Computers and Chemical Engineering 194 (2025) 108976

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Bayesian optimization of gray-box process models using a modified upper confidence bound acquisition function

Joschka Winz *, Florian Fromme, Sebastian Engell

TU Dortmund University, Emil-Figge Str. 70, Dortmund, 44227, Germany

ARTICLE INFO

Keywords:
Bayesian optimization
Surrogate modeling
Gray-box modeling
Process optimization

ABSTRACT

Optimizing complex process models can be challenging due to the computation time required to solve the model equations. A popular technique is to replace difficult-to-evaluate submodels with surrogate models, creating a gray-box process model. Bayesian optimization (BO) is effective for global optimization with minimal function evaluations. However, existing extensions of BO to gray-box models rely on Monte Carlo (MC) sampling, which requires preselecting the number of MC samples, adding complexity. In this paper, we present a novel BO approach for gray-box process models that uses sensitivities instead of MC and can be used to exploit decoupled problems, where multiple submodels can be evaluated independently. The new approach is successfully applied to six benchmark test problems and to a realistic chemical process design problem. It is shown that the proposed methodology is more efficient than other methods and that exploiting the decoupled case additionally reduces the number of required submodel evaluations.

1. Approaches for optimization of gray-box process models

The steady-state optimization of chemical processes is an important task in the design of efficient chemical production plants. But the complexity and the implementation of the stationary model, e.g. in a flowsheet simulator, can result in long computation times and difficulties to use efficient optimization algorithms. The use of metaheuristics in combination with flowsheet simulators usually leads to a large number of calls to the simulator Janus and Engell (2021) and Henrich et al. (2008). An approach to overcome these difficulties that has become popular in recent years (Keßler et al., 2019a; Kaiser and Engell, 2023; Winz and Engell, 2024) is to replace some elements of the overall model that are causing long computation times (usually because of the need for iterative solutions) with surrogate models, e.g. neural networks. The surrogate models are embedded into the original flowsheet model. This approach is similar to hybrid modeling, where a white-box model is extended with a black-box model component that describes a relationship that is not fully understood (Sansana et al., 2021; Shah et al., 2022).

A hypothetical process model and a gray-box representation consisting of both surrogate and mechanistic submodels is shown in Fig. 1. The exemplary process consists of several units that are connected by streams. Each unit is described by a set of equations that describe the relationship between the states of the streams entering or leaving the unit and the design and operation degrees of freedom, e.g. the number of trays in a column and the reflux ratio. The state of a stream i is denoted by the vector $\boldsymbol{y}_i$ and the degrees of freedom of a unit j are denoted as $\boldsymbol{w}_j$. All quantities are assumed to be vectors.

With the introduction of surrogate models, the process model is converted into a gray-box process model as shown in the bottom part of Fig. 1. Here modules are introduced, shown as boxes with dashed outlines. A module can include one unit, a part of a unit, or multiple units. In the hypothetical process, the unit MIX1 is described by simple model equations $\boldsymbol{f}_{\mathrm{MIX1}}$ that determine the mixing of the input streams. For a simple unit, it is common that the model equations can be written in explicit form. R1 serves as an example of a unit that is described by an implicit model $\boldsymbol{f}_{\mathrm{R1}}$ that can be solved easily. Such unit models do not pose a major challenge for process optimization and can be included in a gray-box process model without modifications.

The units HE1, C1 and C2 serve as examples of units which are modeled in proprietary third-party software. In this case, the source code is typically not available, which can lead to two issues. First, derivative information is often not accessible, necessitating the use of finite-difference approximations, which are error-prone for highly nonlinear relationships and require many function evaluations. Second, a large computational overhead typically occurs when calling the third-party software to evaluate the submodel. Both aspects hinder an efficient process optimization. Therefore in the gray-box process model the model equations of these units are replaced by surrogate models.

* Corresponding author.
E-mail address: joschka.winz@tu-dortmund.de (J. Winz).

https://doi.org/10.1016/j.compchemeng.2024.108976
Received 30 August 2024; Received in revised form 10 November 2024; Accepted 5 December 2024
Available online 19 December 2024

For example, the heat exchanger model is replaced by the surrogate model $\hat{\boldsymbol{h}}_1$. This surrogate model is trained on data that was sampled from the corresponding original function $\boldsymbol{h}_{\text{orig},1}$. This original function results from the solution of the model equations of this unit $\boldsymbol{f}_{\text{HE1}}$. The symbol $\boldsymbol{h}$ is used to denote the vector of variables that are computed as the outputs of a surrogate model, in this case $\boldsymbol{h}_1 = \boldsymbol{y}_4$. The evaluation of the value of the output and of the first and second order derivative information is computationally cheap for surrogate models.

The units C1 and C2 are approximated by one surrogate model. In this case, the original function $\boldsymbol{h}_{\text{orig},3}$, combines the model equations of the blocks C1 and C2.

D1 and M1 are assumed to be units that are represented by models that are difficult to solve. With such unit operation models, process optimization becomes challenging because the equations of the unit model have to be solved many times. Therefore, it makes sense to replace these unit models by surrogate models as well. In Fig. 1 two possibilities are shown. The surrogate model can be set up to approximate the complete input–output relationship of a unit, as shown for M1. On the other hand, a surrogate model can be employed to approximate only a subset of the model equations of a unit. This is the case for D1, where the surrogate model describes the composition of two phases in equilibrium as a function of the state of the input stream 4. These compositions are then used to define the states of the streams 5 and 6 that leave the unit by a new set of model equations $\boldsymbol{f}_{\text{D1,new}}$. There are also cases in which a surrogate model is introduced for some of the model equations, but the remaining equations are still implicit. An example of this is the surrogate modeling of fugacity coefficients for pT-flash calculations (Nentwich et al., 2019).

A key aspect in the optimization on the basis of a gray-box process model like the one depicted in Fig. 1 is the accuracy of the surrogate models. If all surrogate models are sufficiently accurate, the overall process can be optimized using global optimization code or local optimizers with different initializations. But in reality the surrogate models are trained on data sets of limited size, and therefore the computed outputs deviate from those of the original models. As sampling the full space of the variables of interest for surrogate models with many inputs is computationally expensive and inefficient, it is desirable to improve the quality of the surrogate models specifically in the region around the optimum of the process, i.e., by adapted resampling and retraining. In this case, the sampling of the original models and the surrogate modeling has to be considered as part of the overall optimization procedure, and the interplay between optimization and re-sampling has to be taken into account. There has been a considerable amount of work on the optimization and retraining of gray-box models in the last two decades, and different approaches to address this sampling – surrogate modeling – optimization interplay have been proposed.

In an early approach, Alexandrov et al. (1998) presented a trust region based approach for surrogate model optimization by realizing that the trust region framework can be employed not just on quadratic approximations, but rather on a variety of approximation models that fulfill certain conditions. This method addresses the sampling – surrogate modeling – optimization interplay by evaluating the original objective function and its gradient in each iteration at the current iterate. The surrogate models are constructed to exactly reproduce this data, and the approximate model is optimized within a trust region. The results of the surrogate model and of the original model are compared and the trust region is adapted. Then, new approximate models are constructed with new samples and the process is iterated. An extension of this approach has been proposed by Eason and Biegler (2016) for the general case with constraints.

In both approaches, the uncertainty of the surrogate model is reflected by the trust region, the size of the trust region is reduced if the accuracy of the surrogate models is low.

A similar approach is presented by Kahrs and Marquardt (2007) that makes use of a validity domain instead of a trust region. In contrast to the trust region approaches, the validity domain is computed globally using all evaluated samples. The objective function is optimized within the validity domain. If the optimum lies on the boundary of the validity domain, the domain is expanded by collecting additional data points and retraining the surrogate model.

The previously mentioned methods are local optimization methods that generate a sequence of iterates that converge to a local optimum. Therefore, the surrogate models are only required to be accurate in the vicinity of this sequence of iterates. Another class of approaches use large sample sets to construct surrogate models that are accurate in the entire input domain of interest.

Many applications follow a one-shot workflow of first generating input–output data of the original submodel that is then replaced by a surrogate model. This data is used to fit surrogate models once. Before using the model in the optimization the accuracy is estimated, for example, using validation data sets. With the surrogate models embedded in the overall model, an objective function is optimized, and the prediction accuracy of the surrogate model is investigated at the computed optimum by comparing the results to the evaluation of the original model.

Applications of this approach include fermentation media optimization (Desai et al., 2008), polymerization and raw acetic anhydride production process optimization (Nascimento et al., 2000), optimization of a pressure swing adsorption process (Sant Anna et al., 2017), and optimization of a queuing system (Chambers and Mount-Campbell, 2002).

When the sampling – surrogate modeling – optimization procedure is only conducted once, special attention has been paid to finding the global optimum due to the fact that the surrogate models often show non-monotonic behavior. For instance, Keßler et al. (2017) globally optimize a chemical process with embedded neural networks as surrogate models using BARON (Tawarmalani and Sahinidis, 2005). They show different reformulations that drastically reduce the computation time. Among them is a piecewise linear reformulation of the neural network activation function. Schweidtmann and Mitsos (2019) showed how deterministic global optimization can be applied efficiently to an optimization problem with embedded neural networks by using McCormick relaxations in a reduced space.

One issue with using global optimization in a one-shot workflow is that there is no guarantee that the true global optimum is not missed due to locally inaccurate surrogate models. To overcome this issue, Davis and Ierapetritou (2007) optimized gray-box models by fitting global Kriging models to sampled input–output data to find multiple promising regions where the global optimum may be situated. Each region is then locally optimized using a response surface methodology (RSM) approach to determine the global optimum.

Caballero and Grossmann (2008) further extended this methodology. In their approach, the steps of sampling, surrogate modeling, and optimization are performed multiple times, each time shrinking the input space in which the data for building the surrogate models is sampled from the original functions around the predicted optimum. In this way, the predictive accuracy of the surrogate models increases locally with each iteration.

This idea has inspired a lot of related work. For instance, Boukouvala and Floudas (2017) presented a method for globally optimizing gray-box models using basis function models as surrogate models. In this way, surrogate models are generated that enable computationally efficient global optimization. The sizes of the input domains in which data is sampled for creating the surrogate models are iteratively reduced, similar to the work of Caballero and Grossmann (2008).

Henao and Maravelias (2011) developed a workflow to optimize gray-box flowsheet models where each unit in a flowsheet is approximated by an ANN model. The input spaces and the ANN models are updated if, after the optimization, the predicted optimum is at the bound of this domain or the problem was infeasible.

