Formulating data-driven surrogate models for process optimization

Ruth Misener [a,1], Lorenz Biegler[b,1]
[a] Department of Computing, Imperial College London, London, UK SW7 2AZ
[b] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA

Abstract Recent developments in data science and machine learning have inspired a new wave of research into data-driven techniques for mathematical optimization. This paper first considers two essential conditions for integrating surrogates into process optimization and discusses achieving those conditions. Next, we consider two perspectives for developing process engineering surrogates: a *surrogate-led* and a *mathematical programming-led* approach. These data-driven surrogate models must be integrated into a larger process optimization problem, so this paper next discusses the *verification problem*, i.e., checking that the optimum of the surrogate corresponds to the optimum of the truth model. Finally, we consider relevant software.

Keywords
Optimization formulations, surrogate modeling, data-driven optimization, optimization under uncertainty, software tools

Introduction

Both data-driven techniques and mathematical optimization have been pillars of process systems engineering (PSE) since its inception (Sargent, 1972; Pistikopoulos et al., 2021). But recent developments in data science and machine learning have inspired a new wave of research into data-driven techniques for mathematical optimization (Ning and You, 2019). This research belongs to larger efforts at the intersection of data science and PSE (Qin and Chiang, 2019; Shang and You, 2019; Tsay and Baldea, 2019; Schweidtmann et al., 2021; Thebelt et al., 2022).

Recent research integrating data-driven techniques into mathematical optimization includes: derivative-free optimization (Rios and Sahinidis, 2013), hybrid data-driven / mechanistic modeling (Von Stosch et al., 2014; Boukouvala et al., 2016), using surrogate models in optimization (Bhosekar and Ierapetritou, 2018; McBride and Sundmacher, 2019), and optimization under uncertainty using data-driven techniques (Ning and You, 2019; Thebelt et al., 2022).

This paper first considers two essential conditions for integrating surrogates into process optimization and discusses achieving those conditions. Next, we consider two perspectives for developing process engineering surrogates:

- A *surrogate-led* perspective first selects an appropriate surrogate, for instance a Gaussian process for its statistical properties, and then develops effective optimization formulations for that particular surrogate model.
- A *mathematical programming-led* perspective selects a specific surrogate model based on its desired optimization properties, e.g. linearity.

The *surrogate-led* approach selects a particular data-driven surrogate model and then develops an optimization formulation for that surrogate model. The second, *mathematical programming-led*, perspective is important for PSE domains where the optimization problems being solved are so difficult that a surrogate model must conform to specific properties.

These data-driven surrogate models must be integrated into a larger process optimization problem, so this paper next discusses the *verification problem*, i.e., checking that the optimum of the surrogate corresponds to the optimum of the truth model. Finally, we consider relevant software.

Surrogates for process optimization

We mention two essential conditions for process optimization surrogates:

Condition 1. Surrogate model accuracy can be enforced by constraining them to desired space so that they do not extrapolate with large errors.

Condition 2. Overfitting data-driven models must be avoided for surrogate models embedded into optimization problems.

These conditions are tied to the stability of optimal solutions, which are partially characterized by Lipschitz continuity (LipC) for input/output relations and data uncertainties. Moreover, the LipC property depends strongly on surrogate model parameters with bounded confidence regions that result from data uncertainties. Otherwise, the optimum determined from a surrogate model is unlikely to be robust to changes in the problem data or input conditions.

Levels of Surrogacy

Truth models for plant system-wide equations can be classified at the physical property, unit and system or plant

[1] Corresponding author. Email: `r.misener@imperial.ac.uk`, `lb01@andrew.cmu.edu`.

levels, through the following form:

$$\begin{aligned}
&y = g(x_{sys}, x_{un}, x_{pp}, u_{sys})\\
&0 = f_{sys}(x_{sys}, x_{un}, x_{pp}, u_{sys}, p_{un})\\
&0 = f_{un}(x_{sys}, x_{un}, x_{pp}, p_{un})\\
&0 = f_{pp}(x_{pp}, x_{un})
\end{aligned} \tag{1}$$

where y and u are system outputs and input vectors, x_l are state variables at level l. At the plant level, the abstraction is the most generalized and seeks to substitute the entire system of Equations (1) with a single surrogate model,

