What's in the Box?

Systematic approaches for inferring algebraic models from experimental data or simulations

Nick Sahinidis[1,2]

Acknowledgments: Yan Zhang[1,2], Alison Cozad[1,2], David Miller[1]

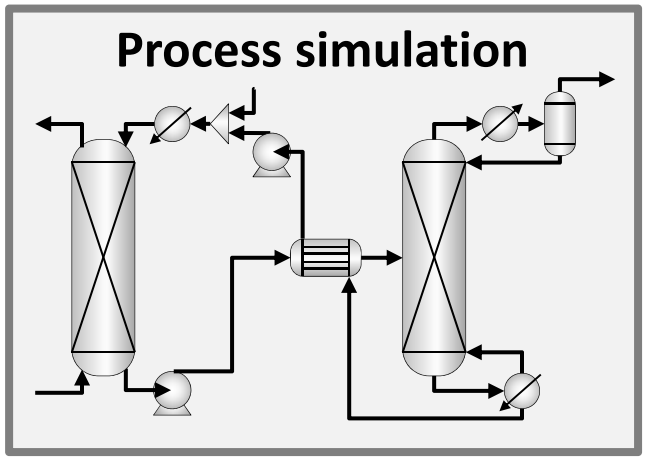
[1]National Energy Technology Laboratory, Pittsburgh, PA,USA

[2]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

LEARNING PROBLEM

- **Build a model of output variables z as a function of input variables x over a specified interval**

$$x \in \mathbb{R}^D \qquad x^l \leq x \leq x^u \qquad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \\ \vdots \\ x_D \end{pmatrix}$$



$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \\ \vdots \\ z_K \end{pmatrix} \qquad z \in \mathbb{R}^K \qquad z = f(x)$$

Independent variables:
Operating conditions, inlet flow properties, unit geometry

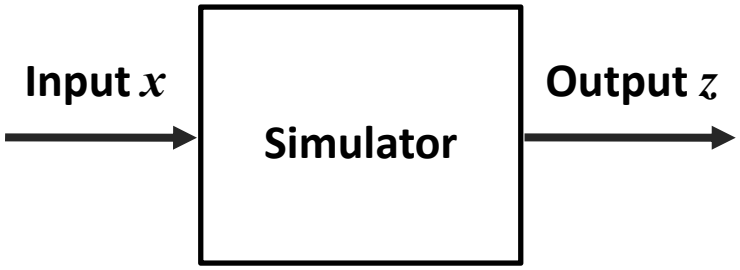
Dependent variables:
Efficiency, outlet flow conditions, conversions, heat flow, etc.

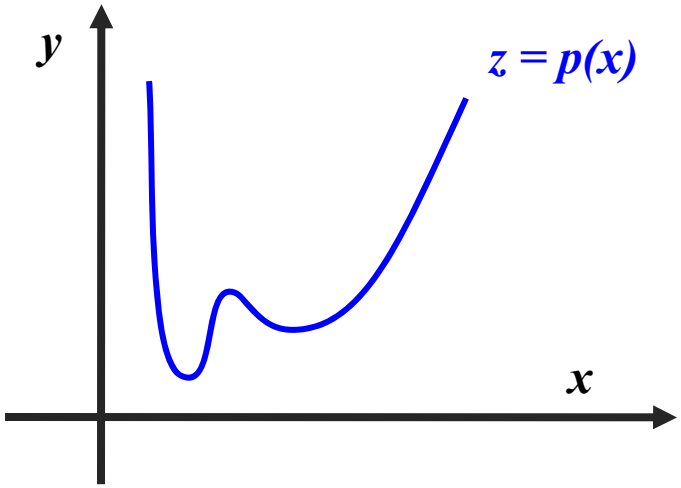
OUTLINE

- Polynomial chaos expansion
  - Best subset selection method
  - Application in risk assessment
- ALAMO: Automatic Learning of Algebraic Models for Optimization
  - Best subset selection method
  - Adaptive sampling
  - Comparisons with least squares and the lasso
  - Application in optimization

POLYNOMIAL CHAOS EXPANSION

- **Build polynomial surrogate models of a given simulator**





Polynomial chaos expansion:

$$z = P(\boldsymbol{x}) = \alpha_0 B_0 + \sum_{j=1}^{M} \alpha_j B_1(x_j) + \sum_{j=1}^{M} \sum_{k=1}^{j} \alpha_{jk} B_2(x_j, x_k) + \sum_{j=1}^{M} \sum_{k=1}^{j} \sum_{h=1}^{k} \alpha_{jkh} B_3(x_j, x_k, x_h) + \ldots$$

Key steps:

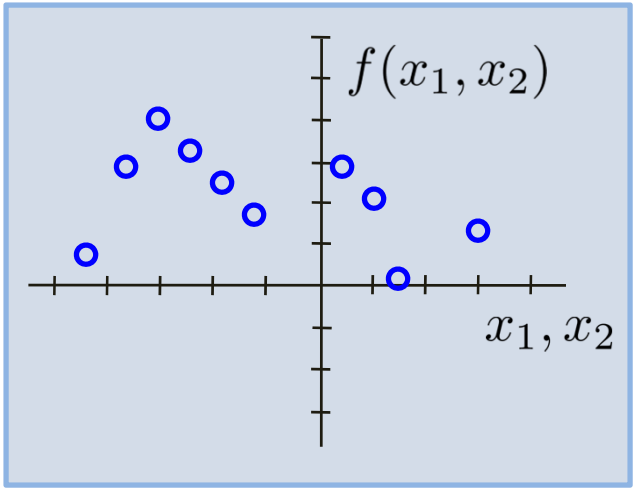
- Choose basis functions $\boldsymbol{B}$
- Determine coefficients $\boldsymbol{\alpha}$

OVERVIEW OF PCE METHODS

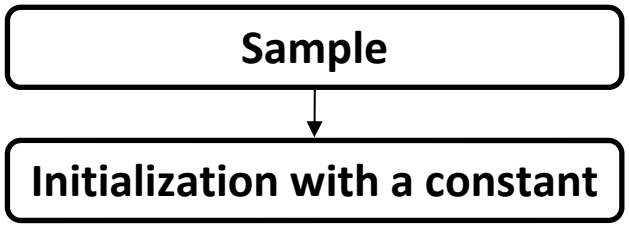
- **Intrusive PCE methods (Xiu and Karniadakis, 2003)**
  - **Substitute PCE's into partial differential equations**
  - **Solve new equations for coefficients**
- **Nonintrusive PCE methods (Webster *et al.,* 1996, Isukapalli *et al.,* 1998, Ghiocel and Ghanem, 2002, Li and Zhang, 2007, Eldred and Burkardt, 2009, Blatman and Sudret, 2010, Oladyshkin *et al.,* 2011)**
  - **No manipulation of partial differential equations**
  - **Estimate coefficients by projection**
  - **Estimate coefficients by fitting curves**

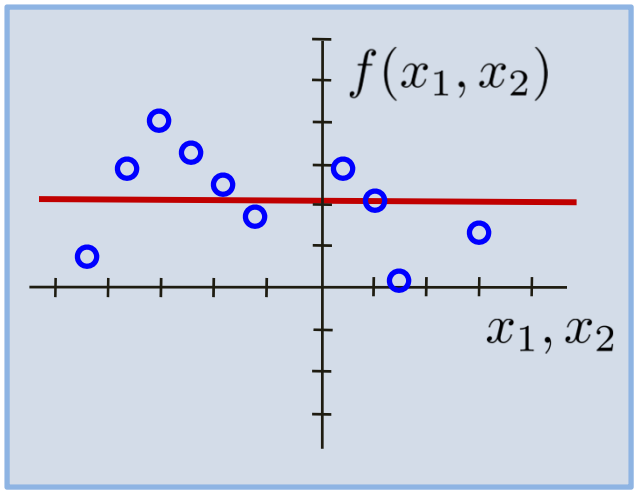
PROPOSED PCE ALGORITHM

Sample



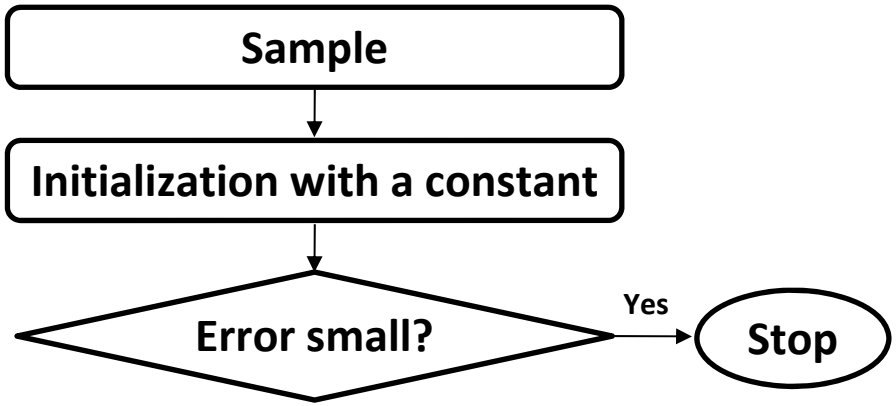
PROPOSED PCE ALGORITHM

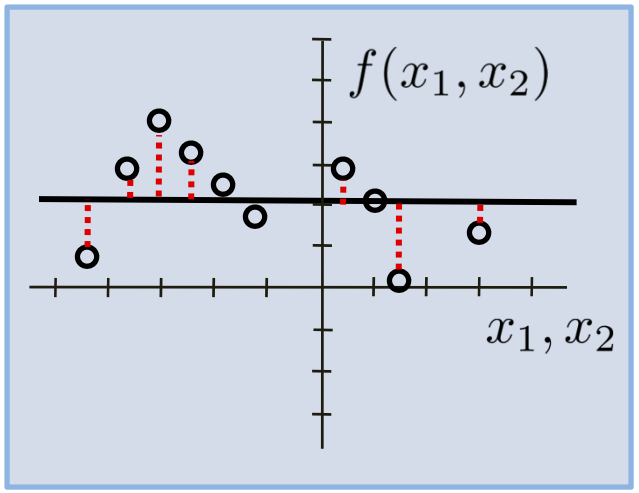




z = constant

PROPOSED PCE ALGORITHM

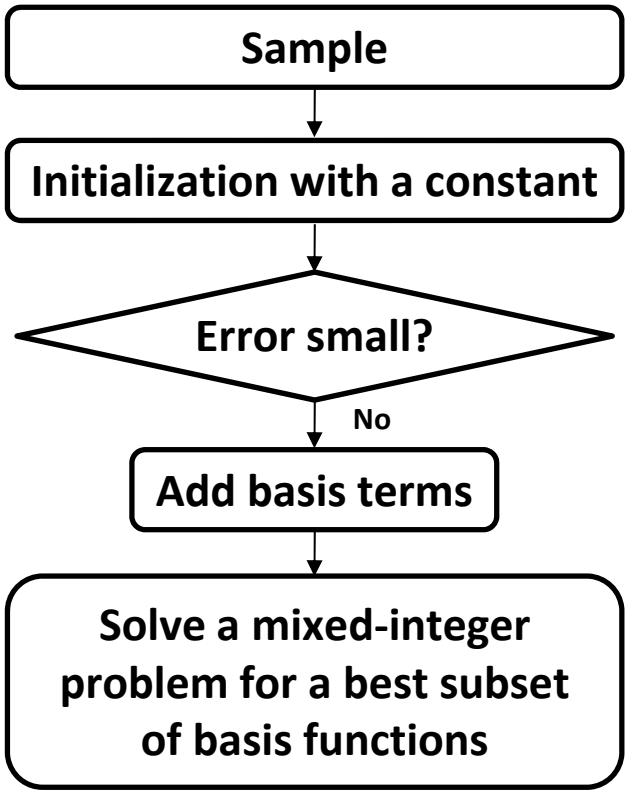




z = constant

Error = ($z_{surrogate} - z_{simulation}$)^2

PROPOSED PCE ALGORITHM



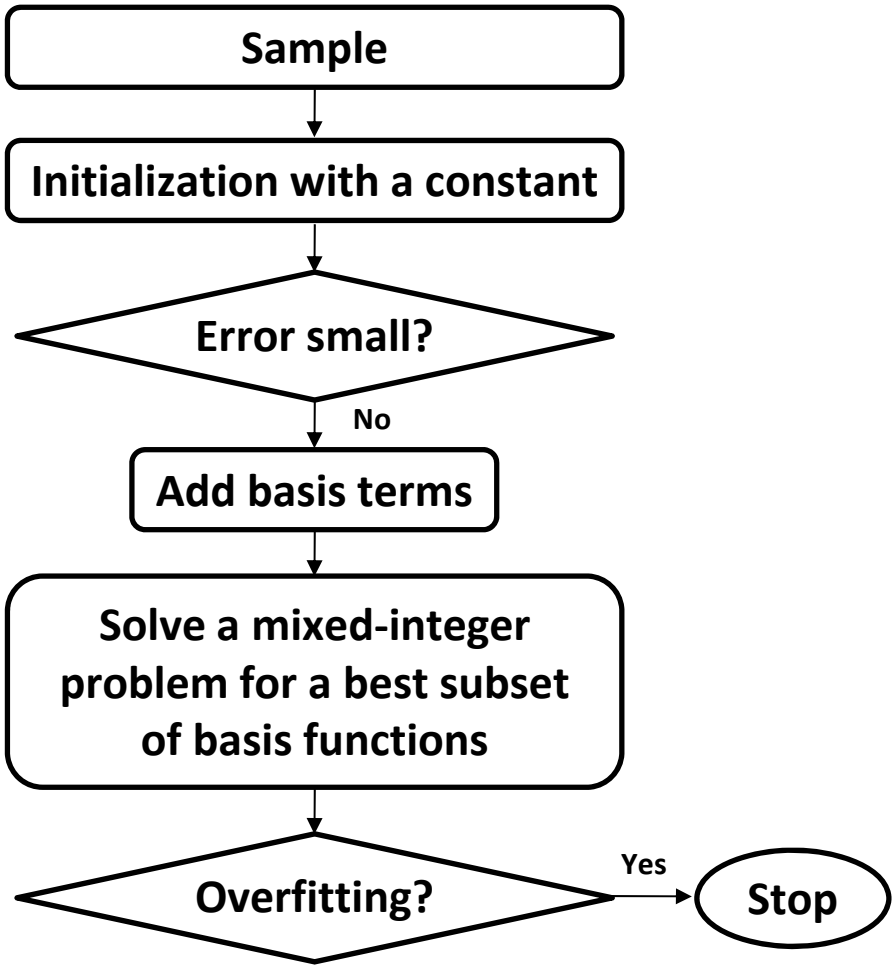
Add higher-order terms to current PCE:

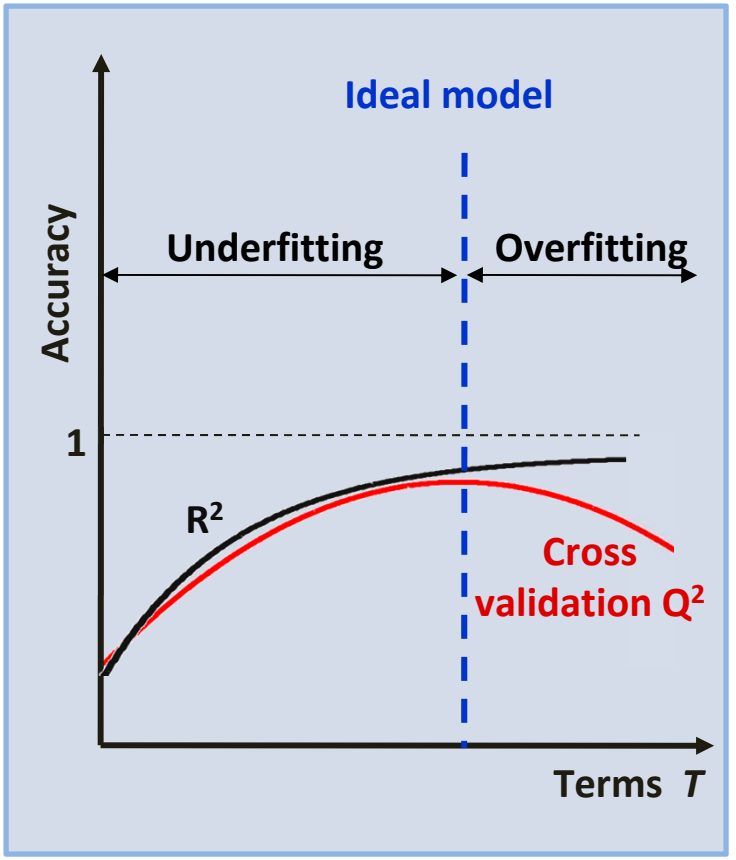
$$z = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2$$

Best subset selected by MIP:

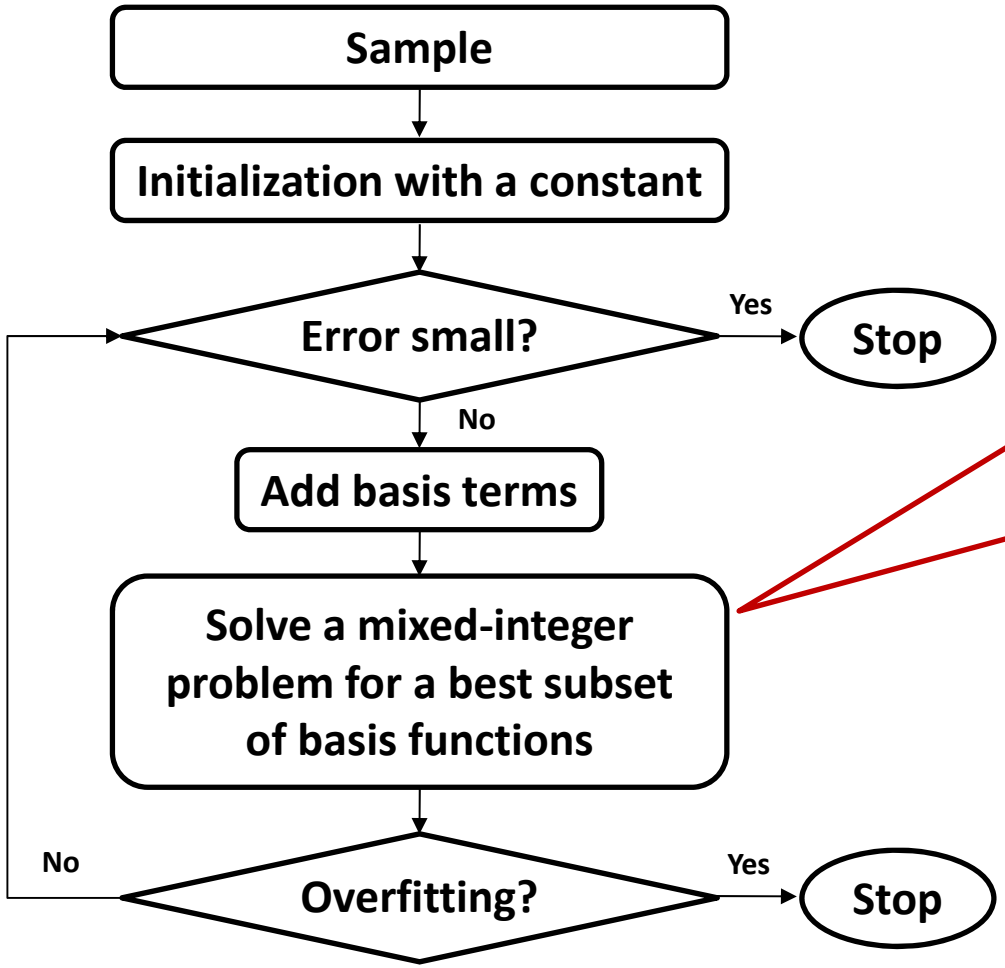
$$z = \alpha_0 + \alpha_2 x_2$$

PROPOSED PCE ALGORITHM





PROPOSED PCE ALGORITHM



Novelty of proposed method:

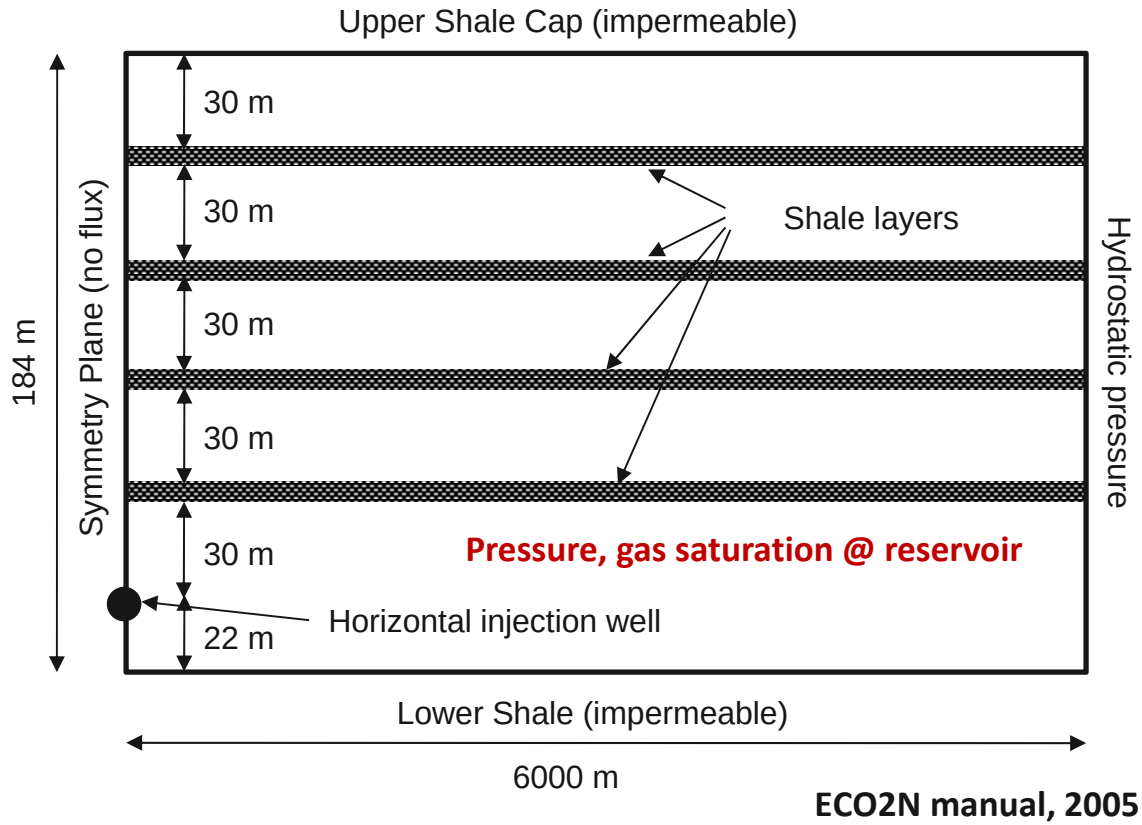
Math programming for basis selection

- **Global optimal subset**
- **Obtains more accurate models**

TESTING ON A BENCHMARK PROBLEM

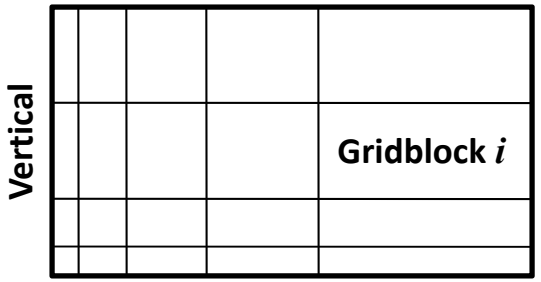
- **CO_2 injection into a deep saline aquifer**
- **Simulated using TOUGH2**

$$\text{mass/pressure/gas saturation} = f\ (\text{porosity, permeability, injection rate})$$

