Modular flowsheet optimization using trust regions and Gaussian process regression

E. I. Sanchez Medina[1], D. Rodriguez Vallejo[3], B. Chachuat[3], K. Sundmacher[1,2], P. Petsagkourakis[4], E. A. del Rio-Chanona[3]

1 Chair for Process Systems Engineering, Otto von Guericke University, Universitätspl. 2, Magdeburg, 39106, Germany
2 Process Systems Engineering, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, Magdeburg, 39106, Germany
3 The Sargent Centre for Process Systems Engineering, Imperial College London, Exhibition Rd., South Kensington, London, SW7 2AZ, UK
4 The Sargent Centre for Process Systems Engineering, University College London, Gower St. Bloomsbury, London, WC1E 6BT, UK

INTRODUCTION

Process flowsheet optimization

- Process flowsheet optimization carried out using **gradient-based** methods requires that the gradient information for all the process models is available.
- **Derivative-free** techniques are often paired to the simulation models for optimization.
- A flowsheet simulation itself may be **computationally demanding**, which calls for robust and sampling-efficient methods to drive the optimization.

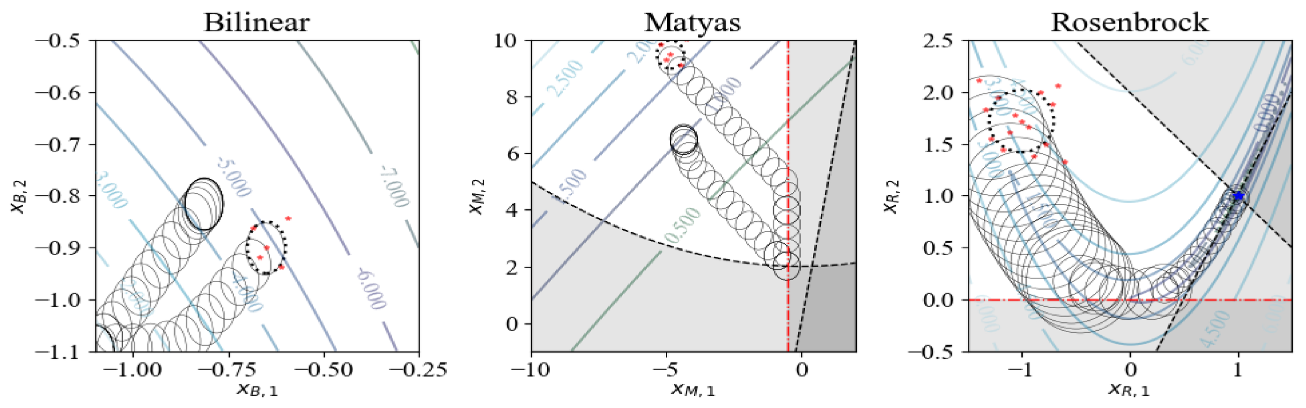
Global approaches

- Proceed by constructing a surrogate model based on flowsheet simulations before optimizing it, often within an iteration where the surrogate is progressively refined [1,2].

True model → Surrogate model

Local approaches

- Maintain an accurate representation of the flowsheet within a trust region, whose position and size are adapted iteratively. This procedure entails reconstructing the surrogate model as the trust region moves towards the local optimum [3,4].