$$\begin{aligned}
&y = g_{surr}(u_{sys}) = g(x_{sys}, u_{sys}) + \varepsilon_y,\\
&0 = f_{sys}(x_{sys}, u_{sys}),
\end{aligned}$$

where ε_y is the approximation error vector of the surrogate model. This approach avoids calculating mass or energy balances and attempts to determine a simulation topology solely based upon the simulation input variables u_{sys}. Such models have the following advantages: First, plant surrogate models may have few input variables (degrees of freedom) for the entire plant and they may lead to high fidelity interpolative plant models. Moreover, they are solvable with simpler derivative-free optimization solvers. On the other hand, extrapolating these surrogates will likely violate conservation laws and other first principle relations. Moreover, these surrogates are not reusable for related cases, as they need to be reconstructed for every specific plant case.

Instead, surrogate models at the unit level can be linked to form a plant model. At the unit level, these represent an intermediate level of surrogacy that satisfy the overall mass and energy balances of the plant, although these surrogate models are not designed to account for conservation or other first principle laws within the unit. The underlying system of Equations (1) is reduced to the form:

$$\begin{aligned}
&y = g(x_{sys}, x_{un}, u_{sys})\\
&0 = f_{sys}(x_{sys}, x_{un}, u_{sys}, p_{un})\\
&0 = f_{surr,un}(x_{sys}, x_{un}, p_{un}) = f_{un}(x_{sys}, x_{un}, p_{un}) + \varepsilon_{un}
\end{aligned} \tag{2}$$

where ε_{un} is the approximation error vector of the surrogate model. Such models have the following features. Unit-level surrogate models with few input variables (degrees of freedom) based on the unit structure, and they can lead to high fidelity interpolative unit models. Moreover, conservation laws hold at plant level and these surrogates are reusable for new plant-level cases. But extrapolating the surrogate model may violate first principle relations and conservation laws in the unit, and surrogate extrapolation errors may lead to convergence failures at the plant level. Also, plant-wide optimization solution of embedded unit surrogates requires mid-scale optimization solvers, which are more computationally expensive.

Finally, surrogate models describing first-principles relations within common process units, e.g., flash calculations and physical properties, represent the lowest level of abstraction, leaving the rest of the unit- and plant-level model equations in place. Equation (1) becomes:

$$\begin{aligned}
&y = g(x_{sys}, x_{un}, x_{pp}, u_{sys})\\
&0 = f_{sys}(x_{sys}, x_{un}, x_{pp}, u_{sys}, p_{un})\\
&0 = f_{un}(x_{sys}, x_{un}, x_{pp}, p_{un})\\
&0 = f_{surr,td}(x_{pp}, x_{un}) = f_{pp}(x_{pp}, x_{un}) + \varepsilon_{pp}
\end{aligned} \tag{3}$$

where ε_{pp} is the approximation error for the surrogate model. These rigorous first-principle or "physics-based" models are now integrated within the unit models and the rest of the process is then modeled with rigorous unit- and plant-level equations. These equations, along with the physics-based surrogates, form an equation-oriented model, which must be solved with a large-scale optimization solver. Such models have the following features. These surrogate models with few input variables (degrees of freedom) for subunit models and can lead to high fidelity interpolative subunit models. Moreover, conservation laws hold at unit and plant level and these surrogates are reusable for new plant-level and unit-level cases. On the other hand, the optimization problem based on Equation (3) must be solved with large-scale optimization solvers, in order to evaluate plant-level and unit-level cases efficiently. Moreover, surrogate model extrapolation errors can lead to convergence failures at both unit and plant levels. Ma et al. (2022) and Goldstein et al. (2022) explore the performance of surrogates at these different modeling levels.

Optimization models of data-driven surrogates *Regression based on polynomials:* Developing polynomial regression models is common and has been extensively reviewed (Bhosekar and Ierapetritou, 2018). Some newer ideas in the process systems engineering literature adaptively select which polynomial regressors to use, e.g., ALAMO (Wilson and Sahinidis, 2017), and build a polynomial surrogate explicitly for global optimization, e.g., ARGONAUT (Boukouvala and Floudas, 2017).

