# Neural Networks for Surrogate-assisted Evolutionary Optimization of Chemical Processes

Tim Janus
*Chemical and Biochemical Engineering*
*TU Dortmund University*
Dortmund, Germany
tim.janus@tu-dortmund.de

Anne Lübbers
*Chemical and Biochemical Engineering*
*TU Dortmund University*
Dortmund, Germany
anne.luebbers@tu-dortmund.de

Sebastian Engell
*Chemical and Biochemical Engineering*
*TU Dortmund University*
Dortmund, Germany
sebastian.engell@tu-dortmund.de

***Abstract*—In the chemical industry commercial process simulators are widely used for process design due to their extensive library of models of plant equipment and thermodynamic properties and their ease of use. Most of these simulators compute the steady-states of complex flowsheets, but their models are inaccessible and derivatives with respect to their model parameters are not available. Evolutionary algorithms are a suitable approach for the global optimization of such black-box models, but they require the evaluation of many individuals. Applications to industrial-size case-studies suffer from high computational times where the numerical simulations consume the majority of the time. This contribution proposes the use of neural networks as surrogate models to guide the evolutionary search. These models are trained multiple times during the evolutionary search and are used to exclude nonpromising individuals and to generate candidate solutions. We demonstrate the performance improvement due to the use of the surrogate models for a medium-size case-study of a chemical plant consisting of a reactor, a liquid-liquid separation and a distillation column. The results show that the required number of simulations can be reduced by 50%.**

***Index Terms*—evolutionary algorithms, surrogate models, optimization, neural networks, chemical processes**

ACRONYMS

| | |
|---|---|
| ANN | artifical neural network. |
| API | application programming interface. |
| CG | candidate-generation. |
| DFO | derivative free optimization. |
| DoF | degree of freedom. |
| DS | decision-support. |
| DS+CG | decision-support and candidate-generation. |
| EA | evolutionary algortihm. |
| ES | evolution strategy. |
| MA | memetic algorithm. |
| mae | mean absolute error. |
| MINLP | Mixed Integer Non-Linear Program. |
| MOO | multi-objective optimization. |
| SVM | support-vector machine. |
| SVR | support-vector regression. |

## I. INTRODUCTION

In 2017, the chemical industry contributed 5.7 trillion US dollars to the gross domestic product (GDP) of the world [1]. Chemical production processes implement a sequence of different so-called unit operations, e. i. distillation and reaction, to create a product from a set of raw materials. The design and revamping of chemical plants is usually performed by an interdisciplinary group of experts. A common approach is the use of steady-state process simulations, heuristic rules and expert knowledge to find feasible and economic process configurations. Steady-state process simulation is a mature field with several commercial products in the market. These simulators differ in the accessibility of the models and in their used technique for the numerical solution of their simulations. The applied techniques can be divided into the sequential-modular and equation-oriented simultaneous approaches. Both are explained well in Biegler et al. [2]. With respect to optimization, it is important to distinguish between simulators with accessible and with inaccessible models. Accessible models provide full information about the underlying equations and variables whereas the inaccessible models act as a black-box that only provides simulation results or failure codes. Therefore, inaccessible models require derivative-free optimization methods.

In a recent overview, Asprion et al. [3] postulated requirements for a modern process simulator and gave an overview of in-house and commercial process simulators. This contribution applies Aspen TECH's process simulator Aspen Plus, which is widely used in industry because of its extensive model library and relatively robust solution techniques. However, competition is growing [4] and new process simulators respond to some of the postulated requirements in [3]. To the best knowledge of the authors all commercially available process simulators either do not provide a public application programming interface (API) to couple external optimization methods or are based on inaccessible model libraries. Furthermore, none of the available process simulators provide global optimization capabilities. Therefore, in this work an external derivative-free optimization method is coupled to Aspen Plus.

Several groups have investigated the optimization of chemical processes based on commercial simulators. One approach is to compute the missing derivatives by numerical differentation and to apply gradient-based optimization. Sundberg et al. E. g. [5] optimized a catalytic cracking process with Aspen HYSYS. Cardella et al. [6] optimized a plant for hydrogen liquefaction with the commercial process simulator UniSim [7]. This procedure does not scale well for large numbers of degrees of freedom (DoFs).

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

Other groups applied derivative free optimization (DFO) methods, Ernst et al. [8] developed a genetic algorithm for chemical process optimization using Aspen Plus. Zimmermann et al. [9] extended this work and investigated how a diverse generation of individuals can improve the convergence of multi-objective optimization (MOO) for chemical processes. Urselmann et al. [10], [11] introduced a memetic algorithm (MA) for chemical process optimization. The MA consists of an evolutionary algortihm (EA) and a local refinement based on nonlinear optimization. The MA was coupled to Aspen Plus [12] to support engineers in the industry in their work on the process design. The MA supports several derivative-free optimization methods as local solvers. Further, an extension for MOO [13] was developed. Janus et al. [14] found that the use of DFO methods for local refinement, e. g. CMA-ES [15] or the solver NOMAD [16], is not worth the effort in comparison to only using an EA and applied the Aspen Plus internal optimizer for constraint satisfaction. Although this approach performs better than derivative-free alternatives, it has technical limitations, e. g. the fact that an arbitrary cost function cannot be formulated. No in-house or commercial process simulator provides a robust global optimization, and the use of derivative-free optimization methods on large plant models coupled to a flowsheet simulator suffers from computation times which are usually not acceptable.