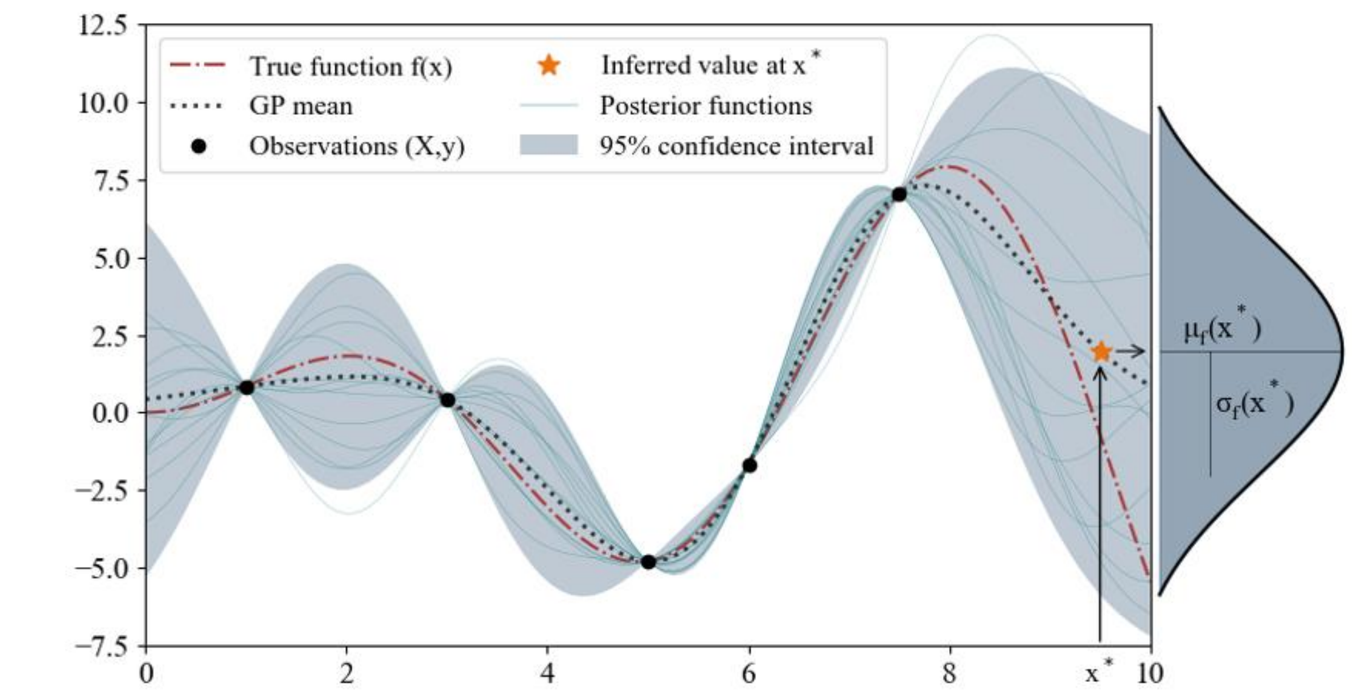


BACKGROUND

Gaussian process regression (GPR)

- Generalization of a multivariate Gaussian distribution to infinite dimensions.
- It can be used to approximate an unknown function $f: \mathbb{R}^{n_x} \to \mathbb{R}$ using noisy. observations

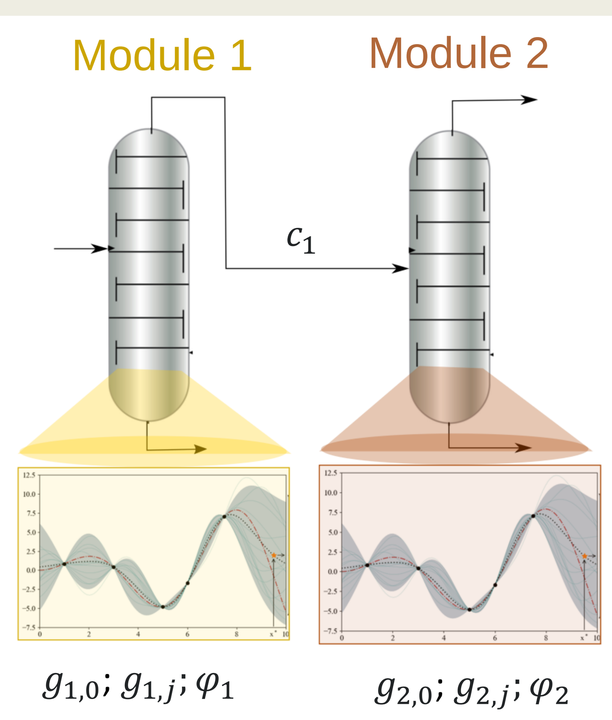
  $y = f(\mathbf{x}) + \varepsilon$ with $y \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^{n_x}$ and $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.
- Therefore, the approxiamtion can be written as $f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot,\cdot))$.
- The mean function $m(\cdot)$ provides prior knowledge about f.
- The covariance function $k(\cdot,\cdot)$ accounts for correlations between function values.
- Inference can be done by conditioning the distribution $f(\boldsymbol{x}^*) \mid \boldsymbol{X}, \boldsymbol{y} \sim \mathcal{N}\left(\mu_f(\boldsymbol{x}^*), \sigma_f^2(\boldsymbol{x}^*)\right)$.
- With posterior $\mu_f(\boldsymbol{x}^*) \coloneqq \boldsymbol{K}^*(\boldsymbol{x}^*, \boldsymbol{X})\boldsymbol{K}(\boldsymbol{X})^{-1}\boldsymbol{y}$ and $\sigma_f^2(\boldsymbol{x}^*) \coloneqq \sigma_n^2 - \boldsymbol{K}^*(\boldsymbol{x}^*, \boldsymbol{X})\boldsymbol{K}(\boldsymbol{X})^{-1}\boldsymbol{K}^*(\boldsymbol{x}^*, \boldsymbol{X})^\intercal$