Neural networks: With machine learning, data-driven models are traditionally based on (deep) neural nets are usually trained using variants of stochastic gradient descent algorithm (SGD), such as adaptive moment estimation (Adam) have been very successful in practice but do not have favorable convergence properties (e.g., global, superlinear) that are standard for modern nonlinear optimization algorithms. Since they are very successful in fitting DNNs (training), it is an open question whether how these DNNs and SGD algorithms compete with modern, large-scale optimization strategies for process engineering models.

This disconnect between existing theoretical analyses of gradient-based algorithms and the practice of training deep neural networks (DNN) has recently been explored by Zhang et al. (2022). Here it is shown that neural network weights, with either differentiable (e.g., sigmoidal) or non-differentiable (ReLU) activation functions, often do not converge to stationary points of the loss function, even though stable convergence to minimum loss, approaching zero, is observed. This stabilization on training loss can be explained through a convergence proof on the DNN weight distributions.

These caveats lead us to take a mathematical programming view of the optimization models applying to several types of data-driven surrogates. Some of the earliest formulations for optimizing over neural networks are big-M mixed-integer programming formulations relevant to ReLU activation functions (Lomuscio and Maganti, 2017; Fischetti and Jo, 2018). Grimstad and Andersson (2019) considered tightening big-M parameters for optimization over NN with ReLU activation functions. Alternative mixed-integer formulations for ReLU activation functions include adding cuts representing the convex hull of a single neural network node (Anderson et al., 2020) or a partition-based formulation that includes a subset of the convex hull constraints (Tsay et al., 2021).

Other mathematical programming formulations for ReLU activation functions include a semidefinite relaxation (Raghunathan et al., 2018) and a quadratic relaxation derived through applying the S-Lemma (Fazlyab et al., 2019). For ReLU activation functions of the form $(r = \max(0,x))$, Yang et al. (2021) considered three formulations:

1. embedded, e.g., $r = a/(1+e^{-ax}) \approx \max(0,x)$, which needs smoothing for a nonlinear optimization solver,

2. binary variables to handle the max functions within a mixed-integer linear strategy,

3. complementarity formulations with $r = x+y$, $0 \geq y \perp x+y \geq 0$ using basic (not relaxed) complementarity formulations within a nonlinear optimization problem.

Yang et al. (2021) also consider convex hull constraints used to prevent extrapolation of neural networks. In addition, Ma, Sahinidis and coworkers Ma et al. (2022) consider data-driven approaches to minimize energy cost of extractive distillation with truth models from an Aspen simulator. First, they apply surrogate-based optimization using ALAMO's generalized linear models, and then also consider neural network models with the ReLU activation function. Second, the use DFO to optimize problems directly using simulation results. In a detailed comparison, they observed that ALAMO performs well for less complex systems, while the ReLU network performs better for complex ones. On the other hand, the most effective performance was obtained with DFO using smooth penalty functions.

Other types of neural network activation functions have also been considered, including binarized NN (Khalil et al., 2018) and a reduced space formulation for nonlinear smooth activation functions (Schweidtmann and Mitsos, 2019). But there is lots more to do in this research area, for instance transforming nonsmooth ReLU activation functions using smoothed mathematical optimization problems with complementarity constraints.

Regression trees: Mixed-integer programming formulations of gradient-boosted regression trees are from Mišić (2020) and Mistry et al. (2021). These formulations are available, for example, in the black-box optimizer ENTMOOT (Thebelt et al., 2021, 2022) and the formulation tool OMLT (Ceccon et al., 2022).

Gaussian processes: Optimization over Gaussian processes can be managed in many different ways. First, Gaussian processes are a natural fit for robust optimization strategies (Bertsimas et al., 2010a,b; Bogunovic et al., 2018; Wiebe and Misener, 2021; Wiebe et al., 2022). Of course, there are infinite possible functions in a Gaussian process, so (depending on the application) we can either optimize over the mean while integrating some notion of uncertainty or use pathwise-conditioning to sample from the Gaussian process posterior (Wilson et al., 2020). Schweidtmann et al. (2021) have also developed a reduced space formulation for global optimization.