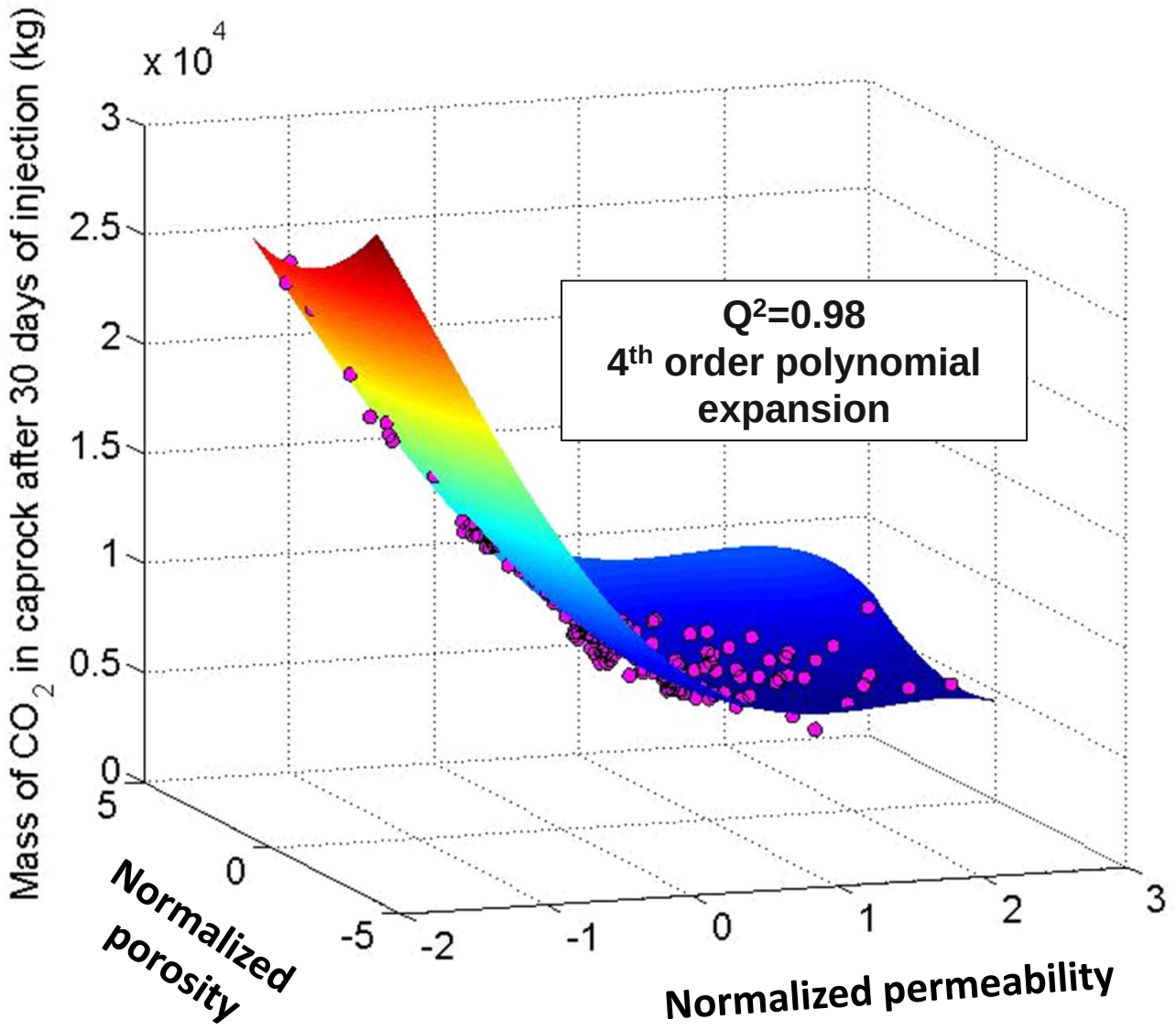


ECO2N manual, 2005

986 gridblocks
986 surrogate models for each output



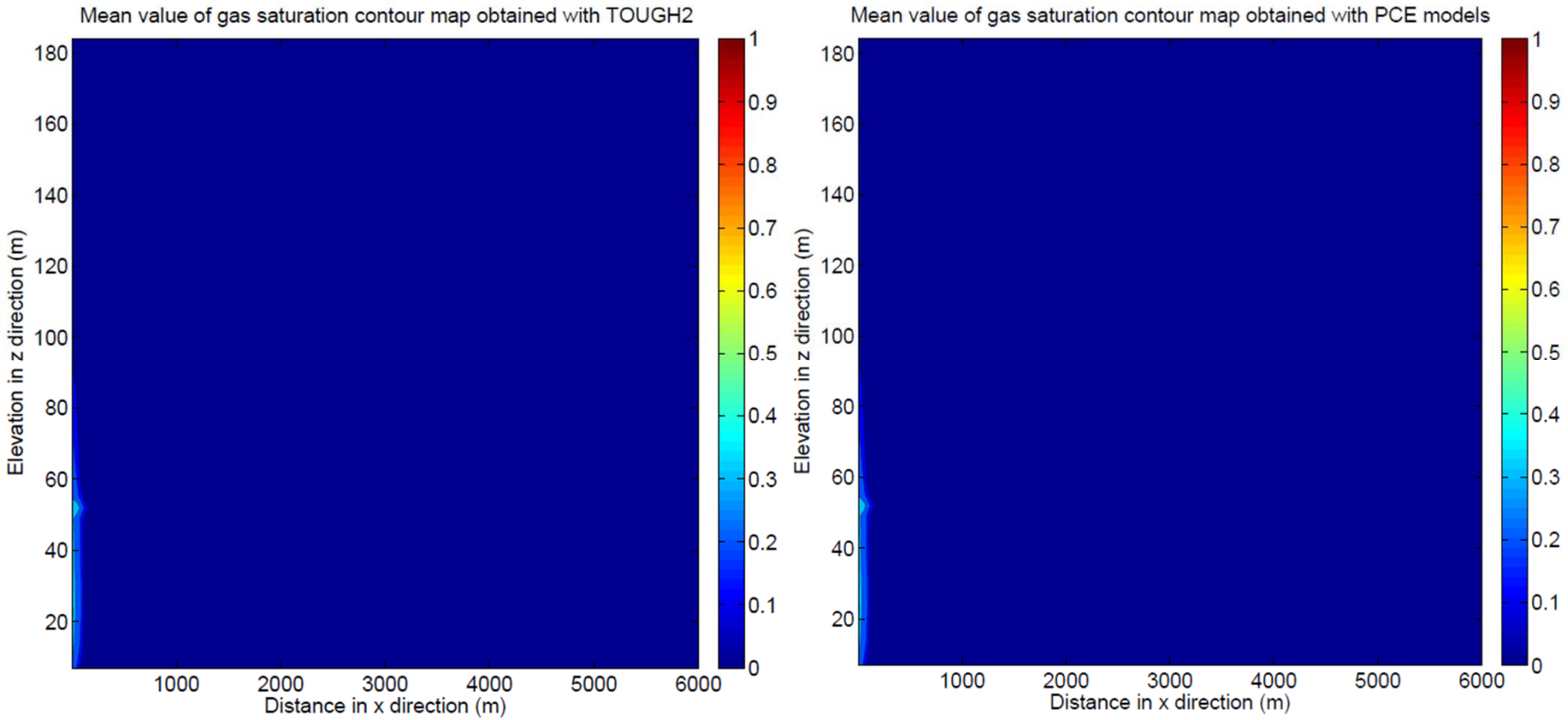
POLYNOMIAL SURROGATE MODEL



COMPARISON AGAINST TOUGH2

For every gridblock i, $i = 1, \ldots, 986$

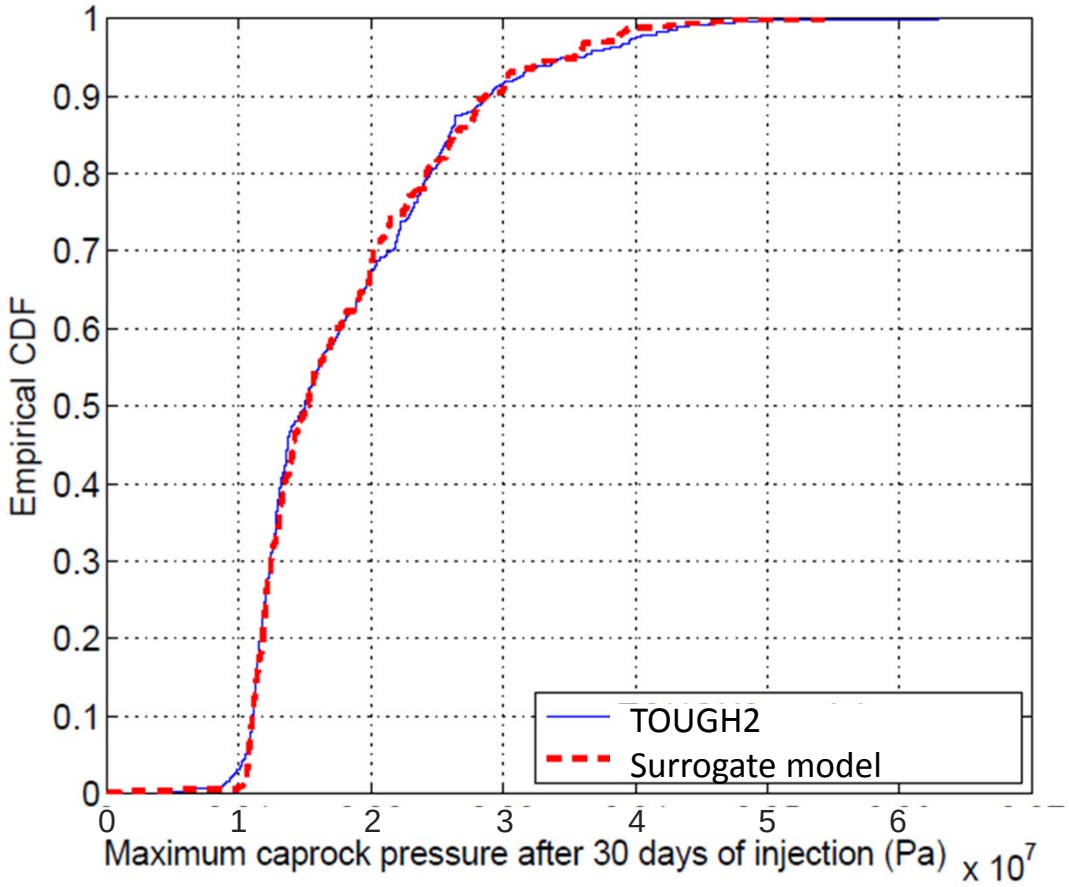
$\mathrm{CO_2}$ saturation in brine $= f_i$ (porosity, permeability, injection rate)



Maximum relative error = 6%

MONTE CARLO SIMULATION

Cumulative probability distribution obtained with 4000 simulations of surrogate models



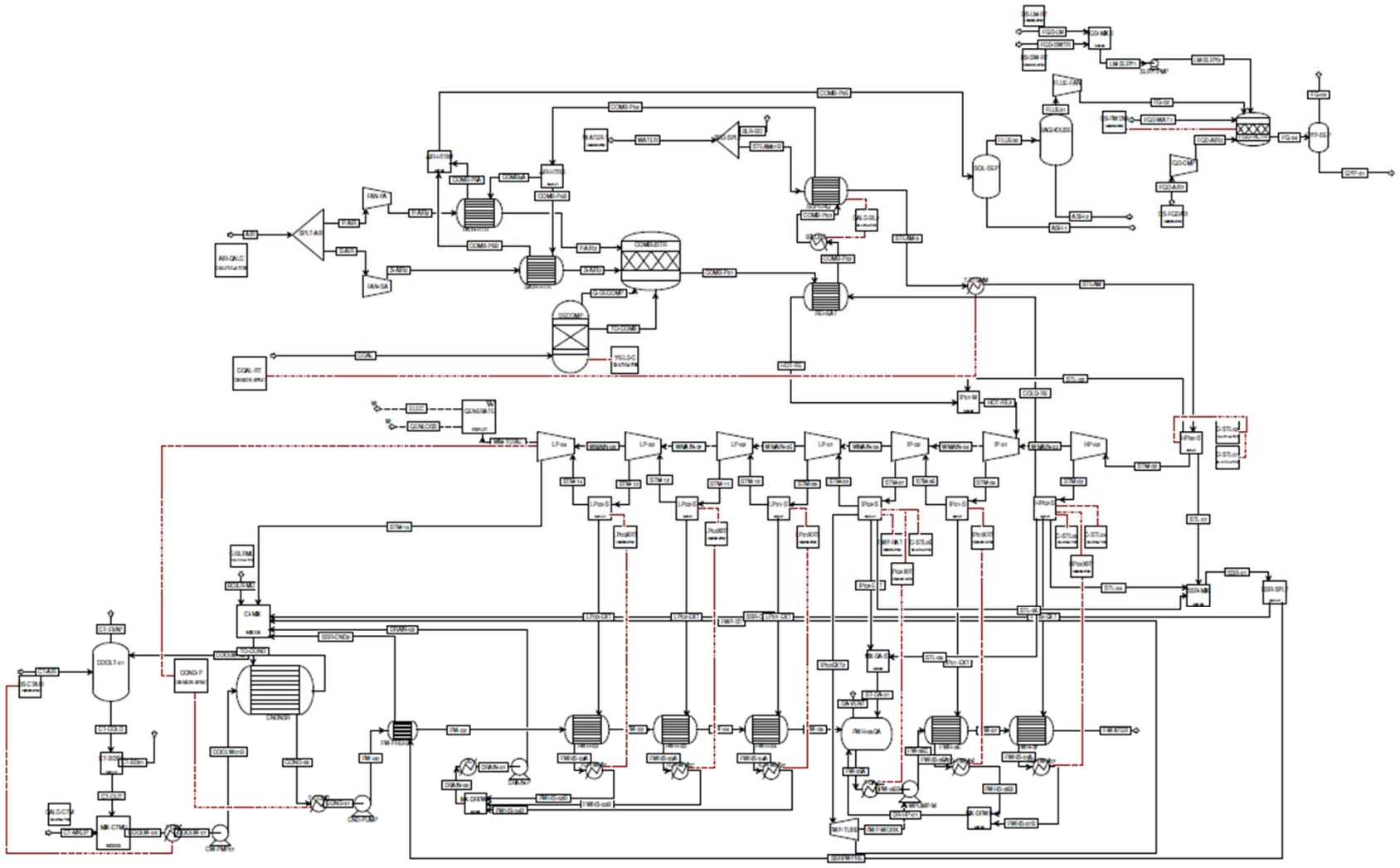
REDUCED-ORDER MODELING GAINS

- **For the benchmark study and SACROC application, surrogate models developed based on the proposed technique were found to be accurate.**

Case	Maximum relative error	CPU time to run surrogates in MATLAB	CPU time to run numerical simulator	Speedup
TOUGH2 Benchmark	6%	30 s	15 mins	10 x
SACROC Oilfield Case	7%	45 s	24 hrs	100 x

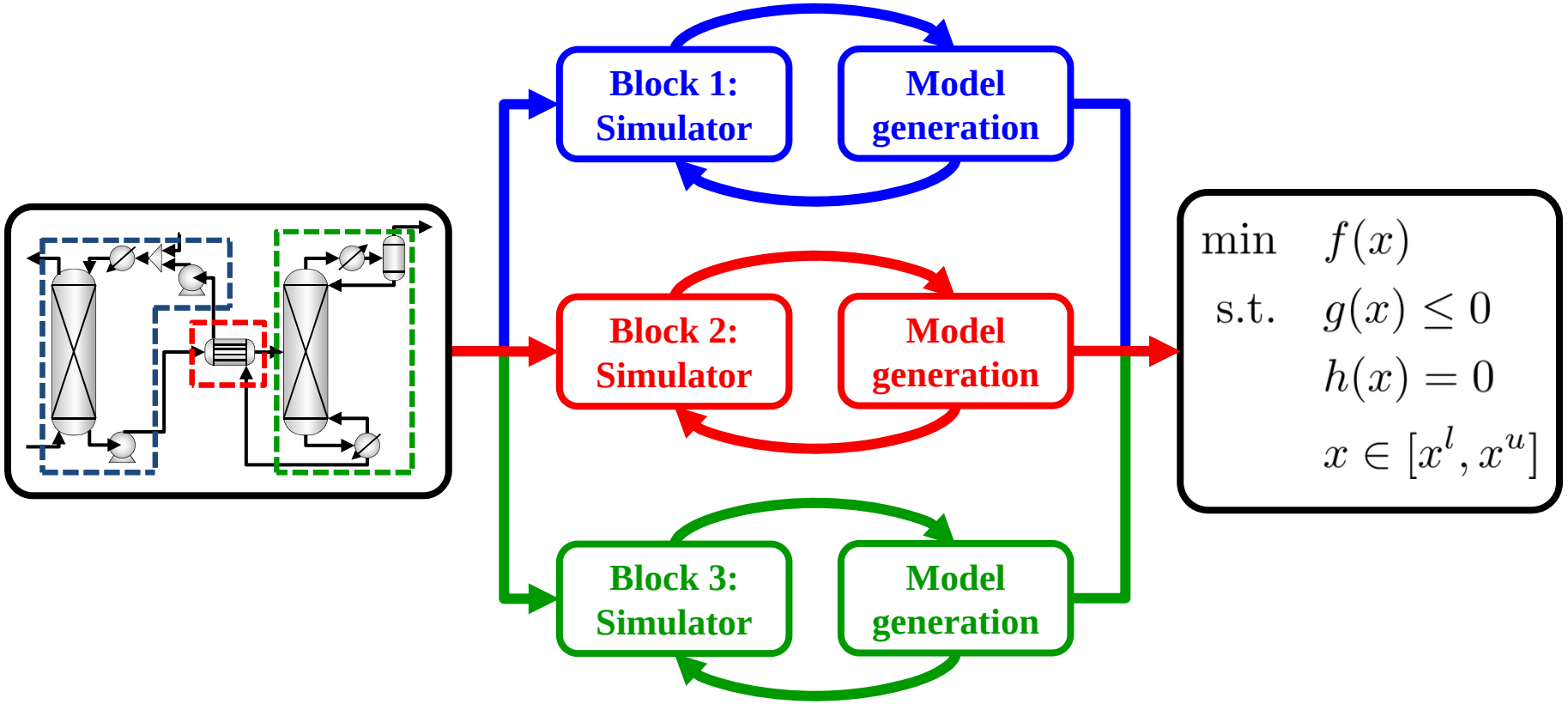
See Zhang and Sahinidis (IECR, 2013)

SIMULATION OPTIMIZATION



Pulverized coal plant Aspen Plus® simulation provided by the National Energy Technology Laboratory

PROCESS DISAGGREGATION



Process Simulation

Disaggregate process into process blocks

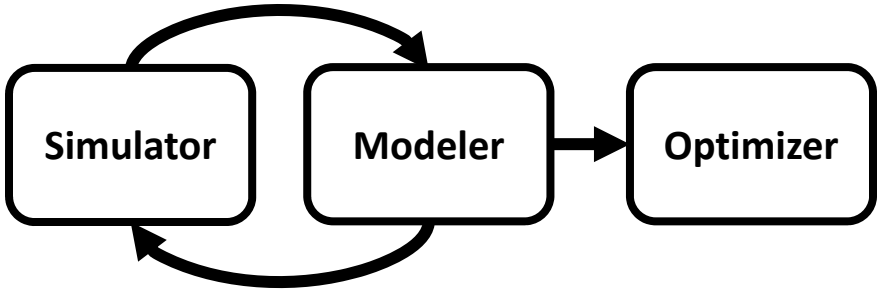
Surrogate Models

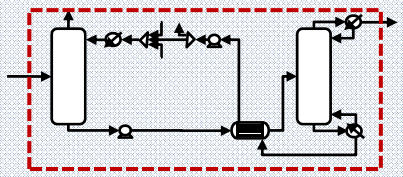
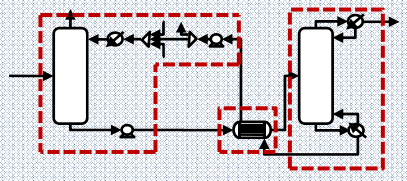
Build simple and accurate models with a functional form tailored for an optimization framework