Keßler et al. (2019b) use surrogate models to optimize distillation column designs. They employ Kriging surrogate models, and the input

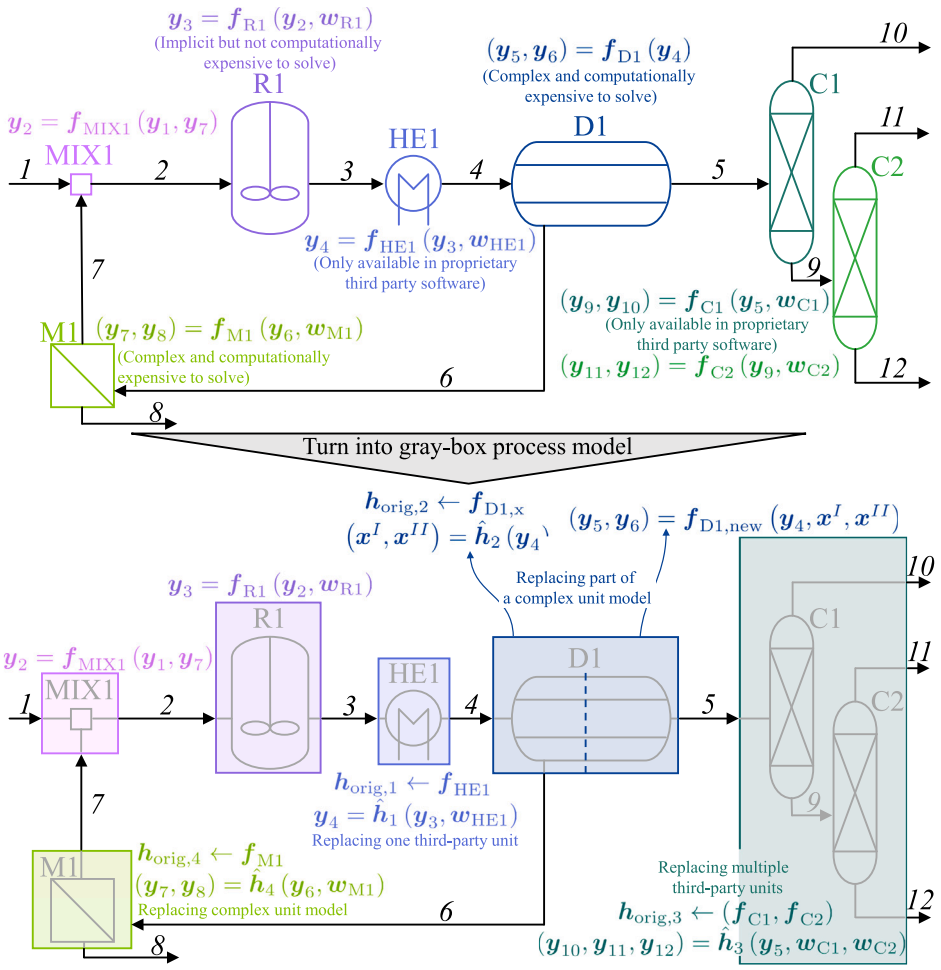


Fig. 1. Modular gray-box flowsheet. Streams are denoted in italic numbers, the names of units are written in non-italic font.

space of the surrogates is iteratively reduced. This work has been extended in Keßler et al. (2019a) by using both explicit and implicit surrogate models depending on the complexity of the system being modeled.

By iteratively shrinking the size of the input spaces of the surrogate models, the surrogate model accuracy is improved during the optimization routine. But, to ensure that the input space always contains the global optimum, a lot of data has to be collected in each iteration. This can lead to excessive sampling also in parts of the input space that are not likely to contain the global optimum.

Another approach to the global optimization of problems with gray-box models is Bayesian optimization (BO). A key concept in BO is to quantify how promising a point in the input space is in terms of optimization, based on the prediction and on the uncertainty of the prediction of one or more surrogate models. Repeatedly, the most promising point within the entire input space is evaluated and added to the training set. This approach has two advantages over the other methods presented. First, the prediction uncertainty of the surrogate models, which is an important metric for gray-box model optimization, is explicitly used to determine whether a point is likely to be the global optimum. Second, the global optimum is not unintentionally discarded by shrinking the bounds of the input space, since the entire input space is considered in each iteration.

2. Bayesian optimization for black-box and gray-box optimization problems

Originally, BO algorithms were proposed to tackle optimization problems of the form (Močkus, 1975; Jones et al., 1998)

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad f(\boldsymbol{x}). \tag{1}$$

Here f is the objective function. f depends on the decision variables $\boldsymbol{x}$ that are bounded to lie in a set $\mathcal{X}$. Typically, $\mathcal{X}$ is defined by box constraints.

A surrogate model $\hat{f}$ is constructed based on an initial set of samples X and Y, which were collected by evaluating the original function f on the input values X. Often in Bayesian optimization, a Gaussian process regression model is used as the surrogate model because the predicted variance is easily accessible.

$\hat{f}$ is then used to find a minimum of f with a minimum number of additional function evaluations. To this end, an acquisition function $\alpha(\boldsymbol{x})$ is optimized to locate a new sample which is added to the sample set, and the surrogate model is recomputed for the extended data set. This acquisition function quantifies how promising the evaluation of the original function f is at a location $\boldsymbol{x}$. Typical choices are the expected value of the improvement over the current best sample f_n^* (EI), the probability to improve the current best sample (PI), or the upper confidence bound of the surrogate model prediction (UCB), as shown in Eq. (2) (Forrester and Keane, 2009).

$$\begin{aligned}
&\text{EI:} && \alpha^{\mathrm{EI}}(\boldsymbol{x}) = \sigma_f(\boldsymbol{x})\varphi\left(\frac{f_n^* - \hat{f}(\boldsymbol{x})}{\sigma_f(\boldsymbol{x})}\right) + \left(f_n^* - \hat{f}(\boldsymbol{x})\right)\Phi\left(\frac{f_n^* - \hat{f}(\boldsymbol{x})}{\sigma_f(\boldsymbol{x})}\right)\\
&\text{PI:} && \alpha^{\mathrm{PI}}(\boldsymbol{x}) = \Phi\left(\frac{f_n^* - \hat{f}(\boldsymbol{x})}{\sigma_f(\boldsymbol{x})}\right)\\
&\text{UCB:} && \alpha^{\mathrm{UCB}}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \kappa\sigma_f(\boldsymbol{x})
\end{aligned} \tag{2}$$

Here Φ denotes the standard normal cumulative distribution function and φ denotes the standard normal probability density function. σ_f is the estimated (local) uncertainty in the surrogate model prediction $\hat{f}$.

The sample set is updated by evaluating f at the optimum of $\alpha(\boldsymbol{x})$, and the next iteration begins by constructing new surrogate models. The basic algorithm of Bayesian optimization is presented below as algorithm 1 (Frazier, 2018). The focus of the work presented here is the application of Bayesian optimization to general problem formulations in which the original functions that are replaced by surrogate models do not represent the objective function directly, but are implicitly contained in the objective function and in additional constraints. Therefore, algorithm 1 is written in a general form with possibly multiple surrogate models. The classic black-box case, see Eq. (1), is a special case of this formulation where there is only one submodel that directly represents the objective function.

Algorithm 1 General Bayesian optimization algorithm (Frazier, 2018)

Require: Maximum number of iterations N

1: Determine n_0 initial samples with a space-filling design, like Latin hypercube sampling, and evaluate the original input-output functions of the submodels to initialize the data sets $\mathcal{D}_i$
2: Set $n \leftarrow n_0$
3: **while** $n < N$ **do**
4: Update the posterior probability distributions, i.e. fit surrogate models to the data sets $\mathcal{D}_i$
5: Optimize the acquisition function to determine the most promising sample location $\boldsymbol{x}^*$
6: Evaluate all submodels at $\boldsymbol{x}^*$ and update the data sets $\mathcal{D}_i$
7: Increment n
8: **end while**
9: Return the best observed sample location

Even though there is some computational effort needed for constructing the new surrogate models and optimizing the acquisition function, this approach is often able to converge to the optimum of f in a small number of iterations if the dimension of $\boldsymbol{x}$ is not too large and the objective function is sufficiently smooth.

The basic approach described above does not include nonlinear constraints. Several researchers have independently developed extensions to the EI acquisition function to handle inequality constraints that are also expensive to evaluate (Schonlau et al., 1998; Gardner et al., 2014; Gelbart et al., 2014). The problem formulation is then

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}) \le 0. \end{aligned} \tag{3}$$

The proposed acquisition function α^{EIC} extends the expected improvement by including a factor that quantifies the probability of the new point being feasible.

$$\alpha^{\mathrm{EIC}}(\boldsymbol{x}) = \alpha^{\mathrm{EI}}(\boldsymbol{x}) \cdot \prod_{i=1}^{n_g} p_{g_i}(\boldsymbol{x}) \tag{4}$$

In this equation, the expected improvement α^{EI}, defined in Eq. (2), is multiplied with the probability that the constraints $\boldsymbol{g}$ are feasible at a location $\boldsymbol{x}$ given the current constraint surrogate models $\hat{g}_i$. This probability is denoted as $p_{g_i}(\boldsymbol{x})$. If the probability distribution of $\hat{g}_i$ is Gaussian, $p_{g_i}(\boldsymbol{x})$ can be computed with the Gaussian cumulative distribution function.

Hernández-Lobato et al. (2015) proposed a predictive entropy search acquisition function with constraints (PESC) for constrained BO. This function does not rely on having found a feasible point, which makes it suitable for highly constrained systems.

Priem et al. (2020) proposed to relax the constraints in constrained black-box BO to enlarge the explored region of the design space.