In the past, surrogate-assisted optimizaton has been used to enhance the convergence of derivative-free optimization approaches [17]–[19]. A common problem here is the generation of training data. Palmer et al. [20] used a small amount of apriori sampled data to train surrogate models to optimize an ammonia synthesis plant. Caballero and Grossmann [21] replaced selected unit operations with surrogate models by training Kriging models based on apriori sampling. Nentwich et al. [22] used support-vector machines (SVMs) for classification and Kriging models for regression as surrogate models to describe the gas solubility in a reactor and the phase behavior of a liquid-liquid mixture.

The framework proposed in this contribution uses surrogate models to reduce the required number of simulations within an optimization based upon black-box models. Two strategies to reduce the number of calls of the simulator have been developed and tested: the exclusion of simulations of nonpromising solutions by a decision-support system and a candidate-generation scheme that generates candidates by a derivative-based optimization on surrogate models. In our approach no data is collected before the optimization, the necessary information for training is collected on the fly while the optimization by the EA is running. The sampled values of the DoFs are widely distributed at first, but then are refined and provide more accurate surrogate models in regions of interest.

The case-study considered in this contribution is a chemical plant consisting of a reactor in which a so-called hydroformylation reaction takes place [23], [24], a liquid-liquid separation [25] by a decanter and a distillation column. Due to the presence of two recycle streams the convergence of the process simulation is not trivial. Some values of the DoFs lead to failed process simulations, and the computational time spent on these is wasted.

Section 2 introduces the method for surrogate-assisted optimization with dynamically adapted neural network models. Section 3 introduces the case-study of the hydroformylation of 1-dodecene to tridecanal. Section 4 presents the results of different variants of the proposed surrogate-enhanced optimization scheme and Section 5 concludes this work and provides ideas for future work.

## II. Method

This contribution extends a MA for chemical process optimization with a decision-support (DS) and a candidate-generation (CG) strategy. Both are based on machine learning techniques to train surrogate models. The extension consists of three parts

1) The on the fly collection of data while optimizing to train the surrogate models.
2) The use of surrogate models as a decision-support system to avoid nonpromising simulations.
3) The generation of solution candidates based on an derivative-based optimization on the trained surrogate models.

Before the description of the three new elements, the extended MA framework is introduced.

### A. Fundamentals: Memetic Algorithm Framework

Chemical process flowsheet optimization with a simulator based on inaccessible models can be formulated as a Mixed Integer Non-Linear Program (MINLP) of the form:

$$\begin{aligned} \min \quad & f(x,y,z) \\ s.t. \quad & e_n(x,y,z) = 0,\ n = 1,\dots,N \\ & n_k(x,y,z) \leqslant 0,\ k = 1,\dots,K \\ & x \in \mathbb{R}^a,\ y \in \mathbb{R}^b,\ z \in \mathbb{N}^c \end{aligned} \tag{1}$$

where $f(x,y,z)$ is an economic cost function, $x$ are state variables of the system which are computed by the process simulator, $y$ are continuous DoFs and $z$ are discrete DoFs. The equality constraints $e_n(x,y,z)$ describe the chemical model of the flowsheet, e. i. mass balances, reaction kinetics. The inequality constraints $n_k(x,y,z) \leqslant 0$ describe process constraints, e. i. the required product purity.

To perform a simulation requires three steps. First the inputs $y$ and $z$ are transferred to the process simulator, then the simulation is invoked, and after a successful simulation the state variables $x$ are received. The simulation fails rather frequently so that no set of state variables $x$ that satisfies the model equations is found. In the failure case neither the cost function $f(x,y,z)$ nor the constraints $n(x)_k < 0$ can be evaluated, i.e. the computational effort of the simulation is wasted.

Figure 1 illustrates a flow chart of the generation loop of the MA. The MA consists of a self-adapting evolution strategy (ES) [26] that is extended by special operators for chemical processes [11]. The offspring generation consists

| Name (unit) | Bounds |
|---|---|
| $T_R(C°)$ | 80:120 |
| $P_R(bar)$ | 15:30 |
| $\tau(h)$ | 0.1:5 |
| $T_E(C°)$ | -5:25 |
| $N_s(none)$ | 4:40 |
| $N_f(none)$ | 2:dyn |
| $D2F(none)$ | 0.6:0.95 |
| $RR(none)$ | 0.1:15 |
| $F_{dmf}(\frac{kmol}{h})$ | 20:100 |
| $F_{dec}(\frac{kmol}{h})$ | 20:100 |

TABLE I
DEGREES OF FREEDOM

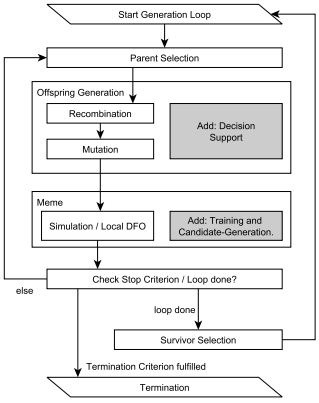


Fig. 1. Memetic algorithm for chemical process optimization, grey boxes show features added to the framework by this contribution

of a recombination operator and a mutation operator. In this contribution, the offspring generation is extended by the DS as shown in Figure 2 (left). The DS is implemented to predict if the generated offspring is promising. If the result is negative, the offspring generation is repeated, otherwise the algorithm continues with the further steps, simulation and CG.

Figure 2 (right) shows the meme that is employed to collect data and to (re)-train the surrogate models. The first training occurs after $N_0$ simulations and thereafter retraining occurs every $N_{it}$ simulations, where both parameters are case-specific. The meme implements the CG by transferring the DoFs of converging simulations to an external derivative-based optimization method that applies the surrogate models. When DS and CG are not enabled, the MA behaves like the original ES.