Selecting a data-driven surrogate based on its optimization properties

The preceeding discussion develops optimization models for data-driven surrogates. But sometimes we wish to focus first on the needs of an application and then chose a corresponding surrogate. Early, foundational work initializing this line of inquiry explored how individual surrogates may fit into larger decision-making problems (Palmer and Realff, 2002; Caballero and Grossmann, 2008; Henao and Maravelias, 2011). Some of the other ideas in this area include: developing a decision tree with desired properties (Bertsimas and Dunn, 2017), selecting ReLU neural networks to expand the applicability of multi-parametric programming (Katz et al., 2020), and Developing a neural network with desired properties (Tsay, 2021).

There has been significant work solving optimization problems based on hybrid data-driven / mechanistic models and the consequences of these algorithms for surrogate models (Eason and Biegler, 2016; Bajaj et al., 2018; Eason and Biegler, 2018; Kim and Boukouvala, 2020). The next subsections further develop these ideas. For optimization, all models are imperfect, but some are useful. As a result, surrogate models ranging from first principles to shortcut models to data-driven models are widely applied in the context of optimization studies.

Strategies for surrogate-based optimization

Derivative-free optimization methods are generally formulated for unconstrained optimization problems, i.e., $min_{x\in\mathbb{R}^n} f(x)$. These methods are either stochastic or deterministic in nature. The former methods are based on opportunistic sampling and selection algorithms, which converge asymptotically, but offer no guarantees for a finite number of samples. On the other hand, deterministive methods are based on generalized pattern searches, which adapt themselves to the response surface, and often provide guarantees to convergence of local optimality of surrogate model. These include the DFO and NOMAD.DIRECT solvers described in Conn et al. (2009). In addition, surrogate models with optimally chosen basis functions are created by the ALAMO solver, which applies an MIP formulation, linear least squares and modified AIC criteria.

For the development of constrained optimization formulations with data-driven models, optimization strategies can be applied with embedded surrogate models, which substitute for high-fidelity (or ”truth”) models are widely performed in process engineering. There, the high-fidelity model is replaced over the entire optimization space with a surrogate model such as polynomial, DNN and Kriging model(Bhosekar and Ierapetritou, 2018). While this approach no longer requires additional evaluation of the high-fidelity model once the surrogate model is established, it is likely that the optimization will lead to extrapolation of the surrogate model. And these extrapolation errors for the surrogate can lead to convergences failures, or termination at a point that is not the optimum of the high-fidelity model. Consequently, it is challenging to maintain the accuracy of the surrogate model over the entire optimization space.

Conditions where the optimum of the surrogate model corresponds to the optimum of the truth model.

For global optimization there are a number of DFO methods with convergence guarantees “in the limit” (see Huyer and Neumeier (2008)). However, unlike conventional methods based on spatial branch and bound search, they do not provide lower bounds and certificates for global solutions.

On the other hand, for local optimization, this challenge can be addressed through locally approximated surrogate models that are updated with recourse to the truth model, as part of the optimization strategy. Starting from unconstrained approaches by Fahl and Sachs (2003) and Conn et al. (2009), Eason and Biegler (2016, 2018) developed the trust region filter (TRF) method for constrained optimization that samples from ”truth models” which have smooth input-output properties. The TRF method iteratively solves sub-problems with local surrogate models under trust region constraints along with stabilizing filter methods. Applied to any surrogate model with smooth input-output properties, this approach guarantees convergence to the target problem with truth models and requires few function evaluation of these box models. In addition, if the derivatives of these high-fidelity model are available, then simpler surrogate models can be used along with first order corrections (FOC). FOC approaches have been to surrogate large models that are expensive in computation such as aerodynamics and pressure swing optimization (Alexandrov et al. (1998); Agarwal and Biegler (2013)) with the derivative information of the original high-fidelity model.

The trust region filter (TRF) method for surrogate-based optimization) has rigorous guarantees of convergence to local optimality for the truth model. These are based on DFO properties in Conn et al. (2009) and are independent of the choice of the surrogate model. On the other hand, performance of the TRF method depends on accuracy of the surrogate model sampling of the truth model is required as TRF proceeds. The TRF method been applied in a number of surrogate-based optimization case studies, where direct optimization of the truth models was prohibitive. These include periodic adsorption processes (Agarwal and Biegler, 2013)), air-fired and oxycombustion power plants (Dowling et al., 2016), surrogate equations of state and MWD models for polymerization (Eason et al., 2018; Kang et al., 2019), hear exchanger networks with surrogated detailed exchanger models (Kazi et al., 2020), real-time optimization of refineries (Chen et al., 2021) and optimization of benzene chlorination processes (Yoshio and Biegler, 2020). Future research will deal with relaxing the TRF properties to include model mismatch and noise from surrogate models, based on work on ε-exact models by Biegler et al. (2014).