In many settings, the part of the model that is expensive to evaluate is not the objective function itself, but the model which computes some terms in the objective function. Astudillo and Frazier (2019) investigated the application of Bayesian optimization to composite functions of the form

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad f(\boldsymbol{h}_{\mathrm{orig}}(\boldsymbol{x})). \tag{5}$$

In this formulation, f denotes the objective function that can be evaluated quickly and for which first and second order derivative information with respect to $\boldsymbol{h}$ is available. Some submodels $\boldsymbol{h}_{\mathrm{orig}}(\boldsymbol{x})$ are expensive to evaluate and thus are replaced by surrogate models. An expected improvement for composite functions (EI-CF) acquisition function is proposed for this case in Astudillo and Frazier (2019). When working with composite functions, the probability distribution of the Gaussian process has to be propagated through the typically nonlinear equation f. Therefore, it is proposed to apply Monte Carlo sampling to estimate the expected improvement or the probability of improvement of f at a given input location. The EI-CF acquisition function from Astudillo and Frazier (2019) is shown in Eq. (6):

$$\alpha^{\text{EI-CF}}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M}\left\{f_n^* - f\left(\hat{\boldsymbol{h}}(\boldsymbol{x}) + \boldsymbol{\Sigma}_{\boldsymbol{h}}(\boldsymbol{x})\boldsymbol{Z}_m\right)\right\}^+. \tag{6}$$

In this equation, M denotes the number of Monte Carlo samples, $\boldsymbol{\Sigma}_{\boldsymbol{h}}$ is the lower Cholesky factor of the variance–covariance matrix of $\hat{\boldsymbol{h}}$. $\boldsymbol{Z}_m$ is drawn randomly from $\mathcal{N}(\boldsymbol{0}_{n_h}, \boldsymbol{I}_{n_h})$, where $\boldsymbol{0}_{n_h}$ and $\boldsymbol{I}_{n_h}$ are the n_h dimensional vector of zeros and the identity matrix. For each $m = 1 \dots M$, a separate random sample is drawn. $\{\cdot\}^+$ is a shorthand notation for $\max\{\cdot, 0\}$.

This approach was extended by Paulson and Lu (2022) in the COBALT framework. It differs from Astudillo and Frazier (2019) by including also composite constraint functions:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}, \boldsymbol{h}_{\mathrm{orig}}(\boldsymbol{A}\boldsymbol{x})) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{h}_{\mathrm{orig}}(\boldsymbol{A}\boldsymbol{x})) \le 0. \end{aligned} \tag{7}$$

Note that the inputs to the original submodels $\boldsymbol{h}_{\mathrm{orig}}$ are linear projections $\boldsymbol{A}$ of the decision variables $\boldsymbol{x}$, where $\boldsymbol{A}$ typically has a low rank. The acquisition takes the form of an extension of a Watson and Barnes (1995) acquisition function, where the constraints are relaxed analogous to the work of Priem et al. (2020). The optimization of the acquisition function takes the form in (8), denoted as mWB2-CF.

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & \alpha^{\text{mWB2-CF}} = f(\boldsymbol{x}, \hat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{x})) - s_n\alpha^{\text{EI-CF}}(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}, \hat{\boldsymbol{h}}(\boldsymbol{A}\boldsymbol{x})) + \tau_n\sqrt{\sigma_g^2(\boldsymbol{x})} \le 0 \end{aligned} \tag{8}$$

Here $\hat{\boldsymbol{h}}$ denotes the mean prediction of a Gaussian process model, s_n and τ_n are scaling factors and EI-CF$_n(\boldsymbol{x})$ is the expected improvement for composite functions as described above. The constraints are relaxed based on their predicted variance, which is linearly propagated from the variances of the Gaussian process models $\boldsymbol{\Sigma}_h^2$:

$$\sigma_g^2(\boldsymbol{x}) = \nabla_h \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\Sigma}_h^2(\boldsymbol{x})\nabla_h \boldsymbol{g}(\boldsymbol{x})^T. \tag{9}$$

In this equation, $\nabla_h \boldsymbol{g}(\boldsymbol{x})$ denotes the Jacobian matrix that contains the derivative of $\boldsymbol{g}$ with respect to $\boldsymbol{h}$.

Lu and Paulson (2023) developed another method called the constrained upper quantile bound acquisition function (CUQB). It uses Monte Carlo sampling to determine the quantile bounds of the objective function and the constraint functions. The optimization problem of the acquisition function is shown in Eq. (10) for minimization problems.

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & l_f(\boldsymbol{x}) \\ \text{s.t.} \quad & l_{g_i}(\boldsymbol{x}) \le 0 \quad \forall i \in [1, n_g] \end{aligned} \tag{10}$$

l_f is the lower quantile bound of the objective function $f(\boldsymbol{x}, \hat{\boldsymbol{h}}(\boldsymbol{x}))$ and is defined as the $\lceil pM \rceil$th highest Monte Carlo sample of f, where $\lceil \cdot \rceil$ denotes the ceiling operator. The samples are generated by drawing $\boldsymbol{Z}_m$ from $\mathcal{N}(\boldsymbol{0}_{n_h}, \boldsymbol{I}_{n_h})$ M times and computing $f\left(\boldsymbol{x}, \hat{\boldsymbol{h}}(\boldsymbol{x}) + \boldsymbol{\Sigma}_{\boldsymbol{h}}(\boldsymbol{x})\boldsymbol{Z}_m\right)$ for each $m = 1 \dots M$. The lower quantile bound l_{g_i} of the constraint $g_i(\boldsymbol{x}, \hat{\boldsymbol{h}}(\boldsymbol{x}))$ is defined analogously. p is the probability level for which the bounds are computed.

The methodologies for Bayesian optimization described above, EI-CF, COBALT, and CUQB, are all based on Monte Carlo sampling. This is a disadvantage because it requires to choose the number of samples, which is not trivial. When too few samples are chosen, the inaccurate propagation of the uncertainty can lead to misclassifying the feasibility of a location or how promising the uncertainty of a surrogate model is with respect to the improvement of the objective function. Thus, to be on the safe side, one should choose a large number of samples. However, this leads to long computation times for evaluating the objective function and the constraints when optimizing the acquisition function. In addition, the optimization problem of minimizing the acquisition function is stochastic if it is based on Monte Carlo sampling. Sample average approximation (SAA) can be used to deal with this problem (Paulson and Lu, 2022). A single random Monte Carlo sample Z_m is drawn in each BO iteration and used in each evaluation of the acquisition function. This makes the optimization problem deterministic, but introduces an optimality gap between the optimum of the SAA problem and the true unbiased optimum (Kleywegt et al., 2002). Stochastic gradient descent (SGD) can be used to avoid this by resampling the Monte Carlo samples each time the sampling function is called (Astudillo and Frazier, 2019). However, SGD does not support nonlinear constraints, and including them requires solving a nonconvex optimization problem for each evaluation of the acquisition function (Paulson and Lu, 2022).

In our earlier work, we proposed to extend the upper confidence bound acquisition function to gray-box models (Winz and Engell, 2021; Winz et al., 2021). The main idea of this approach is to estimate an optimistic objective function value that quantifies the value of the uncertainty in the surrogate models by using sensitivity information. This approach, in contrast to the general methods presented above, does not require any Monte Carlo sampling.

However, in the form presented in Winz and Engell (2021), Winz et al. (2021), the optimization of the acquisition function requires two iterative layers, which is cumbersome, and the sensitivity coefficients were only approximated based on Lagrange multipliers.

In this work we fill the literature gap of an approach to apply BO to gray-box optimization problems without the need of several iterative layers or Monte Carlo sampling. We develop the approach in Winz and Engell (2021), Winz et al. (2021) further by proposing a novel way of determining the sensitivity of the objective function with respect to the uncertainty of the surrogate model, and we show how this acquisition function can be used in a decoupled setting where some or all of the submodels can be evaluated separately. The rest of this paper is structured as follows: The methodology for optimization of structured models that contain a number of surrogate models, which is the main contribution of this work is presented in Section 3.

The proposed approach is tested in two settings. First, it is applied on a set of six well-known test problems that cover a variety of problem types, both constrained and unconstrained. The results show that both the proposed approach and the COBALT approach lead to comparably fast convergence on all test problems. The results on the test problems are reported in Section 4. Second, a challenging real-world case, a flowsheet of a hydroformylation process in a TMS system where the phases of the process change in the different unit operations is optimized in Section 5. The performance of the new approach is shown to be superior compared to the COBALT approach.

A conclusion and an outlook on further research directions are given in Section 6.

3. A new upper confidence bound acquisition function to constrained gray-box model optimization

3.1. Optimization problem formulation

As discussed above, the standard UCB acquisition function α^{UCB} is defined as (Srinivas et al., 2012):

$$\alpha^{\mathrm{UCB}}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \kappa\sigma_f(\boldsymbol{x}). \tag{11}$$

In this definition, $\hat{f}$ denotes the surrogate model prediction of the cost function. To this term a second term that represents the optimism is added, which is composed of the estimated prediction error of the cost function by the surrogate model, σ_f, multiplied by κ. For minimization problems, the scaling has to be non-positive, $\kappa \leq 0$.

We here extend this acquisition function to nonlinear problems with several submodels and constraints of the structure

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}, \boldsymbol{H}_{\mathrm{orig}}(\boldsymbol{x})) \\ \text{s.t.} \quad & g_{eq,i}(\boldsymbol{x}, \boldsymbol{H}_{\mathrm{orig}}(\boldsymbol{x})) = 0,\ i = 1\dots, n_{g_{eq}} \\ & g_{ineq,i}(\boldsymbol{x}, \boldsymbol{H}_{\mathrm{orig}}(\boldsymbol{x})) \leq 0,\ i = 1\dots, n_{g_{ineq}}. \end{aligned} \tag{12}$$

In this formulation, $\boldsymbol{H}_{\mathrm{orig}}$ denotes the vector of all outputs of the original submodels that are approximated by surrogate models. These submodels influence the cost function f and the equality and inequality constraints $g_{eq,i}$ and $g_{ineq,i}$. The vector $\boldsymbol{H}_{\mathrm{orig}}$ consists of n_h individual submodels $\boldsymbol{h}_{\mathrm{orig},i}$ as

$$\boldsymbol{H}_{\mathrm{orig}}(\boldsymbol{x}) = \left(\boldsymbol{h}_{\mathrm{orig},1}(\boldsymbol{z}_1(\boldsymbol{x}))^T, \dots, \boldsymbol{h}_{\mathrm{orig},n_h}(\boldsymbol{z}_{n_h}(\boldsymbol{x}))^T\right)^T. \tag{13}$$

Here $\boldsymbol{h}_{\mathrm{orig},i}(\boldsymbol{z}_i) \in \mathbb{R}^{n_{Hi}}$ denotes the vector of outputs of the ith submodel. It is assumed that these submodels can be evaluated separately and depend on individual sets of input variables $\boldsymbol{z}_i(\boldsymbol{x}) \in \mathbb{R}^{n_{z_i}}$. The dependency of the cost and constraint functions on $\boldsymbol{x}$ represents the remaining, non-approximated part of the overall model.

