Adaptively exploring the feature space of flowsheets

J. Höller,[1] M. Bubel,[1] R. Heese,[1] P. O. Ludl,[1] P. Schwartz,[1]
J. Schwientek,[1] N. Asprion,[2] M. Wlotzka,[2] and M. Bortz[1]

[1]*Fraunhofer Institute for Industrial Mathematics (ITWM),*
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
[2]*Chemical and Process Engineering, BASF SE,*
Carl-Bosch-Str. 38, 67056 Ludwigshafen, Germany

(Dated: June 21, 2023)

Abstract

Simulation and optimization of chemical flowsheets rely on the solution of a large number of non-linear equations. Finding such solutions can be supported by constructing machine-learning based surrogates, relating features and outputs by simple, explicit functions. In order to generate training data for those surrogates computationally efficiently, schemes to adaptively sample the feature space are mandatory. In this article, we present a novel family of utility functions to favor an adaptive, Bayesian exploration of the feature space in order to identify regions that are convergent, fulfill customized inequality constraints and are Pareto-optimal with respect to conflicting objectives. The benefit is illustrated by small toy-examples as well as by industrially relevant chemical flowsheets.

I. INTRODUCTION

Flowsheets of chemical production processes are typically modelled using the MESH-equations [1]. For stationary processes, a system of coupled nonlinear equations results, whose size scales with the number of components and the number of theoretical stages used in the model. For industrial production plants, the system can contain up to an order of $10^3 - 10^6$ equations, containing mass and heat balances as well as thermodynamic information and the flowsheet topology; cf. [2] for examples of industrial applications. Since these systems are generally underdetermined, they are complemented with user-defined equality specifications so that a fully determined system is obtained. Additionally, inequality constraints may be needed to accurately describe the allowed operating window of the process.

The set of equality specifications describes the process design and they are denoted as features in the following. The complete solution vector of the system is denoted as the vector of process variables. Some of its entries (or functions thereof) can be chosen as objectives for process optimization.

Generally, for a given set of equality specifications, it is, due to nonlinearities, a priori unknown whether a solution exists at all. If no solution is found for a given set of feature values, i.e. if the simulation is divergent, this may have two reasons: Either no solution exists for that set, or the initial point used in the simulation is not within the convergence radius of an existing, but unknown solution. In black-box simulations, these two cannot be distinguished, which can lead to false conclusions about the simulation outcome: Although a solution exists, the simulation may fail, and the simulated point is erroneously considered as not being covered by the simulation model.

The idea pursued here is to explore the feature space with two goals: Avoiding unnecessary simulations at points where no solution exists and realizing simulations in the most informative regions of the feature space.

In order to favor sampling at informative points in feature space, a Bayesian, i.e. adaptive exploration strategy is used. A succinct summary is given in the following; for further details, we refer to the reviews [3], [4] and references therein.

Let x denote a vector in feature space and $f(x)$ the corresponding output vector obtained from the rigorous flowsheet simulation. The Bayes strategy employed here then

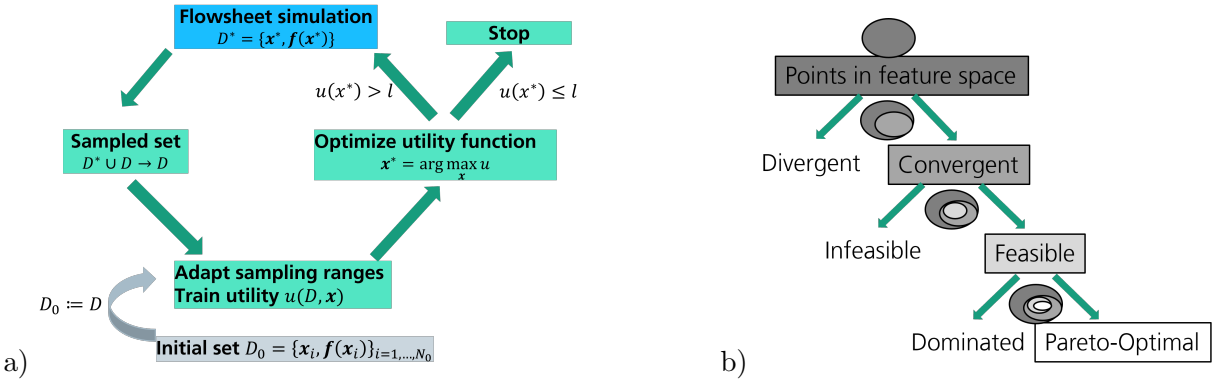

Figure 1: a) The Bayes Loop; see the main text for details. b) Use cases for adaptive exploration covered in this work.

works as follows (cf. Fig. 1):

An initial data set $D_0 = \{(x_i, f(x_i))_{i=1,\ldots,N_0}\}$ consisting of features and their output values is created and used to adapt the sampling ranges and train a utility function u, based on surrogate models relating feature and output spaces. The utility function encodes the information gain and thus is maximized, so that a new data point with feature vector $x^* = \arg\max_x u$ and output vector $f(x^*)$ is obtained. A simple stopping rule consists in terminating the procedure if the optimal value $u(x^*)$ is less then a lower acceptance threshold l. Otherwise, $(x^*, f(x^*))$ is added to the current set of samples, and the next iteration starts by adapting the sampling ranges.

Two assets are important to emphasize: First, depending on the utility function, different use cases can be dealt with, cf. Fig. 1 b). Not only is it possible to resolve the border of convergent simulations in feature space, but, in the presence of inequality constraints, one may also explore the feasible region, defined by solutions obeying the inequality constraints, within the convergent domain. Going one step further, one may also explore Pareto-optimal solutions in the convergent and feasible regions.

Second, the sampling scheme is adaptive, as opposed to uniform schemes like, for example, Latin Hypercube or Sobol sampling [5]. Cf. [6] for an introduction into surrogate modelling of processes based on uniform and adaptive schemes. This avoids simulations which are not informative for the use case at hand. This was also exploited by [7] to deal with parametric model uncertainties.

The novel contribution presented in this work is a significant improvement of utility

functions compared to previous work [8], [9]. These utility functions cover different use cases (convergence, feasibility, optimization, cf. Fig. 1) and are combined by adapting the frequency of their subsequent calls to the desired use case. These calls include an automated selection of the domain where the training of the surrogate and the maximization of the utility functions are carried out. Furthermore, a numerically efficient procedure to deal with multicriteria optimization problems within the Bayesian scheme is presented. The benefit is demonstrated by simple toy examples and by industrially relevant flowsheets.

Different previous works exist that deal with the search for the feasible domain of black-box simulations. This problem itself is well-known [10], [11] with many applications. In recent years, the use of machine learning techniques to address this problem by setting up surrogates based on simulation data [12] became increasingly popular and successful: Kriging models [13], [14] and radial basis functions [15] in combination with an expected improvement function [16] are used in analogy to Bayes optimization [4] for the feasibility exploration with surrogates. A summary of different approaches has been published recently [17].

However, our approach presented in this manuscript differs from those known approaches by combining binary information about convergence and a continuous measure of feasibility fulfillment. We substantially improve the utility functions from Ref. [8] and [9] in order to increase the information carried in the individual samples, thus increasing computational efficiency. Furthermore, we propose heuristics to adapt the domains to the actual interesting regions of the feature space, which leads to a high degree of reliability of the utility function's predictions for samples. As outlined in Fig. 1, the key idea of the original method is to iteratively design computer experiments (i.e., flowsheet simulation runs) and use the data to improve machine learning models for predicting the convergence behavior of the simulation given the design variables. Such loops are typically employed in feasibility exploration strategies and are closely related to Bayes optimization methods [17].