Optimization Model

Add algebraic constraints design specs, heat/mass balances, and logic constraints

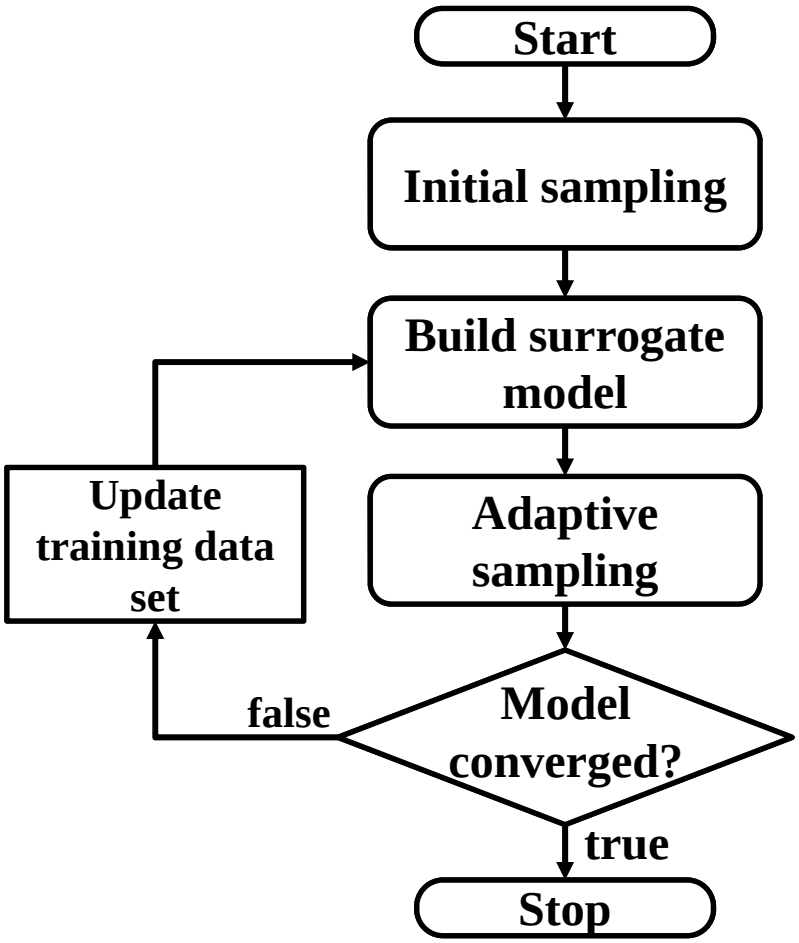
RECENT WORK IN CHEMICAL ENG

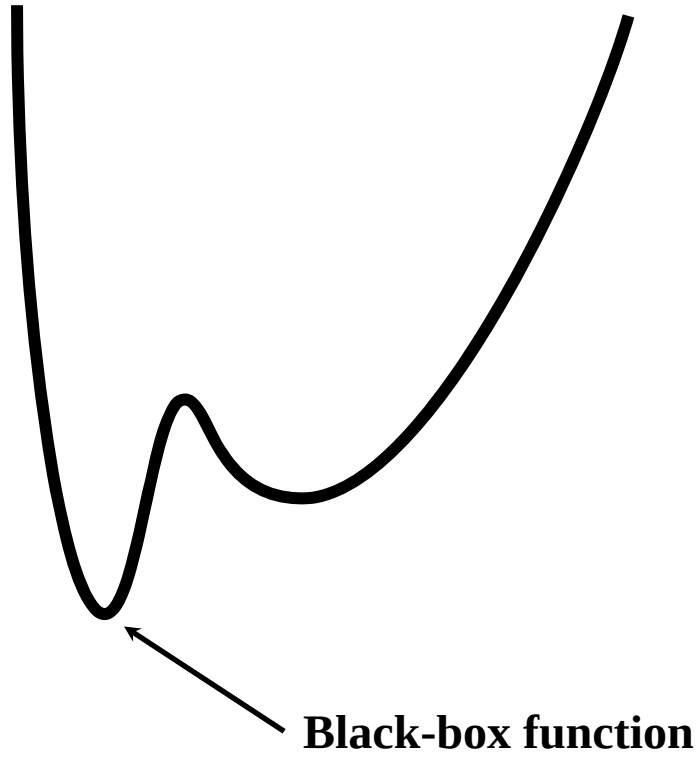


	Kriging	Neural nets	Polynomial-based
Full process 	▪ Palmer and Realff, 2002 ▪ Huang et al., 2006 ▪ Davis and Ierapetritou, 2012	▪ Michalopoulos et al., 2001	▪ Palmer and Realff, 2002
Disaggregated 	▪ Caballero and Grossmann, 2008	▪ Henao and Maravelias, 2011	

ALAMO

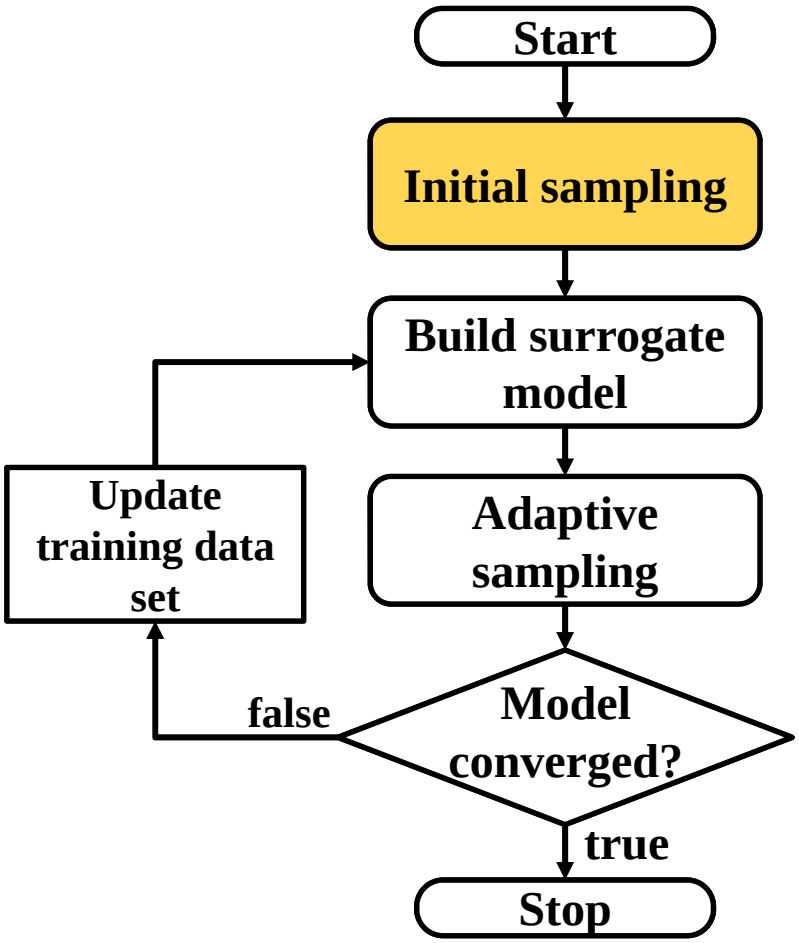
Automated Learning of Algebraic Models for Optimization

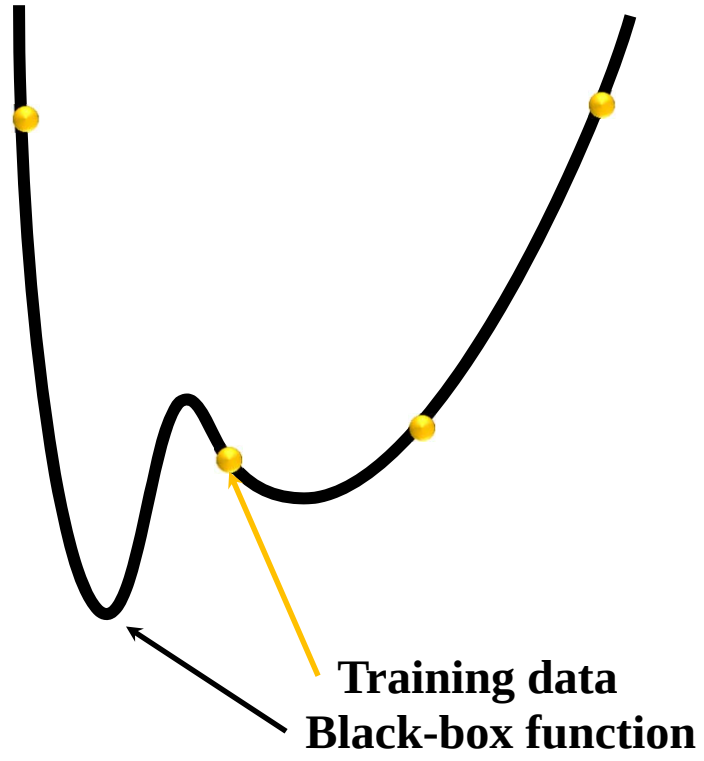




ALAMO

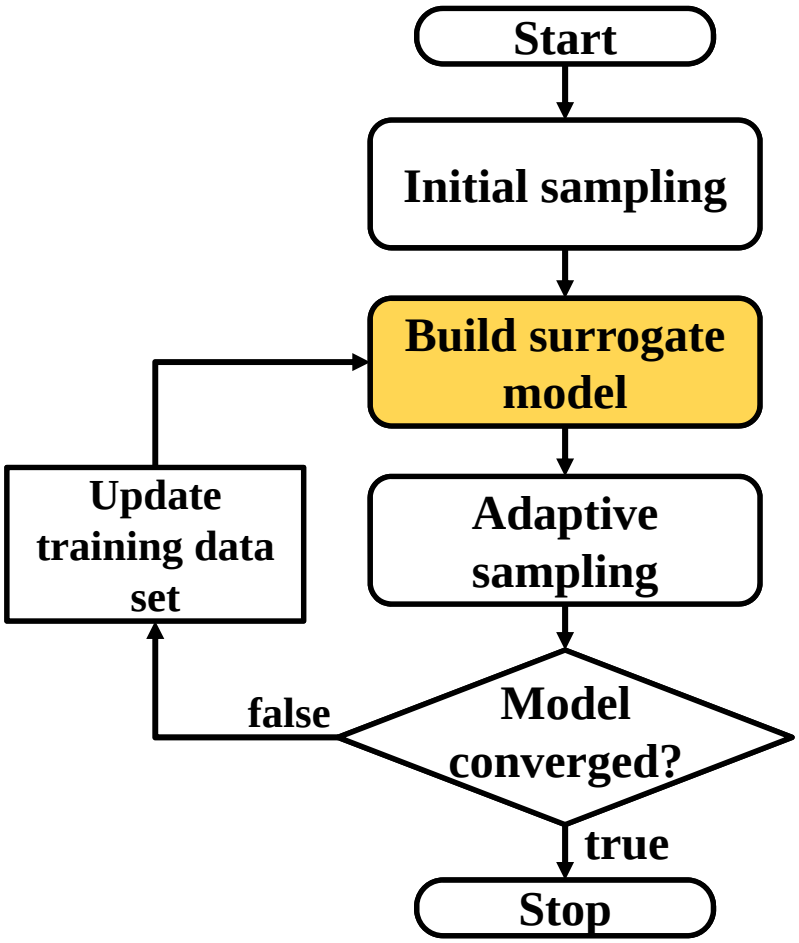
Automated Learning of Algebraic Models for Optimization

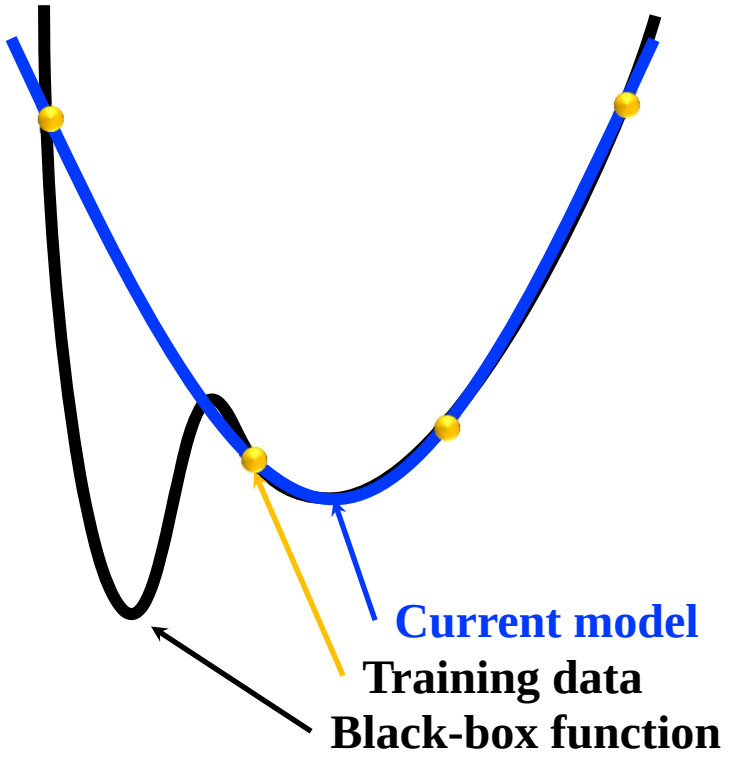




ALAMO

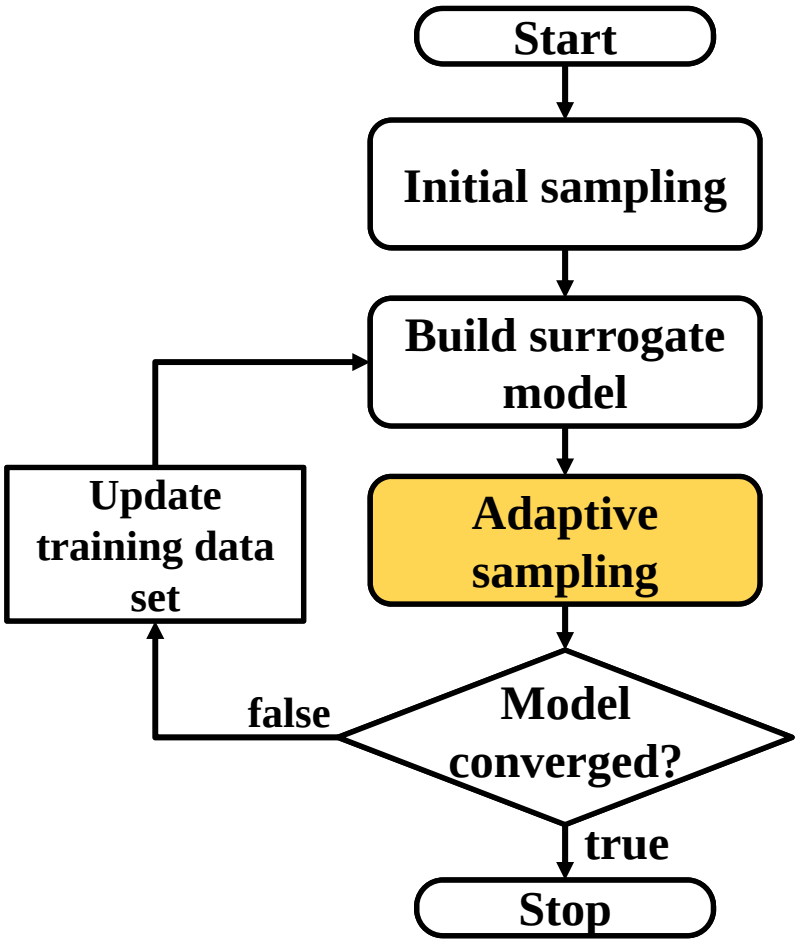
Automated Learning of Algebraic Models for Optimization