3.2. The gray-box upper confidence bound acquisition function

To apply the Bayesian optimization idea to the constrained optimization problem (12), the following optimization problem is formulated using a modified acquisition function which is solved in each iteration:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & a^{\text{GB-UCB}} = f(\boldsymbol{x}, \hat{\boldsymbol{H}}(\boldsymbol{x})) + \kappa \sum_{i=1}^{n_h}\sum_{j=1}^{n_{Hi}} \left|s_{i,j}(\boldsymbol{z}_i(\boldsymbol{x}))\right| \sigma_{i,j}(\boldsymbol{z}_i(\boldsymbol{x})) \\ \text{s.t.} \quad & g_{eq,i}(\boldsymbol{x}, \hat{\boldsymbol{H}}(\boldsymbol{x})) = 0,\ i = 1\dots, n_{g_{eq}} \\ & g_{ineq,i}(\boldsymbol{x}, \hat{\boldsymbol{H}}(\boldsymbol{x})) \leq 0,\ i = 1\dots, n_{g_{ineq}}. \end{aligned} \tag{14}$$

In this function, the submodels are described by the vector of surrogate model outputs $\hat{\boldsymbol{H}}$, which contains the surrogate model outputs of each individual surrogate model according to

$$\hat{\boldsymbol{H}}(\boldsymbol{x}) = \left(\hat{\boldsymbol{h}}_1(\boldsymbol{z}_1(\boldsymbol{x}))^T, \dots, \hat{\boldsymbol{h}}_{n_h}(\boldsymbol{z}_{n_h}(\boldsymbol{x}))^T\right)^T. \tag{15}$$

The uncertainty of the jth output of the ith submodel prediction is denoted as $\sigma_{i,j}$. It is multiplied with the sensitivity of the objective function to the error of this surrogate model, $s_{i,j}$.

The key idea thus is to add a term into the cost function that describes how much the objective function could be reduced if the surrogate model error turned out in a favorable way. This is analog to the traditional UCB-case, see Eq. (11).

The sensitivity of the objective function to each surrogate model output error is needed to quantify the optimism that is generated by the uncertainty of the surrogate model. This sensitivity, denoted as $s_{i,j}$, plays a key role in the algorithm, and computing it is not always trivial. In our previous work (Winz and Engell, 2021; Winz et al., 2021) we defined the sensitivity via Lagrange multipliers. However, this requires an additional inner loop that repeatedly solves the optimization problem in line 5 of algorithm 1 for each additional sample. As this is computationally expensive, a different approach is presented in this work.

First, for the unconstrained case ($n_{g_{eq}} = n_{g_{ineq}} = 0$) and assuming that the dependency of f on $\boldsymbol{H}$ is known in a closed functional form, the sensitivity can be readily computed by evaluating the derivative of the objective function with respect to the surrogate model prediction error $\Delta\boldsymbol{H}$.

$$\boldsymbol{s}_{unconstrained} = \left.\frac{df(\boldsymbol{x}, \hat{\boldsymbol{H}}(\boldsymbol{x}) + \Delta\boldsymbol{H})}{d\Delta\boldsymbol{H}}\right|_{\Delta\boldsymbol{H}=0} \tag{16}$$

$\boldsymbol{s}$ denotes the vector of the sensitivities of all submodel outputs as

$$\boldsymbol{s} = \left(s_{1,1}, \dots, s_{1,n_{H1}}, \dots, s_{n_h,1}, \dots, s_{n_h,n_{Hn_h}} \right)^T. \tag{17}$$

If constraints limit the feasible region of the input space, the sensitivity is not given by this expression. This can easily be seen by considering a problem with equality constraints only:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & h_{\text{orig}}(\boldsymbol{x}) = 0. \end{aligned} \tag{18}$$

In this case, even if $\boldsymbol{s}_{unconstrained} = 0$, the surrogate model $\hat{h}$ that approximates h_{orig} has an influence on the optimal $\boldsymbol{x}$ value. To quantify this fact, the influence of a perturbation in $\boldsymbol{H}$, denoted as $\boldsymbol{\Delta H}$, on $\boldsymbol{x}$ through the active constraints is included. If $\boldsymbol{x_0}$ satisfies the active constraints $g_{active,i}$ without a perturbation $\boldsymbol{\Delta H} = 0$, then the feasible value of $\boldsymbol{x}$ in the presence of the perturbation $\boldsymbol{\Delta H}$ that is closest to $\boldsymbol{x_0}$ is the solution of the following optimization problem, assuming that all active/inactive constraints remain active/inactive:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & (x - x_0)^T \boldsymbol{W} (x - x_0) \\ \text{s.t.} \quad & g_{active,i}(\boldsymbol{x}, \boldsymbol{H}(\boldsymbol{x}) + \boldsymbol{\Delta H}) = 0, \; i = 1 \dots, n_{g_{active}}, \end{aligned} \tag{19}$$

with a diagonal weighting matrix $\boldsymbol{W}$ with positive nonzero entries on the diagonal. When the original optimization problem is well scaled, $\boldsymbol{W}$ can be chosen as an identity matrix. Linearizing the constraints yields a quadratic programming problem:

$$\begin{aligned} \min_{(\boldsymbol{x}-\boldsymbol{x}_0)} \quad & (\boldsymbol{x} - \boldsymbol{x}_0)^T \boldsymbol{W} (\boldsymbol{x} - \boldsymbol{x}_0) \\ \text{s.t.} \quad & \boldsymbol{g}_{active}(\boldsymbol{x}_0, \boldsymbol{H}_0) + \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{\Delta H}} \boldsymbol{\Delta H} = 0. \end{aligned} \tag{20}$$

The solution of this problem can be written in closed form:

$$\begin{aligned} (\boldsymbol{x} - \boldsymbol{x}_0)^* = & - \boldsymbol{W}^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}}^T \left(\frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}} \boldsymbol{W}^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}}^T \right)^{-1} \boldsymbol{g}_{active}(\boldsymbol{x}_0, \boldsymbol{H}_0) \\ & \underbrace{- \boldsymbol{W}^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}}^T \left(\frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}} \boldsymbol{W}^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}}^T \right)^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{\Delta H}} \boldsymbol{\Delta H}}_{:= s_{hx}} \end{aligned} \tag{21}$$

$\boldsymbol{s}_{hx}$ is the sensitivity of $\boldsymbol{x}$ w.r.t. $\boldsymbol{\Delta H}$ due to the active constraints. This change in $\boldsymbol{x}$ has an influence on the objective function f, so all in all the sensitivity is defined as

$$\boldsymbol{s} = \frac{df}{d\boldsymbol{\Delta H}} - \frac{df}{d\boldsymbol{x}} \boldsymbol{W}^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}}^T \left(\underbrace{\frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}} \boldsymbol{W}^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{x}}^T}_{B} \right)^{-1} \frac{d\boldsymbol{g}_{active}}{d\boldsymbol{\Delta H}}. \tag{22}$$

Note, that the matrix B is only invertible if all active constraints depend on $\boldsymbol{x}$ and the gradients of the active constraints are linearly independent. The former criterion excludes constant constraints which are handled separately in the optimization, the latter is a standard criterion in nonlinear optimization. Imposing it in all iterations may make it necessary to remove linearly dependent constraints for the calculation of $\boldsymbol{s}$. In addition, the condition number of B can become large if the constraints are highly nonlinear or stiff. Possible ways to deal with this problem include reformulating the constraints or regularizing the matrix B, for example by adding a constant to the diagonal to reduce the condition number. In this case the computation is more stable, but the values of the sensitivities are slightly modified.

The evaluation of the sensitivities, in particular the computation of the derivatives, is done using CasADi (Andersson et al., 2019). This can be computationally expensive due to matrix inversion, which leads to a computational effort that scales rapidly with matrix size. This is a problem for systems with many active constraints. In our experience, this effect did not cause problems for the application for flowsheet models of moderate size.

To find inputs $\boldsymbol{x}$ that are promising to evaluate, the optimization problem (14) is solved. The workflow of the outer Bayesian optimization loop and the inner acquisition function optimization loop is visualized in Fig. 2.

The outer loop follows the steps shown in algorithm 1. The inner loop optimizes the acquisition function (14) to determine the location of the next sample. Note that for each evaluation of the acquisition function, the sensitivity is computed using the current set of active constraints. Therefore, the acquisition function is not continuous in regions where the set of active constraints changes, which may pose problems for the numerical solver. From our experience using IPOPT (Wächter and Biegler, 2006) this was not an issue.

This approach assumes that the evaluation of the original functions is deterministic, i.e., that there is no aleatoric (inherent) uncertainty. The idea of the GB-UCB acquisition function is to quantify how much better an objective function value could be if the epistemic uncertainty of the surrogate model prediction is realized in a favorable way. The focus is on the epistemic uncertainty because it can be reduced by further sampling at this point. In general, it is not trivial to distinguish between epistemic and aleatoric uncertainty, but this would be necessary to evaluate the optimism based only on epistemic uncertainty.

Often, Gaussian process models are used as surrogate models due to the fact that the estimated prediction error is readily available. The formulation in Eq. (14) includes also cases in which a surrogate model represents a discrete output space, i.e. a classification surrogate model. In this case, Gaussian process models cannot be used efficiently because the prediction of the probability of an input belonging to a specific class cannot be written in closed form due to the fact that the likelihood for the classification task is not conjugate to the Gaussian prior. In this case, we recommend the use of support vector machines or neural network models. The classification surrogate model prediction error can be estimated using specific techniques like Bayesian neural networks (Arbel et al., 2023) or statistical techniques as the bootstrap or the jackknife variance.

3.3. Exploiting the model structure

In many cases, there is not only one, but there are several submodels included in the optimization problem that can be evaluated separately. This can be exploited to reduce the total number and the computational effort of original function evaluations necessary for convergence.

For the constrained black-box case Gelbart et al. (2014) propose a two step approach. First, an acquisition function is optimized to determine a promising sample location. Afterwards, a criterion based on entropy search is used to determine the submodel that is evaluated. Hernández-Lobato et al. (2015) stated that this approach does not take advantage of all available information and proposed an acquisition function that is additive over the submodels. In this additive acquisition function, the contribution of each submodel can be easily quantified and the submodel with the highest contribution to the entropy based acquisition function can be evaluated.