Apart from exploring convergence and feasibility, the Bayes scheme has been employed to address optimization problems. It can be seen as a stochastic globalization strategy [18], [19]. Apart from exploring the borders of convergence and feasibility, this is the third use-case that is dealt with in this manuscript: Bayesian optimization is used in

order to generate starting values for a local solver working on the flowsheet simulation. In order to place the Bayesian scheme into a multicriteria context, typically the dominated hypervolume improvement is considered [20], [21]. Since its calculation can be rather time consuming [22], here a rather simple, yet efficient method to incorporate conflicting objective functions is introduced.

Adaptive exploration schemes are also used as adaptive design of experiment strategies; the review [23] contains a comparison with uniform procedures. Applications are described, for example, for the field of material design in [24] and for chemical synthesis in [25].

The remainder of this paper is organized as follows. The following section describes the utility functions, individually for each use case. The third section contains application examples from chemical engineering. The paper ends with a conclusion and outlook section. Details on the adaption of the domain and on the building blocks of the utility functions are deferred to appendices.

II. UTILITY FUNCTIONS FOR CONVERGENCE, FEASIBILITY AND PARETO SAMPLING

This section describes how the utility terms given in appendix B are combined to a utility function to carry out adaptive exploration strategies for the use cases convergence sampling, feasibility sampling and Pareto optimization sampling. All these terms are designed such that they are maximized, with a smallest possible value equal to 0. This makes it possible to define a stopping rule, which is reached if maximizing the utility function does not yield a value larger than some lower acceptance threshold l, which will be given in each use case below. If no improvement beyond l is found, either hyperparameters of the utility terms are adjusted, or the sampling is stopped, as described in the following.

A. Convergence sampling

Using the utility terms defined in appendix B, the utility function for convergence sampling is set up as:

$$u_c = \log\left[A\ (R + R_b\, S)\right] \tag{1}$$

with a lower acceptance threshold $l_c = -0.5$. It involves utility terms for attraction A (with attraction radius r_A), repulsion R (with repulsion radius r), repulsion R_b (with repulsion radius r_b) and the entropy S.

The second term, namely $A\,R_b\,S$, favorizes samples with a certain nearest-neighbours distance close to the border, where the entropy term is close to one. The reason for including an additional term $A\,R$ is as follows: If only very few points have been sampled, the contribution from S can be small far away from those points, meaning that the prediction of convergence is certain. However, this prediction can be wrong, so an additional globalization term $A\,R$ is included, with radii $r_b < r < r_A$ in Eq. (1). A simple toy example illustrating the behaviour of the Bayesian sampling is given below, in Sec. III A.

The importance of R_b is handled dynamically: Initially, a larger value of r_b is chosen, so that expanding the domain is prioritized over refining the border. If it is not possible to find samples that way, that is, if no improvement leading to values larger than l_c is possible, the border radius is reduced, and another utility optimization is started with the already trained surrogates. The sampling stops if r_b reaches a lower bound and no values of u_c beyond l_c are found.

B. Feasibility sampling

The utility function for feasibility sampling, u_f, shall avoid samples too far away from and too close to existent data, and respect convergence and feasibility. It is chosen as

$$u_f = \log(A\,R\,P_C) + \alpha \log(P_F) \tag{2}$$

and is composed of the utility terms for attraction A, repulsion R, the probability for convergence P_C and the probability of feasibility by constraint fulfilment P_F; the lower acceptance threshold is set to $l_f = -0.5$.

The argument of the first logarithm on the right hand side of Eq. (2) favors samples with a constant distance to each other, namely at the maximum value of $A\,R$, within the convergent region. The argument of the second logarithm yields a significant contribution when at the same time, these samples fulfill the constraints.

The factor α is added for stability. The common approach for expected improvement under independent constraints corresponds to $\alpha = 1$. However, if no feasible samples are

known, the regressor is likely to predict very low probability to be feasible over the entire domain, and the acceptance threshold for the utility cannot be fulfilled at all. In this case $\alpha < 1$ is used to reduce the importance of constraint fullfilment for the present step and allow the algorithm to slowly progress towards feasible samples.

If there are only very few feasible samples, the regressor may not recognize them reliably and again never predict a feasible sample. In this case $\alpha < 1$ based upon the following heuristic is chosen: Calculate the mean predicted probability to be feasible over all feasible samples as

$$P_{\log} = \sum_{constraints} \log\left(\frac{1}{n_F}\sum_{i=0}^{n_F} P_{Fi}\right) \tag{3}$$

If $P_{\log} < -0.2$, we use $\alpha = -\frac{0.2}{P_{\log}}$. If utility optimization does not lead to values larger than l_f, a new optimization with a reduced repulsion radius or reduced α, but the same surrogates is started. The procedure stops when no values beyond l_f are found and the repulsion radius and α reach their lower bounds.

C. Pareto Optimization sampling

The utility function for optimization is chosen as

$$u_o = \log(\alpha\, A\, P_C\, P_F\, EI)\,, \tag{4}$$

with a lower acceptance threshold $l_o = -2.5$. The product $A\, P_C\, P_F$ favors samples that are convergent, feasible and close to known solutions. Closeness means that known solutions can be used as initialization for the nonlinear solver, thus enhancing convergence. Eq. (4) also contains the expected improvement EI and a scale factor α, included here in order to define a problem-independent lower acceptance threshold. It is chosen as the standard deviation of the equally weighted sum of all N_o many objectives over the data set used for the 'Zoom to non-dominated domain'.

Additionally, the following criteria to accept a sampled point are applied: Sample suggestions with $u_o \in [-1, -0.5)$ need a minimal distance of 2% to all existing samples, and for $l_o \leq u_o < -1$ they need 10% in order to be accepted. Furthermore, samples of such low utility are discarded if there are any others available.

To generate samples within a multicriteria setting consisting of N_o many objective functions, the procedure is as follows. With probability 50% just one single objective is selected for the EI (equal chance for each, resulting in a probability of $1/(2N_o)$ for a given objective function). In the remaining 50%, each objective is selected with 50% chance. If this selects none, as objective function, the equally weighted sum is chosen in the EI.

Thereafter, the domain adaption for Pareto-optimality sampling, cf. Sec. A 3, is performed for the objective function and all constraints. Then the surrogates are trained, the current best sample with respect to EI is identified and the utility is optimized.

In all application examples studied so far, this heuristic procedure yields good starting values, especially when it comes to problems that require globalization strategies. As already stated in the introduction, this is the very goal of the approach presented here: Finding good starting values so that a local optimization solver working on the flowsheet simulation is likely to converge to the globally optimal solutions, using adaptive scalarization schemes [26]. Since our aim is not to approximate the Pareto boundary accurately within the Bayesian framework, we restrain from computationally costly Pareto hypervolume improvements [20], [22].

D. Combined sampling strategies

The strategies described above can be combined by successive calls in the Bayes loop. For example, for exploration of the design space, feasibility and convergence sampling runs are done iteratively. To explore and optimize simultaneously, all three utilities are maximized iteratively one after the other until the termination criterion is met. The numbers of repeated calls of the sampling strategies are hyper-parameters that are determined by experience.

III. APPLICATION TO EXAMPLES FROM CHEMICAL ENGINEERING

A. Toy example

As a first step the reasoning behind the utility function Eq. (1) will be motivated on the basis of a simple analytical example from a previous work [9]. In this example there

are two features $x = (x_1, x_2) \in [-2, 2] \times [-2, 2]$. The convergent, divergent and feasible regions are defined as

- divergent for $||x||_2 > 1$,
- convergent for $||x||_2 \leq 1$ and
- feasible for $||x||_2 \leq 1$ and $x_1 x_2 > 0.1$.

and are depicted in Fig. 2.