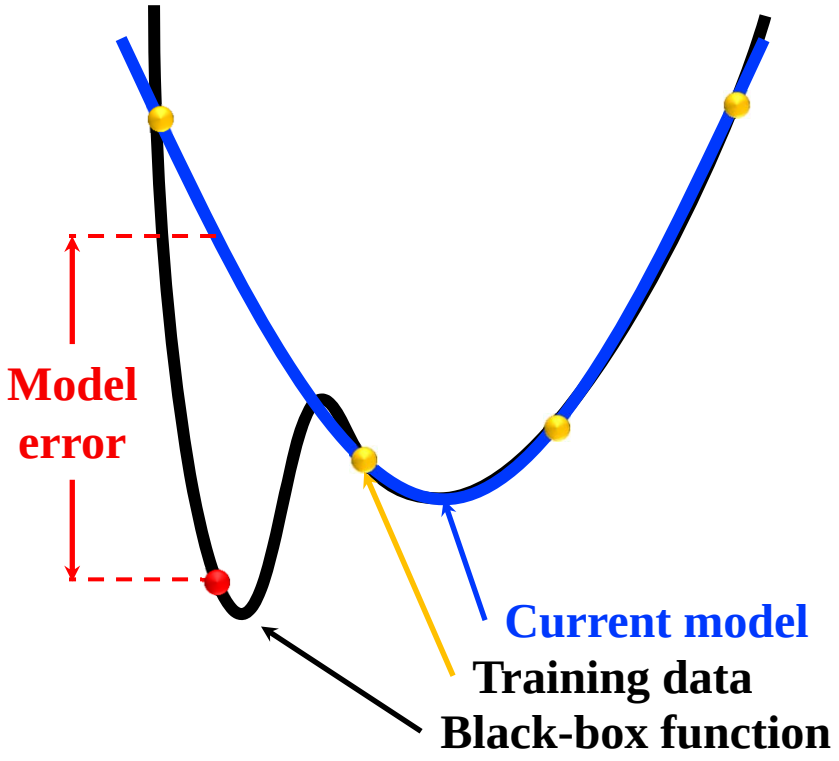




ALAMO

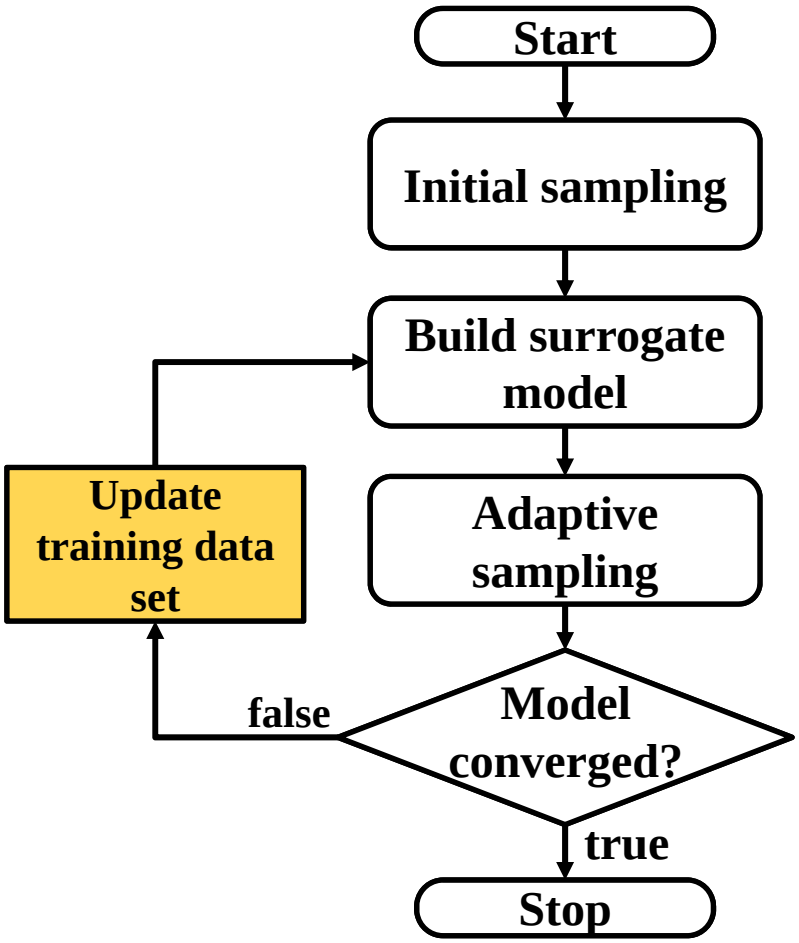
Automated Learning of Algebraic Models for Optimization

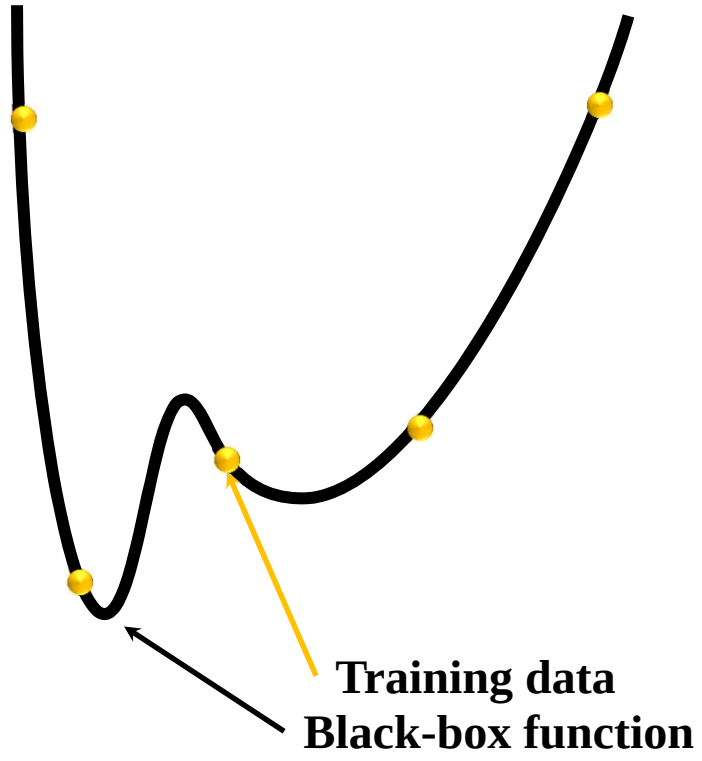




ALAMO

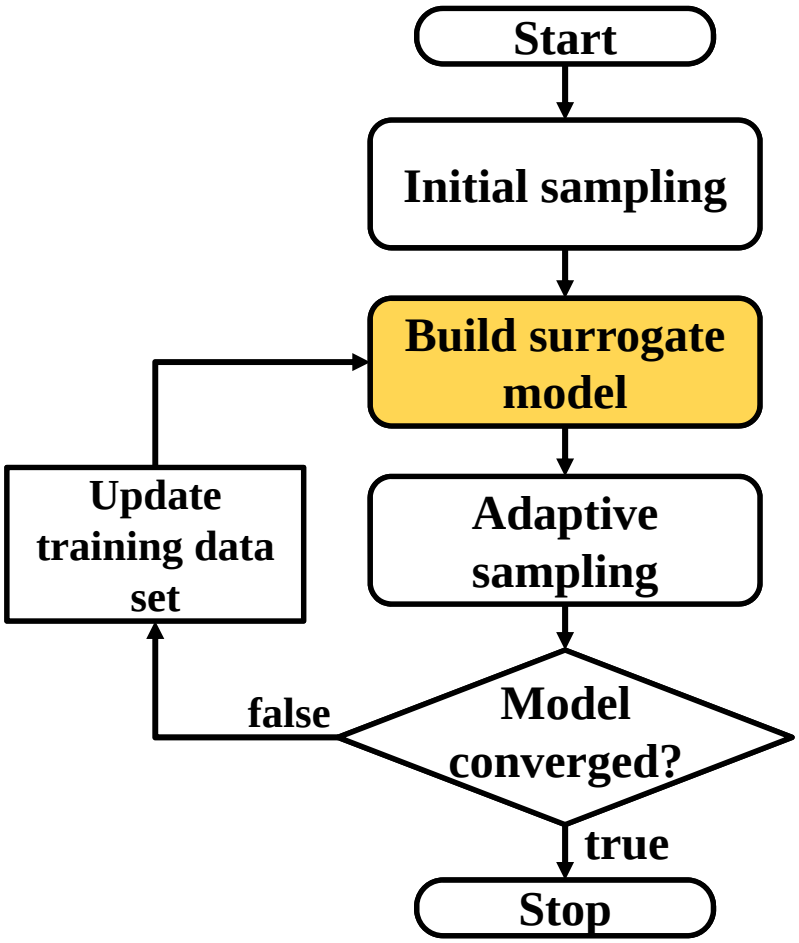
Automated Learning of Algebraic Models for Optimization

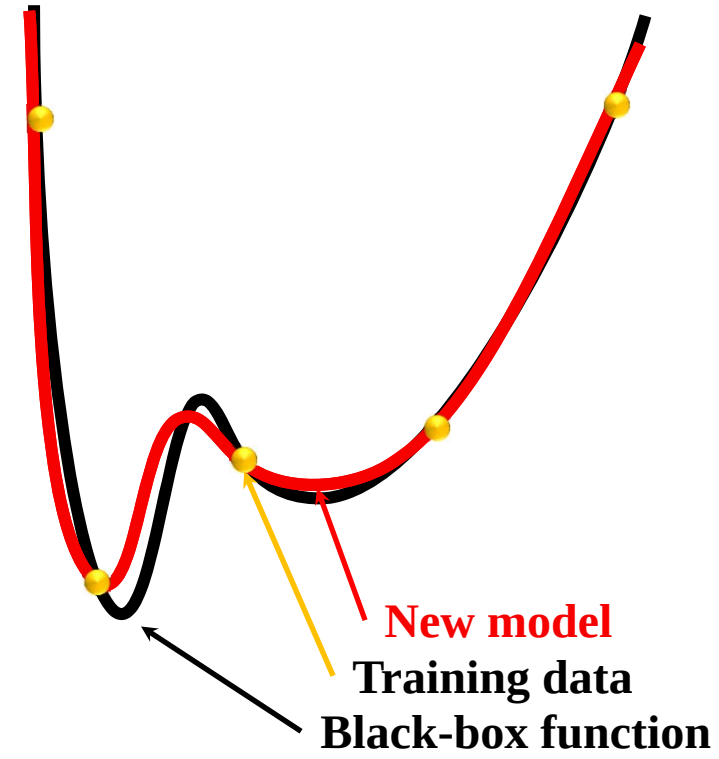




ALAMO

Automated Learning of Algebraic Models for Optimization





MODEL IDENTIFICATION

- **Goal: Identify the functional form and complexity of the surrogate models**

$$z = f(x)$$

- **Functional form:**
  - **General functional form is unknown: Our method will identify models with combinations of simple basis functions**

Category		$X_j(x)$
I.	Polynomial	$(x_d)^{\alpha}$
II.	Multinomial	$\prod_{d \in \mathcal{D}' \subseteq \mathcal{D}} (x_d)^{\alpha_d}$
III.	Exponential and logarithmic forms	$\exp\left(\frac{x_d}{\gamma}\right)^{\alpha}, \log\left(\frac{x_d}{\gamma}\right)^{\alpha}$
IV.	Expected bases	From experience, simple inspection, physical phenomena, etc.