### B. Data-sampling and Artificial Neural Networks

A database $H$ is generated on the fly while the ES is running. Every simulation result is stored in the database such that the information is accessible to train surrogate models. We use artifical neural networks (ANNs) as surrogate models and access the MATLAB Statistics and Machine Learning Toolbox via the COM-Interface for the training and querying of the surrogate models. ANN is a generic term for a class of data-based models. This contribution uses multilayer feed-forward networks as surrogate models. The structure of a feed-forward network is defined by the number of hidden layers $L_h$ and by the number of nodes in each layer $S_i$ with $i \in \{1, ..., L_h\}$. The number of nodes in the hidden layers are subject to optimization but the nodes in the input and output layer depend on the arguments and on the output of the trained function and are therefore given a priori.

Any state variable $x$, any constraint $n(x, y, z)_i$ or the cost function $f$ in (1), can be predicted by a neural network. In general the influence of the state variables differs strongly between the applications, e. g. the catalyst loss that is very important for the proccess considered here [14] may be of lesser importance or not present at all in other processes. Therefore we propose predictions that are applicable to most case-studies. These are the classification of the simulation outcome $P_{class}$ and a set of $k$ regressions $P_{reg,k}$ that estimates the purity constraints of a process.

### C. Application of the Decision-Support Strategy

The DS indicates if the simulation of a chosen set of DoFs is promising by querying the trained surrogate models. A classifier $P_{class}$ predicts if the simulation of an individual will fail, i. e. the process simulation will not converge. For each purity constraint of the case-study, a regression can be trained that predicts the fraction of the constrained component in the corresponding stream, i. e. the purity of a specific component.

We train a classifier that learns a binary prediction of the simulation outcome (fail, converged). Hereby one represents converged and zero represents failed simulations. The ANN predicts a number between zero and one and the DS predicts a failing simulation if $P_{class} < s_{class}$. In the default case $s_{class}$ is $0.5$ but it can be set differently for a more conservative or more aggressive exclusion behavior.

For predicting the satisfactions of the constraint, we propose a rule that combines an ANN prediction with a dynamic threshold. The threshold depends on the prediction accuracy of the model, measured by the mean absolute error (mae), with respect to the full training data set. In addition, the standard deviation of the mae over the set of feasible data (2) is calculated. A security constant $s$ is defined to describe a confidence interval. These values are recalculated after each training.

$$s_{feas,k} = \sqrt{\frac{1}{N_{feas} - 1} \sum (|t_i - P_{reg,k}| - mae)^2} \quad (2)$$

For each constraints $c_k$ a positive evaluation of (3) indicates a promising simulation. By the subtraction of $mae$, the error of the trained model is taken into account. Under the assumption of a normal distributed model error, a confidence interval of $1.96$ assures a 95 % probability of predicting a solution, which lies exactly on the constraint. The constant $s$ can be adjusted to trade-off the number of false negatives and saved simulations.

$$P_{reg,k}(A) \geqslant c_k - mae_k - s \cdot s_{feas,k}, \ \forall k \in \{1, ..., K\} \quad (3)$$

For the final rule of the DS we propose the combination of the classifier $P_{class}$ with the regressions $P_{req,k}$ that predicts the purity constraints of the process as a conjunction:

$$P = \begin{cases} 1 \text{ if } \forall k : \{P_{reg,k} \geqslant c_k - mae_k - s \cdot s_{feas,k}\} \\ \qquad \wedge P_{class} \geqslant s_{class} \\ 0 \text{ otherwise.} \end{cases}$$

The left side of Figure 2 shows the modification of the offspring generation to include the DS. After the mutation of a newly generated offspring, the decision-support is queried. In case of a negative result, a new offspring is generated by using the same operators, until a promising offspring has been generated. The different magnitudes of computational effort for simulation compared to EA operations render the time

spend in offspring generation negligible even for hundreds of offsprings.

### D. Candidate-Generation by Optimization on Surrogates

The candidate-generation (CG) searches new candidates by derivative-based optimization on surrogate models. We compute points which minimizes the distance of the values of a subset of the constraints to their boundaries and satisfy all other constraints. The motivation for this is that in many cases the optimum with respect to some cost function is located at those boundaries, e. g. on the bound for product purities, a higher purification commonly implies higher energy and/or investment costs for the unit operation.

$$\min \quad \sum_{k=1}^{p} |c_k - P_{reg,k}(y,z)|, \ f \in [1,..,K] \tag{4}$$

$$s.t. \ \ c_i - P_{reg,i}(y,z) \leqslant 0, \ f < i < K \tag{5}$$

$$P_{class}(x,z) \geqslant s_{class} \tag{6}$$

$$y_{min} \leqslant y \leqslant y_{max}, \ z_{min} \leqslant z \leqslant z_{max}$$

The optimization is subject to box-constraints that define the boundaries of the DoFs. The classification (6) increases the probability for the simulation of the generated candidates to converge. The optimization is initialized with a value from the offspring generated by the ES, whereby all DoFs are converted to continuous values. The discrete DoF are treated as continuous inputs of the ANN and the result of the optimization is rounded to the next integers. The new candidate is first evaluated by a simulation of the corresponding flowsheet. By applying surrogate models, the optimization problem can be solved fast, as the trained surrogate models provide derivatives.

The right side of Figure 2 shows a meme that implements the candidate-generation. The meme is invoked after the offspring generation. First, the offspring is simulated and stored in the training database $H$. If the simulation fails, the meme terminates, otherwise the offspring is marked as an initial point. The candidate-generation solves an optimization problem to improve this initial point. In this contribution we apply the MATLAB solver fmincon but the concept is applicable to other gradient-based solvers as well. The generated candidate is then validated by a process simulation and the result of this simulation is stored in the training database $H$.