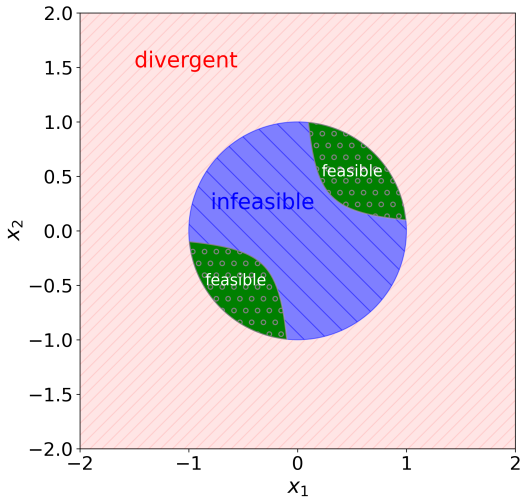

Figure 2: Domains of the toy example from [9].

In Fig. 3 the improved sampling strategies as sketched above are compared to a result from a previous work [9]. Therefore, first the same utility functions as in [9] are adopted, but the sampling is done with the improved initialization and range selection as described in Sec. A. The applied range heuristic avoids sampling very far from the convergent domain. Still, an accumulation of divergent samples at the boundary of the feasible domain appears.

In a second step, cf. the right panel in Fig. 3, not only the range selection has been done, but at the same time the improved utility functions for convergence and feasibility sampling, cf. Sec. II, have been used. This leads to a much more homogeneous resolution of both the convergence border and the feasible domain.

Using the simple toy model, the structure of the utility function for convergence sampling Eq. (1) can be made plausible as follows. Maximizing the entropy S alone would

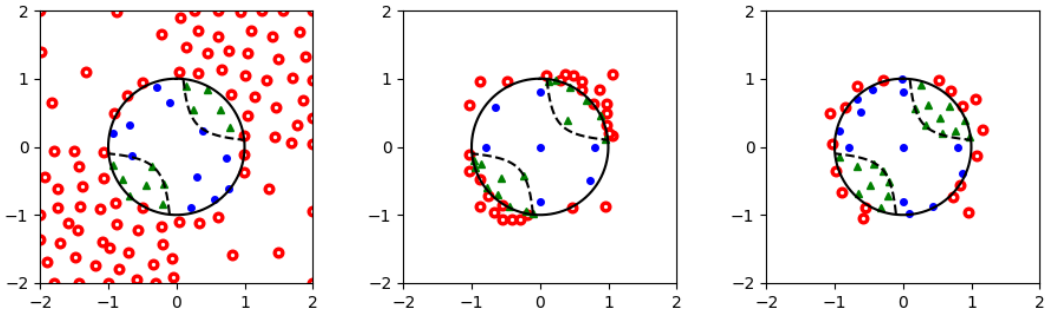

Figure 3: Left: Exploration strategy from [9] (15 random initial points + 45 adaptive samples with weights $(w_s, w_r, w_c) = (1, 5, 1)$). Middle: Exploration strategy with utility from [9] and initialization and range heuristics described in Sec. A. Right: Exploration with utility from this work, Eq. (1), and with range heuristics. Axis labels are as in Fig. 2.

suggest points on the assumed border between convergent and divergent regions, independently of already sampled points. Therefore, a repulsion term R_b should be included multiplicatively. However, as illustrated in Fig. 4, a utility of the form $R\,S$ still does not lead to the desired behaviour: If only very few divergent points have been met at a certain stage of the exploration, and these divergent points are not distributed evenly along the convergence border, the sampling would take quite a long time to resolve the complete boundary, if this happens at all. Therefore, an additional repulsion term R is added, with a radius $r > r_b$, thus favoring samples further away from the known ones. Fundamentally, the roles of R and S are not symmetric: It does make sense to sample for $S = 0$, i.e. for a certain prediction of the convergence behaviour, if the distance between samples is large. Finally, the overall distance to the known samples is kept limited by a global attraction term A entering as a prefactor.

The combination of utility terms in Eqs. (1) and (2) serves to direct the samples towards the most relevant positions. Fig. 5 shows this for two specific steps. In both cases the utility plot shows rather high values outside of the domain selected for the corresponding step, marked by a white frame. However, due to the range heuristics, it is not necessary to have a reasonable utility (and, in consequence, accurate surrogate models) in the whole, but only in the selected domain - a significantly simpler task. Despite of the rather small number of samples available at the used step the utility functions do highlight relevant

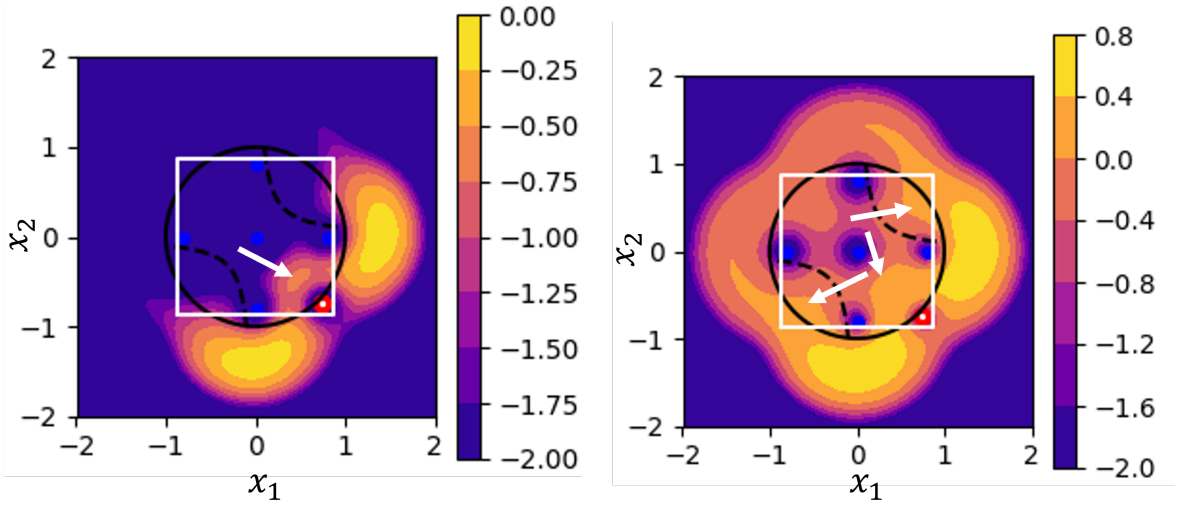

Figure 4: Comparison of a utility function containing only $R_b S$ (left panel) versus the suggestion in (1) (right panel). In both cases, the same six datapoints (as shown) have been used to train both utilities. The maxima of the utility function within the white boxes are marked by white arrows; the values of the utility functions are given by the color scale.

positions for new samples efficiently.

B. Methanol water flash

In this example, a flash fed with a liquid mixture of 0.5 kmol/h water and 0.5 kmol/h methanol at 25°C is considered. As equality specifications, values for the flash pressure $p \in [0.5, 5]$ bar and the heat duty $\dot{Q} \in [-2.0, +1.0]$ kW are set. Solving the MESH equations then determines the flash temperature T. Sampling is initialized with the star-like heuristic, see Sec. A 1, starting from the convergent point $(p, \dot{Q}) = (0.5\ \text{bar}, 0.5\ \text{kW})$. Thereafter, the sampling strategy based on the convergence utility of Eq. (1) is applied to explore the input space $(p, \dot{Q})$ with 200 adaptive points. The resulting data points in the design space are shown in Fig. 6. The convergence utility is able to explore a large part of the convergent subdomain of the design space.