The GB-UCB acquisition function is also additive in nature. Thus, to quantify how important it is to evaluate one surrogate model compared to another one, the optimism generated by the submodel i is defined as

$$o_i(\boldsymbol{z}_i) = \kappa \sum_{j=1}^{n_{Hi}} \left| s_{i,j}(\boldsymbol{z}_i) \right| \sigma_{i,j}(\boldsymbol{z}_i). \tag{23}$$

This can be used as a criterion for deciding which submodel to evaluate to generate more training data. A possible approach is to evaluate a submodel i with probability p_i as

$$p_i = \frac{o_i}{\sum_{k=1}^{n_h} o_k}. \tag{24}$$

This approach has the advantage that each submodel has a chance to be evaluated, such that the sampling is not restricted to one or a few submodels, but still submodels with a large contribution to the optimism are preferred.

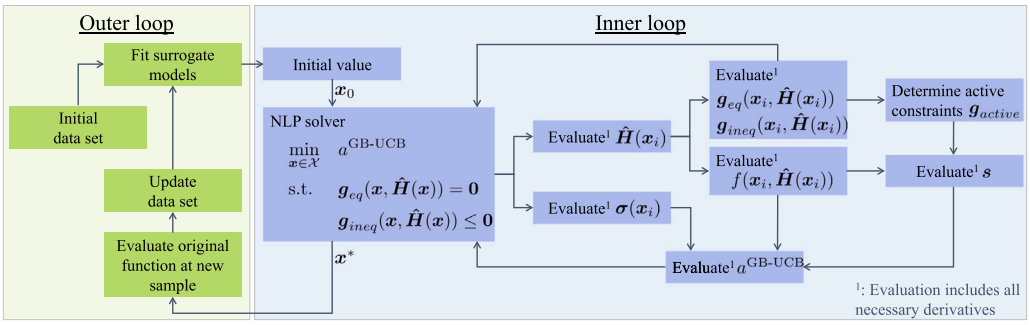


Fig. 2. Workflow of the Bayesian optimization algorithm with the GB-UCB acquisition function.

3.4. Visualization of the GB-UCB acquisition function

To illustrate the different acquisition functions a toy problem of the following form is considered:

$$\min_{x\in[0,10]} \quad f(x,h(z=x)) = (h-0.5)(\cos(x)+0.5)+0.1x \quad \text{with} \quad h(x)=\sin(x). \tag{25}$$

The true optimum of the function is -0.909 at $x = 5.461$. 8 samples were drawn from $h(x)$ to construct the surrogate model $\hat{h}(x)$. The fit is shown in Fig. 3, together with the resulting objective function values that result from the approximation of the surrogate model. The surrogate model is a Gaussian process regression model. The values of different acquisition functions are shown for comparison.

From Fig. 3 it can be seen that the predictions of the surrogate model are accurate for values of x up to roughly 4.5. For higher values of x, the estimated variance is significant. This is also reflected in the approximation of the objective function. The estimated variance is propagated through the nonlinear function f using Monte Carlo samples drawn from $\mathcal{N}(\hat{h}(x),\sigma_h^2(x))$. Still the true objective function is well represented for low values of x.

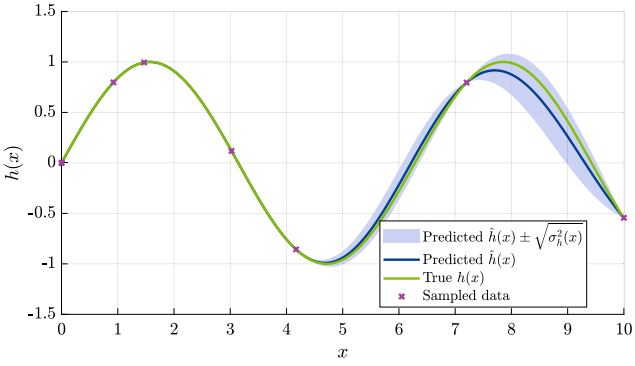
In the third plot the normalized acquisition functions EI-CF (Astudillo and Frazier, 2019), mWB2-CF (Paulson and Lu, 2022), and GB-UCB from this work are shown together with the normalized predicted objective function value. Comparing the different approaches, EI-CF takes on values near 0 over a large range of x values. In fact, if no Monte Carlo sample leads to improvement, the EI-CF acquisition function is exactly 0 with a gradient of 0. This can make optimizing this acquisition function challenging (Paulson and Lu, 2022). The mWB2-CF and GB-UCB acquisition functions do not share this feature since they contain the predicted objective function. An interesting difference between mWB2-CF and GB-UCB is that even in regions far from the optimum, here for example in $x \in [7,10]$, GB-UCB gives incentives larger than $f(\hat{\boldsymbol{h}}(\boldsymbol{x}))$, though this does not influence the choice of the next sample in this case.

All in all, it can be seen that for all acquisition functions the same region around 5.5 is correctly identified as being promising for further sampling.

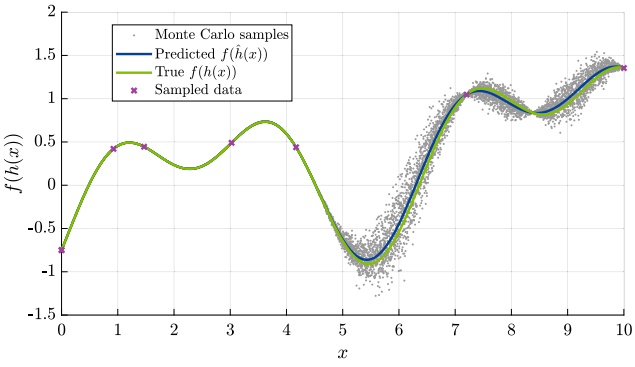
4. Results for test problems

In this section, the EI-CF, GB-UCB, and mWB2-CF acquisition functions are tested on six test problems taken from Paulson and Lu (2022). The exact definitions of the optimization problems can be found in Paulson and Lu (2022).

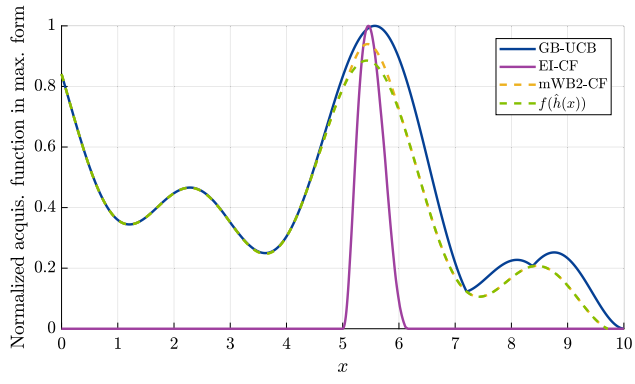
The set of test problems includes three nonlinearly constrained and three box-constrained problems. The characteristics of the test problems are shown in Table 1. n_g and $n_{g,\text{active}}$ denote the numbers of nonlinear constraints and of nonlinear constraints that are active at the optimum. n_h, n_z, n_x indicate the number of submodels, of inputs of the submodels, and of decision variables, the number of local minima is represented by $n_{\text{local minima}}$.



(a) Gaussian process model predictions



(b) Approximation of objective function



(c) Comparison of acquisition functions

Fig. 3. Visualization of the GB-UCB acquisition function and comparison to other acquisition functions for a simple optimization problem.

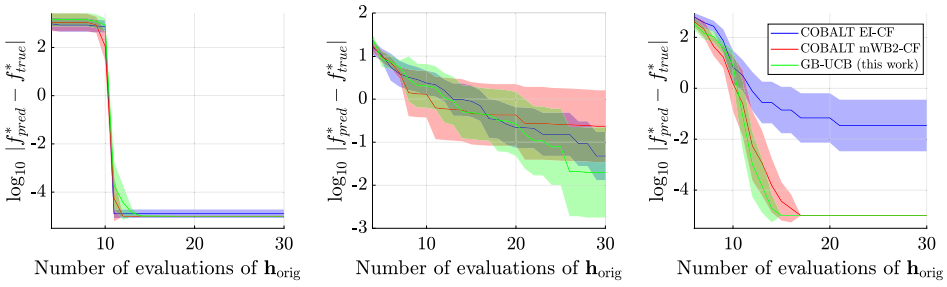


Fig. 4. Results for box-constrained test problems using different acquisition functions.

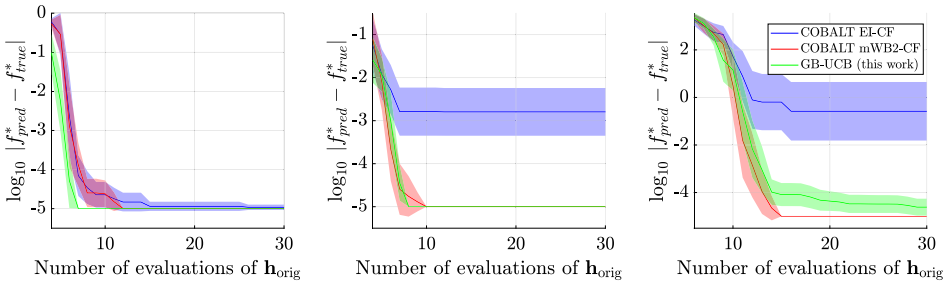


Fig. 5. Results for nonlinearly constrained test problems using different algorithms.

Table 1
Characteristics of test problems from Paulson and Lu (2022).

Name	n_g	$n_{g,\text{active}}$	n_h	n_z	n_x	$n_{\text{local minima}}$	Reference
Goldstein–Price	0	0	2	2	2	4	Dixon and Szego (1978)
Rastrigin	0	0	1	1	3	1331	Rastrigin (1974)
Rosenbrock	0	0	4	4	6	2	Rosenbrock (1960)
Toy-Hydrology	2	1	1	1	2	3	Gramacy et al. (2015)
Rosen–Suzuki	3	2	2	2	4	1	Hock and Schittkowski (1980)
Colville	6	2	4	4	5	1	Rijckaert and Martens (1978)

As it can be seen, the benchmark problems are quite diverse. There are problems with and without nonlinear constraints. Also the number of local minima is quite different, with the Rastrigin function exhibiting a large number of local minima. For the nonlinearly constrained problems, both mono- and multimodal test problems are included. In all nonlinearly constrained problems, some but not all constraints are active at the global optimum.

The results for the box-constrained problems are shown in Fig. 4. The shown results were generated by applying algorithm 1 using different acquisition functions and constraint handling techniques. The initial sample set is taken as $n_0 = \max\{3, n_z + 1\}$ points distributed using Latin hypercube sampling as in Paulson and Lu (2022). The experiments were repeated 20 times. The Gaussian processes were computed using the Statistics and Machine Learning Toolbox in MATLAB. The design parameter κ is set to a fixed value of -1.96, which is used for all results in this work. For the COBALT approach 100 Monte Carlo samples are computed for each evaluation of the acquisition function. The error metric is taken as the logarithm of the difference between the best feasible objective function value that has been observed so far and the true global optimum. The methodology of optimizing the acquisition function is taken from Paulson and Lu (2022): First, a global screening is conducted by evaluating 10,000 samples distributed evenly in the input space using Latin hypercube sampling. The sample that has the best objective function value and a constraint violation less than 10^{-4} is then used as the initial value for a local search using IPOPT (Wächter and Biegler, 2006).

