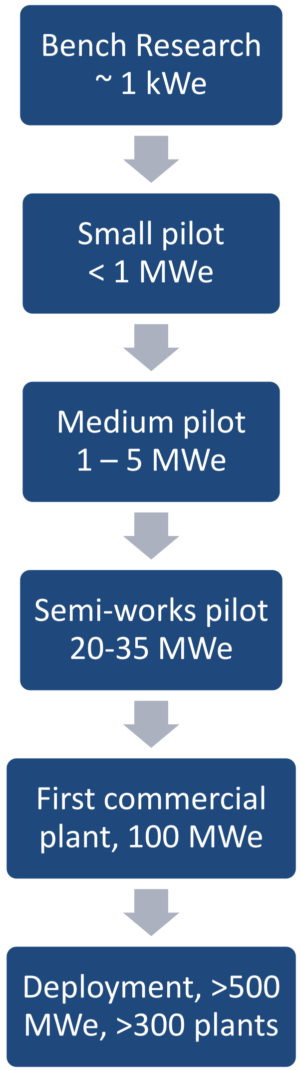
ALAMO: Automatic Learning of Algebraic Models for Optimization

Alison Cozad[1,2], Nick Sahinidis[1,2], David Miller[2]

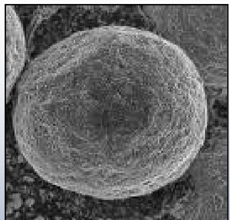
[1]National Energy Technology Laboratory, Pittsburgh, PA,USA
[2]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

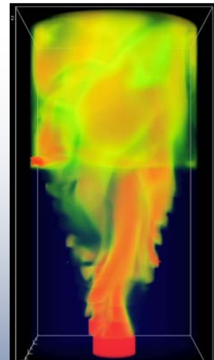
CARBON CAPTURE CHALLENGE

- **The traditional pathway from discovery to commercialization of energy technologies can be quite long, i.e., ~ 2-3 decades**
- **President's plan requires that barriers to the widespread, safe, and cost-effective deployment of CCS be overcome within 10 years**
- **To help realize the President's objectives, new approaches are needed for taking carbon capture concepts from lab to power plant, quickly, and at low cost and risk**
- **CCSI will accelerate the development of carbon capture technology, from discovery through deployment, with the help of science-based simulations**



CARBON CAPTURE SIMULATION INITIATIVE





Identify promising concepts → **Reduce the time for design & troubleshooting** → **Quantify the technical risk, to enable reaching larger scales, earlier** → **Stabilize the cost during commercial deployment**

National Labs

Berkeley Lab

Pacific Northwest National Laboratory

NETL

Los Alamos National Laboratory EST. 1943

Lawrence Livermore National Laboratory

Academia

Carnegie Mellon

Princeton University

West Virginia University

Boston University

Institute for Clean and Secure Energy, The University of Utah

Industry

Fluor

ADA Environmental Solutions

B&W the babcock & wilcox company

GE

EPRI Electric Power Research Institute

Air Products

Burns & McDonnell Since 1898

Southern California Edison, An Edison International Company, San Onofre Nuclear Generating Station