MODULAR FLOWSHEET OPTIMIZATION PROBLEM

$$
\begin{aligned}
\min_{(\mathbf{x}_1,\mathbf{y}_1,\dots,\mathbf{x}_m,\mathbf{y}_m)} \quad & \sum_{i=1}^{m} g_{i,0}(\mathbf{x}_i,\mathbf{y}_i) \\
\text{s.t.} \quad & g_{i,j}(\mathbf{x}_i,\mathbf{y}_i) \le 0, \quad j = 1\dots n_{g_i},\ i = 1\dots m \\
& c_j(\mathbf{x}_1,\mathbf{y}_1,\dots,\mathbf{x}_m,\mathbf{y}_m) = 0, \quad j = 1\dots n_c \\
& \mathbf{y}_i = \varphi_i(\mathbf{x}_i,\theta_i) \\
& (\mathbf{x}_1,\dots,\mathbf{x}_m) \in \mathcal{X} \\
& (\mathbf{y}_1,\dots,\mathbf{y}_m) \in \mathcal{Y},\ i = 1\dots m
\end{aligned}
$$

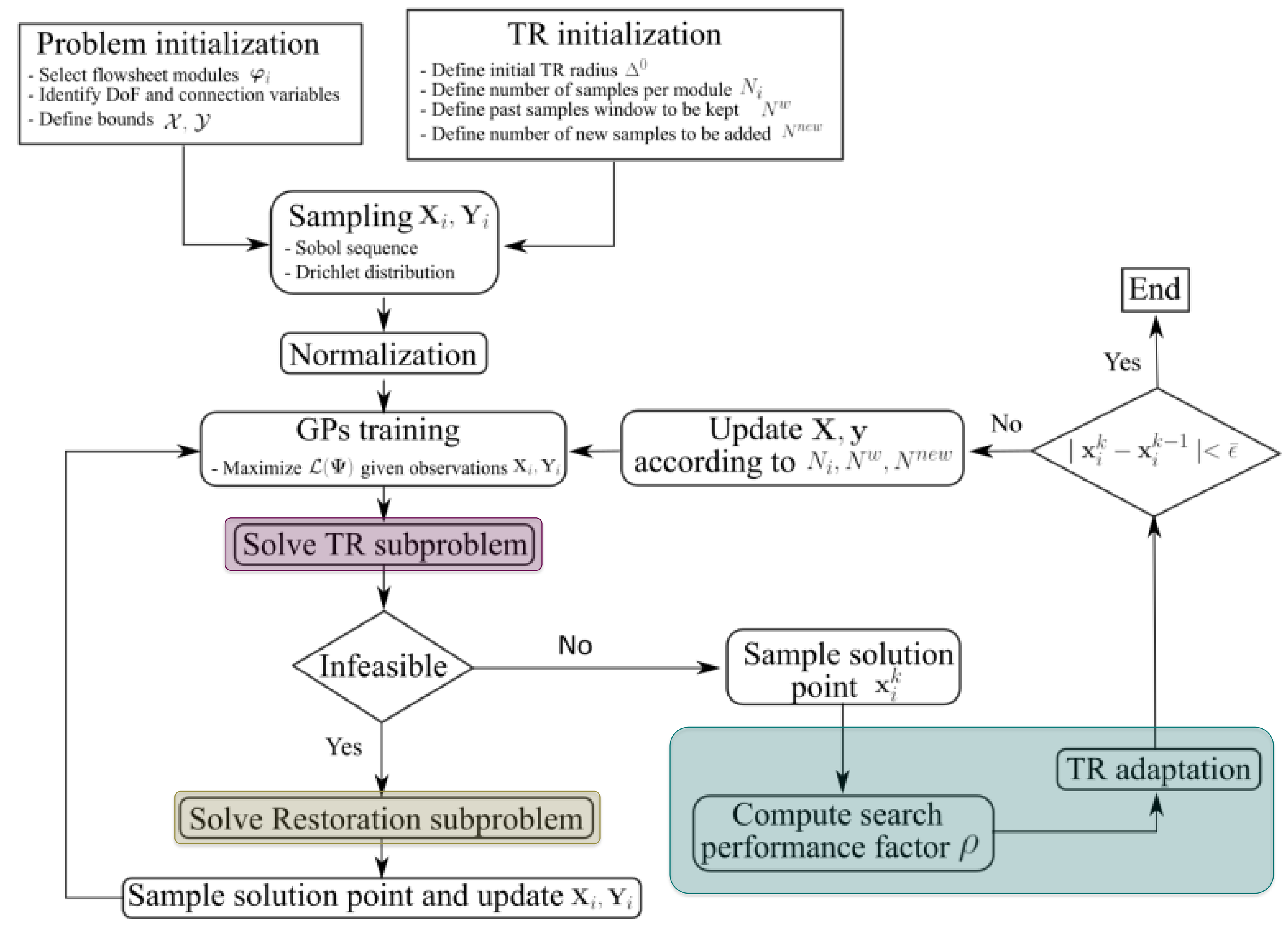
- Contribution to cost function $g_{i,0}$
- Operating/design constraints $g_{i,j}$
- Input/output mapping φ_i
- Connection stream constraints c_j