BEST SUBSET SELECTION

Step 1: Define a large set of potential basis functions

$$\hat{z}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 \frac{x_1}{x_2} + \beta_5 \frac{x_2}{x_1} + \beta_6 e^{x_1} + \beta_7 e^{x_2} + \dots$$

BEST SUBSET SELECTION

Step 1: Define a large set of potential basis functions

$$\hat{z}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 \frac{x_1}{x_2} + \beta_5 \frac{x_2}{x_1} + \beta_6 e^{x_1} + \beta_7 e^{x_2} + \dots$$

Step 2: Model reduction

$$\hat{z}(x) = \beta_0 + \beta_2 x_2 + \beta_5 \frac{x_2}{x_1} + \beta_7 e^{x_2}$$

BEST SUBSET SELECTION

Step 1: Define a large set of potential basis functions

$$\hat{z}(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 \frac{x_1}{x_2} + \beta_5 \frac{x_2}{x_1} + \beta_6 e^{x_1} + \beta_7 e^{x_2} + \dots$$

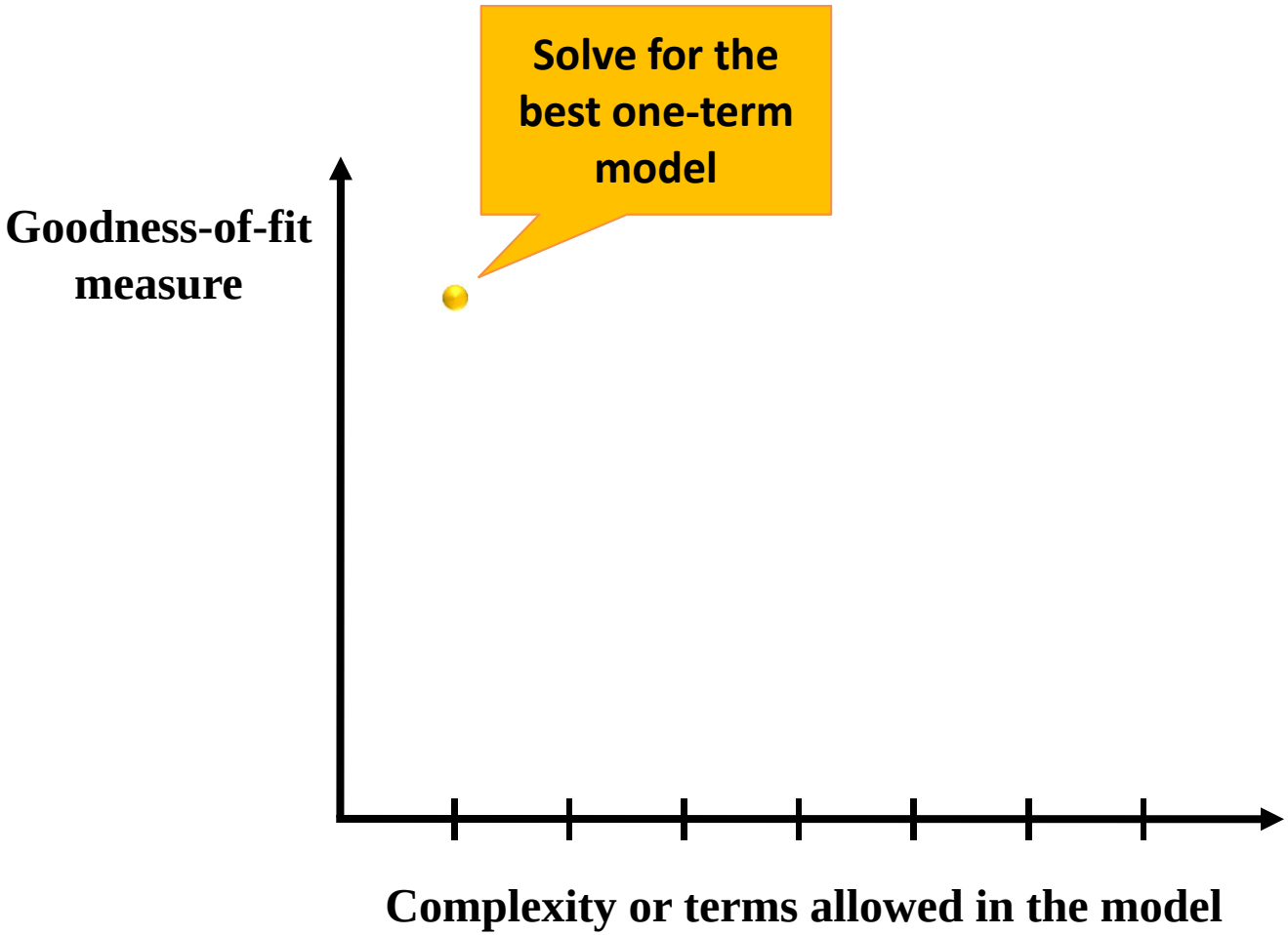
Step 2: Model reduction

$$\hat{z}(x) = \beta_0 + \beta_2 x_2 + \beta_5 \frac{x_2}{x_1} + \beta_7 e^{x_2}$$

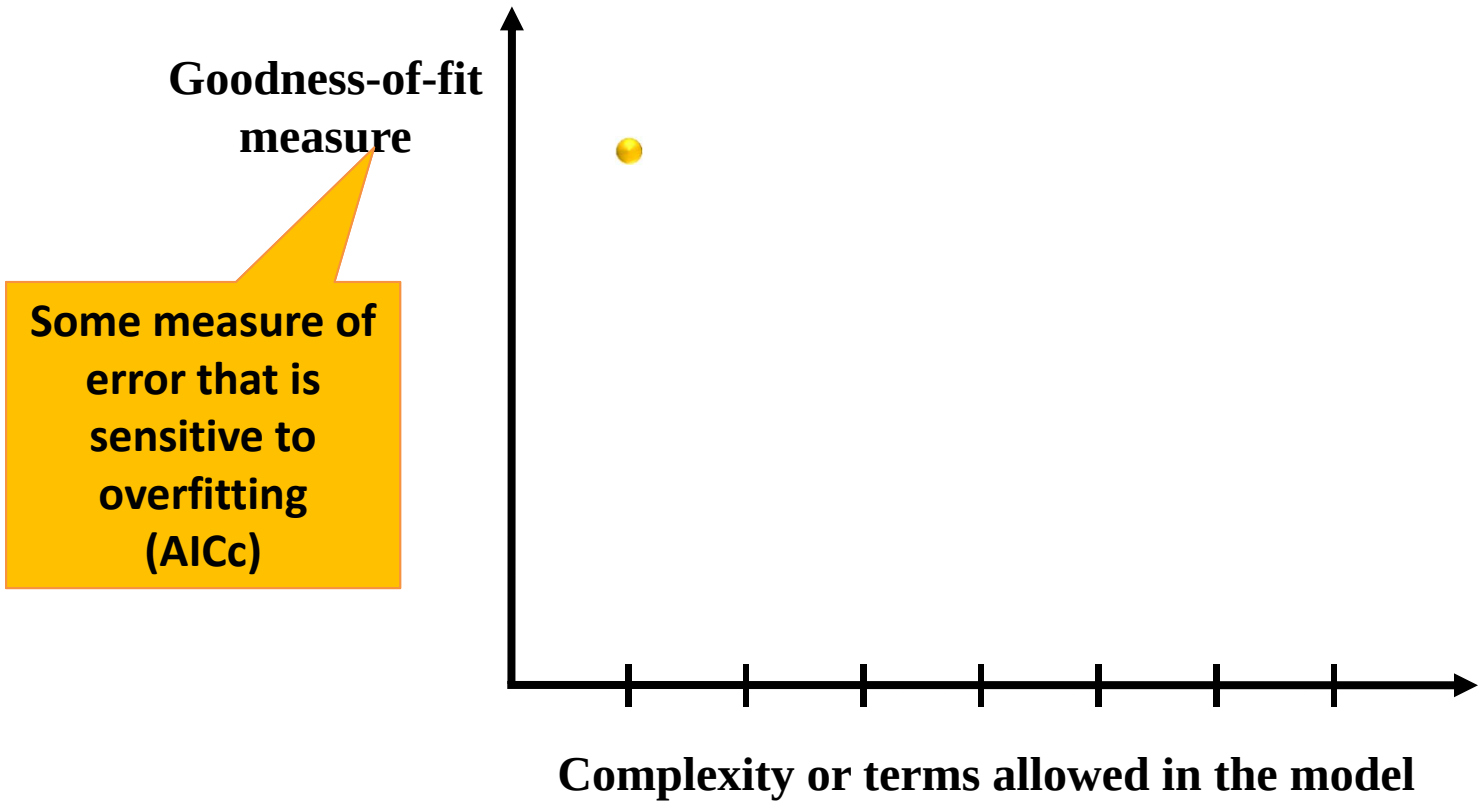
To identify a simple functional form we need to solve two problems:

1. **Model Sizing**
2. **Basis function selection**

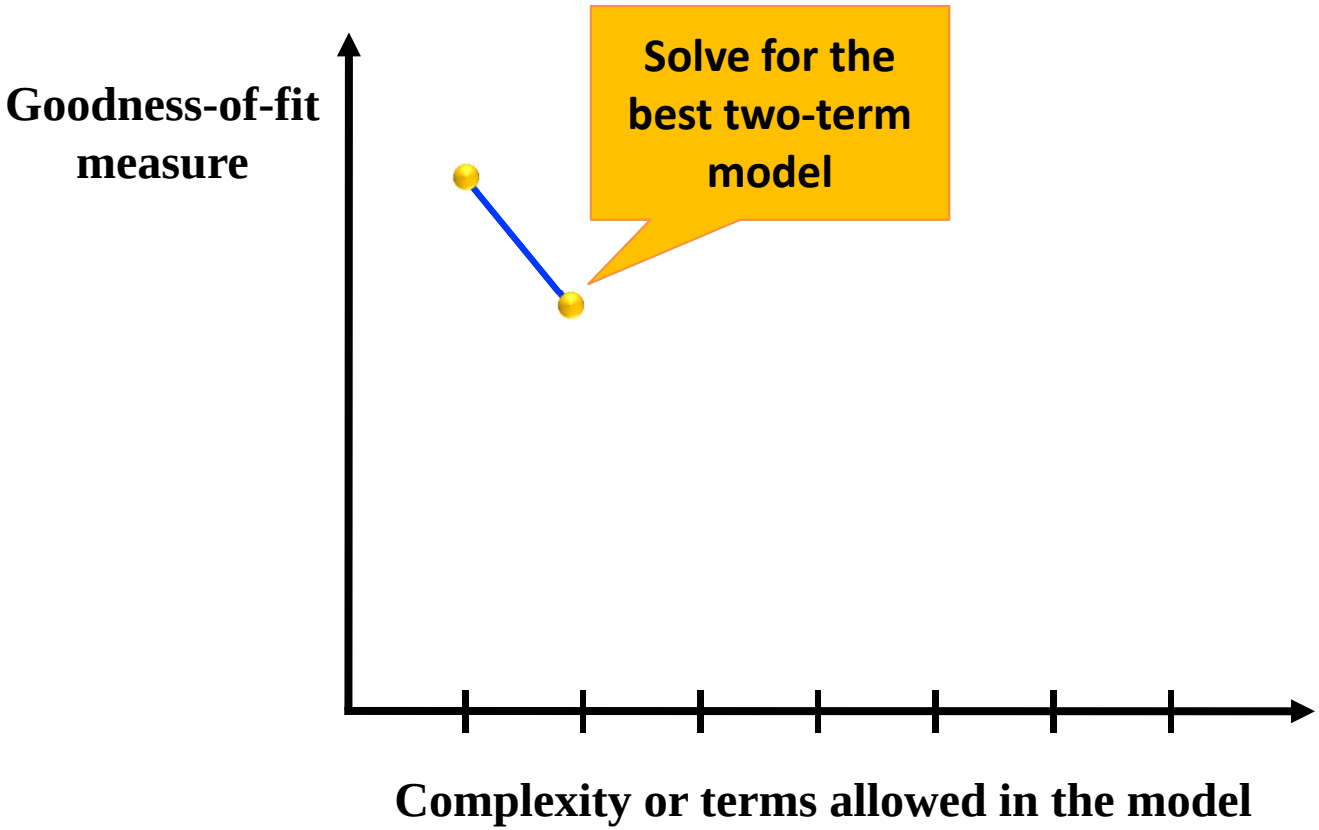
MODEL SIZING



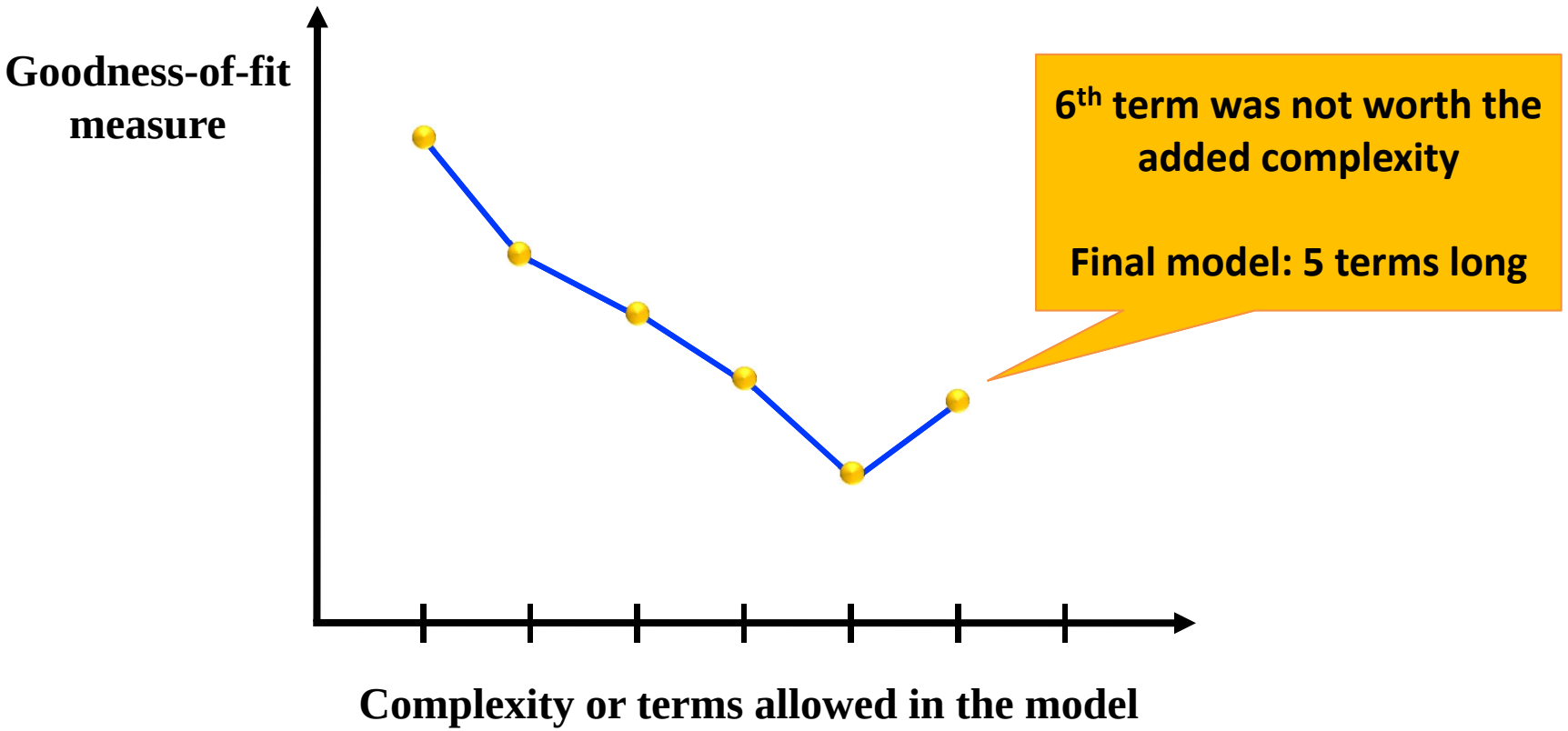
MODEL SIZING



MODEL SIZING

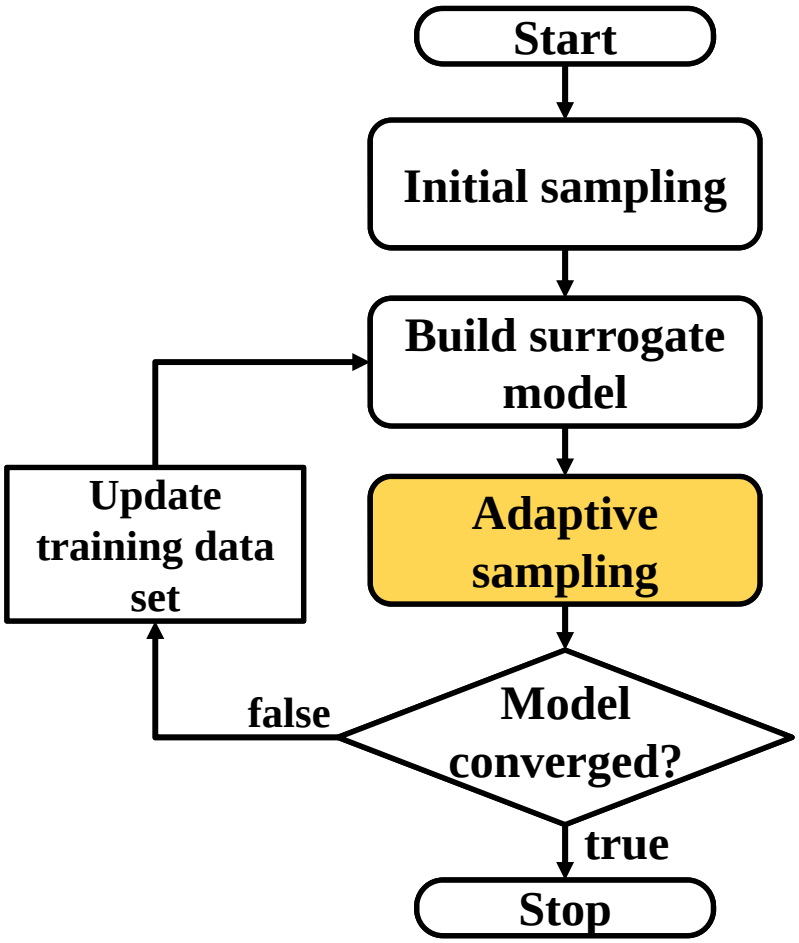


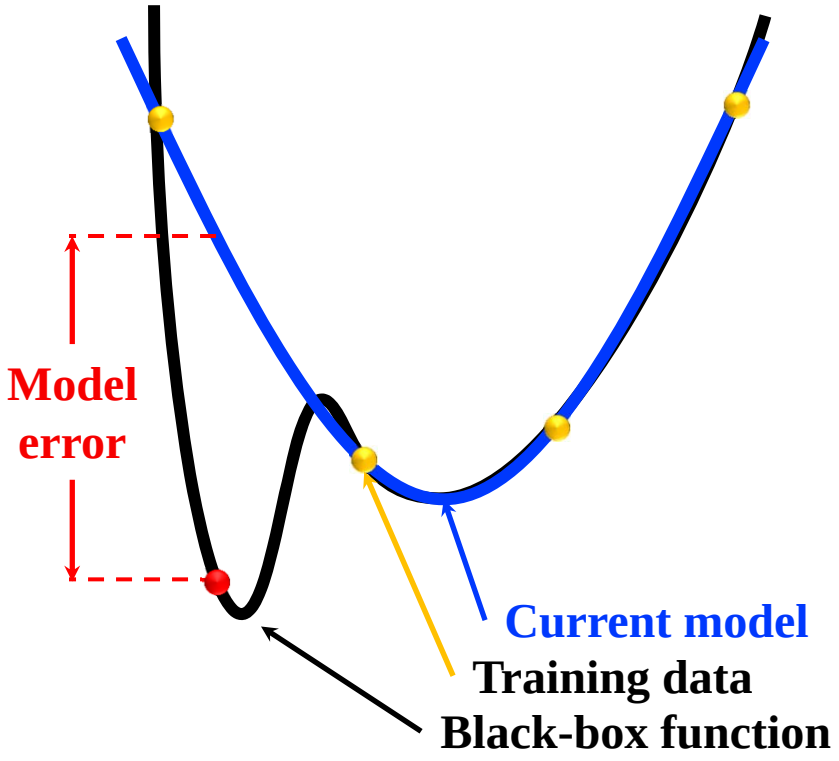
MODEL SIZING



ALAMO

Automated Learning of Algebraic Models for Optimization

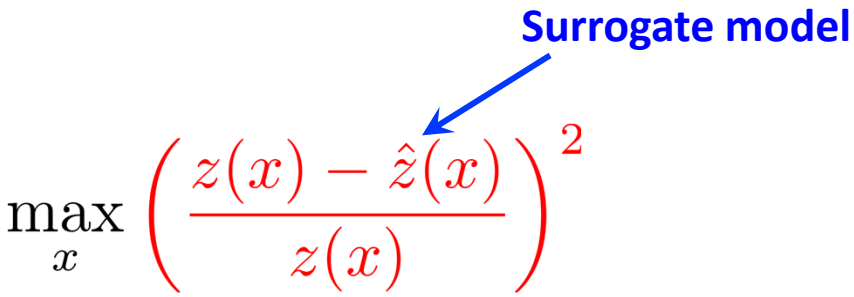




ERROR MAXIMIZATION SAMPLING

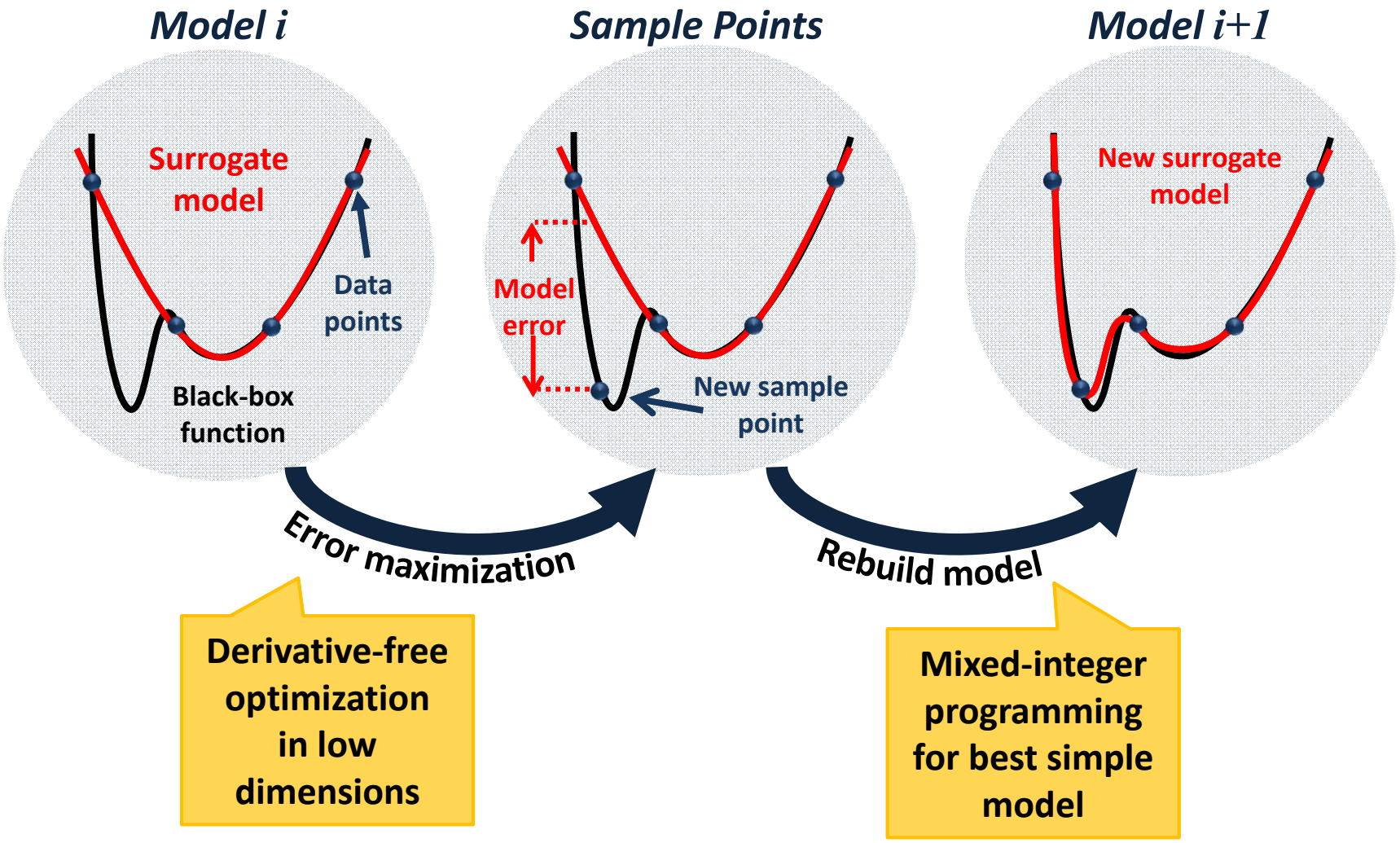
- **New goal: Search the problem space for areas of model inconsistency or model mismatch**

- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**



  - **Optimized using a black-box or derivative-free solver (SNOBFIT)** [Huyer and Neumaier, 08]
  - **Derivative-free solvers work well in low-dimensional spaces** [Rios and Sahinidis, 12]

SYNOPSIS



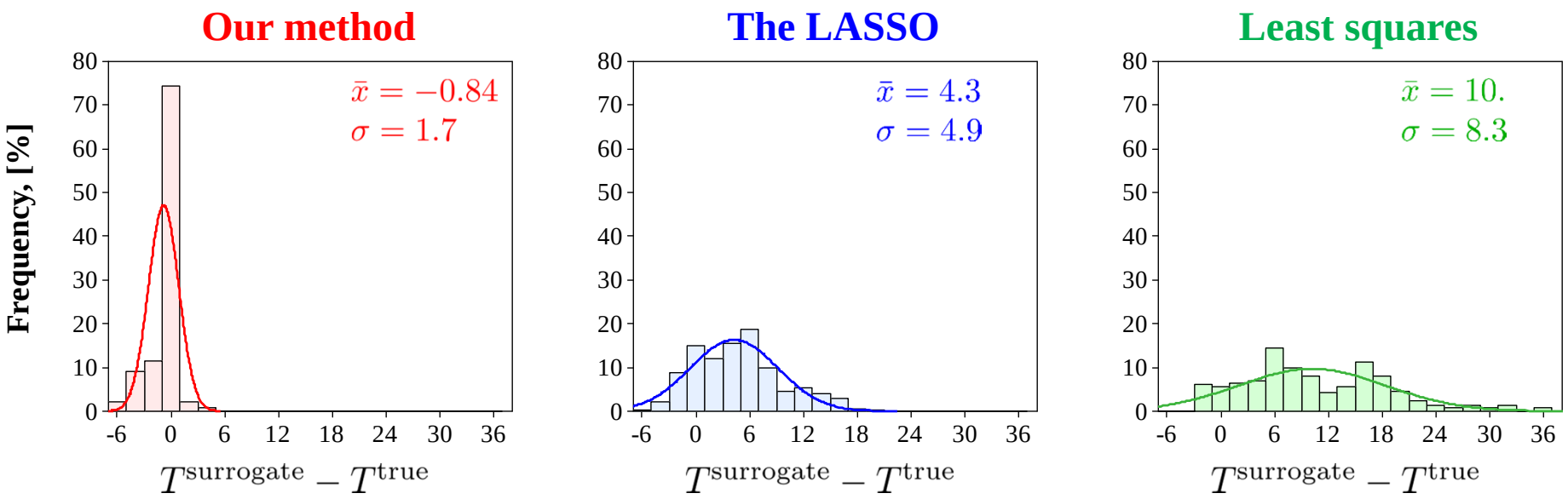
See paper 589b (Th 8:50 am)

COMPUTATIONAL TESTING

- **Paper 589b: Test the accuracy, efficiency, and model simplicity**
- **Modeling methods compared**
  - **MIP – Proposed methodology**
  - **LASSO – The lasso regularization**
  - **OLR – Ordinary least-squares regression**
- **Sampling methods compared**
  - **EMS – Proposed error maximization technique**
  - **SLH – Single Latin hypercube (no feedback)**