Alstom

Southern Company

Duke Energy

URS

Boeing

DuPont

ExxonMobil

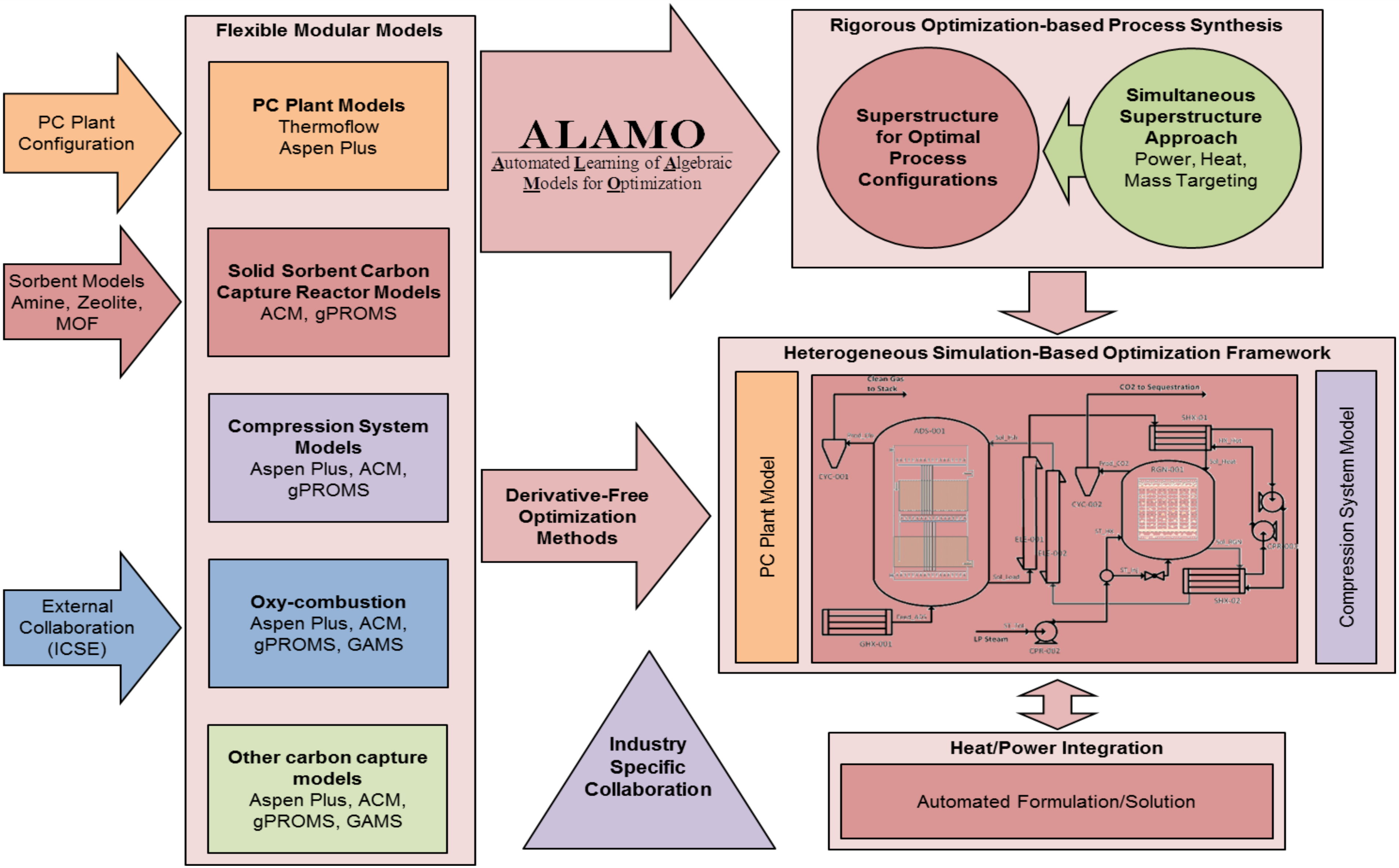
Chevron

Phillips 66

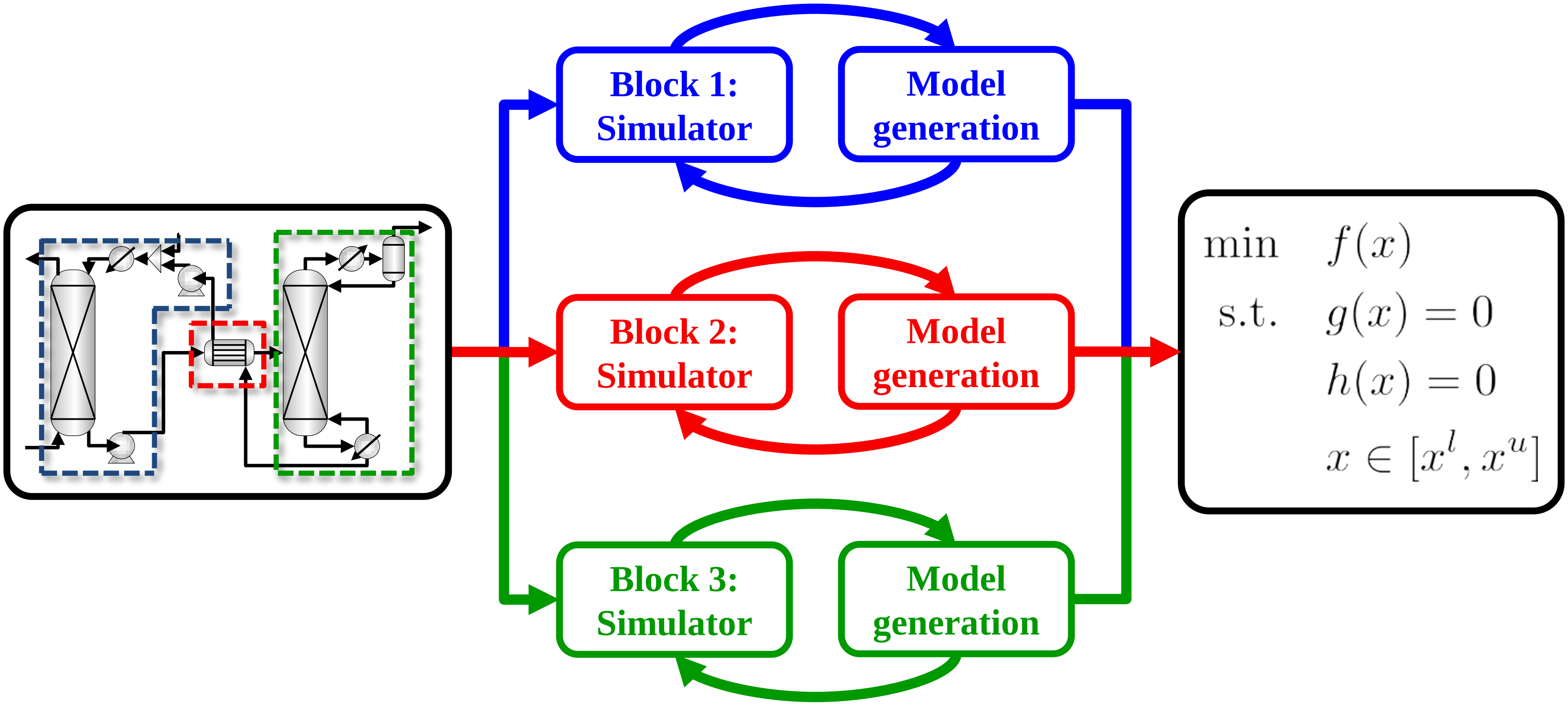
WorleyParsons resources & energy

Eastman

CENTRAL ACTIVITY: OPTIMIZATION



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

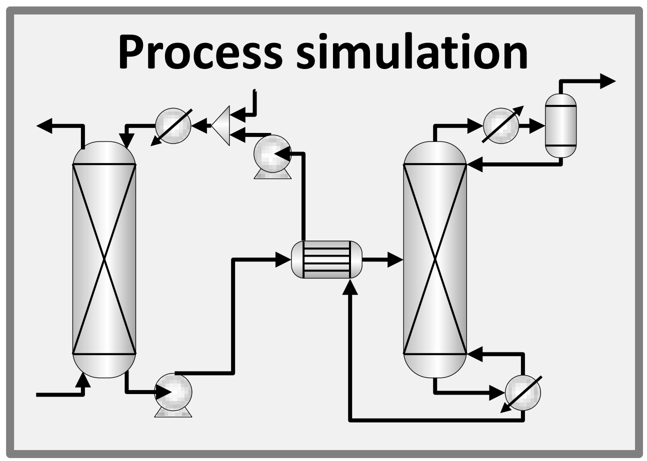
Optimization Model

Add algebraic constraints *h(x)*=0: design specs, heat/mass balances, and logic constraints

LEARNING PROBLEM STATEMENT

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

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



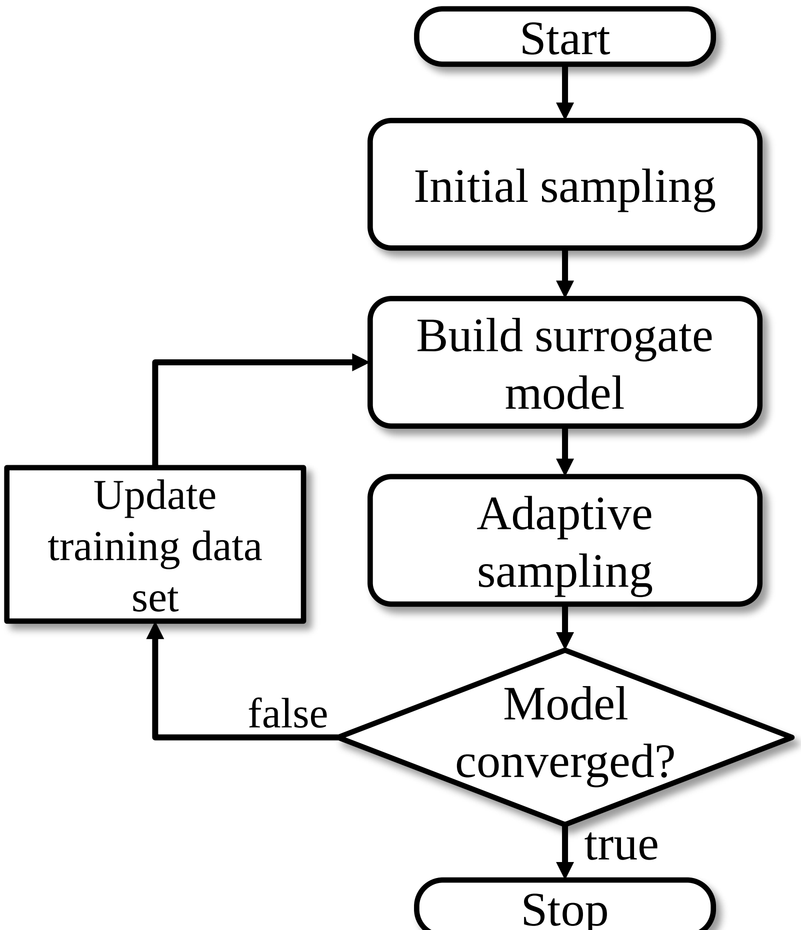
$$\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \\ \vdots \\ z_K \end{pmatrix} \qquad z \in \mathbb{R}^K \qquad z = f(x)$$

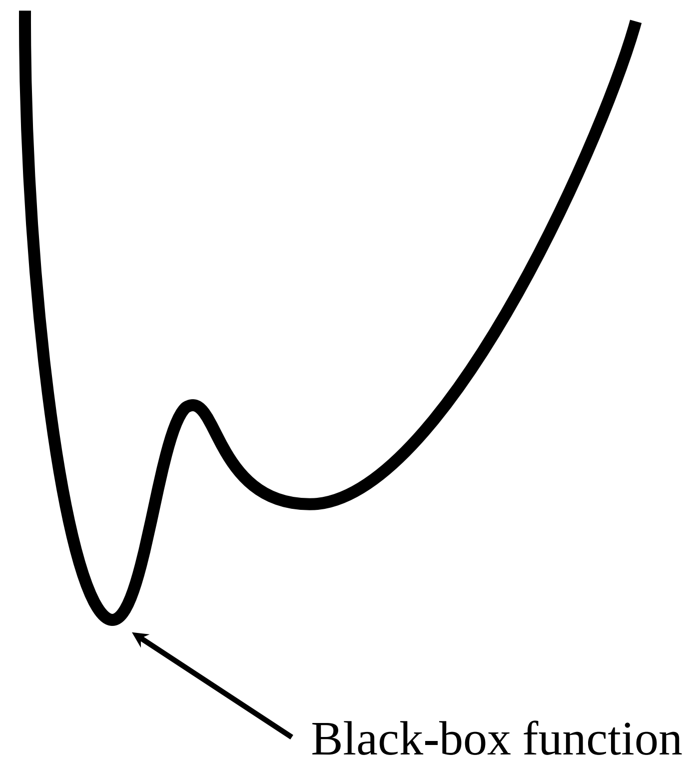
Independent variables:
Operating conditions, inlet flow properties, unit geometry