We close by mentioning some of the software in this area. The tools ALAMO (Wilson and Sahinidis, 2017), ARGONAUT (Boukouvala and Floudas, 2017), and ENTMOOT (Thebelt et al., 2021, 2022) seek to optimize over a combination of mechanistic, model-based equations and data-driven surrogates. Codes like reluMIP (Lueg et al., 2021) and OMLT (Ceccon et al., 2022) allow users to directly add data-driven surrogates to Pyomo (Bynum et al., 2021).

Conclusions

This paper has offered brief glimpse into formulating data-driven surrogates models as mathematical programming formulations for process optimization. There is still much work to be done in this area.

References

Agarwal, A. and L. T. Biegler (2013). A trust-region framework for constrained optimization using reduced order modeling. *Optim. Eng. 14, 1*, 3–36.

Alexandrov, N. M., J. E. Dennis, R. M. Lewis, and V. Torczon (1998). A trust-region framework for managing the use of approximation models in optimization. *Structural Optimization 15(1)*, 16–23.

Anderson, R., J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma (2020). Strong mixed-integer programming formulations for trained neural networks. *Math. Prog. 183*, 3–39.

Bajaj, I., S. S. Iyer, and M. F. Hasan (2018). A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point. *Comp. Chem. Eng. 116*, 306–321.

Bertsimas, D. and J. Dunn (2017). Optimal classification trees. *Machine Learning 106*(7), 1039–1082.

Bertsimas, D., O. Nohadani, and K. M. Teo (2010a). Nonconvex robust optimization for problems with constraints. *INFORMS J. Computing 22*(1), 44–58.

Bertsimas, D., O. Nohadani, and K. M. Teo (2010b). Robust optimization for unconstrained simulation-based problems. *Oper. Res. 58*(1), 161–178.

Bhosekar, A. and M. Ierapetritou (2018). Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Comp. Chem. Eng. 108*, 250–267.

Biegler, L. T., Y.-D. Lang, and W.-J. Lin (2014). Multiscale optimization for process systems engineering. *Comp. Chem. Eng. 60, 10*, 17– 30.

Bogunovic, I., J. Scarlett, S. Jegelka, and V. Cevher (2018). Adversarially robust optimization with Gaussian processes. *NeurIPS 31*.

Boukouvala, F. and C. A. Floudas (2017). Argonaut: Algorithms for global optimization of constrained grey-box computational problems. *Optim. Lett. 11*(5), 895–913.

Boukouvala, F., R. Misener, and C. A. Floudas (2016). Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO. *Eur. J. Oper. Res. 252*(3), 701–727.

Bynum, M. L., G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Siirola, J.-P. Watson, and D. L. Woodruff (2021). *Pyomo - Optimization Modeling in Python*, Volume 67. Springer Nature.

Caballero, J. A. and I. E. Grossmann (2008). An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE Journal 54*(10), 2633–2650.

Ceccon, F., J. Jalving, J. Haddad, A. Thebelt, C. Tsay, C. D. Laird, and R. Misener (2022). OMLT: Optimization & machine learning toolkit. *arXiv:2202.02414*.

Chen, X., K. Wu, A. Bai, C. M. Masuku, J. Niederberger, F. S. Liporace, and L. T. Biegler (2021). Real-time refinery optimization with reduced-order fluidized catalytic cracker model and surrogate-based trust region filter method. *Comp. Chem. Eng. 153*, 107455.

Conn, A. R., K. Scheinberg, and L. N. Vicente (2009). *Introduction to derivative-free optimization*. Phila., PA: SIAM.

Dowling, A. W., J. P. Eason, J. Ma, D. C. Miller, and L. T. Biegler (2016). Equation-based design, integration, and optimization of oxycombustion power systems. In M. Martin (Ed.), *Alternative Energy Sources and Technologies*, Switzerland, pp. 119–158. Springer.

Eason, J. P. and L. T. Biegler (2016). A trust region filter method for glass box/black box optimization. *AIChE Journal 62*(9), 3124–3136.