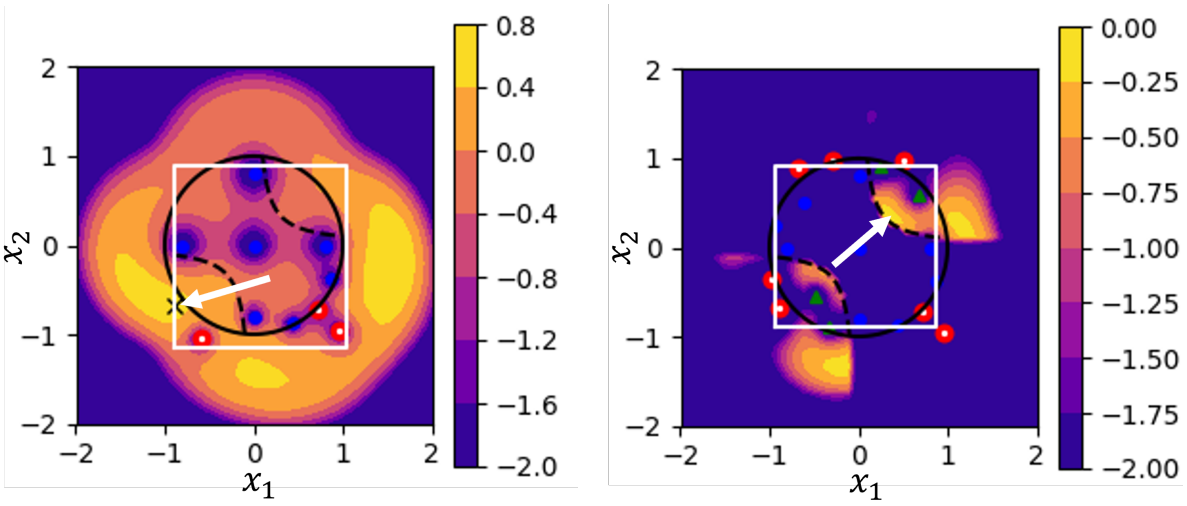

Figure 5: Contour plot of the utility function for border (left) and feasible (right) sampling at two selected steps in the example. The maxima of the utility functions within the selected domains (white frames) are marked by arrows.

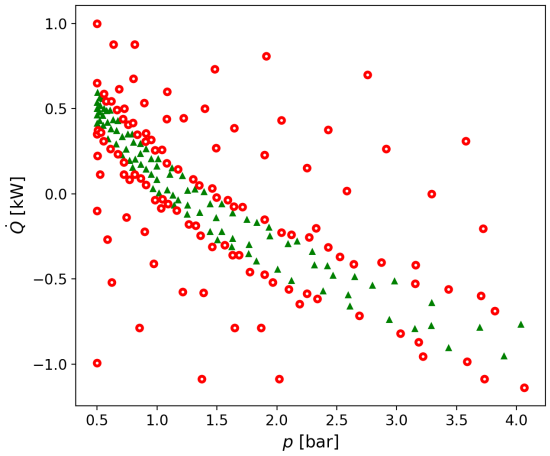

Figure 6: Exploration of the design variable space $(p, \dot{Q})$ of the methanol water flash with the convergence utility of Eq. (1).

C. Multi-objective Bayes optimization

We now want to turn to higher dimensional problems, which also involve optimization with respect to multiple objectives. In the following we illustrate how a prior optimization-driven exploration of the input space can significantly improve the approximation of the Pareto boundary.

1. The OSY test problem

First, we consider the following optimization problem with a six-dimensional input space, two objective functions, and six restrictions from [27]

$$\begin{aligned}
\text{OSY}: \min_{\mathbf{x}} \quad & \begin{pmatrix} f_1(\mathbf{x}) \equiv -[25(x_1-2)^2 + (x_2-2)^2 + (x_3-1)^2 + (x_4-4)^2 + (x_5-1)^2] \\ f_2(\mathbf{x}) \equiv x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \end{pmatrix} \\
\text{s.t.} \quad & c_1(\mathbf{x}) \equiv x_1 + x_2 - 2 \geq 0, \quad c_2(\mathbf{x}) \equiv 6 - x_1 - x_2 \geq 0, \\
& c_3(\mathbf{x}) \equiv 2 - x_2 + x_1 \geq 0, \quad c_4(\mathbf{x}) \equiv 2 - x_1 + 3x_2 \geq 0, \\
& c_5(\mathbf{x}) \equiv 4 - (x_3 - 3)^2 - x_4 \geq 0, \quad c_6(\mathbf{x}) \equiv (x_5 - 3)^2 + x_6 - 4 \geq 0, \\
& 0 \leq x_1, x_2, x_6 \leq 10, 1 \leq x_3, x_5 \leq 5, 0 \leq x_4 \leq 6,
\end{aligned} \tag{5}$$

whose solution is the Pareto boundary shown in Fig. 7 (see [27] for details). Due to the non-convexity of the feasible set and the concavity of the first objective function, we have a non-convex bi-criteria optimization problem at hand. This means that, on the one hand, we have to choose an appropriate scalarization strategy, and on the other hand, we have to compute a globally optimal solution in each scalarization. The former is quite unproblematic: here we used a hybrid approximation (approximating the convex regions of the Pareto boundary using weighted-sum and the non-convex parts using Pascoletti-Serafini scalarization; see [26] for details). Achieving the latter, however, in general is difficult—either computationally expensive when using deterministic methods of global optimization or, in the case of applying heuristics, finding the global optimum is not guaranteed. We compared the following three globalization approaches:

(NG) *no globalization* at the beginning, only initialization of subsequent scalarizations with the scalarization-specific best already calculated solution (warm-starting)

(MG) *multi-start* of the extreme compromise calculations f_1^* and f_2^* using 10 random starting points and subsequent initialization as in (NG)

(EG) *optimization-driven exploration* of the input space using 35 initial evaluations and 11 adaptive samples (incl. three weighted-sum starts) and subsequent initialization as in (NG)

The starting point in all three cases was the infeasible solution $\mathbf{x} = (1.3, 2.0, 2.0, 2.0, 2.0)$.

Fig. 7 shows the Pareto boundary approximations achieved by means of the three globalization strategies.

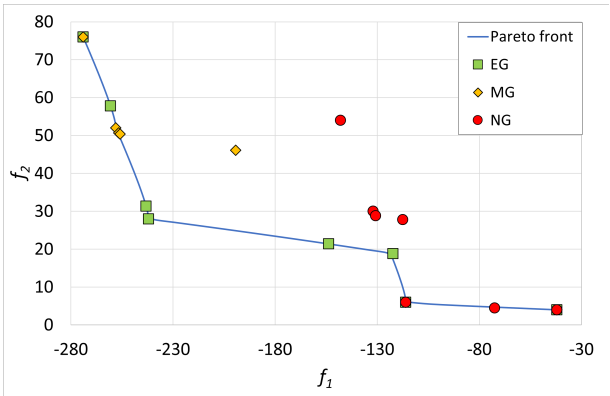

Figure 7: OSY problem: Pareto boundary (blue line) and its approximations – without globalization (red circles), by extreme compromises multi-start globalization (orange diamonds), and exploration-based globalization (green squares)

Strategy (NG) provides Pareto-optimal solutions only in the lower right part. Otherwise, the resulting boundary is too small and only a poor approximation of the actual boundary. The multi-start strategy (MG) provides globally optimal extreme compromises, but the following (not multi-started) scalarizations lead in general only to suboptimal solutions. For the multi-start, of course, more computation time is required. The exploration-based strategy (EG) provides Pareto-optimal points for all scalarizations and this with similar running times as with strategy (NG).