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

ALAMO

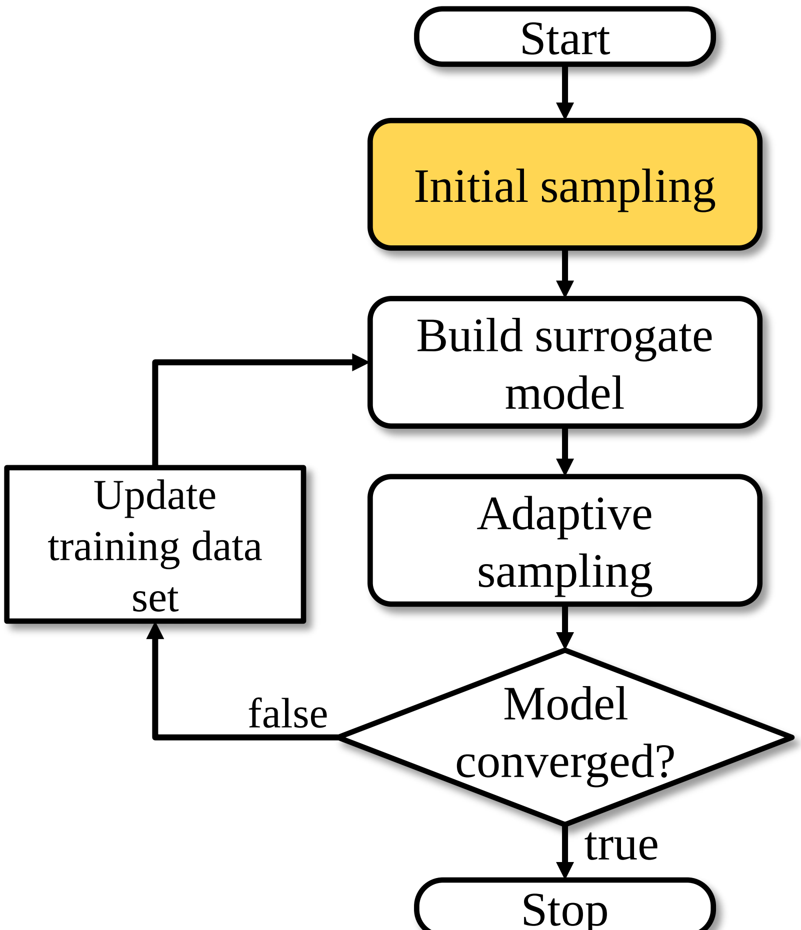
Automated **L**earning of **A**lgebraic **M**odels for **O**ptimization

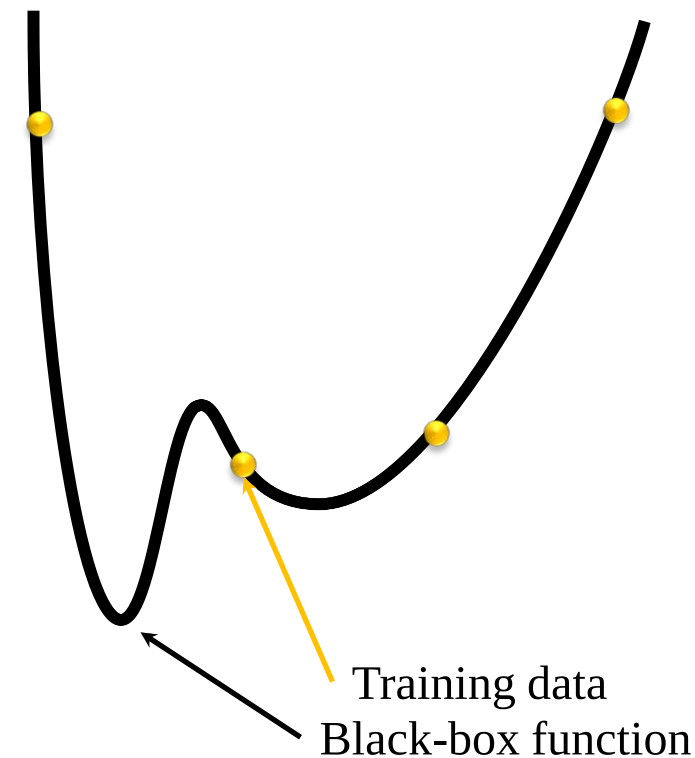




ALAMO

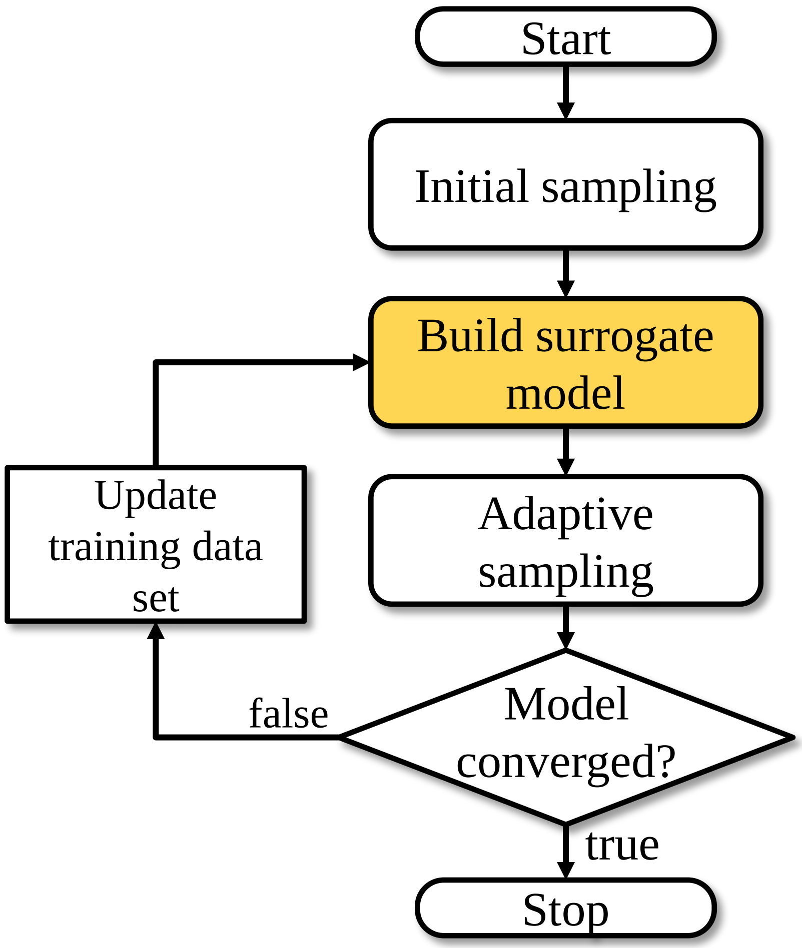
Automated Learning of Algebraic Models for Optimization

