regions improves the optimizer. As expected, `CBO-FRCHEI` has a higher success rate than `CBO-CHEI` for all benchmark problems. Evaluating failures is a waste of resources because no value for the cost function is obtained, and the candidate configuration is never considered the best experiment. In addition, this data point does not contribute to training the surrogates due to the missing target value. For this reason, the optimizer might get stuck into failure regions, where the objective function surrogate considers there is a substantial expected improvement. In Figure 12 it can be visually verified that `CHEI` has more failure evaluations than random sampling and all other methods, while `FRCHEI` has the least failure rate. Clearly, `CHEI` does not make progress, over-exploits the input space and falls into many failure regions. `FRCHEI` provides a good balance between exploration and exploitation, which is a desired characteristic for Bayesian optimization methods.

It can be further seen in Figure 11 that `FRCEI` provides similar performance and sometimes slightly outperforms `FRCHEI`. The idea behind hierarchical priors is to be able to better describe the model discrepancy towards outliers while preserving the closed-form solution of the acquisition function. One side effect of this approach is the inflation of the predictive posterior uncertainty that might lead to over-exploration of the design space, which is critical for Bayesian optimization in high-dimensional discrete problems. Nevertheless, we still

employ $\mathcal{TP}$ regression to our framework since it is a more robust and flexible probabilistic model. It does not increase the implementation complexity and computation time, and may work better for other problems.

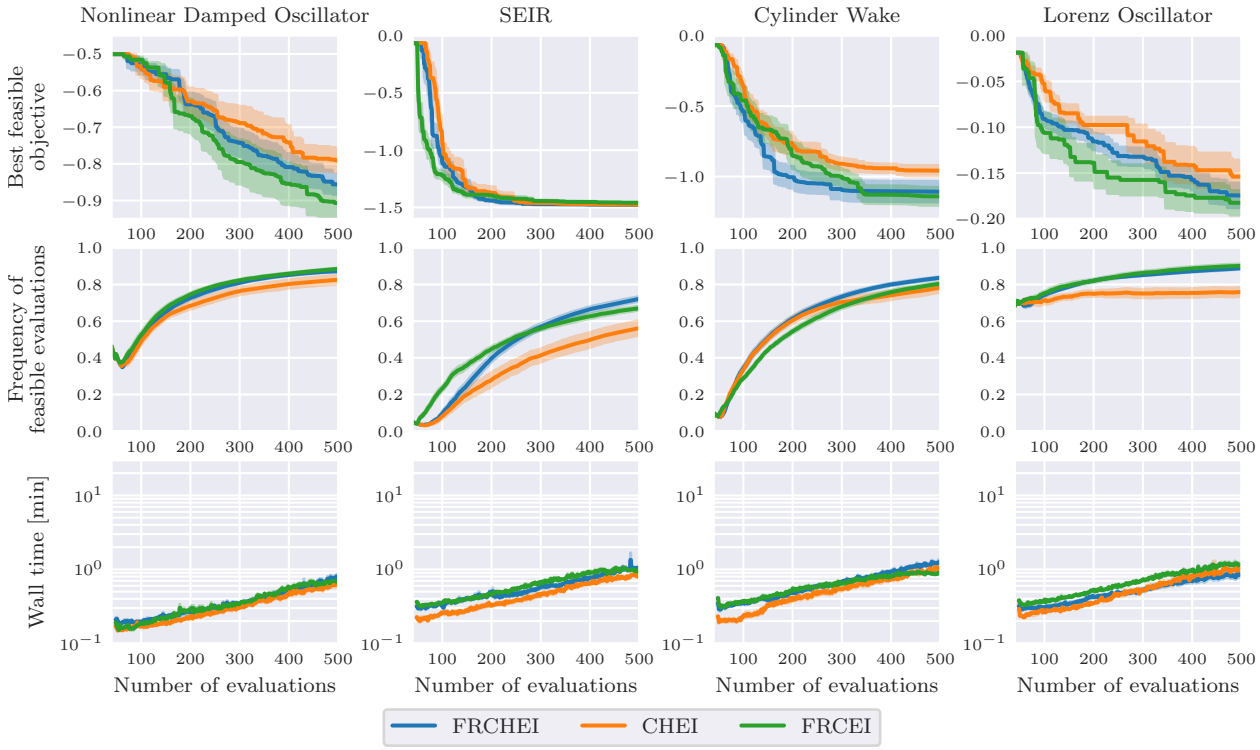
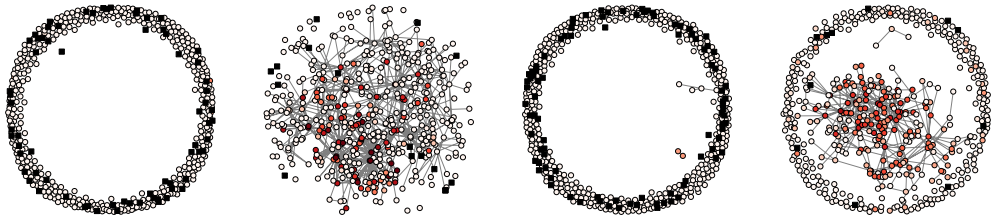