## III. Case Study

We consider the case-study of the homogenously catalyzed hydroformylation of 1-dodecene in a thermomorphic solvent system. This process has been investigated in two miniplants in the Collaborative Research Center DFG Transregio SFB 63 "Integrated chemical processes in liquid multi-phase systems" InPROMPT. The process produces n-tridecanal in a reactor by hydroformylation of a mixture of the reactants 1-dodecene and the solvents DMF and decane. The reaction is catalyzed by an expensive homogenous rhodium catalyst which has to be recovered from the product stream. The phase behavior of the mixture of DMF and decane is temperature-dependent and this is used to switch between a homogenous mixture in the

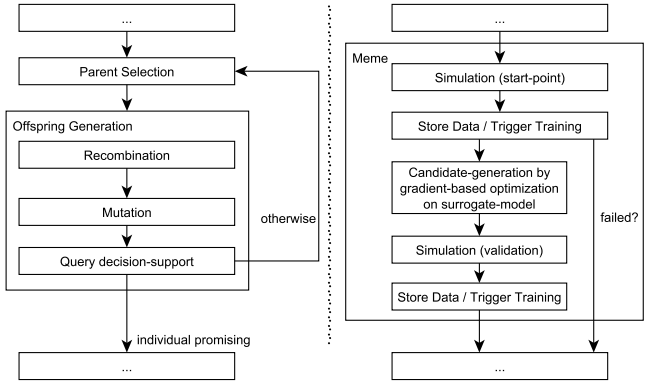


Fig. 2. Offspring generation with decision-support (left), Meme for candidate-generation (right)

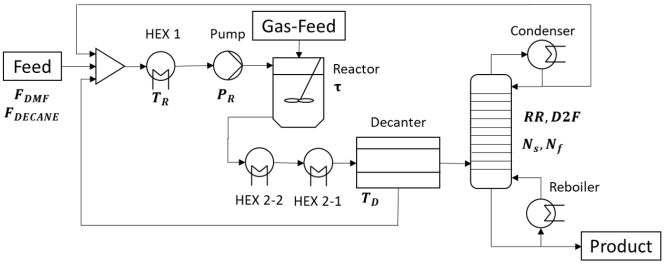


Fig. 3. Process Flowsheet - Hydroformylation of 1-dodecene to tridecanal

reactor and a mixture with two liquid phases [25] where the polar phase containing the catalyst is recycled to the reactor.

Figure 3 shows the process flowsheet and the DoFs are highlighted in bold. A feed stream of the reactant 1-dodecene and the solvents DMF and decane is heated up and pressurized to create a homogenous mixture before it is fed into the reactor. The reactor is also fed with carbon monoxide and hydrogen. Beside the main reaction of 1-dodecene to n-tridecanal, four side reactions are occuring in the reactor [24]. After the reaction, the reacted mixture is cooled down by a cascade of two heat exchangers. The first heat exchanger Hex 2-2 is operated with cooling water and the second heat exchanger Hex 2-1 is operated with ammonia to achieve temperatures below five degrees Celsius. The mixture is fed into the liquid-liquid separator. The polar DMF rich phase contains the catalyst and is fed back into the reactor. The decane rich phase contains the remaining reactant, the product and the byproducts and is fed into the distillation column for product purification. The top stream of the column is fed back to the reactor and a product-rich liquid stream is obtained at the bottom of the column.

The model of the case-study consists of approximately 2200 equations as stated by the equation-oriented user interface of Aspen Plus. As the model library of the process simulator Aspen Plus is not accessible, neither the exact equations nor derivatives are known. An inequality constraint describes the requirement of a product purity of 99 mol %. Table I lists the ten DoFs of the case-study. The flowrate of the solvents

| Param. | Value | Description |
|---|---|---|
| $\mu$ | 10 | Generation size of $\mu$. |
| $\lambda$ | 20 | Offsprings per generation. |
| $\kappa$ | 5 | Maximum age individuals |
| $N_0$ | 500 | Simulations before first training. |
| $N_{it}$ | 200 | Retrain after $N_{it}$ simulations. |
| $s$ | 1.96 | Width of the confidence interval. |
| $s_{class}$ | 0.5 | Threshold for classification. |
| $S_i$ | 5 | Number of nodes per layer. |
| $L_{class}$ | 2 | Hidden layers for classification. |
| $L_{reg}$ | 2 | Hidden layers for regression. |

TABLE II
SHARED META-PARAMETERS OF THE OPTIMIZATION

DMF $F_{dmf}$ and decane $F_{dec}$, the temperature $T_R$, pressure $P_R$ and residence time $\tau$ of the reactor, and the temperature $T_E$ of the decanter are DoFs. In addition, the distillation column provides two integer and two continuous DoFs, the number of theoretical stages $N_s$ and the feed stage $N_f$, and the distillate to feed ratio $D2F$ and the reflux ratio $RR$ (both continuous). The cost function (7) is an indicator of the production cost per ton product.

$$f(x, y, z) = \frac{C_{OU} + C_{Mat}}{8000 \cdot \dot{m}_{Product}} + Penalty \tag{7}$$

$C_{OU}$ equals the sum of the capital and utility costs of the seven unit operations, as proposed by Turton [27].

$$Penalty(v) = \begin{cases} v = 0 & 0 \\ v \neq 0 & 4000 + v \cdot 10000 \end{cases} \tag{8}$$

Equation (8) represents a cost for violating the product purity constraints by an amount of $v$. It is modeled by a linear function with a large constant, such that every solution that violates the constraint leads to higher costs than those that fulfill the constraint. Another source for discontinuities is the optional use of the coolant ammonia in the heat exchanger Hex 2-1. $C_{Mat}$ is very sensitive to the loss of catalyst. The coupling between the different vessels via recycle streams and the structural DoFs in the column lead to a multi-modal economic cost function.