It can be seen that the three acquisition functions lead to slightly different results in these test cases. The EI-CF acquisition function shows higher errors for the Rosenbrock test problem, which is probably due to the fact that this acquisition function has a zero gradient for a large portion of the input space (see Fig. 3(c)), which makes it difficult to identify the global optimum of the acquisition function. For the shown tests GB-UCB performs at least as good as COBALT using either the EI-CF or mWB2-CF acquisition function. For the Rosenbrock problem slightly faster convergence can be observed.

The results for the nonlinearly constrained test problems are shown in Fig. 5. A proposed optimum of each iteration is only accepted if it is feasible.

While EI-CF results in higher errors, GB-UCB and mWB2-CF show similar results. One difference can be observed for the first iterations. Here the value of GB-UCB is lower for the Toy-Hydrology and Rosen–Suzuki test problems. This can be explained by the fact that for GB-UCB the predicted constraints are strictly enforced, while they are relaxed in the COBALT approach. This leads to the initial samples having a higher probability of not being feasible on the true constraint function. While the differences are small, GB-UCB converges faster for the Toy-Hydrology problem, slightly faster for the Rosen–Suzuki problem, and slightly slower for the Colville problem. In general, both the GB-UCB and the COBALT mWB2-CF approach lead to satisfactory results on all six test problems, while GB-UCB has the advantage that it does not rely on Monte Carlo sampling.

5. Optimization of a process for the hydroformylation of 1-dodecene

5.1. Process model

In this section, the application of the different Bayesian optimization algorithms with different acquisition functions to a realistic process optimization problem is investigated. The case study chosen is the process of the hydroformylation of 1-dodecene in a thermomorphic solvent system. This case study has also been considered in previous

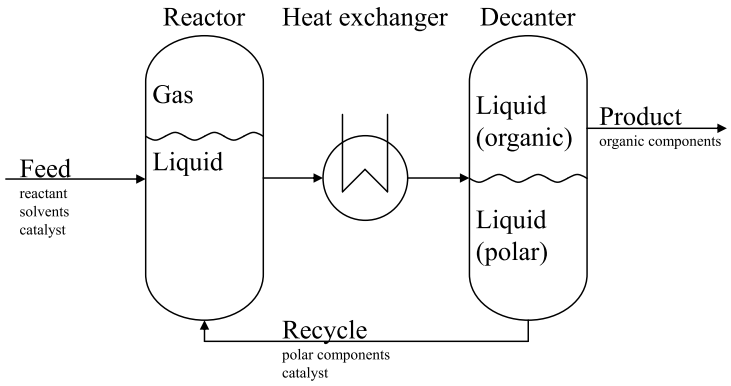


Fig. 6. Flowsheet of considered process.

work (Nentwich et al., 2019; Winz et al., 2021; Nentwich, 2022), where more details about the process can be found.

In this process, a hydroformylation reaction is performed in a thermomorphic solvent system (Zagajewski et al., 2014). A simplified flowsheet is shown in Fig. 6.

This process is set up in such a way that the losses of the expensive rhodium-based catalyst are minimized. This is achieved by designing the solvent system such that the liquid phase in the reactor is homogeneous, i.e. it does not show any liquid–liquid phase separation for the concentrations that occur in the reactor. When the mixture is cooled down in the heat exchanger, the liquid phase becomes unstable and separates into a polar phase containing the catalyst and an organic phase. The polar phase is recycled back to the reactor. The organic phase contains the product, 1-tridecanal, and unconverted dodecene, and is sent downstream for further purification and recycling of the unused feed.

In this process, there are two thermodynamic equilibria that are important for modeling and optimization. The first is the gas–liquid-equilibrium (GLE) in the reactor, which describes the solubility of the synthesis gas (CO and H_2) in the liquid mixture. The second thermodynamic equilibrium is the liquid–liquid-equilibrium (LLE) in the decanter. It describes the stability of the liquid in the decanter and the concentrations of the two phases if the content of the decanter separates into two phases. As these equilibria have a strong influence on the process, they have to be modeled accurately. The perturbed-chain statistical-associating-fluid-theory (PC-SAFT) equation of state can be used for this purpose as shown by Schäfer et al. (2012), Vogelpohl et al. (2014), Merchan and Wozny (2016). However, evaluating the equilibria with PC-SAFT is computationally expensive, which makes it difficult to optimize the process when PC-SAFT models are included. This is a common problem and much work has been dedicated to computing surrogate models to approximate equilibrium calculations of modern thermodynamic models in general (see e.g. Keßler et al. (2017), Iftakher et al. (2022)) and for this specific process (Nentwich and Engell, 2019; Kunde et al., 2019; Kaiser and Engell, 2023).

In this process, there are seven quantities that are described by surrogate models: the two solubilities of the synthesis gas c_{CO}, c_{H_2}, the Boolean variable of the decanter feed stability PS and the phase distribution coefficients that describe the separation in the decanter, κ_i, where $i = 1 \dots 4$ corresponds to the liquid components decane, DMF, 1-dodecene and 1-tridecanal. The phase distribution coefficients κ are defined as

$$\kappa_i = \frac{x_i^I}{x_i^D} \frac{x_i^D - x_i^{II}}{x_i^I - x_i^{II}}, \tag{26}$$

where $x_i^{I/II}$ denotes the molar fraction of the ith component in phase I or II, the molar fractions of the feed into the decanter are denoted by x_i^D. These phase distribution coefficients are used as outputs of the surrogate models instead of the molar fractions of both phases for two reasons. On the one hand the set of output variables is smaller since there are only half as many phase distribution coefficients as molar fractions in a two-phase system, and also each combination of phase distribution coefficients in $[0, 1]$ maps to a valid set of molar fractions. So no additional constraints are needed to enforce the condition that the sum of the molar fractions is equal to one or to maintain the mass balance.

Table 2
Surrogate models.

Submodel	Symbol	Input quantities $z(x)$	Range	Surrogate model
GLE	c_{CO}	$\{T^R, p^R, x_1^R, x_2^R, x_3^R, y_{CO}^R\}$	$[0, \infty)$	Gaussian process
	c_{H_2}		$[0, \infty)$	Gaussian process
LLE	PS	$\{T^D, x_1^D, x_2^D, x_3^D\}$	$\{0, 1\}$	Neural network
	κ_1		$[0, 1]$	Gaussian process
	κ_2		$[0, 1]$	Gaussian process
	κ_3		$[0, 1]$	Gaussian process
	κ_4		$[0, 1]$	Gaussian process

Table 3
Degrees of freedom for process optimization.

Symbol	Range	Meaning
$\dot{n}_1^{Feed}$	$[6, 20]$ $\frac{kmol}{h}$	Molar flow rate of decane in the feed
$\dot{n}_2^{Feed}$	$[0, 20]$ $\frac{kmol}{h}$	Molar flow rate of DMF in the feed
$\dot{n}_3^{Feed}$	$[0, 20]$ $\frac{kmol}{h}$	Molar flow rate of 1-dodecene in the feed
$\dot{n}_{cat}^{Feed}$	$[0, 1]$ $\frac{kmol}{h}$	Molar flow rate of the catalyst in the feed
T^R	$[90, 115]$ °C	Reactor temperature
T^D	$[5, 30]$ °C	Decanter temperature
p^R	$[10, 50]$ bar	Reactor pressure
V^R	$[1, 30]$ m^3	Reactor volume
y_{CO}^R	$[0,1]$	Gas molar fraction of CO in the reactor

The inputs of the LLE and GLE submodels are different. Table 2 shows an overview of the individual surrogate models.

In this table T, p and x with a superscript R or T denote the temperature, pressure and molar fraction in the reactor or decanter unit. E.g. y_{CO}^R is the molar fraction of CO in the binary gas phase in the reactor.

All continuous quantities are modeled as Gaussian processes, while the discrete variable PS is modeled using a classification neural network. The prediction variance of the neural network is approximated by the jackknife variance, see Nentwich and Engell (2019). Note that both submodels can be evaluated separately. Thus, this is a decoupled problem.

To investigate the performance of the Bayesian optimization algorithms, the production cost per ton of 1-tridecanal is minimized by adjusting the process degrees of freedom that are listed in Table 3.

Besides these degrees of freedom, there are eight dependent variables that are needed to define the implicit kinetic model of the reactor which is taken from Kiedorf et al. (2014).

The total cost C_{total} per product $\dot{m}_4$ is defined as the sum of three contributions: the investment costs C_{CAPEX}, depreciated over ten years, and the annual cost of raw materials C_{raw} and utilities C_{utility}. The annual production is assumed to be 10,000 tons.

$$C_{\text{total}} = \frac{1}{\dot{m}_4}\left(C_{\text{raw}} + C_{\text{utility}} + \frac{1}{10}C_{\text{CAPEX}}\right) \tag{27}$$

The parameters for quantifying the individual cost terms are taken from Turton et al. (2008) and Hentschel et al. (2014).

The mole balances around the reactor lead to nonlinear constraints of the process model. The feed of the decanter is constrained to be unstable, as predicted by the neural network. In addition, the yearly production imposes an equality constraint. Therefore, this process model contains not only inequality, but also equality constraints, of which not all are influenced by a surrogate model output.

5.2. Optimization results

Before applying the Bayesian optimization algorithms, initial data sets of 30 points of the GLE and LLE output quantities are generated. Both BO algorithms are applied as described in Section 4, except for the screening, which has been slightly adapted. In this optimization problem, there are equality constraints that are statistically never exactly satisfied with random screening points. Therefore, the screening is adjusted so that the initial point is chosen as the sample with the lowest objective function value, with an added penalty of the sum of the squared constraint violations scaled by 10^2.

For the COBALT algorithm, the value of τ in Eq. (8) is defined by the heuristic shown in Eq. (28) taken from Paulson and Lu (2022):

$$\tau_n = -3\left(1 - \frac{n}{N}\right). \tag{28}$$

Here n denotes the current iteration, N denotes the number of the final iteration. Since the parameter τ can have a strong influence on the results, the algorithm was applied with several values of N.