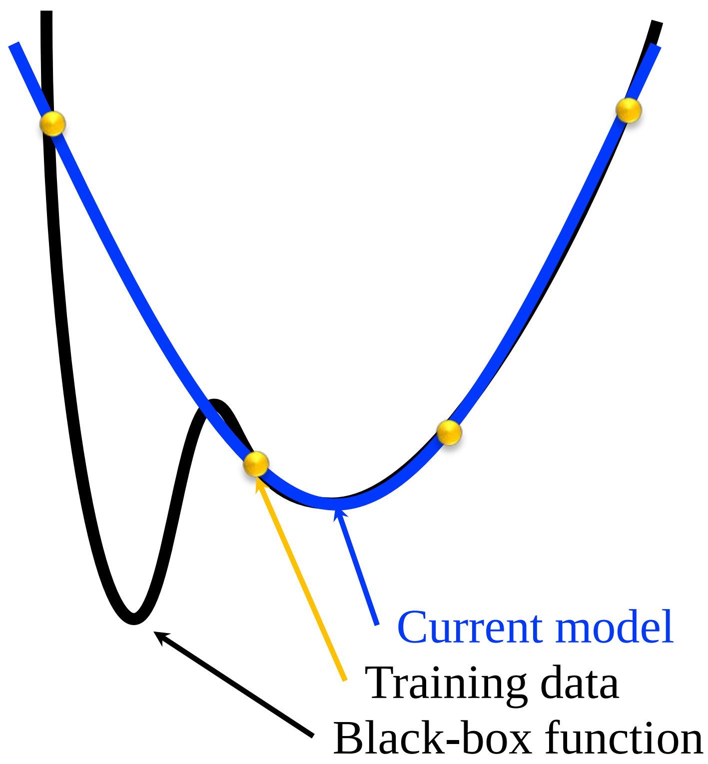




ALAMO

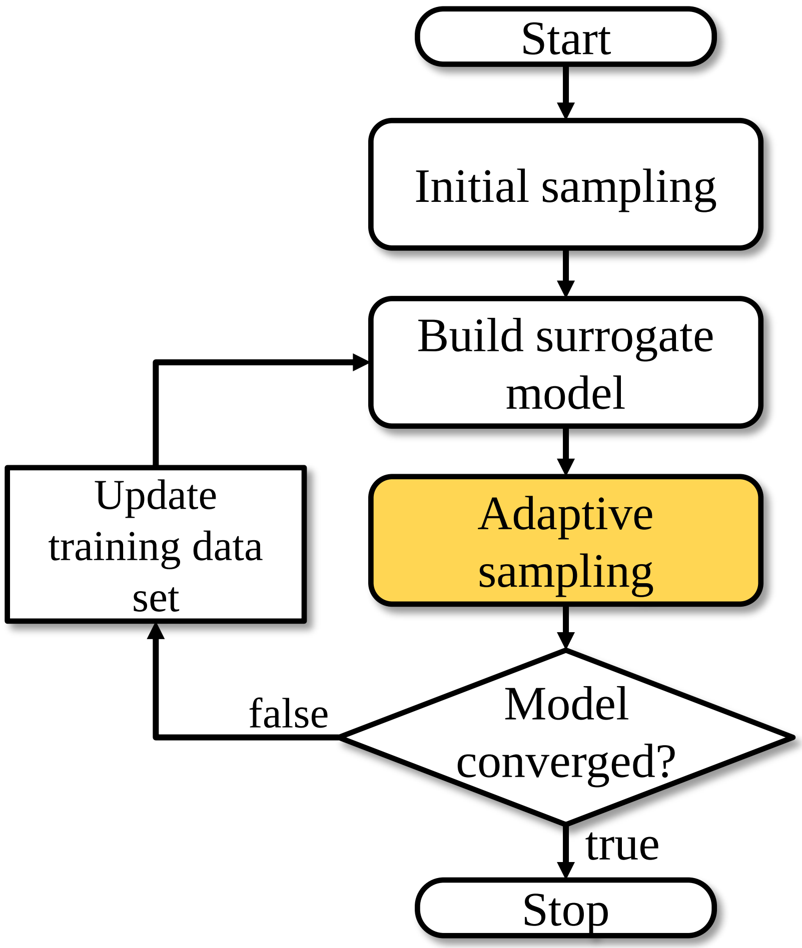
Automated Learning of Algebraic Models for Optimization

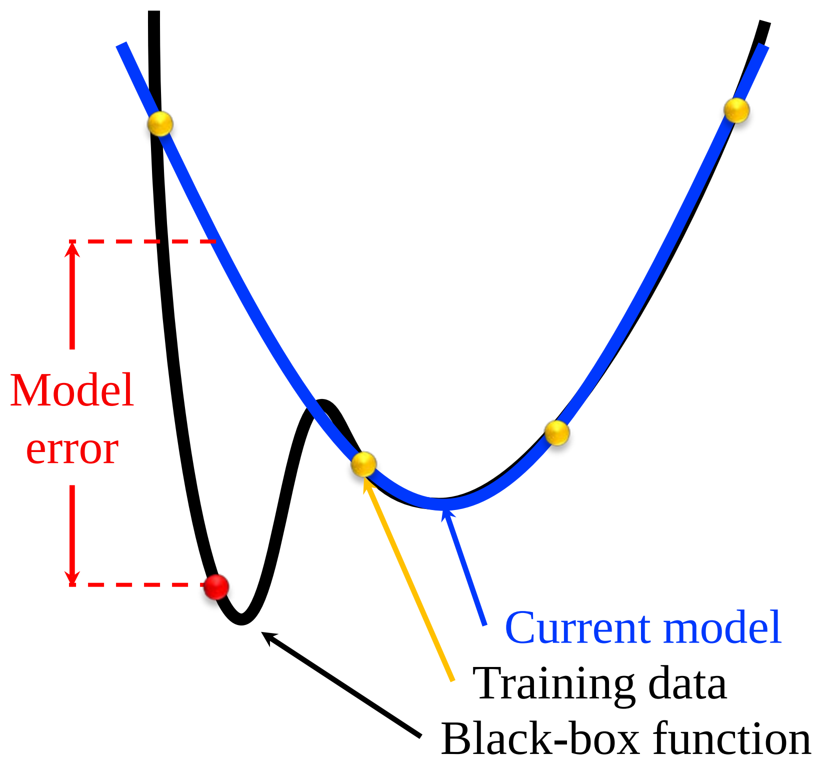




ALAMO

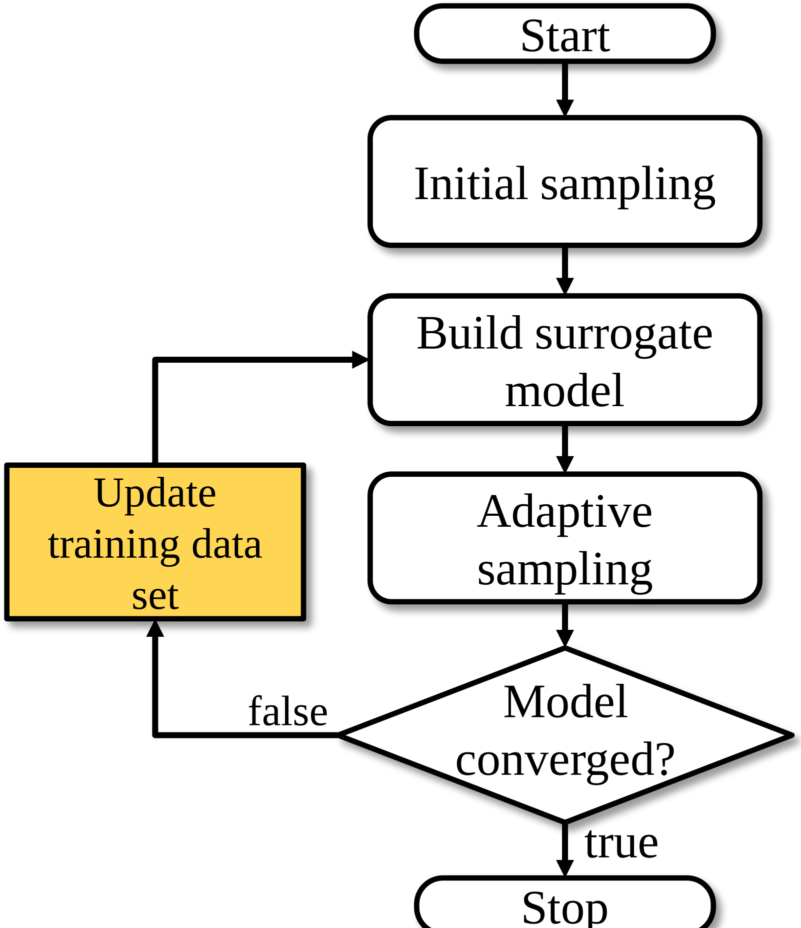
Automated **L**earning of **A**lgebraic **M**odels for **O**ptimization