## IV. RESULTS

We investigated the performance of four variants of the optimization method. The reference variant applies an ES without modifications of the offspring generation or the meme. There is a variant each for employing decision-support (DS) and candidate-generation (CG) individually. The fourth variant applies both strategies decision-support and candidate-generation (DS+CG). The parameters in Table II are shared by the variants and they were determined by preliminary studies. The experiments were repeated ten times for each variant.

We compare the experiments with respect to the minimum cost value over the number of required simulations. The training time of the ANN is negligible for the investigated case-study but further investigations of the performance of the derivative-based optimization on surrogate models will be done in future work. Figures 4 and 5 show convergence plots for the different variants. The x-axis indicates the progress of the optimization measured by the number of simulations and the y-axis indicates the cost per ton of product in Euro per ton. As the experiment was repeated ten times for each variant the diagram shows statistic metrics. The solid line represents the average and the dotted lines represents the min and max cost of the experiments. The transparent areas show the standard-deviation of the cost between the runs. The Figures are divided into three regions. The left most region is called training region and it comprises $N_0 = 500$ simulations. After the black dotted line, the surrogate models are trained for the first time. The second region comprises additional 1000 simulations and the line marks the numbers of simulations that are needed to converge the surrogate-based variants. The right most dotted line marks the number of simulations that are needed for the reference ES to converge, 3000.

In the first region, a large variance of the performance of the optimization can be observed, i. e. the algorithm is in exploration mode. Every variant behaves as the ES variant as no surrogate models are trained yet. In the second region, the variances of the surrogate-based strategies shrink strongly, so that the surrogate-assisted methods have all converged at the end of this region. The CG variant does not converge to the best known solution. The reference variant ES needs the entire third region to converge, i .e. DS+CG requires about half the number of simulations. Table IV shows the Wilcoxon tests for the different variants at different numbers of simulations. Hereby a one indicates that the right variant of the test is significantly better than the left variant, e. g. in the first row, DS is significantly better than ES after 1100 simulations. The first row shows that there is a significant improvement for the variant DS+CG over ES starting at 700 simulations. DS is better than ES at 1100, 1200, 1500 and 3000 simulations, but not for 1300 and 1400 simulations. CG is not better than the reference case ES but becomes worse at 3000 simulations as shown by the last row. The following two rows show that DS+CG is at some points better than DS and also better than CG starting at simulation 700. DS+CG found a better solution than the best known solution found in previous work by Janus et al. [14]. Table IV summarizes the performance of the ten experiments per variant and shows that DS and DS+CG converge faster than the reference case but CG is not capable to converge to the best known solution even after 12.000 simulations.

Table V shows the average ratio of failed/infeasible/feasible simulations for the different variants and gives an indication for the reason of CG. In the reference case, 25 % of the simulations fail. The DS halves the number of failed simulations. However, when applying CG, the number of failed simulation almost doubles, while the combination DS+CG reduces the number of failed simulations by 8 % with respect to the reference case.

At first sight, the worse performance of the CG variant

| | Test after x Simulations | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Test** | **600** | **700** | **800** | **900** | **1000** | **1100** | **1200** | **1300** | **1400** | **1500** | **3000** |
| ES vs. DS | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| ES vs. CG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ES vs. DS+CG | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| DS vs DS+CG | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| CG vs DS+CG | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| CG vs. ES | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

TABLE III
RESULTS OF WILCOXON TEST - A ONE INDICATES THAT THE MEDIAN OF THE RIGHT TEST VARIANT IS SIGNIFICANTLY BETTER THAN THE MEDIAN OF THE LEFT. TESTS ARE ON A CONFIDENCE INTERVAL OF 95%.

| | Costs [€ / T] | | | [€ / T] |
|---|---|---|---|---|
| **Variant** | **Min.** | **Avg.** | **Max.** | **Std. deviation** |
| ES | 3469 | 3484 | 3512 | 12.9 |
| DS | 3457 | 3467 | 3474 | 5.1 |
| CG | 3486 | 3556 | 3623 | 28.1 |
| DS+CG | 3438 | 3463 | 3480 | 6.5 |

TABLE IV
SUMMARY OF THE RESULTS - AFTER 12.000 SIMULATIONS

| **Variant** | **failed** | **infeasible** | **feasible** |
|---|---|---|---|
| ES | 25 % | 51 % | 25 % |
| DS | 12 % | 50 % | 38 % |
| CG | 46 % | 36 % | 18 % |
| DS+CG | 17 % | 58 % | 25 % |

TABLE V
AVERAGE RATIO OF FAILED, INFEASIBLE AND FEASIBLE SIMULATIONS

is surprising, but without filter to reduce the number of failing simulations, CG proposes to visit areas of the search space with a high probability of simulation failures. The worse number of infeasible simulations with DS+CG is not a disadvantage because it shows that the strategy explores the boundaries of the feasible space more aggressively than ES or DS alone which makes sense as the best solutions often are at these boundaries. Also the interaction of an ES and the candidate-generation may disturb the self-adaptation of the underlying ES.

## V. CONCLUSION AND FUTURE WORK

Global optimization for chemical process flowsheets on the basis of commercial process simulators, e. g. Aspen Plus, is not yet applied in industry, due to high computational times. This contribution is a building block to achieve a reduction of the time needed for the optimization of complex flowsheets from weeks to days for industrial-sized case studies. Derivative-free optimization methods require a large number of simulations which are time consuming, therefore the reduction of the number of simulator calls is crucial.

Machine learning methods can be used to train surrogate models, e. g. artifical neural networks (ANNs), for classification and regression. These can be used for filtering simulator calls and for derivative-based optimization.