ACKNOWLEDGEMENTS

E. I. S. M. is also affiliated with the International Max Planck Research School for Advanced Methods in Process and Systems Engineering - IMPRS ProEng at the Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg.

TRUST REGIONS AND GPR FOR OPTIMIZATION



Trust region subproblem

$$
\begin{aligned}
(\mathbf{d}_1^{k+1}, \hat{\mathbf{y}}_1^{k+1} \dots, \mathbf{d}_m^{k+1}, \hat{\mathbf{y}}_m^{k+1}) \in \underset{\mathbf{d}_1,\hat{\mathbf{y}}_1\dots,\mathbf{d}_m,\hat{\mathbf{y}}_m}{\arg\min} & \sum_{i=1}^{m} g_{i,0}\left(\mathbf{x}_i^k + \mathbf{d}_i, \hat{\mathbf{y}}_i\right) \\
\text{s.t.}\ & g_{i,j}\left(\mathbf{x}_i^k + \mathbf{d}_i, \hat{\mathbf{y}}_i\right) \le 0, \quad j = 1\dots n_{g_i},\ i = 1\dots m \\
& c_j\left(\mathbf{x}_i^k + \mathbf{d}_i, \hat{\mathbf{y}}_i\right) = 0, \quad j = 1\dots n_c,\ i = 1\dots m \\
& \hat{\mathbf{y}}_i = \mu_{\varphi_i}(\mathbf{x}_i^k + \mathbf{d}_i, \theta_i), \quad i = 1\dots m \\
& \|\mathbf{d}_i^{DoF}\| \le \Delta^k \\
& \mathbf{d}_i = \{\mathbf{d}_i^{DoF}, \mathbf{d}_i^{Dep}\},\ \ \mathbf{x}_i^k = \{\mathbf{x}_i^{k,DoF}, \mathbf{x}_i^{k,Dep}\} \\
& (\mathbf{x}_1^k + \mathbf{d}_1, \dots, \mathbf{x}_m^k + \mathbf{d}_m) \in \mathcal{X} \\
& (\hat{\mathbf{y}}_1 \dots \hat{\mathbf{y}}_m) \in \mathcal{Y}
\end{aligned}
$$