Eason, J. P. and L. T. Biegler (2018). Advanced trust region optimization strategies for glass box/black box models. *AIChE Journal 64*(11), 3934–3943.

Eason, J. P., J.-Y. Kang, X. Chen, and L. T. Biegler (2018). Surrogate equations of state for equation-oriented optimization of polymerization processes. *44*, 781–786.

Fahl, M. and E. W. Sachs (2003). Reduced-order modelling approaches to pde-constrained optimization based on proper orthogonal decomposition. In *Large-scale PDE-Constrained Optimization*, Heidelberg, Germany, pp. 268–280. Springer.

Fazlyab, M., M. Morari, and G. J. Pappas (2019). Probabilistic verification and reachability analysis of neural networks via semidefinite programming. In *2019 IEEE 58th CDC Conference*, pp. 2726–2731.

Fischetti, M. and J. Jo (2018). Deep neural networks and mixed integer linear optimization. *Constraints 23*(3), 296.

Goldstein, D., M. Heyer, D. Jakobs, E. S. Schultz, and L. T. Biegler (2022). Multilevel surrogate modeling of an amine scrubbing process. *AIChE Journal 68, 6*, e17705.

Grimstad, B. and H. Andersson (2019). ReLU networks as surrogate models in mixed-integer linear programs. *Comp. Chem. Eng. 131*, 106580.

Henao, C. A. and C. T. Maravelias (2011). Surrogate-based superstructure optimization framework. *AIChE Journal 57*(5), 1216–1232.

Huyer, W. and A. Neumeier (2008). Snobfit – stable noisy optimization by branch and fit. *ACM Trans. Math. Software 35 (2)*.

Kang, J., Z. Shao, X. Chen, and L. T. Biegler (2019). Reduced order models for dynamic molecular weight distribution in polymerization processes. *Comp. Chem. Eng. 126*, 280–291.

Katz, J., I. Pappas, S. Avraamidou, and E. N. Pistikopoulos (2020). Integrating deep learning models and multiparametric programming. *Comp. Chem. Eng. 136*, 106801.

Kazi, S., M. Short, and L. T. Biegler (2020). Heat exchanger network optimization including detailed heat exchanger models using trust region methods. *Computer Aided Chemical Engineering 48*, 1147–1152.

Khalil, E. B., A. Gupta, and B. Dilkina (2018). Combinatorial attacks on binarized neural networks. In *International Conference on Learning Representations*.

Kim, S. H. and F. Boukouvala (2020). Surrogate-based optimization for mixed-integer nonlinear problems. *Comp. Chem. Eng. 140*, 106847.

Lomuscio, A. and L. Maganti (2017). An approach to reachability analysis for feed-forward ReLU neural networks. *arXiv preprint arXiv:1706.07351*.

Lueg, L., B. Grimstad, A. Mitsos, and A. M. Schweidtmann (2021). reluMIP: Open source tool for MILP optimization of ReLU neural networks. `https://github.com/ChemEngAI/ReLU_ANN_MILP` *1.0.0*.

Ma, K., N. V. Sahinidis, R. Bindlish, S. J. Bury, R. Haghpanah, and S. Rajagopalan (2022). Data-driven strategies for extractive distillation unit optimization. *Comp. Chem. Eng. to appear*.

McBride, K. and K. Sundmacher (2019). Overview of surrogate modeling in chemical process engineering. *Chemie Ingenieur Technik 91*(3), 228–239.

Mišić, V. V. (2020). Optimization of tree ensembles. *Oper. Res. 68*(5), 1605–1624.

Mistry, M., D. Letsios, G. Krennrich, R. M. Lee, and R. Misener (2021). Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. *INFORMS J. Comput. 33*(3), 1103–1119.

Ning, C. and F. You (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Comp. Chem. Eng. 125*, 434–448.

Palmer, K. and M. Realff (2002). Metamodeling approach to optimization of steady-state flowsheet simulations: Model generation. *Chemical Engineering Research & Design 80*(7), 760–772.

Pistikopoulos, E. N., A. Barbosa-Povoa, J. H. Lee, R. Misener, A. Mitsos, G. V. Reklaitis, V. Venkatasubramanian, F. You, and R. Gani (2021). Process systems engineering – the generation next? *Comp. Chem. Eng. 147*, 107252.