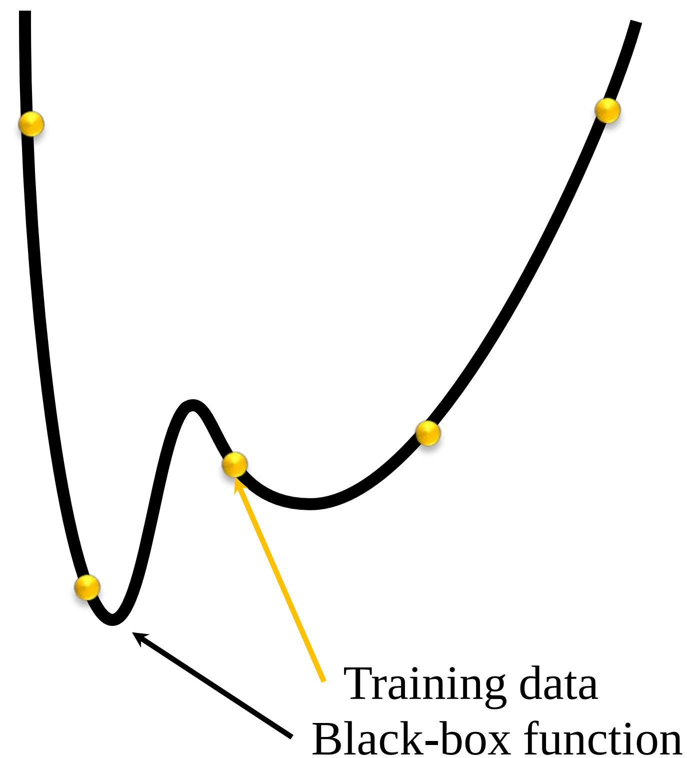




ALAMO

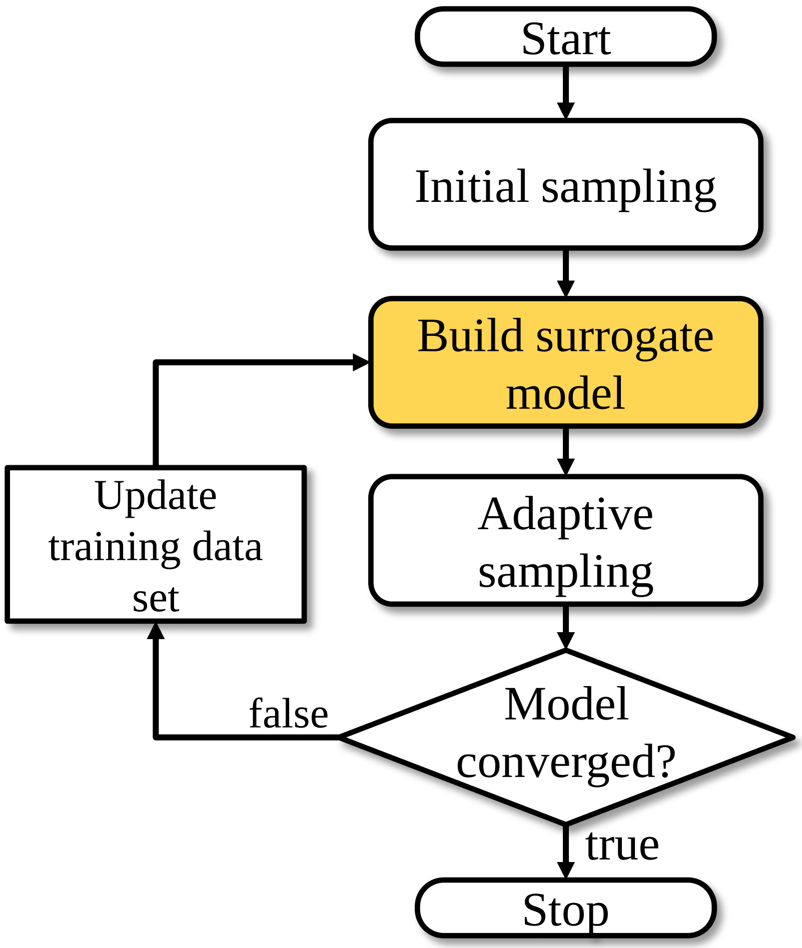
Automated Learning of Algebraic Models for Optimization

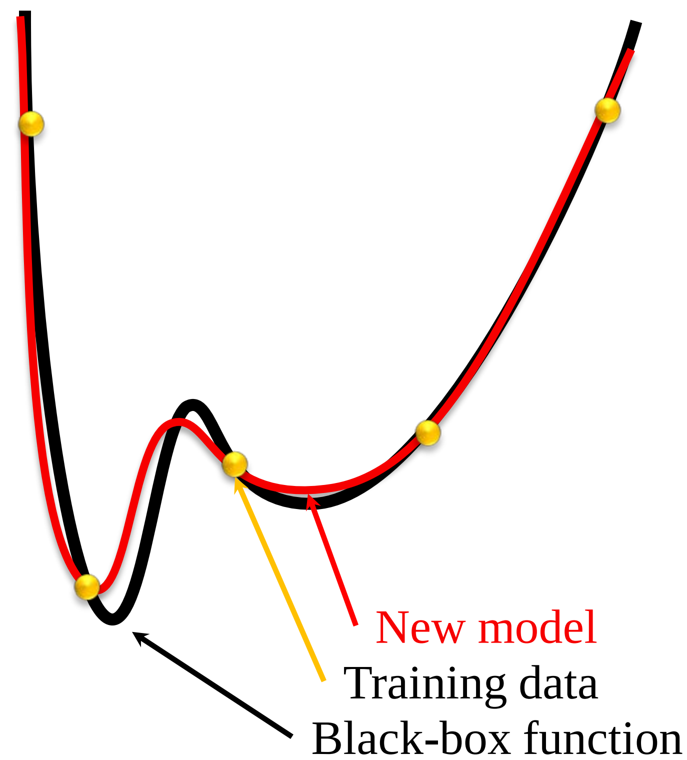




ALAMO

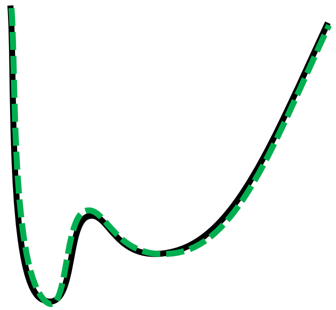
Automated Learning of Algebraic Models for Optimization





HOW TO BUILD THE SURROGATES

- **We aim to build surrogate models that are**
  - **Accurate**
    - ***We want to reflect the true nature of the simulation***
  - **Tailored for algebraic optimization**



$$\hat{f}(x) = \sum_{i=1}^{n} \gamma_i exp\left(\frac{\|x\|}{\sigma^2}\right) + \beta_0 + \beta_1\, x + \ldots$$

$$\hat{f}(x) = \beta_1\, x + \beta_2\, x^2 + \beta_3\, x^3 + \beta_4\, e^x$$

  - **Generated from a minimal data set**

MODEL IDENTIFICATION

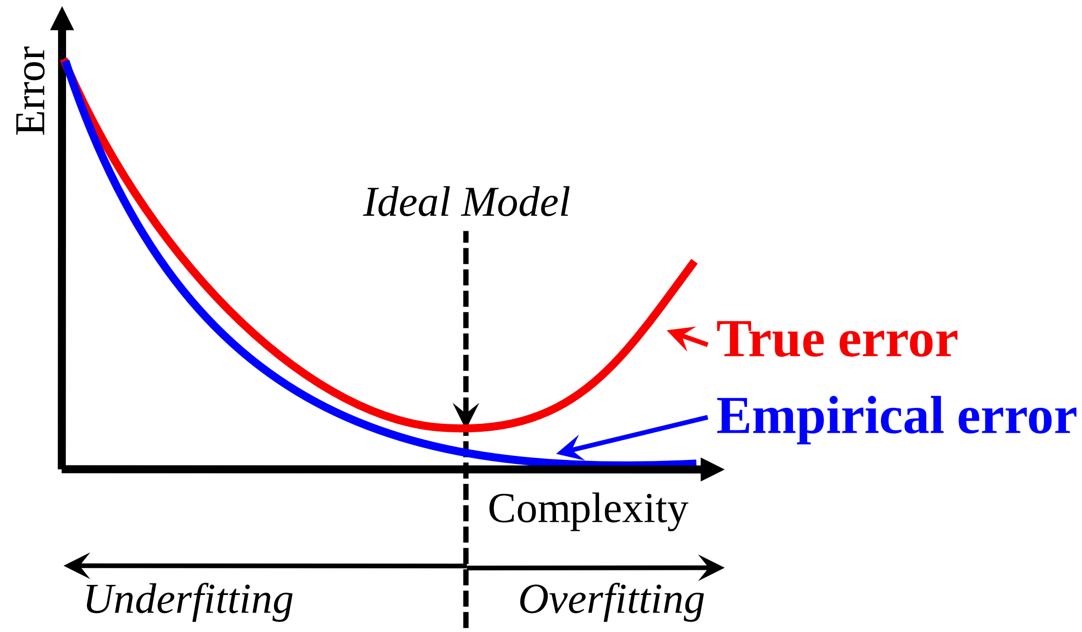
- **Goal: Identify the functional form and complexity of the surrogate models**

$$z = f(x)$$

- **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.