In this contribution we applied on the fly training of ANNs as surrogate-models and introduced two strategies, namely decision-support (DS) and candidate-generation (CG), that are based on the classification of the simulation outcome and a derivative-based optimization based on the surrogate-models.

These strategies speed up the convergence of EA-based optimization of flowsheets using an external process simulator. For the case-study of the hydroformylation of 1-dodecene to n-tridecanal it was shown that the proposed strategies halve the number of required simulations.

We observed that caution is necessary when coupling a self-adapting optimization method with an optimization-based candidate-generation because the extrapolation of the latter may lead into search regions that have a high rate of simulation failures and the candidate-generation may only visit a sub space of the entire search-space. In this case-study, the training time of the ANN was small compared to the simulation time, and training is only performed every 200 simulations, therefore the time needed to train the ANN was not a concern.

The investigated case-study has ten DoFs but in industrial applications typically thirty or more DoFs are present [19]. In this case, the time needed to train the ANN as well as the required number of data points may become an issue. A solution for this challenge can be dimensionality reduction, such that the surrogate models are trained for the different unit operations individually. Similar modularization techniques were proposed by Caballero et al. [21] or Quirante et al. [28] and are promising directions for future work.

In industrial case-studies multiple constraints are present, and they are often not independent and possibly contradictory. A case-study with multiple and partly contradicting constraints may lead to a search region that contains multiple local optima. With a set of local optima the derivative-based solver may need multiple intial points or an iterative approach to find the optimum. The solver NOMAD [16] proposes the usage of different barrier strategies, e.g. progressive and extreme, to tackle problems with multiple constraints in an iterative

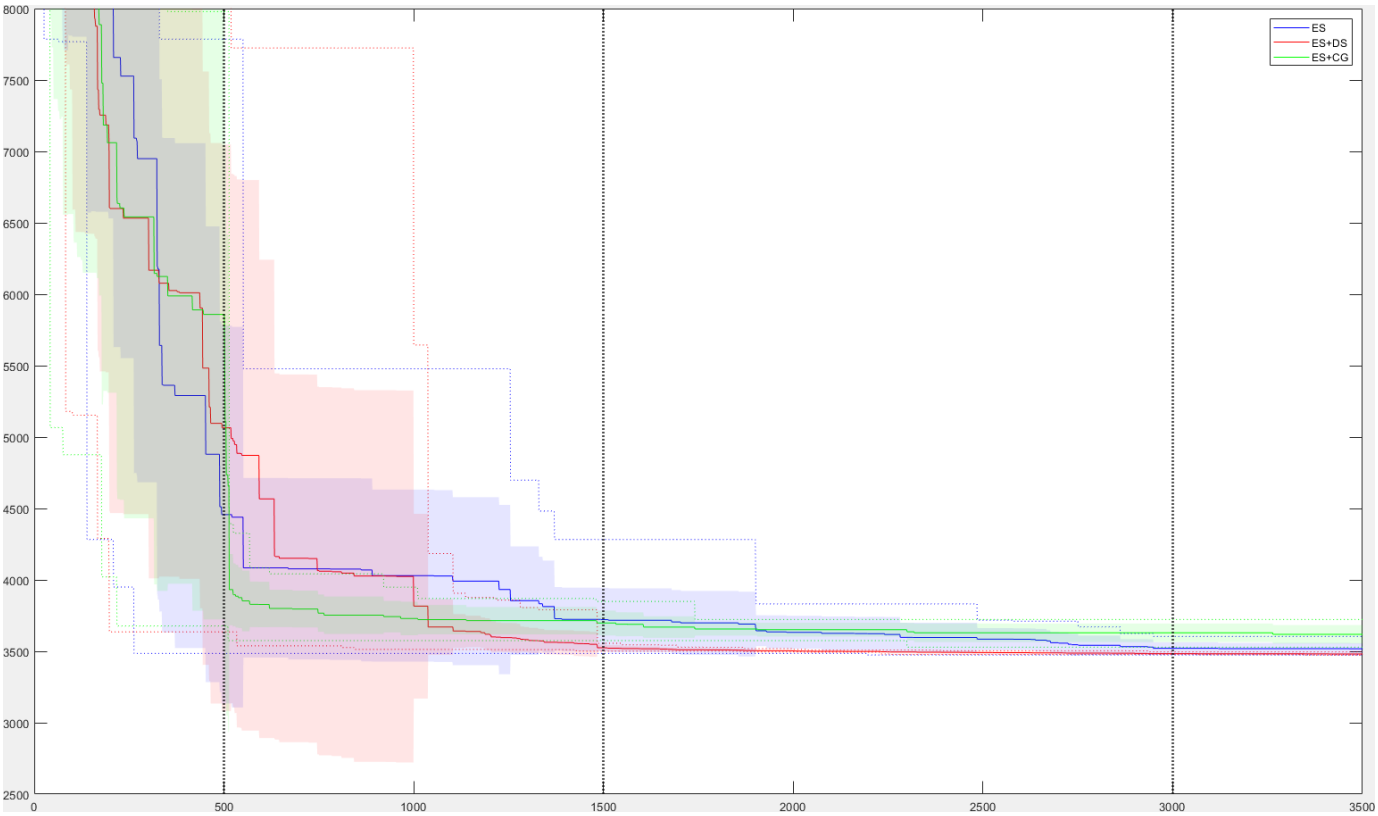


Fig. 4. Convergence plot for ten repeats, solid lines represent average, dotted lines represent the min and max and the transparent region shows the standard-deviation of the experiments best found solution - comparing the surrogate-assisted variants with the reference variant ES (blue)