Restoration subproblem

$$
\begin{aligned}
(\mathbf{d}_1^{k+1}, \hat{\mathbf{y}}_1^{k+1} \dots, \mathbf{d}_m^{k+1}, \hat{\mathbf{y}}_m^{k+1}) \in \underset{\mathbf{d}_1,\hat{\mathbf{y}}_1\dots,\mathbf{d}_m,\hat{\mathbf{y}}_m}{\arg\min} & \sum_{i=1}^{m} \left\|\mathbf{g}_i\left(\mathbf{x}_i^k + \mathbf{d}_i, \hat{\mathbf{y}}_i\right)\right\|_\infty \\
\text{s.t.}\ & c_j\left(\mathbf{x}_i^k + \mathbf{d}_i, \hat{\mathbf{y}}_i\right) = 0, \quad i = 1\dots m,\quad j = 1\dots n_c \\
& \hat{\mathbf{y}}_i = \mu_{\varphi_i}(\mathbf{x}_i^k + \mathbf{d}_i, \theta_i), \quad i = 1\dots m \\
& \|\mathbf{d}_i^{DoF}\| \le \Delta^k \\
& \mathbf{d}_i = \{\mathbf{d}_i^{DoF}, \mathbf{d}_i^{Dep}\},\ \ \mathbf{x}_i^k = \{\mathbf{x}_i^{k,DoF}, \mathbf{x}_i^{k,Dep}\} \\
& (\mathbf{x}_1^k + \mathbf{d}_1, \dots, \mathbf{x}_m^k + \mathbf{d}_m) \in \mathcal{X} \\
& (\hat{\mathbf{y}}_1 \dots \hat{\mathbf{y}}_m) \in \mathcal{Y}
\end{aligned}
$$

Search performance factor and adaptation rules

$$
\rho^{k+1} \coloneqq \frac{\sum_{i=1}^{m} g_{i,0}\left(\mathbf{x}_i^k, \varphi_i(\mathbf{x}_i^k, \theta_i)\right) - g_{i,0}\left(\mathbf{x}_i^k + \mathbf{d}_i^{k+1}, \varphi_i(\mathbf{x}_i^k + \mathbf{d}_i^{k+1}, \theta_i)\right)}{\sum_{i=1}^{m} g_{i,0}\left(\mathbf{x}_i^k, \mu_{\varphi_i}(\mathbf{x}_i^k, \theta_i)\right) - g_{i,0}\left(\mathbf{x}_i^k + \mathbf{d}_i^{k+1}, \mu_{\varphi_i}(\mathbf{x}_i^k + \mathbf{d}_i^{k+1}, \theta_i)\right)}
$$

TR radius adaptation

If $\rho^{k+1} < \eta_2:\ \Delta^{k+1} \coloneqq \Delta^k \cdot \xi_1$

else if $\rho^{k+1} > \eta_3$ and $\|\mathbf{d}_i^{DoF}\| = \Delta_i^k:\ \Delta^{k+1} \coloneqq min\{\bar{\Delta}, \Delta^k \cdot \xi_2\}$