OVERFITTING AND TRUE ERROR



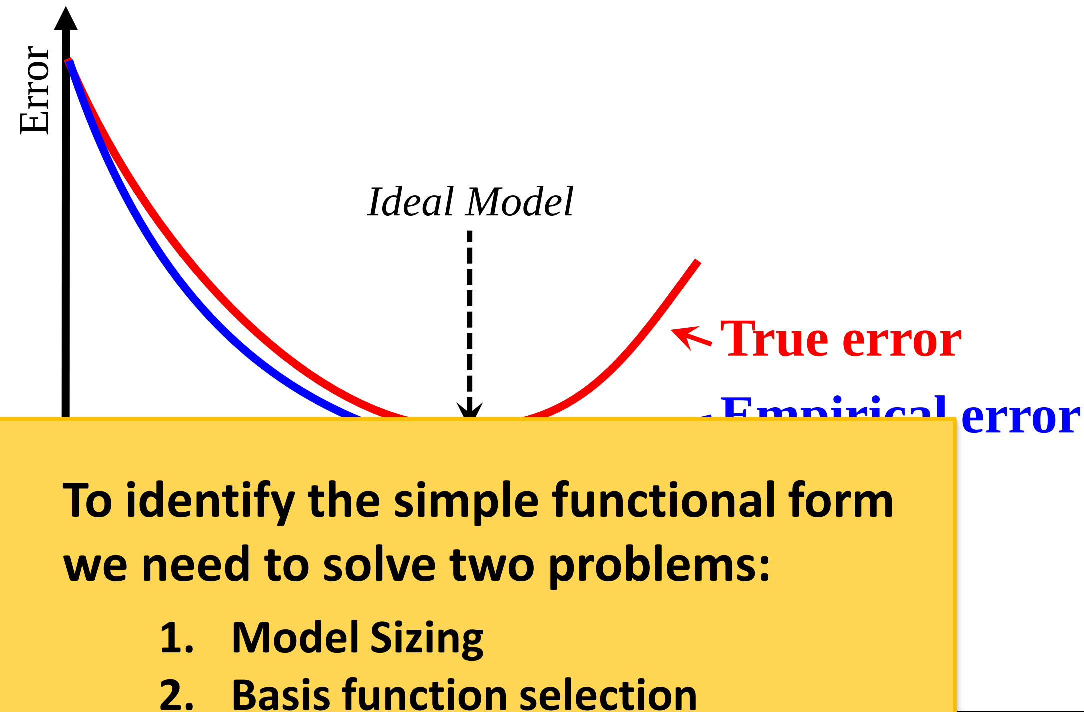
Step 1: Define a large set of potential basis functions

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

Step 2: Model reduction

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

OVERFITTING AND TRUE ERROR



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

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

Step 1: Define a large set of … functions

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

Step 2: Model reduction

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

BASIS FUNCTION SELECTION

$$\min \quad SE = \sum_{i=1}^{N} \left| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right|$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$$

$$-U(1 - y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1 - y_j) \qquad j \in \mathcal{B}$$

$$\beta^l y_j \leq \beta_j \leq \beta^u y_j \qquad j \in \mathcal{B}$$

$$y_j = \{0, 1\} \qquad j \in \mathcal{B}$$

BASIS FUNCTION SELECTION

Find the model with the least error

$$\min \quad SE = \sum_{i=1}^{N} \left| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right|$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$$

$$-U(1-y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1-y_j) \qquad j \in \mathcal{B}$$

$$\beta^l y_j \leq \beta_j \leq \beta^u y_j \qquad j \in \mathcal{B}$$

$$y_j = \{0, 1\} \qquad j \in \mathcal{B}$$

BASIS FUNCTION SELECTION

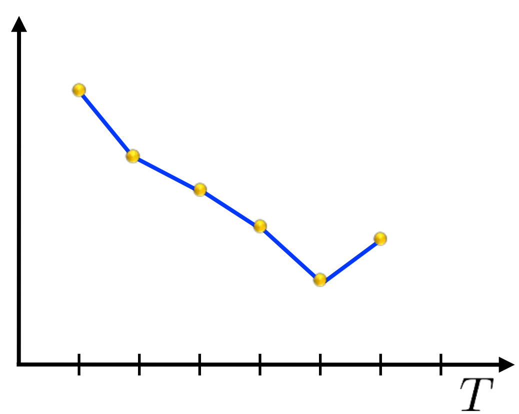
$$\min \quad SE = \sum_{i=1}^{N} \left| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right|$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$$

$$-U(1-y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1-y_j) \qquad j \in \mathcal{B}$$

$$\beta^l y_j \leq \beta_j \leq \beta^u y_j$$

$$y_j = \{0, 1\}$$



We will solve this model for increasing T until we determine a model

BASIS FUNCTION SELECTION

$$\min \quad SE = \sum_{i=1}^{N} \left| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right|$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$$

$$-U(1-y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1-y_j) \qquad j \in \mathcal{B}$$

$$\beta^l y_j \leq \beta_j \leq \beta^u y_j \qquad j \in \mathcal{B}$$

$$y_j = \{0, 1\} \qquad j \in \mathcal{B}$$

$$\left(\begin{array}{c} \mathbf{y_j = 1} \\ \textbf{Basis function used in the model} \\ \beta_j \text{ is chosen to satisfy a least squares regression} \\ \textit{(assumes loose bounds on } \beta_j\textit{)} \end{array} \right) \vee \left(\begin{array}{c} \mathbf{y_j = 0} \\ \textbf{Basis function NOT used in the model} \\ \beta_j = 0 \end{array} \right)$$

BASIS FUNCTION SELECTION

$$\min \quad SE = \sum_{i=1}^{N} \left| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right|$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$$

$$-U(1-y_j) \le \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \le U(1-y_j) \qquad j \in \mathcal{B}$$

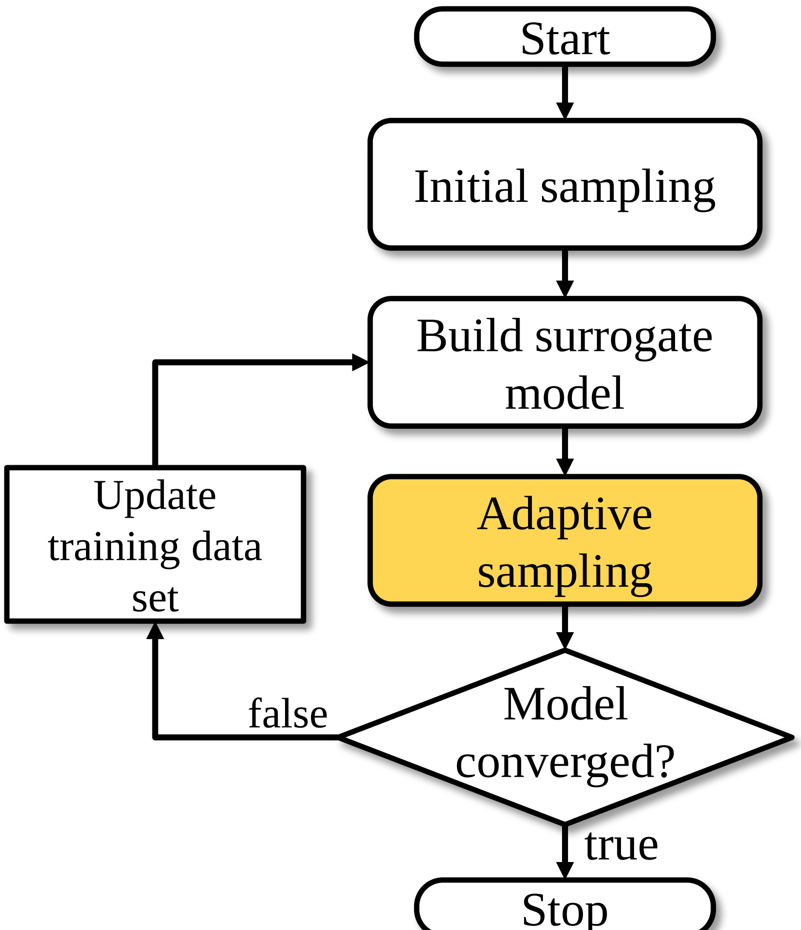
$$\beta^l y_j \le \beta_j \le \beta^u y_j \qquad j \in \mathcal{B}$$

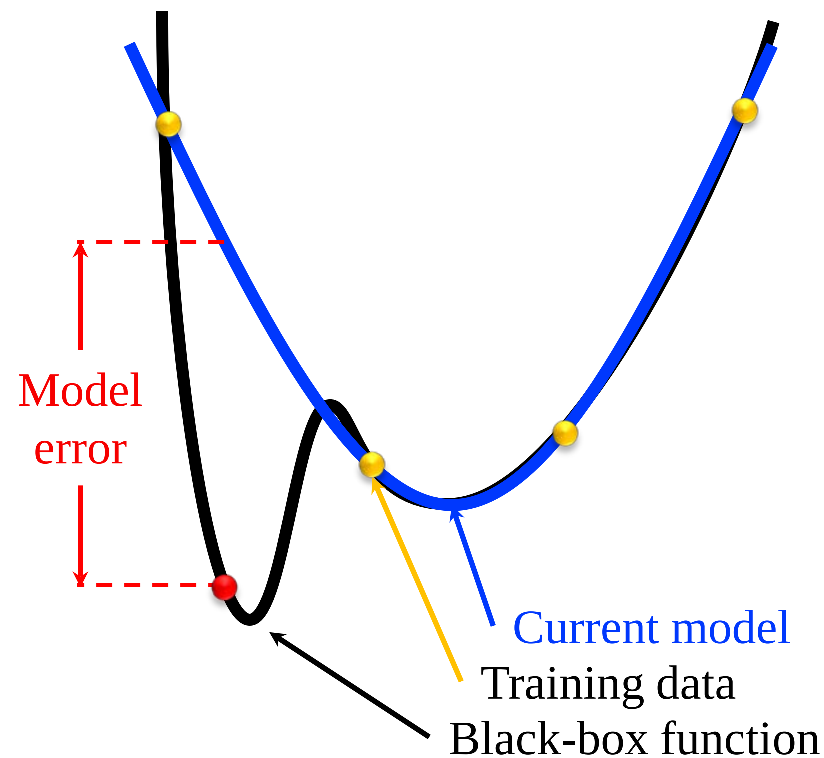
$$y_j = \{0,1\} \qquad j \in \mathcal{B}$$