2. Separation of Chloroform and Acetone with entrainer Benzene

In process engineering tasks the situation is even more complicated. Not only that in general there are also non-convex, multi-objective optimization problems. The often equation-based simulation of the flow sheet is mostly given as a black box. As a consequence, in each optimization iteration the system of equations has to be solved exactly. In the case of ambiguous solvability, this leads to further local optima. But, in the case of – sometimes only numerical – divergences it results in 'getting stuck' Pareto point calculations and, thus, in poor Pareto boundary approximations.

We would like to demonstrate this briefly for the separation of Chloroform and Acetone

using Benzene as entrainer and consider the flow sheet shown in Fig. 8. As conflicting measures of CAPEX and OPEX, we consider the sum of equilibrium stages N and the total heat duty Q, both of which are minimized. This results in a NQ-curve for the total flow sheet. Therefore, as features, the reflux ratios, the splits, the number of stages, and the feed heights of all three columns are chosen. In addition to the natural constraints that the feeds must be within the columns, we require that the withdrawn product streams as shown in Fig. 8 should have a minimum purity of 95%. Thus, we have a mixed-integer, multi-criteria optimization problem with two objective functions, twelve decision variables, and six restrictions at hand.

If the starting point is too far away from one or both extreme compromises, divergences occur when solving the system of equations and, thus, the extreme compromise calculation terminates too early. By means of a prior exploration of the input space, on the other hand, a family of initial solutions covering a large convergence range can be generated, from which scalarization-specific best starting points can be selected. Using 101 (optimization-driven) adaptive samples (and 71 initial evaluations), we were able to compute the Pareto boundary approximation shown in Fig. 8, while without exploration one reference run fails.

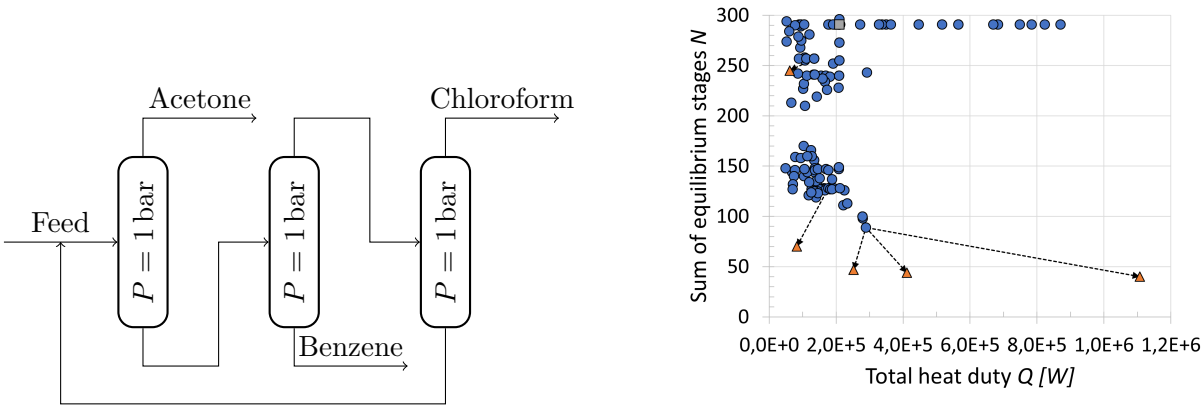

Figure 8: Entrainer distillation of Chloroform and Acetone via Benzene. Left: Flow sheet topology; right: Pareto points (orange triangles) obtained from exploration results (blue circles) with scalarization-specific start point selection (dashed black lines) instead of using original starting point (grey square).

Thus, a prior, optimization-driven exploration of the input space can not only provide good approximations of Pareto boundaries in acceptable time, but also enable Pareto

point computations at all.

3. *Pressure swing distillation*

As a further example the pressure swing distillation of Chloroform and Acetone is considered. The feed consists of a binary mixture of 0.86 mol/mol Chloroform and 0.14 mol/mol Acetone at a rate of 1250 kg/h. These two components form an azeotrope and thus cannot be separated directly by a single distillation column. In this case the mass fractions of both products (Chloroform and Acetone) were maximized subject to a limit of the total heat duty of 1.5 MW. As features once again the reflux ratios $R_{1,2}$ and splits $S_{1,2}$ of the two columns were used ($R_{1,2} \in [1, 80]$, $S_{1,2} \in [0.01, 0.99]$), but now the total number of theoretical stages and the feed stages are kept fixed.

Using adaptive exploration with focus on Pareto-optimal samples as described in Sec. II a large number of simulations in the proper scope for a pareto boundary was obtained. These samples provide very reasonable initial values for a stringent multi-criteria optimization algorithm. The comparatively higher number of samples around the extreme compromises visible in Fig. 9 is intended by the construction of the Pareto sampling utility, because the extreme compromises span the range of the objectives and are more prone to getting stuck in local minima.

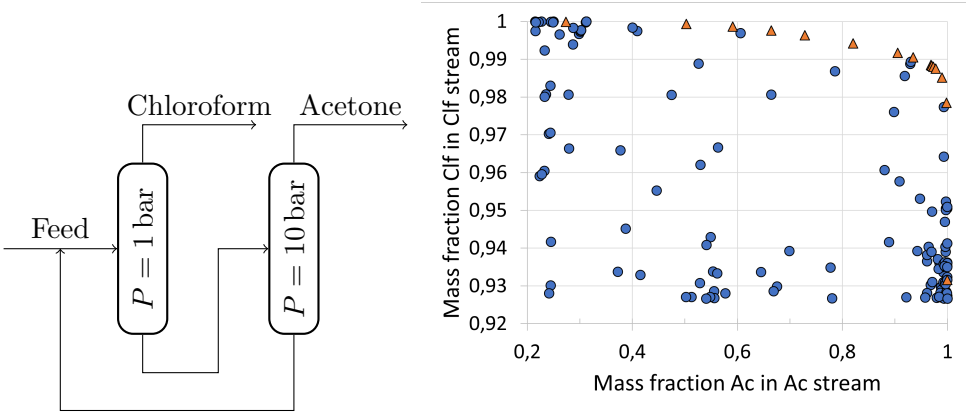

Figure 9: Left: Topology of flow sheet for pressure swing distillation; right: Result from adaptive Pareto sampling (blue circles). The rigorous Pareto boundary obtained by sandwiching (see [26]) is shown by orange triangles for reference.

IV. CONCLUSION

Although conceptually simple, the success of the Bayesian strategy to adaptively explore the feature space of chemical flow sheet simulations essentially depends on the utility function. Choosing the utility function adequately is crucial to obtain the most informative points in few iterations. We have given examples for utility functions that cover the use cases of exploring convergent, feasible and Pareto-optimal regions. As a surrogate underlying the utility function, Gaussian processes have been used, because they offer a simple structure of the surrogate with only few hyper parameters.

Evidence was given that the Bayesian scheme with the utility functions presented above can support optimization based on rigorous heavy-load simulations by generating reliable starting points. Although the application examples have a chemical engineering background, the procedure itself is independent from this application domain and may prove useful in other domains as well.