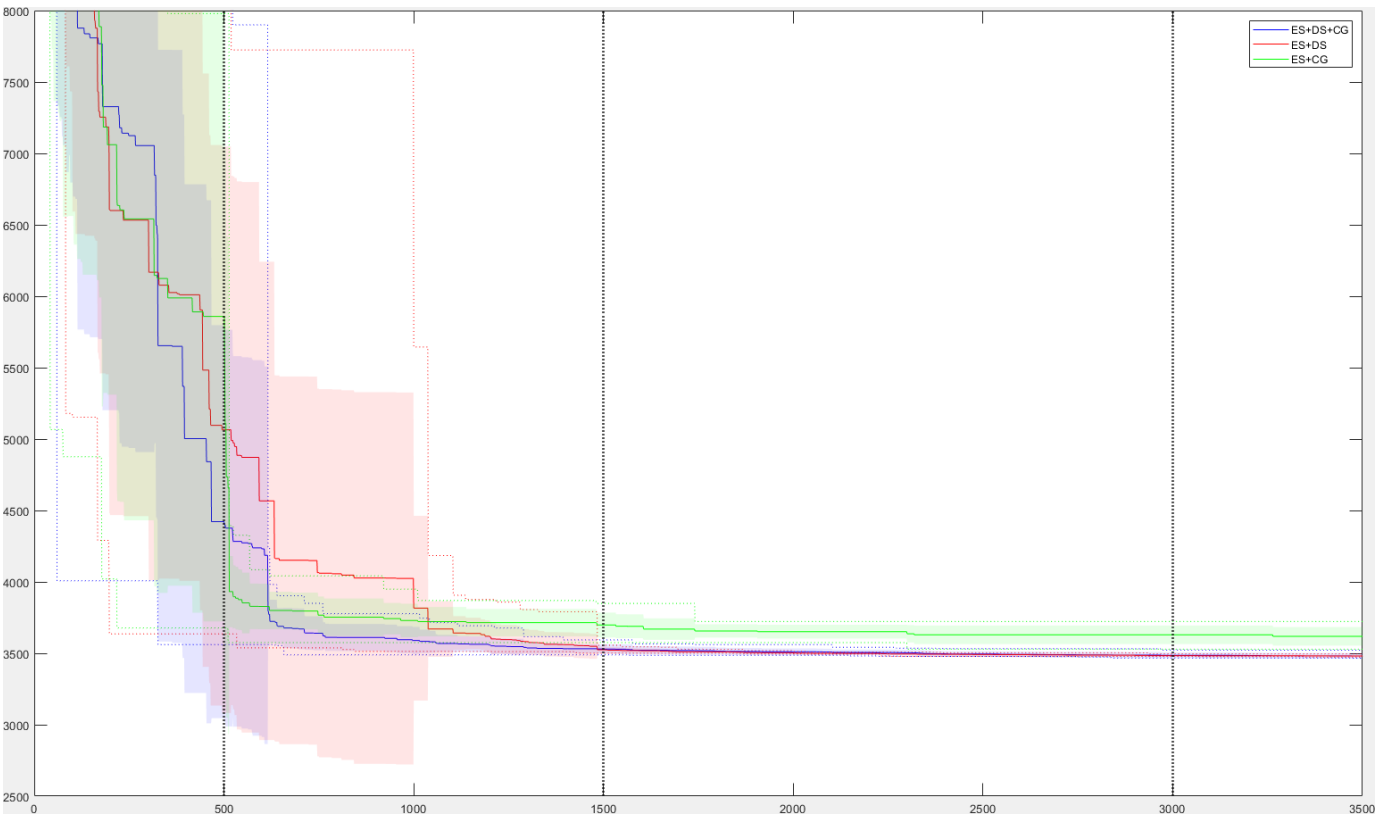


Fig. 5. Convergence plot for ten repeats, solid lines represent average, dotted lines represent the min and max and the transparent region shows the standard-deviation of the experiments best found solution - Comparing the surrogate-assisted variants with the combination of both strategies (blue)

manner. Pan et. al [29] proposes an evolutionary algorithm to generate a population of candidate solutions by applying evolutionary MOO on surrogate models. Both approaches are promising directions for future research. Besides ANNs, other data-based model types can be employed. SVMs and support-vector regression (SVR)s are based on a hyperplane that separates classes or allows a distance measure of arbitrary data points. This hyperplane may be projected to the design-space, e. g. the 2d-space of distillate to feed and reflux ratio of a distillation column. By this, the landscape which is enclosed by the projected hyperplane can be validated by an engineer and thereby increases the trust in the optimization.

Parallelization is a promising option to further reduce the computation time [17]. It can be used to evaluate the candidates in the population or to train several surrogate models in parallel.

## Acknowledgment

Gefördert durch die Deutsche Forschungsgemeinschaft (DFG) - TRR 63 "Integrierte chemische Prozesse in flüssigen Mehrphasensystemen" (Teilprojekt D1) - 56091768. funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - TRR 63 "Integrated Chemical Processes in Liquid Multiphase Systems (subprojects D1) - 56091768

The authors thank Mirko Skiborowski and Corina Nentwich for the stimulating discussions. Additionally, the support of Maximilian Cegla, Christian Klanke, Clemens Lindscheidt, Marina Rantanen-Modeer and Vassilios Yfantis has been highly appreciated.

## References

[1] Oxford Economics, "The Global Chemical Industry: Catalyzing Growth and Addressing Our World's Sustainability Challenges," no. March, pp. 1–29, 2019.

[2] L. Biegler, I. Grossmann, and A. Westerberg, *Systematic methods for chemical process design*. Springer-Verlag New York, 12 1997.

[3] N. Asprion and M. Bortz, "Process Modeling, Simulation and Optimization: From Single Solutions to a Multitude of Solutions to Support Decision Making," *Chemie-Ingenieur-Technik*, vol. 90, no. 11, pp. 1727–1738, 2018.