$\left(\begin{array}{c} \mathbf{y_j = 1} \\ \textbf{Basis function used in the model} \\ \beta_j \text{ is chosen to satisfy a least squares regression} \\ \textit{(assumes loose bounds on } \beta_j\textit{)} \end{array} \right) \vee \left(\begin{array}{c} \mathbf{y_j = 0} \\ \textbf{Basis function NOT used in the model} \\ \beta_j = 0 \end{array} \right)$

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**

Surrogate model

$$\max_{x} \left(\frac{z(x) - \hat{z}(x)}{z(x)} \right)^2$$

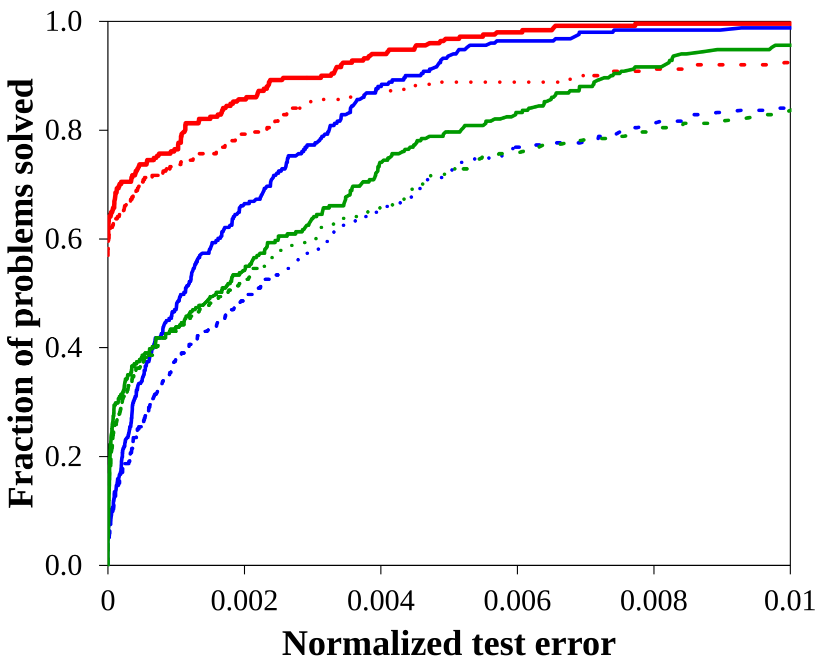
  – **Optimized using a black-box or derivative-free solver (SNOBFIT)** [Huyer and Neumaier, 08]

COMPUTATIONAL TESTING

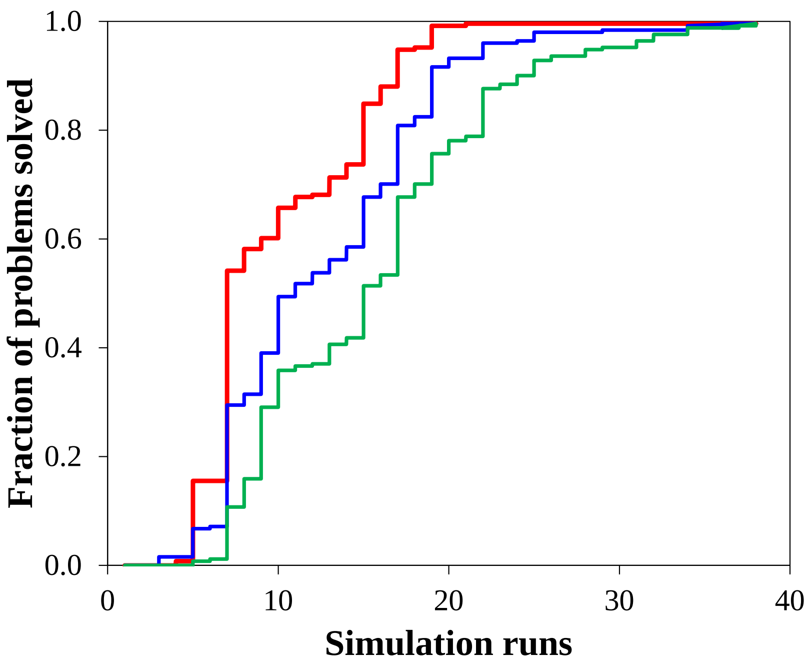
- **Modeling methods compared**
  - **MIP – Proposed methodology**
  - **EBS – Exhaustive best subset method**
    - ***Note: due to high CPU times this was only tested on smaller problems***
  - **LASSO – The lasso regularization**
  - **OLR – Ordinary least-squares regression**
- **Sampling methods compared**
  - **DFO – Proposed error maximization technique**
  - **SLH – Single Latin hypercube (no feedback)**

COMPUTATIONAL EXPERIMENTS

Model accuracy



Modeling efficiency

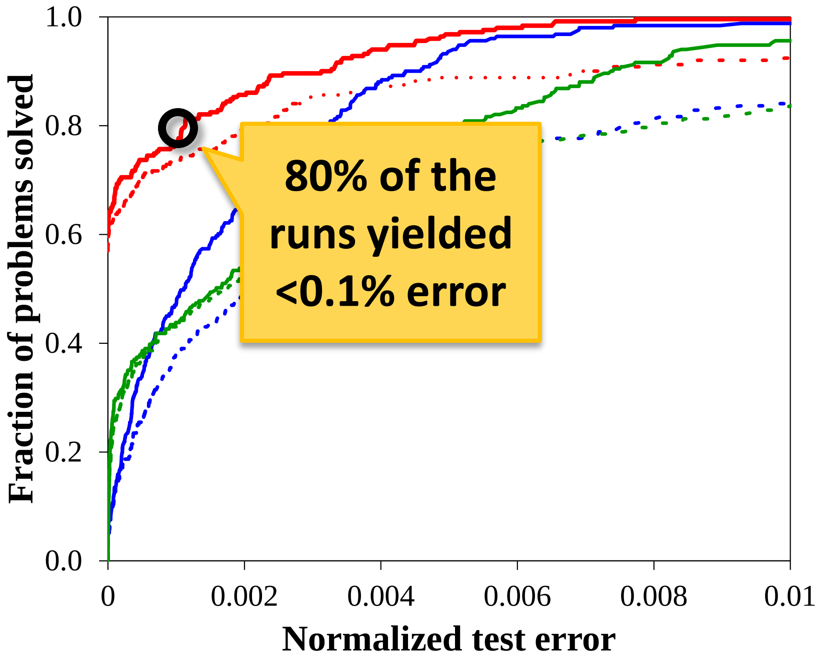


Modeling methods		
Our method	LASSO	Least squares

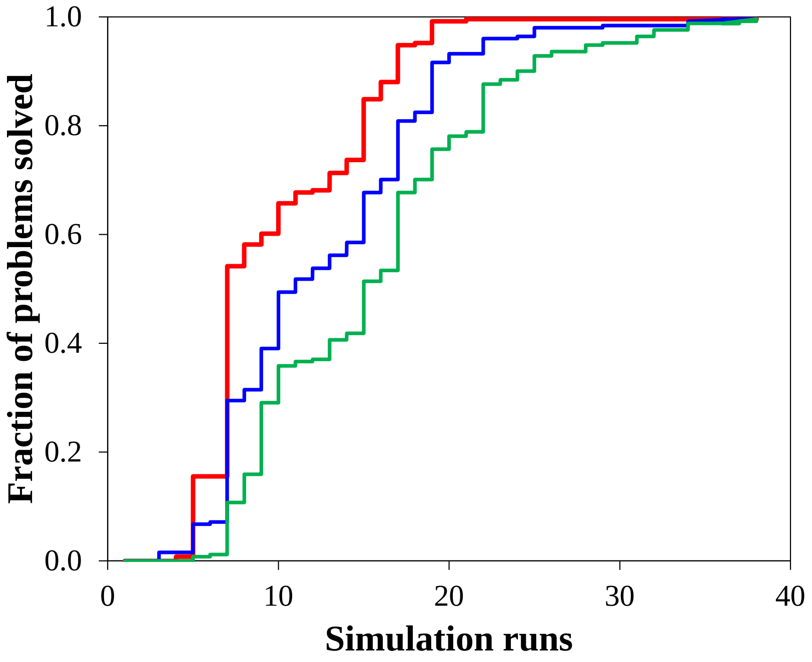
Sampling methods	
Error maximization	Single Latin hypercube