else : $\Delta^{k+1} \coloneqq \Delta^k$

TR center adaptation

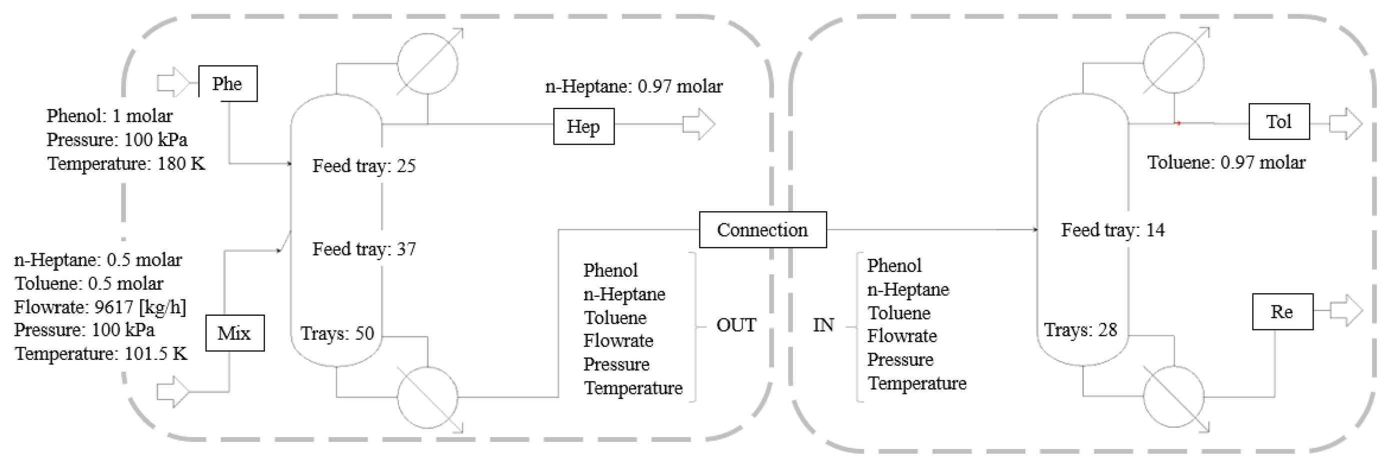
If $\rho^{k+1} < \eta_1:\ \mathbf{x}_i^{k+1} \coloneqq \mathbf{x}_i^k$

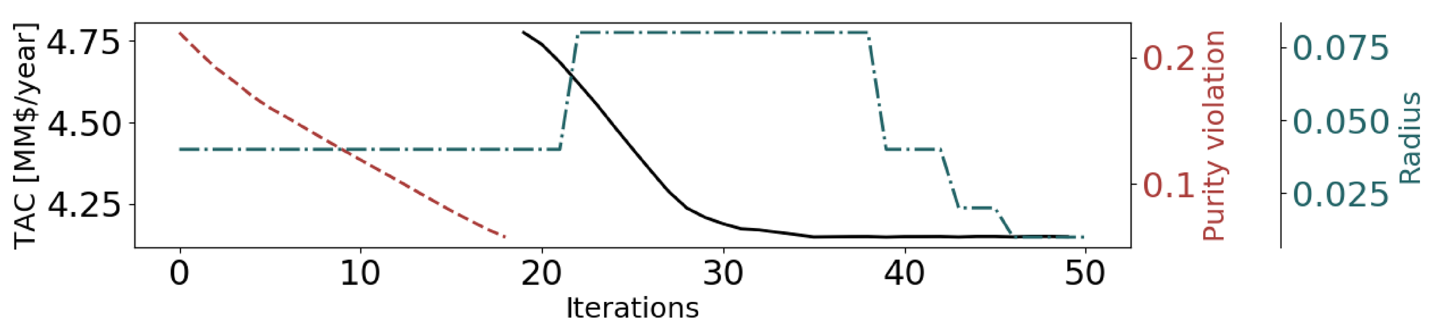
else : $\mathbf{x}_i^{k+1} \coloneqq \mathbf{x}_i^k + \mathbf{d}_i^{k+1}$

COMPUTATIONAL CASE STUDY

Minimization of TAC for extractive distillation system

- Two individual Aspen-HYSYS v9.0 simulation files.
- NLP solved with IPOPT using CasADi.
- Using Matern 3/2 and $\eta_1 = \eta_2 = 0.1$, $\eta_3 = 0.9$, $\xi_1 = 0.5$, $\xi_2 = 2$, $\Delta_1^0 = 0.02$, $\Delta_2^0 = 0.025$





- Optimization without the necessity of a complete flowsheet simulation.
- It can handle hybrid flowsheets with black-box modules and closed-form models.

REFERENCES

[1] Caballero, J.A., et al., (2008). *AIChE journal*, DOI: 10.1002/aic.11579.
[2] Keßler, T., et al., (2019). *Chemical Engineering Science*, DOI: 10.1016/j.ces.2018.12.002
[3] Eason, J.P., et al., (2018). *AIChE Journal*, DOI: 10.1002/aic.16364
[4] Bajaj, I., et al., (2018). *Computers & Chemical Engineering*, DOI: 10.1016/j.compchemeng.2017.12.011.