Appendix A: Adapting the domain

1. Initialization

To create the initial sample set D_0, cf. Fig. 1, we assume that at least one convergent simulation exists. From this convergent simulation, a star-like design with homotopy is performed as sketched in Fig. 10 as follows. For each dimension of the design space the two directions $+e_i$ and $-e_i$ are considered, all separately and independently. In each direction at first one step of fixed length (25% of the total range in that direction) is performed and simulated. If that step converges another step in the same direction is taken, if it diverges the step size is reduced (to 25%) and then simulated again. If a step leads out of the allowed design space, the step is limited to the boundary instead. This procedure is repeated until either a maximum number of simulations is performed, or both a convergent and a divergent point have been found.

Here it is assumed that from this initialization procedure, at least one additional convergent simulation is recovered, so that at least two convergent simulations are available for the zooming and adaptive strategies explained in the following sections.

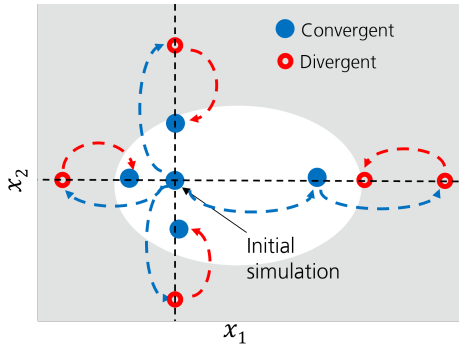

Figure 10: Star-like initialization pattern in feature space. The convergent domain is depicted as a white ellipse, the grey area denotes the divergent region. Arrows denote the points chosen for simulation. If the simulation is divergent, backward steps are performed in order to obtain convergent simulations. Straight black lines illustrate the star-like pattern.

2. Domain adaption for convergence sampling

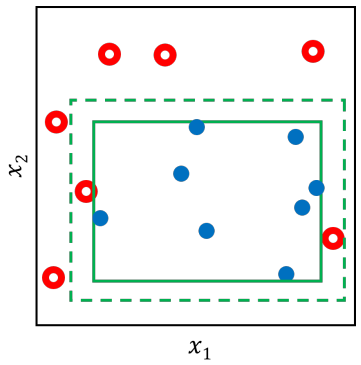

Figure 11: Zoom for convergence sampling in feature space. Blue circles (red open circles) denote convergent (divergent) points. The minimal enclosing box is shown in green; by adding a small margin, the dashed green box results.

The ranges for the sampling of the feature space defined initially by the user may be quite large compared to the actual convergent region. This may lead to strongly imbalanced data sets, and different length scales, inducing unnecessarily large errors of the classifier.

For this reason, the actual domain (that is the region within which the optimization of the utility function is done) should not be much larger than the region covered by convergent data points. It is therefore chosen to be only moderately larger than the box-

shaped domain created by the convergent samples. Therefore, the minimal box containing all convergent samples is determined and increased by a small margin, cf. Fig. 11.

3. Domain adaption for Pareto-optimality sampling

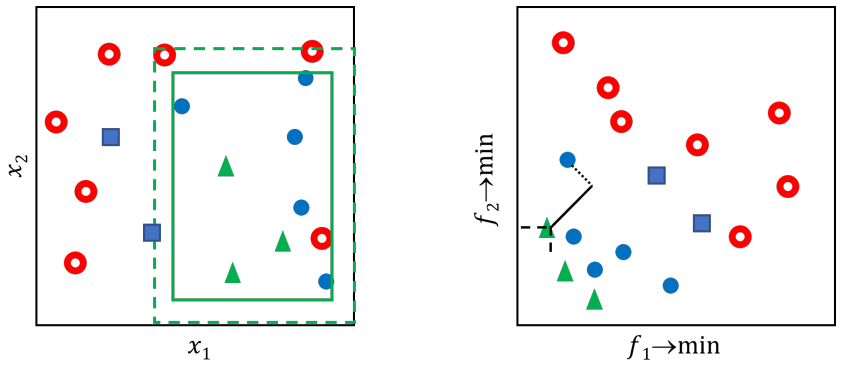

Figure 12: Adaption of domain for sampling Pareto sets. Green triangles: Pareto-optimal points; blue circles and red open circles denote dominated points, both in feature (left) and objective space (right). Only the Pareto points and the dominated points depicted as blue circles are used to select the domain for surrogate training in feature space, shown as green full rectangle on the left. See the main text for further details.

For sampling strategies which have one or more optimization objectives, it is desirable to train the surrogates only on those samples that are most favourable with respect to the optimization targets. In this way, the surrogates achieve better accuracy in the neighbourhood of the currently best-performing samples. Fig. 12 shows a sketch of this situation. All data points which are reasonably good with respect to both objectives are located in a rather small part of the feature space.

To include points which only weakly violate possible inequality constraints but perform well in the objectives, constraint violations are included as additional objectives.

From this enlarged set of objectives, the strictly non-dominated samples, i.e. the Pareto points, are obtained, shown as green triangles in Fig. 12. The ranges for each objective obtained from these Pareto points is used as normalization factor for each objective, respectively.

The domain for the training of the surrogate shall contain the features of the Pareto

points and of those points that are not too far away from them. The distance to the Pareto set is measured along the domination cone center $(1, 1, 1, \ldots)$, cf. Fig. 12, where the domination cone is shown by dashed lines, and distance along $(1, 1)$ of one exemplary point as full line. All points are sorted ascending with respect to this distance; the Pareto points all have value 0.

This allows to make a partition of all samples into three subsets P, D_1, D_2. The subset P contains the Pareto points, D_1 and D_2 contain the first and second halves of the dominated points. Now the standard deviation σ_{D_1} of the distance to the Pareto set is evaluated on D_1, and only those points from D_1 and D_2 are taken to define the sampling domain with a distance smaller than $\alpha \cdot \sigma_{D_1}$ with a tuning factor α. In practice, the choice $\alpha = 5$ leads to good exploration results. In Fig. 12, these points are shown as blue circles. Blue squares are points contained in D_1, but with distance $> \alpha\, \sigma_{D_1}$. Points in D_2 are not considered to evaluate the threshold which effectively removes outliers.

The points in feature space belonging to the samples obtained in this way are used to obtain a minimal enclosing box (green box in Fig. 12), which, after adding a small margin (dashed box in Fig. 12), defines the domain for new samples.

4. Split domain for many samples

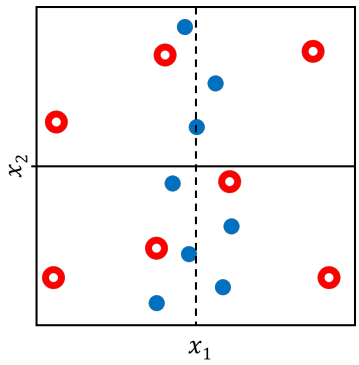

Figure 13: Splitting the domain in feature space along the dimension with the largest range of convergent samples (blue circles): The preferred (rejected) split is shown as full (dashed) line.

The computational effort of training surrogates grows with the number of training points. In order to nevertheless allow sampling of a large number of points, an internal

domain split mechanism is used. If the number of samples reaches a critical value N_c, the domain will automatically be split into smaller parts. Then, any sampling strategy can be run independently on the smaller parts. Finally, sample suggestions from all parts are merged. In this way, for very large data sets the training times for the surrogates and the time needed to solve the optimization problems grows roughly only linearly with the number of points.

The split happens at the median value along one dimension. Three criteria are taken into account in order to select this dimension, namely:

- the difference D between the median of the projected points on one feature and the corresponding center of the design space,
- the fraction F of projected samples within a range of $\pm 5\,\%$ around the median,
- the ratio R of the zoomed and the original domain for convergence sampling (cf. Fig. 11) is closest to one.