Qin, S. J. and L. H. Chiang (2019). Advances and opportunities in machine learning for process data analytics. *Comp. Chem. Eng. 126*, 465–473.

Raghunathan, A., J. Steinhardt, and P. S. Liang (2018). Semidefinite relaxations for certifying robustness to adversarial examples. In *NeurIPS*, Volume 31, pp. 10877–10887.

Rios, L. M. and N. V. Sahinidis (2013). Derivative-free optimization: a review of algorithms and comparison of software implementations. *J. Glob. Optim. 56*(3), 1247–1293.

Sargent, R. (1972). Forecasts and trends in systems engineering. *The Chemical Engineer 262*, 226–230.

Schweidtmann, A. M., D. Bongartz, D. Grothe, T. Kerkenhoff, X. Lin, J. Najman, and A. Mitsos (2021). Deterministic global optimization with Gaussian processes embedded. *Math. Prog. Computation 13*(3), 553–581.

Schweidtmann, A. M., E. Esche, A. Fischer, M. Kloft, J.-U. Repke, S. Sager, and A. Mitsos (2021). Machine learning in chemical engineering: A perspective. *Chemie Ingenieur Technik 93*(12), 2029–2039.

Schweidtmann, A. M. and A. Mitsos (2019). Deterministic global optimization with artificial neural networks embedded. *J. Optim. Theory Appl. 180*(3), 925–948.

Shang, C. and F. You (2019). Data analytics and machine learning for smart process manufacturing: recent advances and perspectives in the big data era. *Engineering 5*(6), 1010–1016.

Thebelt, A., J. Kronqvist, M. Mistry, R. M. Lee, N. Sudermann-Merx, and R. Misener (2021). ENTMOOT: A framework for optimization over ensemble tree models. *Comp. Chem. Eng. 151*, 107343.

Thebelt, A., C. Tsay, R. M. Lee, N. Sudermann-Merx, D. Walz, T. Tranter, and R. Misener (2022). Multi-objective constrained optimization for energy applications via tree ensembles. *Applied Energy 306*, 118061.

Thebelt, A., J. Wiebe, J. Kronqvist, C. Tsay, and R. Misener (2022). Maximizing information from chemical engineering data sets: Applications to machine learning. *Chem. Eng. Sci. 252*, 117469.

Tsay, C. (2021). Sobolev trained neural network surrogate models for optimization. *Comp. Chem. Eng. 153*, 107419.

Tsay, C. and M. Baldea (2019). 110th anniversary: Using data to bridge the time and length scales of process systems. *Ind. Eng. Chem. Res. 58*, 16696–16708.

Tsay, C., J. Kronqvist, A. Thebelt, and R. Misener (2021). Partition-based formulations for mixed-integer optimization of trained ReLU neural networks. In *NeurIPS*.

Von Stosch, M., R. Oliveira, J. Peres, and S. F. de Azevedo (2014). Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Comp. Chem. Eng. 60*, 86–101.

Wiebe, J., I. Cecílio, J. Dunlop, and R. Misener (2022). A robust approach to warped Gaussian process-constrained optimization. *Math. Prog.*, 1–35.

Wiebe, J. and R. Misener (2021). ROmodel: Modeling robust optimization problems in Pyomo. *Optim. Eng.*, 1–22.

Wilson, J., V. Borovitskiy, A. Terenin, P. Mostowsky, and M. Deisenroth (2020). Efficiently sampling functions from Gaussian process posteriors. In *ICML*, pp. 10292–10302.

Wilson, Z. T. and N. V. Sahinidis (2017). The ALAMO approach to machine learning. *Comp. Chem. Eng. 106*, 785.

Yang, D., P. Balaprakash, and S. Leyffer (2021). Modeling design and control problems involving neural network surrogates. *arXiv preprint arXiv:2111.10489*.

Yoshio, N. and L. T. Biegler (2020). Demand-based optimization of a chlorobenzene process with high-fidelity and surrogate reactor models under trust region strategies. *AIChE Journal*, e17054.

Zhang, J., H. Li, S. Sra, and A. Jadbabaie (2022). Neural network weights do not converge to stationary points: An invariant measure perspective. In *ICML*.