[4] aspentech, "Annual report 2017." http://ir.aspentech.com/static-files/9d13c31e-1e32-409f-b681-3a989a7c937e, December 2017.

[5] J. Sundberg, S. Standl, T. von Aretin, M. Tonigold, S. Rehfeldt, O. Hinrichsen, and H. Klein, "Optimal process for catalytic cracking of higher olefins on ZSM-5," *Chemical Engineering Journal*, vol. 348, no. October 2017, pp. 84–94, 2018.

[6] U. Cardella, L. Decker, J. Sundberg, and H. Klein, "Process optimization for large-scale hydrogen liquefaction," *International Journal of Hydrogen Energy*, 2017.

[7] I. Thomas, *A Process Unit Modeling Framework within a Heterogeneous Simulation Environment*, vol. 29. Elsevier B.V., 2011.

[8] P. Ernst, K. Zimmermann, and G. Fieg, "Multi-objective Optimization-Tool for the Universal Application in Chemical Process Design," *Chemical Engineering and Technology*, vol. 40, no. 10, pp. 1867–1875, 2017.

[9] K. Zimmermann and G. Fieg, "Development of a Diversity-Preserving Strategy for the Pareto Optimization in Chemical Process Design," *Chemie-Ingenieur-Technik*, vol. 89, no. 10, pp. 1297–1305, 2017.

[10] M. Urselmann, S. Barkmann, G. Sand, and S. Engell, "A memetic algorithm for global optimization in chemical process synthesis problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 659–683, 2011.

[11] M. Urselmann and S. Engell, "Design of memetic algorithms for the efficient optimization of chemical process synthesis problems with structural restrictions," *Computers and Chemical Engineering*, vol. 72, pp. 87–108, 2015.

[12] M. Urselmann, C. Foussette, T. Janus, S. Tlatlik, A. Gottschalk, M. T. Emmerich, S. Engell, and T. Bäck, "Selection of a DFO Method for the Efficient Solution of Continuous Constrained Sub-Problems within a Memetic Algorithm for Chemical Process Synthesis," *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference - GECCO '16*, pp. 1029–1036, 2016.

[13] M. Urselmann, T. Janus, C. Foussette, S. Tlatlik, A. Gottschalk, M. T. Emmerich, T. Bäck, and S. Engell, "Derivative-Free Chemical Process Synthesis by Memetic Algorithms Coupled to Aspen Plus Process Models," *Computer Aided Chemical Engineering*, vol. 38, pp. 187–192, 2016.

[14] T. Janus, M. Cegla, S. Barkmann, and S. Engell, "Optimization of a hydroformulation process in a thermomorphic solvent system using a commercial steady-state process simulator and a memetic algorithm," in *Computer Aided Chemical Engineering*, vol. 46, pp. 469–474, Elsevier Ltd, 2019.

[15] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[16] S. Ebastien and L. E. Digabel, "Algorithm 909 : NOMAD : Nonlinear Optimization with the," vol. 37, no. 4, pp. 1–15, 2011.

[17] R. T. Haftka, D. Villanueva, and A. Chaudhuri, "Parallel surrogate-assisted global optimization with expensive functions a survey," *Structural and Multidisciplinary Optimization*, vol. 54, no. 1, pp. 3–13, 2016.

[18] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling , feasibility analysis , and optimization : A," *Computers and Chemical Engineering*, 2017.

[19] K. McBride and K. Sundmacher, "Overview of Surrogate Modeling in Chemical Process Engineering," *Chemie-Ingenieur-Technik*, vol. 91, no. 3, pp. 228–239, 2019.

[20] K. Palmer and M. Realff, "Optimization and validation of steady-state flowsheet simulation metamodels," *Chemical Engineering Research and Design*, vol. 80, no. 7, pp. 773–782, 2002.

[21] J. A. Caballero and I. E. Grossmann, "An algorithm for the use of surrogate models in modular flowsheet optimization," *AIChE Journal*, vol. 54, pp. 2633–2650, oct 2008.

[22] C. Nentwich and S. Engell, "Surrogate modeling of phase equilibrium calculations using adaptive sampling," *Computers & Chemical Engineering*, vol. 126, pp. 204–217, 2019.

[23] Y. Brunsch and A. Behr, "Angewandte Temperature-Controlled Catalyst Recycling in Homogeneous Transition-Metal Catalysis : Minimization of Catalyst Leaching **," *Angewandte Chemie International Edition*, pp. 1586–1589, 2013.

[24] G. Kiedorf, D. M. Hoang, A. Müller, A. Jörke, J. Markert, H. Arellano-garcia, A. Seidel-morgenstern, and C. Hamel, "Kinetics of 1-dodecene hydroformylation in a thermomorphic solvent system using a rhodium-biphephos catalyst," *Chemical Engineering Science*, vol. 115, pp. 31–48, 2014.

[25] V. A. Merchan and G. Wozny, "Comparative Evaluation of Rigorous Thermodynamic Models for the Description of the Hydroformylation of 1-Dodecene in a Thermomorphic Solvent System," *Industrial and Engineering Chemistry Research*, vol. 55, no. 1, pp. 293–310, 2016.

[26] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[27] R. Turton, R. C. Bailie, W. B. Whiting, and J. A. Shaeiwitz, *Analysis, synthesis and design of chemical processes*. Pearson Education, 2008.

[28] N. Quirante and J. A. Caballero, "Large scale optimization of a sour water stripping plant using surrogate models," *Computers and Chemical Engineering*, vol. 92, pp. 143–162, 2016.

[29] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 1, pp. 74–88, 2019.