A small value for D ensures that the two domains resulting from the split have similar volume. The fraction F is a measure for those samples that have to be used to train surrogates in both domains and should be small. A large value of R means that convergent samples cover a large range in that dimension, which should favor a split, cf. Fig. 13. To take all three criteria into account, we choose the dimension with the smallest value of $D + F - R$.

Appendix B: Utility terms

The three adaptive sampling strategies (convergence sampling, feasibility sampling, and optimization sampling) are based on the optimization of the corresponding utility terms, cf. Sec. I and Fig. 1.

We first describe these single utility terms and then comment on how these terms are combined to the entire utility function. The utility terms are supposed to be maximized.

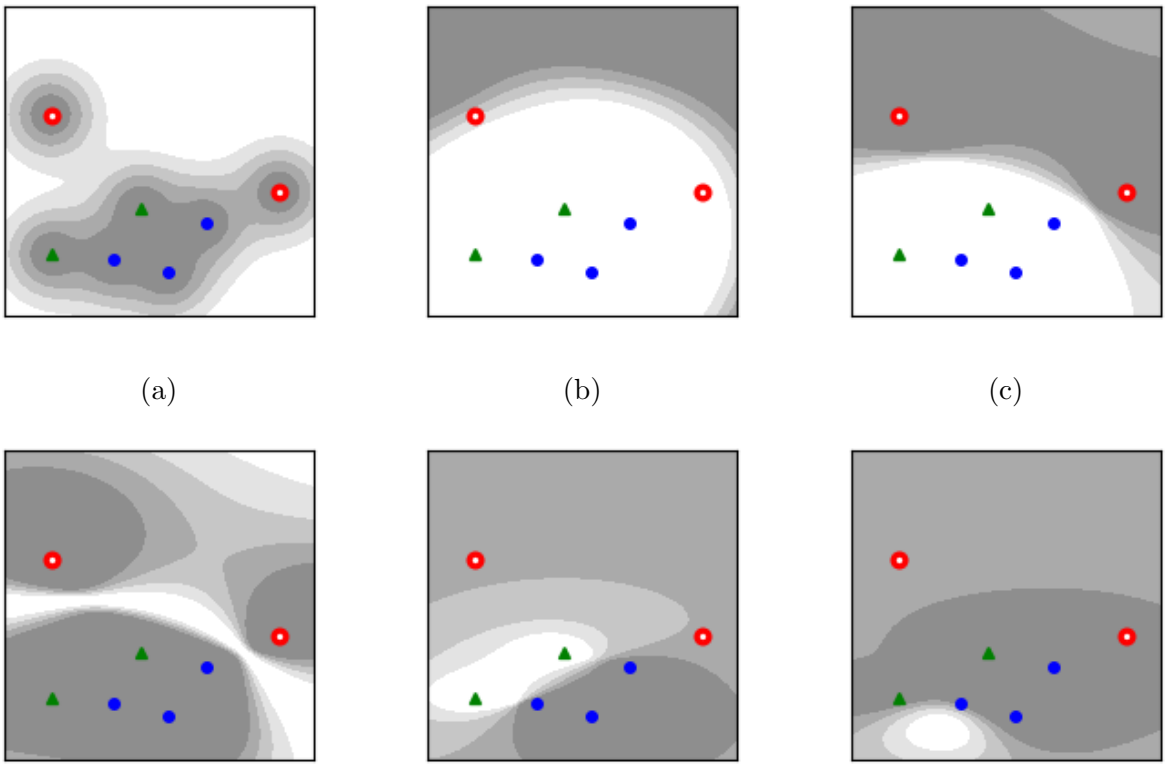

Figure 14: (a) Repulsion, (b) attraction, (c) convergence, (d) entropy, (e) feasibility and (f) expected improvement utility terms with points in feature space. Regions of high utility are shown in white, dark regions have low utility. The types of data points are divergent (red open circles), convergent infeasible (blue circles) and convergent feasible (green triangles). Axis labels have been omitted.

1. Repulsion

The repulsion term can be interpreted as repelling the suggested point from the already existing ones. This is illustrated in Fig. 14: The term vanishes close to the existing data points (in the input space) and increases the farther away one is from them. It is implemented as a product over all n samples at design variable values x_i (the same ones used to train the convergence classification model).

$$R(x) = \prod_{i=1}^{n} \left(1 - e^{-\gamma\left(\frac{x-x_i}{r\,s}\right)^2}\right) \tag{B1}$$

Different hyperparameters are involved in R: The prefactor in the exponential $\gamma = -\log(0.05)$, which means $R(x) = 0.95$ for a single sample if the distance to that sample equals $r\,s$. The default value for the repulsion radius is chosen as $r = 0.25$. The scale s is the length of the selected design space region (cf. Sec. A: 'Domain adaption for convergence sampling' does affect the scale but 'Domain adaption for Pareto-optimality sampling' and 'Split domain for many samples' do not).

2. Attraction

The attraction term in the utility serves to avoid long jumps. It was introduced in order to evaluate inputs too far away from already known points. This is particularly useful for flowsheet simulations in order to initialize the nonlinear equation solver with solutions not too far away from the current point. The attraction term is defined similar to the repulsion term in the following way:

$$A = 1 - \prod_{i=0}^{n_C} \left(1 - e^{-\gamma\left(\frac{x-x_i}{r_A\,s}\right)^6}\right) \tag{B2}$$

Here, only the n_C convergent samples are used as centers x_i and the radius r_A must assume a value significantly larger than the repulsion radius r (at least twice). Otherwise, the product AR in the utility function is too small for the threshold.

The attraction term is visualized in Fig. 14b for a two-dimensional example: For distances on the scale of the attraction radius r_A or smaller, the term is large (white) but decreases rapidly for distances larger than r_A (light grey), approaching a constant value of 0 (dark grey) for very large distances.

3. Convergence

The calibrated classifier CASIMAC [28] is used to predict the probability of convergence P_C (cf. Fig. 14c). From this, entropy is calculated as:

$$S = -\frac{1}{\log 2}\left(P_C \log P_C + (1 - P_C)\log(1 - P_C)\right) \tag{B3}$$

It takes its maximal value $S = 1$ in places where the predicted convergence probability is $P_C = 1/2$, i.e. at the expected border between convergent and divergent regions. It

therefore draws the suggested points towards this border. Illustrated in two dimensions in Fig. 14d, one can see that the entropy is large (white area) close to the estimated border between the divergent (red) and convergent (non-red) points. In regions which are classified with high confidence (corresponding to $P_C = 1$ or $P_C = 0$), the entropy term approaches zero (dark grey).

4. Feasibility

The probability of feasibility under a given constraint is calculated from the distribution predicted by the regression model. In the case of Gaussian Process Regression, this distribution is a normal distribution with mean μ, corresponding to the predition of the surrogate, and its standard deviation σ. The probability $P_{\geq c}$ to respect a lower bound c as inequality constraint then takes the form [29]:

$$P_{\geq c} = \frac{1}{\sqrt{2\pi}\sigma} \int_c^\infty e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x = \frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{\mathrm{c}}{\sqrt{2}\sigma} - \mu\right)\right) \tag{B4}$$

If upper and lower bound are present for one output, the probability is calculated as

$$P_{[c_0,c_1]} = \frac{1}{\sqrt{2\pi}\sigma} \int_{c_0}^{c_1} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x = P_{\geq c_0} + P_{\leq c_1} - 1 \tag{B5}$$

For multiple constraints the product of these probabilities is used; the corresponding utility term encoding the probability of feasibility by constraint fulfilment is denoted by P_F. The interpretation of this term is straight forward: If it is present in the utility function it draws points towards the expected region of constraint fulfilment. In Fig. 14e, the convergent points fulfilling the constraints are shown in green, and the probability of convergence is highest (white) there. Maximizing the utility will therefore draw the samples towards the white region populated by the green points.