Figure 11: Optimization results for different variants of `CBO-FRCHEI`. `CBO-CHEI` is the variant that does not handle failures, and `CBO-FRCEI` is the variant that implements a $\mathcal{GP}$ instead of $\mathcal{TP}$ regression.



(a) RS (b) SA (c) CBO-CHEI (d) CBO-FRCHEI

Figure 12: Graph for the configurations evaluated during optimization of the Nonlinear Damped Oscillator experiment for different methods. Each node represents one configuration evaluated. Nodes are connected if the hamming distance between them is equal to 1, i.e., if they only differ in only one variable. Failures are displayed as black squares. Color map ranges from white (high objective values) to red (low objective values). Our method `CBO-FRCHEI` presents the best optimization results by balancing well between exploration and exploitation.

Moreover, we investigate the effect of varying the batch size in Figure 13. We choose the batch size equal 2, since it provides a good trade-off between optimization performance and wall time. Throughout this paper, all the other experiments with `CBO-FRCHEI` and its variants have been conducted with batch size equal 2. It is also important to note that we do not employ more involving batch evaluation strategies, such as Kriging believer strategy (Ginsbourger et al., 2010), which can potentially increase even more the performance at the cost of increasing the wall time.

Finally, in Figure 14, we compare the performance of the proposed method with different kernels. Evidently, combining the degree 2 polynomial kernel with the discrete diffusion kernel as in Equation 13 provides the best results. Note that the wall time for the polynomial kernel scales almost linearly with the number of samples but is expensive due to the number of multiplications that arise due to the large number of features resulting from the second-order polynomial combination of inputs. The discrete diffusion kernel results in a lower wall time at low number of evaluations but increases at a higher rate. This is due to the complexity of calculating pair-wise delta functions between all inputs in the dataset. In terms of computational complexity, the final performance should be evaluated individually for the problem at hand and depends on the number of samples and the number of input dimensions.