Additionally, to exploit the option of decoupling, the GB-UCB acquisition function is optimized, but only one submodel is evaluated in each iteration. The submodel to be retrained is chosen randomly with the probability shown in Eq. (24). The results are shown in Fig. 7.

In this figure, the number of function evaluations represents the sum of the evaluations of the LLE and GLE submodel. It can be seen that both GB-UCB and COBALT reduce the error to low values. For the COBALT runs with different numbers of iterations, it can be seen that the convergence is initially slow, but more rapid as the iteration number gets closer to the final iteration. This is likely due to the fact that close to the end $\tau \to 0$, which leads to less exploration as the constraints are not relaxed. At the final iterations of COBALT, the error is quite similar to GB-UCB without exploiting decoupling. This can be explained by the fact that GB-UCB does not relax constraints when the surrogate models are uncertain, but rather provides incentives in the cost function. This corresponds to taking $\tau = 0$ for all iterations of COBALT. So for this problem GB-UCB converges faster and more uniformly than COBALT.

In Fig. 7 it can also be seen that exploiting the decoupled structure of the submodels reduces the necessary number of function evaluations significantly. This may indicate that there is a large discrepancy between the submodels: either one submodel is significantly easier to approximate, or one submodel has a much lower influence on the objective function. Differentiating between these cases is difficult as the sensitivity is specific to the submodel and typically not unitless. But what can be analyzed is the number of times each individual submodel has been evaluated up to iteration n, denoted as $n_{\text{LLE},n}$ and $n_{\text{GLE},n}$. To this end, the difference between the amount of evaluations of the two submodels over the algorithm iterations is shown in Fig. 8.

It can be clearly seen that the number of samples that are used to improve the accuracy of the LLE submodel is much higher than the number of samples of the GLE submodel. Especially in the early iterations, the LLE submodel is evaluated almost exclusively. This is

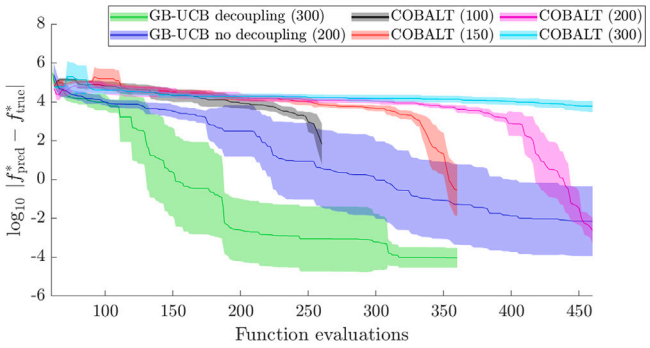


Fig. 7. Results of hydroformylation case study using different algorithms. The final numbers of iterations are given in parentheses.

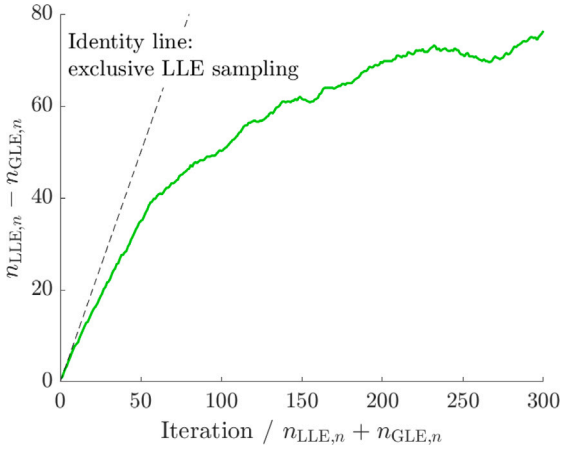


Fig. 8. Comparison of the number of evaluations of individual submodels.

in contrast to the fact that the GLE submodel has six input variables, while the LLE only depends on four inputs. Thus, the input space of the GLE is much larger. This shows the importance of exploiting the decoupling algorithmically, as the number of necessary function evaluations can effectively be reduced, rather than using intuition about which submodel requires more samples.

6. Conclusion and outlook

In this work, we present an approach for applying the upper-confidence-bound acquisition function to gray-box models (GB-UCB) with several surrogate models and inequality and equality constraints. This is achieved by including a sensitivity term that describes the influence of each surrogate model on the objective function. This approach is well suited to exploit the fact that the submodels may be evaluated independently.

The method is tested on six test problems that contain both problems with and without nonlinear constraints. We showed that on these test problems GB-UCB and the reference algorithm COBALT mWB2-CF lead to similar and satisfactory results.

Furthermore, both algorithms were applied to a realistic and demanding case study, the process of the hydroformylation of 1-dodecene in a thermomorphic solvent system. This flowsheet optimization problem contains more degrees of freedom and significantly more (active) constraints than the test problems. We show that GB-UCB yields better results compared to COBALT mWB2-CF, which is probably due to the fact that the relaxation of the constraints in COBALT leads to over-exploration of the search space. This effect is more significant if there are more constraints present, which is common for flowsheet optimization problems. Additionally, exploiting the fact that the submodels can

be evaluated and re-trained separately reduced the amount of submodel evaluations significantly in the flowsheet case study. This happens if it pays off more to increase the accuracy of one of the submodels than to increase the accuracy of the other, in this case in contrast to the fact that it has less input variables. This shows that the approach can identify and exploit asymmetry in the influence and the difficulty of the approximation of different submodels correctly.

Further work should address the reduction of the number of iterations that are needed for convergence. For larger flowsheets, it may not be feasible to perform hundreds of flowsheet optimizations, many of which make use of inaccurate surrogate models. This issue could be addressed by developing a heuristic to determine the number of samples in the initial set for which no optimization has to be performed.

In addition, a cluster sampling approach could reduce the required number of iterations by selecting multiple samples around the optimum in each algorithm iteration.

CRediT authorship contribution statement

Joschka Winz: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Conceptualization. **Florian Fromme:** Investigation. **Sebastian Engell:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research has been supported by the project "KI-Inkubator-Labore in der Prozessindustrie - KEEN", funded by the Bundesministerium für Wirtschaft und Klimaschutz (BMWK), Germany under grant number 01MK20014T. This support is gratefully acknowledged.

Data availability

Data will be made available on request.

References

Alexandrov, N.M., Dennis, J.E., Lewis, R.M., Torczon, V., 1998. A trust-region framework for managing the use of approximation models in optimization. Struct. Optim. 15 (1), 16–23. http://dx.doi.org/10.1007/BF01197433.

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M., 2019. CasADi: A software framework for nonlinear optimization and optimal control. Math. Program. Comput. 11 (1), 1–36. http://dx.doi.org/10.1007/s12532-018-0139-4.

Arbel, J., Pitas, K., Vladimirova, M., Fortuin, V., 2023. A primer on Bayesian neural networks: review and debates. http://dx.doi.org/10.48550/arXiv.2309.16314, arXiv:2309.16314.

Astudillo, R., Frazier, P.I., 2019. Bayesian optimization of composite functions. In: 36th International Conference on Machine Learning, Vol. 2019-June. ICML 2019, pp. 547–556, arXiv:1906.01537.

Boukouvala, F., Floudas, C.A., 2017. ARGONAUT: AlgoRithms for global optimization of coNstrAined grey-box compUTational problems. Optim. Lett. 11 (5), 895–913. http://dx.doi.org/10.1007/s11590-016-1028-2.

Caballero, J.A., Grossmann, I.E., 2008. An algorithm for the use of surrogate models in modular flowsheet optimization. AIChE J. 54 (10), 2633–2650. http://dx.doi.org/10.1002/aic.11579.

Chambers, M., Mount-Campbell, C.A., 2002. Process optimization via neural network metamodeling. Int. J. Prod. Econ. 79 (2), 93–100. http://dx.doi.org/10.1016/S0925-5273(00)00188-2.

Davis, E., Ierapetritou, M., 2007. A kriging method for the solution of nonlinear programs with black-box functions. AIChE J. 53 (8), 2001–2012. http://dx.doi.org/10.1002/aic.11228.

Desai, K.M., Survase, S.A., Saudagar, P.S., Lele, S., Singhal, R.S., 2008. Comparison of artificial neural network (ANN) and response surface methodology (RSM) in fermentation media optimization: Case study of fermentative production of scleroglucan. Biochem. Eng. J. 41 (3), 266–273. http://dx.doi.org/10.1016/j.bej.2008.05.009.

Dixon, L.C.W., Szego, G.P., 1978. Towards Global Optimization. Elsevier Science Ltd.

Eason, J.P., Biegler, L.T., 2016. A trust region filter method for glass box/black box optimization. AIChE J. 62 (9), 3124–3136. http://dx.doi.org/10.1002/aic.15325.

Forrester, A.I., Keane, A.J., 2009. Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. 45 (1–3), 50–79. http://dx.doi.org/10.1016/j.paerosci.2008.11.001.

Frazier, P.I., 2018. A tutorial on Bayesian optimization. http://dx.doi.org/10.48550/ARXIV.1807.02811.

Gardner, J.R., Kusner, M.J., Xu, Z.E., Weinberger, K.Q., Cunningham, J.P., 2014. Bayesian optimization with inequality constraints. In: ICML. pp. 937–945.

Gelbart, M.A., Snoek, J., Adams, R.P., 2014. Bayesian optimization with unknown constraints. http://dx.doi.org/10.48550/arXiv.1403.5607, arXiv:1403.5607.

Gramacy, R.B., Gray, G.A., Digabel, S.L., Lee, H.K.H., Ranjan, P., Wells, G., Wild, S.M., 2015. Modeling an augmented Lagrangian for blackbox constrained optimization. http://dx.doi.org/10.48550/arXiv.1403.4890, arXiv:1403.4890.

Henao, C.A., Maravelias, C.T., 2011. Surrogate-based superstructure optimization framework. AIChE J. 57 (5), 1216–1232. http://dx.doi.org/10.1002/aic.12341.

Henrich, F., Bouvy, C., Kausch, C., Lucas, K., Preuß, M., Rudolph, G., Roosen, P., 2008. Economic optimization of non-sharp separation sequences by means of evolutionary algorithms. Comput. Chem. Eng. 32 (7), 1411–1432. http://dx.doi.org/10.1016/j.compchemeng.2007.06.009.

Hentschel, B., Peschel, A., Freund, H., Sundmacher, K., 2014. Simultaneous design of the optimal reaction and process concept for multiphase systems. Chem. Eng. Sci. 115, 69–87. http://dx.doi.org/10.1016/j.ces.2013.09.046.