5. Optimization

For optimization within the adaptive sampling framework, a term suggesting interesting points from the point of view of target optimization is needed. A standard choice is the expected improvement which is calculated by the formula [29]

$$EI(x) = \sigma(x)(Z\phi(Z) + \varphi(Z)) \tag{B6}$$

$$Z = \left(\mu(x) - l(x_{best})\right)/\sigma(x) \tag{B7}$$

where $\phi(Z)$ and $\varphi(Z)$ are the standard normal cumulative distribution function and probability density function, respectively. The best known value of the target function is denoted by $l(x_{best})$ and, $\mu(x)$ and $\sigma(x)$ are the surrogate prediction for the target and the error at x.

Expected improvement can be interpreted as to be high in regions where either the target takes on large values (it is assumed here that the target is to be maximized) or the model error is large, so that a good target value cannot be excluded.

In Fig. 14f, expected improvement is highest in the white region, which is in the region of feasible points with good target values, but at the same time not too close to the existing data points (where the expected model error is low and therefore sampling there will not improve the model).

References

[1] L T Biegler, I E Grossmann, and A W Westerberg. Systematic methods for chemical process design. Prentice Hall, Dec. 1997. URL: https://www.osti.gov/biblio/293030.

[2] Michael Bortz and Norbert Asprion, eds. Simulation and Optimization in Process Engineering. Elsevier, 2022.

[3] Eric Brochu, Vlad M. Cora, and Nando de Freitas. "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning". In: ArXiv abs/1012.2599 (2010).

[4] P. I. Frazier. A tutorial on Bayesian optimization. 2018. URL: arXiv:1807.02811.

[5] I. M. Sobol. "On the distribution of points in a cube and the approximate evaluation of integrals". In: USSR Computational Mathematics and Mathematical Physics 7.4 (1967), pp. 86–112. ISSN: 0041-5553. DOI: https://doi.org/10.1016/0041-5553(67)90144-9. URL: https://www.sciencedirect.com/science/article/pii/0041555367901449.

[6] Enrique Del Castillo. Process Optimization. A Statistical Approach. International Series in Operations Research Management Science. Springer New York, NY, 2007.

[7] Jasdeep Mandur and Hector Budman. "Robust optimization of chemical processes using Bayesian description of parametric uncertainty". In: Journal of Process Control 24.2

(2014). ADCHEM 2012 Special Issue, pp. 422–430. ISSN: 0959-1524. DOI: https://doi.org/10.1016/j.jprocont.2013.10.004. URL: https://www.sciencedirect.com/science/article/pii/S0959152413002102.

[8] Raoul Heese et al. "Optimized data exploration applied to the simulation of a chemical process". In: Comp. Chem. Eng. 124 (2019), pp. 326–342.

[9] Patrick Ludl et al. "Using machine learning models to explore the solution space of large nonlinear systems underlying flowsheet simulations with constraints". In: Frontiers of Chemical Science and Engineering 16 (2022). DOI: 10.1007/s11705-021-2073-7.

[10] I.E. Grossmann and R.W.H. Sargent. "Optimum design of chemical plants with uncertain parameters". In: AIChE Journal 24 (1978), pp. 1021–1028.

[11] K. P. Halemane and I. E. Grossmann. "Optimal process design under uncertainty". In: AIChE Journal 29 (1983), pp. 425–433.

[12] Kevin McBride and Kai Sundmacher. "Overview of Surrogate Modeling in Chemical Process Engineering". In: Chemie Ingenieur Technik 91.3 (2019), pp. 228–239. DOI: https://doi.org/10.1002/cite.201800091. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.201800091. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cite.201800091.

[13] F. Boukouvala and M.G. Ierapetritou. "Feasibility analysis of black-box processes using an adaptive sampling Kriging-based method". In: Comp. Chem. Eng 36 (2012), pp. 358–368.

[14] F. Boukouvala dn M. G. Ierapetritou. "Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function". In: AIChE Journal 60 (2014), pp. 2462–2474.

[15] Wang and M. Ierapetritou. "A novel feasibility analysis method for black-box processes using a radial basis function adaptive sampling approach". In: AIChE Journal 63 (2017), pp. 532–550.

[16] A. Rogers and M. Ierapetritou. "Feasibility and flexibility analysis of black-box processes Part 1: Surrogate-based feasibility analysis". In: Chem. Eng. Science 137 (2015), pp. 986–1004.

[17] B. Shahriari et al. "Taking the Human Out of the Loop: A Review of Bayesian Optimization". In: Proc. IEEE 104 (1 2016), pp. 148–175.

[18] David Eriksson et al. “Scalable Global Optimization via Local Bayesian Optimization”. In: Advances in Neural Information Processing Systems. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf.

[19] Artur M. Schweidtmann et al. Maximizing the acquisition function of Bayesian optimization to guaranteed Veröffentlicht auf dem Publikationsserver der RWTH Aachen University. 3. International Conference on Machine Learning an AI in (bio)Chemical Engineering, online, 8 Jul 2020 - 8 Jul 2020, July 8, 2020. DOI: 10.18154/RWTH-2020-10117. URL: https://publications.rwth-aachen.de/record/804199.

[20] Maximilian Balandat et al. “BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization”. In: Advances in Neural Information Processing Systems. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21524–21538. URL: https://proceedings.neurips.cc/paper/2020/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

[21] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. “Parallel Bayesian Optimization of Multiple Noisy Objectives with Expected Hypervolume Improvement”. In: Advances in Neural Information Processing Systems. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 2187–2200. URL: https://proceedings.neurips.cc/paper/2021/file/11704817e347269b7254e744b5e22dac-Paper.pdf.

[22] R. Heese and M. Bortz. “Adaptive Sampling of Pareto Frontiers with Binary Constraints Using Regression and Classification”. In: 2020 25th International Conference on Pattern Recognition (ICPR). Los Alamitos, CA, USA: IEEE Computer Society, Jan. 2021, pp. 3404–3411. DOI: 10.1109/ICPR48806.2021.9412217. URL: https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412217.

[23] Stewart Greenhill et al. “Bayesian Optimization for Adaptive Experimental Design: A Review”. In: IEEE Access 8 (2020), pp. 13937–13948. DOI: 10.1109/ACCESS.2020.2966228.

[24] Marcus M. Noack et al. “Autonomous materials discovery driven by Gaussian process regression with inhomogeneous measurement noise and anisotropic kernels”. In: Scientific Reports 10 (2020), p. 17663.

[25] Benjamin J. Shields et al. “Bayesian reaction optimization as a tool for chemical synthesis”. In: Nature 590 (2021), pp. 89–96.

[26] M. Bortz et al. “Multi-criteria optimization in chemical process design and decision support by navigation on Pareto sets”. In: Computers Chemical Engineering 60 (2014), pp. 354–363. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2013.09.015. URL: https://www.sciencedirect.com/science/article/pii/S0098135413002962.

[27] J. Blank. OSY test problem. 2022. URL: https://pymoo.org/problems/multi/osy.html (visited on 05/08/2023).

[28] Raoul Heese et al. “Calibrated simplex-mapping classification”. In: PlosOne 18 (1 2023), e027987.

[29] J. R. Gardner et al. “Bayesian optimization with inequality constraints”. In: ICML’14: Proceedings of the 31st International Conference on International Conference on Machine Lear 32 (2014).