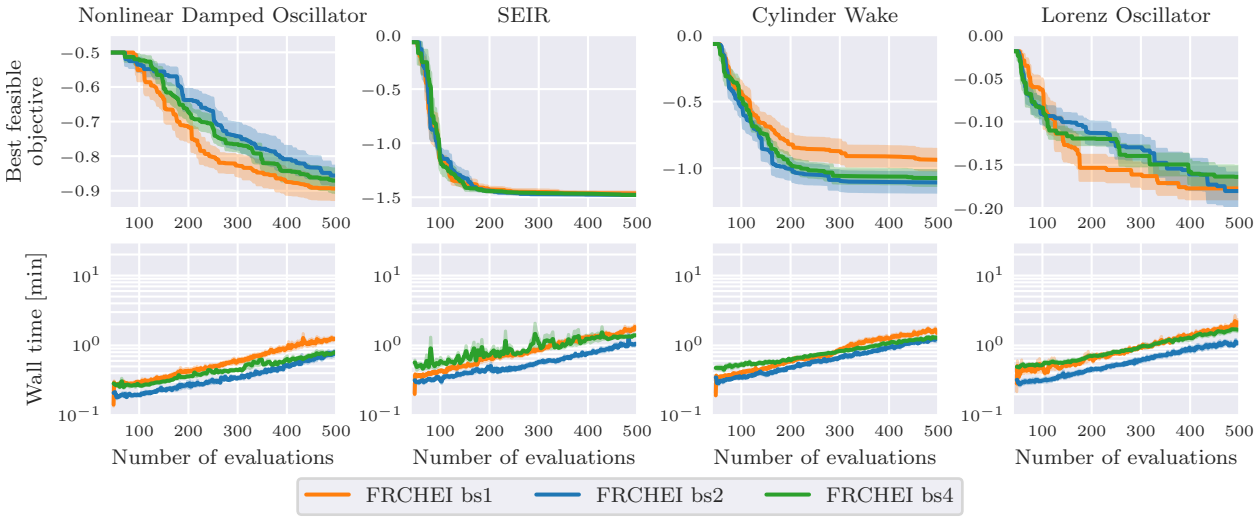


Figure 13: Optimization results for `CBO-FRCHEI` method for different batch sizes of 1, 2 and 4.

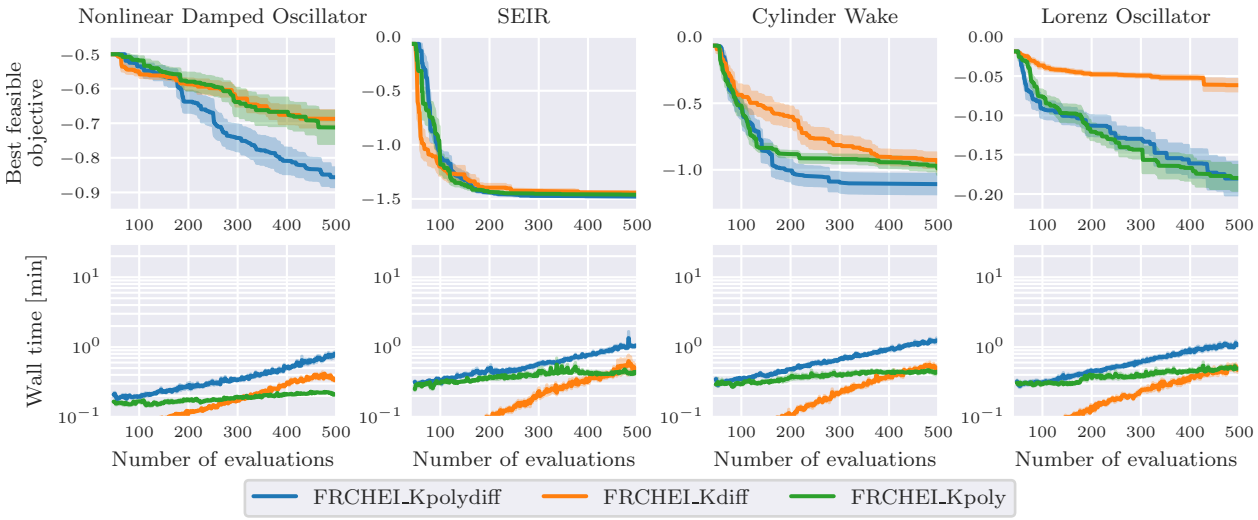


Figure 14: Optimization results for `CBO-FRCHEI` method with different kernel functions: Kpoly is the polynomial kernel, Kdiff the discrete diffusion kernel and Kpolydiff the combination of the two.