Hernández-Lobato, J.M., Gelbart, M.A., Hoffman, M.W., Adams, R.P., Ghahramani, Z., 2015. Predictive entropy search for Bayesian optimization with unknown constraints. arXiv:1502.05312.

Hock, W., Schittkowski, K., 1980. Test examples for nonlinear programming codes. J. Optim. Theory Appl. 30 (1), 127–129. http://dx.doi.org/10.1007/BF00934594.

Iftakher, A., Aras, C.M., Monjur, M.S., Hasan, M.M.F., 2022. Data-driven approximation of thermodynamic phase equilibria. AIChE J. 68 (6), e17624. http://dx.doi.org/10.1002/aic.17624.

Janus, T., Engell, S., 2021. Iterative process design with surrogate-assisted global flowsheet optimization. Chem. Ing. Tech. 93 (12), 2019–2028. http://dx.doi.org/10.1002/cite.202100095.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient global optimization of expensive black-box functions. J. Global Optim. 13 (4), 455–492. http://dx.doi.org/10.1023/A:1008306431147, arXiv:astro-ph/0005074v1.

Kahrs, O., Marquardt, W., 2007. The validity domain of hybrid models and its application in process optimization. Chem. Eng. Process. Process Intensif. 46 (11), 1054–1066. http://dx.doi.org/10.1016/j.cep.2007.02.031.

Kaiser, S., Engell, S., 2023. An integrated approach to fast model-based process design: Integrating superstructure optimization under uncertainties and optimal design of experiments. Chem. Eng. Sci. 269, 118453. http://dx.doi.org/10.1016/j.ces.2023.118453.

Keßler, T., Kunde, C., McBride, K., Mertens, N., Michaels, D., Sundmacher, K., Kienle, A., 2019a. Global optimization of distillation columns using explicit and implicit surrogate models. Chem. Eng. Sci. 197, 235–245. http://dx.doi.org/10.1016/j.ces.2018.12.002.

Keßler, T., Kunde, C., Mertens, N., Michaels, D., Kienle, A., 2019b. Global optimization of distillation columns using surrogate models. SN Appl. Sci. 1 (1), 11. http://dx.doi.org/10.1007/s42452-018-0008-9.

Keßler, T., Mertens, N., Kunde, C., Nentwich, C., Michaels, D., Engell, S., Kienle, A., 2017. Efficient global optimization of a novel hydroformylation process. In: Computer Aided Chemical Engineering, vol. 40, Elsevier, pp. 2113–2118. http://dx.doi.org/10.1016/B978-0-444-63965-3.50354-8.

Kiedorf, G., Hoang, D.M., Müller, A., Jörke, A., Markert, J., Arellano-Garcia, H., Seidel-Morgenstern, A., Hamel, C., 2014. Kinetics of 1-dodecene hydroformylation in a thermomorphic solvent system using a rhodium-biphephos catalyst. Chem. Eng. Sci. 115, 31–48. http://dx.doi.org/10.1016/j.ces.2013.06.027.

Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T., 2002. The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. 12 (2), 479–502. http://dx.doi.org/10.1137/S1052623499363220.

Kunde, C., Keßler, T., Linke, S., McBride, K., Sundmacher, K., Kienle, A., 2019. Surrogate modeling for liquid–liquid equilibria using a parameterization of the binodal curve. Processes 7 (10), 753. http://dx.doi.org/10.3390/pr7100753.

Lu, C., Paulson, J.A., 2023. No-regret constrained Bayesian optimization of noisy and expensive hybrid models using differentiable quantile function approximations. J. Process Control 131, 103085. http://dx.doi.org/10.1016/j.jprocont.2023.103085.

Merchan, V.A., Wozny, G., 2016. Comparative evaluation of rigorous thermodynamic models for the description of the hydroformylation of 1-Dodecene in a thermomorphic solvent system. Ind. Eng. Chem. Res. 55 (1), 293–310. http://dx.doi.org/10.1021/acs.iecr.5b03328.

Močkus, J., 1975. On Bayesian methods for seeking the extremum. In: Marchuk, G.I. (Ed.), Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974. In: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 400–404.

Nascimento, C.A.O., Giudici, R., Guardani, R., 2000. Neural network based approach for optimization of industrial chemical processes. Comput. Chem. Eng. 24 (9), 2303–2314. http://dx.doi.org/10.1016/S0098-1354(00)00587-1.

Nentwich, C., 2022. Surrogate modeling of phase equilibrium calculations using adaptive sampling. In: Schriftenreihe Des Lehrstuhls Für Systemdynamik Und Prozessführung, vol. 2022,1, Shaker Verlag, Düren.

Nentwich, C., Engell, S., 2019. Surrogate modeling of phase equilibrium calculations using adaptive sampling. Comput. Chem. Eng. 126, 204–217. http://dx.doi.org/10.1016/j.compchemeng.2019.04.006.

Nentwich, C., Varela, C., Engell, S., 2019. Optimization of chemical processes applying surrogate models for phase equilibrium calculations. In: Proceedings of the International Joint Conference on Neural Networks. 2019-July, Institute of Electrical and Electronics Engineers Inc., pp. 1–8. http://dx.doi.org/10.1109/IJCNN.2019.8851816.

Paulson, J.A., Lu, C., 2022. COBALT: constrained Bayesian optimization of computationally expensive grey-box models exploiting derivative information. Comput. Chem. Eng. 160, 107700. http://dx.doi.org/10.1016/j.compchemeng.2022.107700.

Priem, R., Bartoli, N., Diouane, Y., Sgueglia, A., 2020. Upper trust bound feasibility criterion for mixed constrained Bayesian optimization with application to aircraft design. Aerosp. Sci. Technol. 105, 105980. http://dx.doi.org/10.1016/j.ast.2020.105980.

Rastrigin, L.A., 1974. Systems of Extremal Control. Nauka.

Rijckaert, M.J., Martens, X.M., 1978. Comparison of generalized geometric programming algorithms. J. Optim. Theory Appl. 26 (2), 205–242. http://dx.doi.org/10.1007/BF00933404.

Rosenbrock, H.H., 1960. An automatic method for finding the greatest or least value of a function. Comput. J. 3 (3), 175–184.

Sansana, J., Joswiak, M.N., Castillo, I., Wang, Z., Rendall, R., Chiang, L.H., Reis, M.S., 2021. Recent trends on hybrid modeling for industry 4.0. Comput. Chem. Eng. 151, 107365. http://dx.doi.org/10.1016/j.compchemeng.2021.107365.

Sant Anna, H.R., Barreto, A.G., Tavares, F.W., De Souza, M.B., 2017. Machine learning model and optimization of a PSA unit for methane-nitrogen separation. Comput. Chem. Eng. 104, 377–391. http://dx.doi.org/10.1016/j.compchemeng.2017.05.006.

Schäfer, E., Brunsch, Y., Sadowski, G., Behr, A., 2012. Hydroformylation of 1-dodecene in the thermomorphic solvent system dimethylformamide/decane. Phase behavior-reaction performance-catalyst recycling. Ind. Eng. Chem. Res. 51 (31), 10296–10306. http://dx.doi.org/10.1021/ie300484q.

Schonlau, M., Welch, W.J., Jones, D.R., 1998. Global versus local search in constrained optimization of computer models. In: Institute of Mathematical Statistics Lecture Notes - Monograph Series, Institute of Mathematical Statistics, Hayward, CA, pp. 11–25. http://dx.doi.org/10.1214/lnms/1215456182.

Schweidtmann, A.M., Mitsos, A., 2019. Deterministic global optimization with artificial neural networks embedded. J. Optim. Theory Appl. 180 (3), 925–948. http://dx.doi.org/10.1007/s10957-018-1396-0.

Shah, P., Sheriff, M.Z., Bangi, M.S.F., Kravaris, C., Kwon, J.S.-I., Botre, C., Hirota, J., 2022. Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters. Chem. Eng. J. 441, 135643. http://dx.doi.org/10.1016/j.cej.2022.135643.

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W., 2012. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Trans. Inform. Theory 58 (5), 3250–3265. http://dx.doi.org/10.1109/TIT.2011.2182033.

Tawarmalani, M., Sahinidis, N.V., 2005. A polyhedral branch-and-cut approach to global optimization. Math. Program. 103 (2), 225–249. http://dx.doi.org/10.1007/s10107-005-0581-8.

Turton, R., Bailie, R.C., Whiting, W.B., Shaeiwitz, J.A., 2008. Analysis, Synthesis and Design of Chemical Processes. Prentice Hall.

Vogelpohl, C., Brandenbusch, C., Sadowski, G., 2014. High-pressure gas solubility in multicomponent solvent systems for hydroformylation. Part II: Syngas solubility. J. Supercrit. Fluids 88, 74–84. http://dx.doi.org/10.1016/j.supflu.2014.01.017.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106 (1), 25–57. http://dx.doi.org/10.1007/s10107-004-0559-y.

Watson, A.G., Barnes, R.J., 1995. Infill sampling criteria to locate extremes. Math. Geol. 27 (5), 589–608. http://dx.doi.org/10.1007/BF02093902.

Winz, J., Engell, S., 2021. Optimization based sampling for gray-box modeling using a modified upper confidence bound acquisition function. In: Türkay, M., Gani, R. (Eds.), Computer Aided Chemical Engineering. In: 31 European Symposium on Computer Aided Process Engineering, vol. 50, Elsevier, pp. 953–958. http://dx.doi.org/10.1016/B978-0-323-88506-5.50147-9.

Winz, J., Engell, S., 2024. Data-efficient surrogate modeling of thermodynamic equilibria using Sobolev training, data augmentation and adaptive sampling. Chem. Eng. Sci. 299, 120461. http://dx.doi.org/10.1016/j.ces.2024.120461.

Winz, J., Nentwich, C., Engell, S., 2021. Surrogate modeling of thermodynamic equilibria: applications, sampling and optimization. Chem. Ing. Tech. 93 (12), 1898–1906. http://dx.doi.org/10.1002/cite.202100092.

Zagajewski, M., Behr, A., Sasse, P., Wittmann, J., 2014. Continuously operated miniplant for the rhodium catalyzed hydroformylation of 1-dodecene in a thermomorphic multicomponent solvent system (TMS). Special Issue: InPROMPT – Integrated Chemical Processes with Liquid Multiphase Systems, Chem. Eng. Sci. Special Issue: InPROMPT – Integrated Chemical Processes with Liquid Multiphase Systems, 115, 88–94.http://dx.doi.org/10.1016/j.ces.2013.09.033,