MODEL SIZING RESULTS

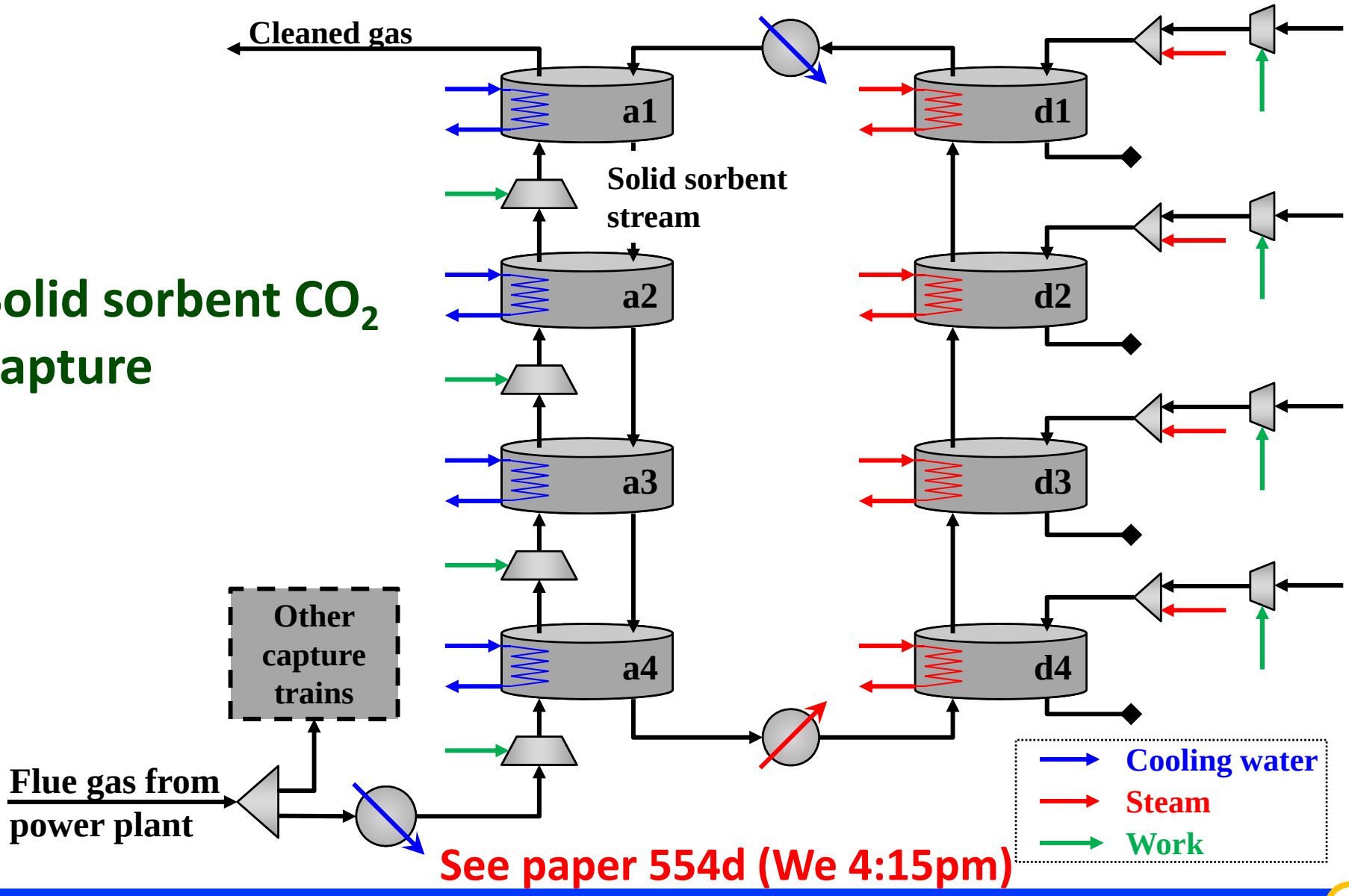
[Number of terms in the surrogate model] – [Number of terms in the true function]



45 problems with 2-10 available bases, 5 repeats

SUPERSTRUCTURE OPTIMIZATION

Solid sorbent CO_2 capture



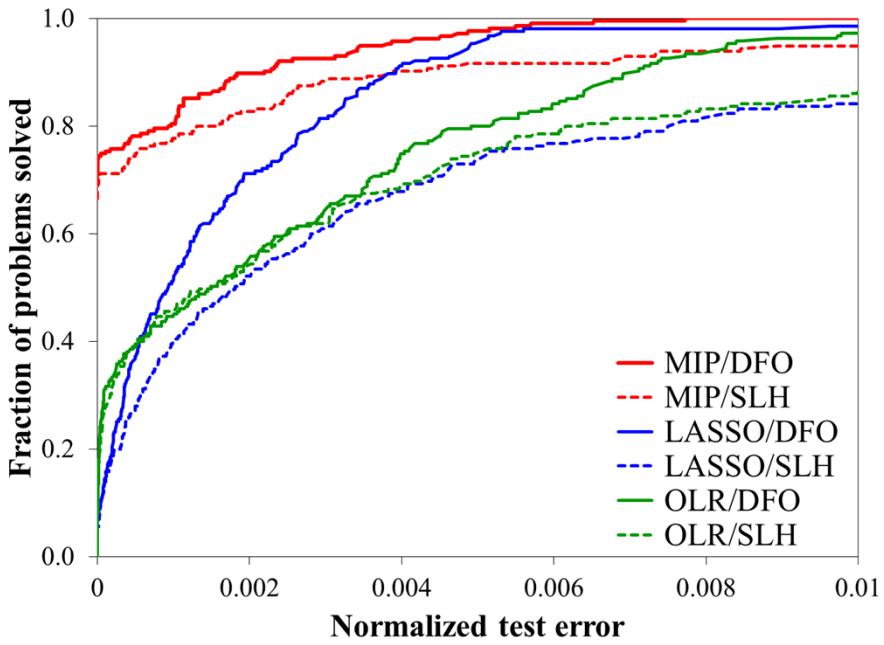
See paper 554d (We 4:15pm)

CONCLUSIONS

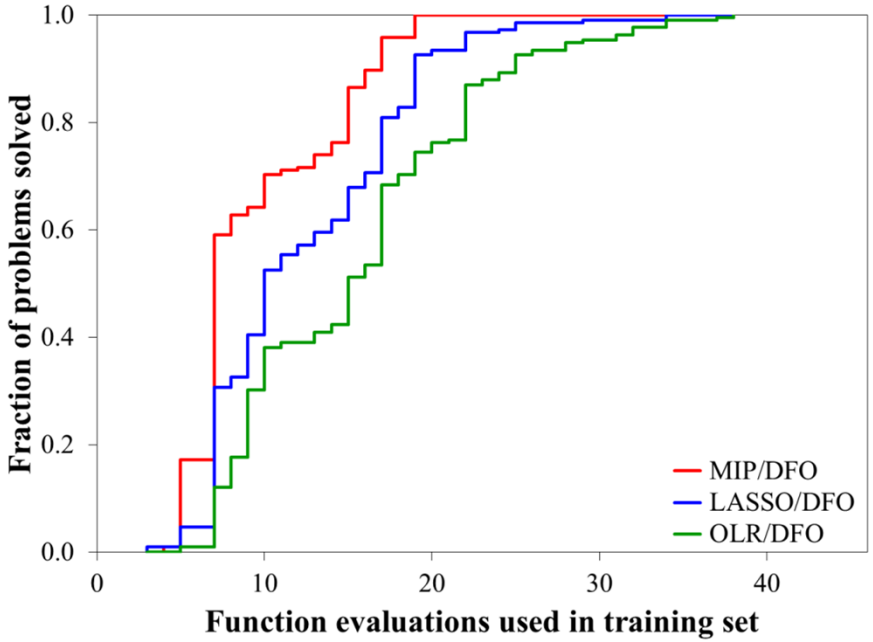
- **Best subset methods provide models that are**
  - ✓ **Accurate representations of black-box models**
- **ALAMO provides algebraic models that are**
  - ✓ **Generated from a minimal number of function evaluations**
  - ✓ **Tractable in an optimization framework (low-complexity models)**
- **Surrogate models can then be incorporated within an optimization or risk assessment framework**
- **Learning algorithms are domain independent**
- **ALAMO site: archimedes.cheme.cmu.edu/?q=alamo**

RESULTS

Model accuracy



Modeling efficiency



Modeling methods

- Our method
- LASSO
- Least squares

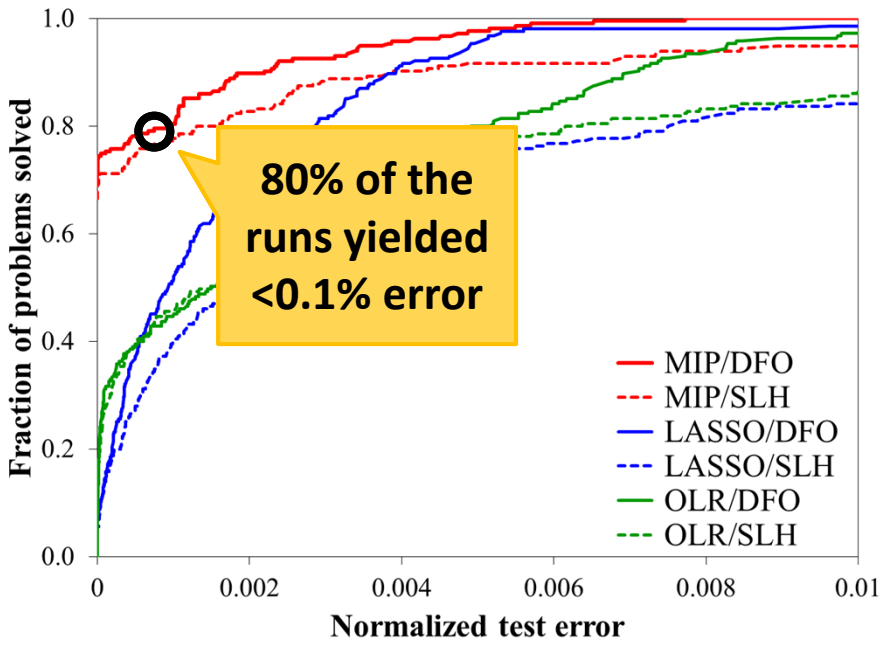
Sampling methods

- Error maximization
- Single Latin hypercube

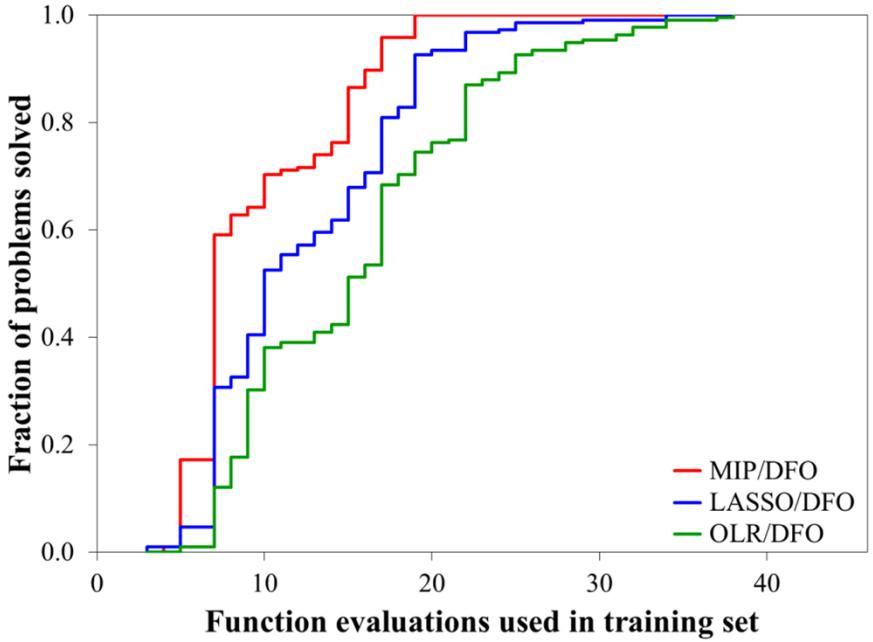
45 test problems, repeated 5 times, tested against 1000 independent data points

RESULTS

Model accuracy



Modeling efficiency



Modeling methods

- Our method
- LASSO
- Least squares

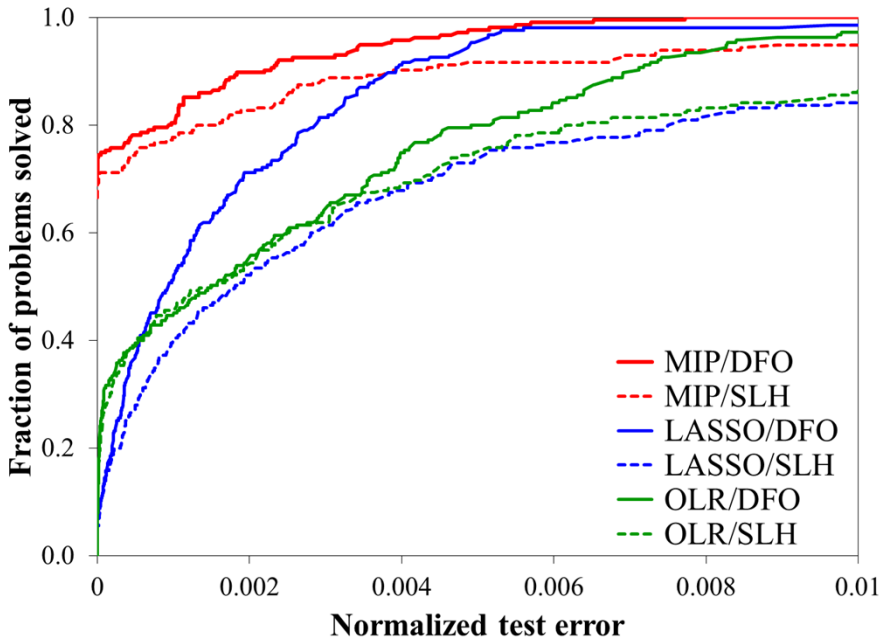
Sampling methods

- Error maximization
- Single Latin hypercube

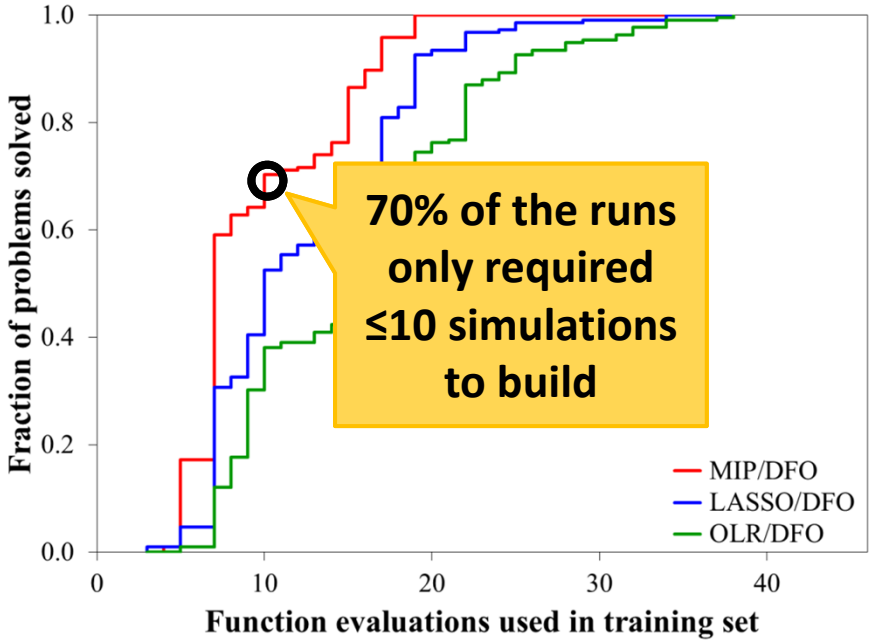
45 test problems, repeated 5 times, tested against 1000 independent data points

RESULTS

Model accuracy



Modeling efficiency



Modeling methods

- Our method
- LASSO
- Least squares

Sampling methods

- Error maximization
- Single Latin hypercube

45 test problems, repeated 5 times, tested against 1000 independent data points