COMPUTATIONAL EXPERIMENTS

Model accuracy



Modeling efficiency



Modeling methods		
Our method	LASSO	Least squares

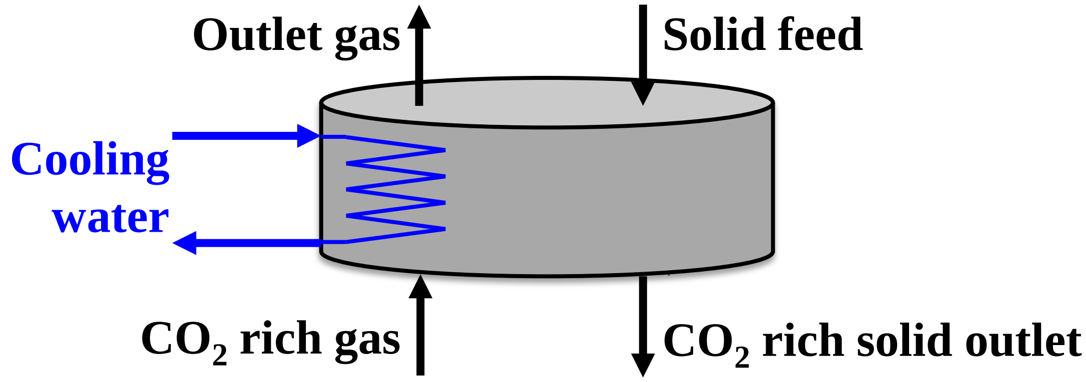
Sampling methods	
Error maximization	Single Latin hypercube

MODEL COMPLEXITY

No. inputs	No. true terms	MIP/ DFO	MIP/ SLH	EBS/ DFO	EBS/ SLH	LASSO/ DFO	LASSO/ SLH	OLR/ DFO	OLR/ SLH
2	2	2	[2, 2]	2	2	[6, 8]	[6, 11]	[12, 15]	[12, 15]
2	3	3	3	3	3	[5, 12]	[5, 10]	[12, 14]	[12, 14]
2	4	[3, 4]	[3, 4]	[3, 4]	[3, 4]	[8, 11]	[8, 10]	[11, 12]	[11, 12]
2	5	[2, 4]	[2, 4]	[2, 5]	[2, 5]	[3, 12]	[4, 11]	[10, 16]	[10, 16]
2	6	[5, 6]	[6, 6]	[5, 6]	[6, 6]	[7, 10]	[6, 7]	[11, 13]	[11, 13]
2	7	[4, 6]	[4, 6]	[4, 7]	[4, 7]	[7, 11]	[6, 12]	[8, 13]	[8, 13]
2	8	[4, 5]	[5, 6]	[4, 5]	[5, 6]	[6, 8]	[6, 9]	[10, 15]	[10, 15]
2	9	[4, 6]	[4, 6]	NA	NA	[6, 14]	[7, 12]	[10, 17]	[10, 17]
2	10	[4, 8]	[4, 8]	NA	NA	[5, 14]	[7, 14]	[10, 14]	[10, 14]
3	2	[2, 3]	[2, 3]	NA	NA	[6, 12]	[7, 13]	[27, 29]	[27, 29]
3	3	[3, 3]	[3, 3]	NA	NA	[8, 16]	[7, 15]	[19, 22]	[19, 22]
3	4	4	[3, 4]	NA	NA	[10, 13]	[9, 10]	[16, 21]	[16, 21]
3	5	5	5	NA	NA	[11, 17]	[9, 15]	[15, 23]	[15, 23]
3	6	[5, 6]	[6, 6]	NA	NA	[9, 18]	[10, 13]	[15, 26]	[15, 26]
3	7	7	[7, 8]	NA	NA	[10, 22]	[10, 22]	22	22

BUBBLING FLUIDIZED BED

Bubbling fluidized bed adsorber diagram



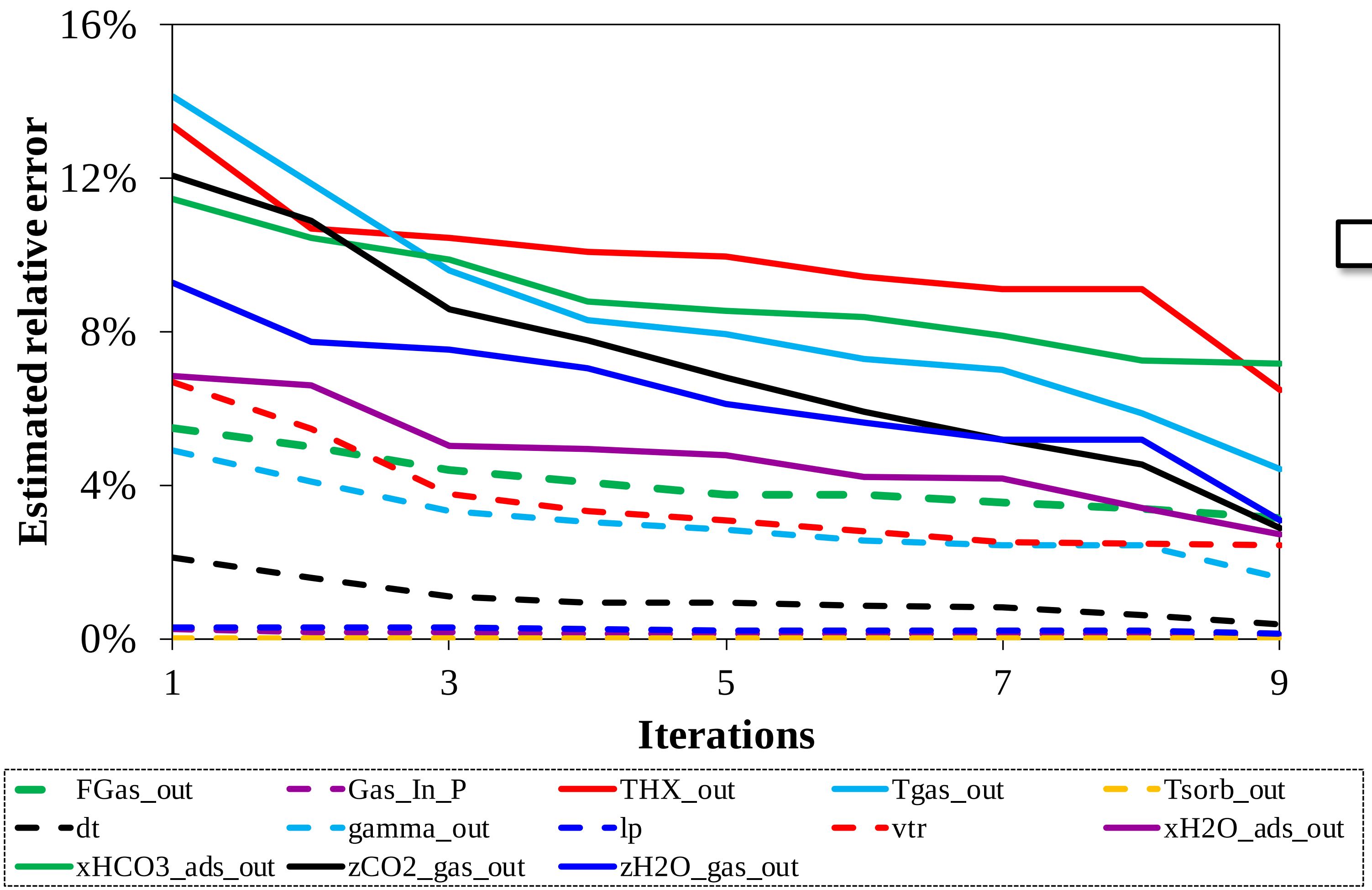
- **Model inputs (14 total)**
  - **Geometry (3)**
  - **Operating conditions (4)**
  - **Gas mole fractions (2)**
  - **Solid compositions (2)**
  - **Flow rates (4)**

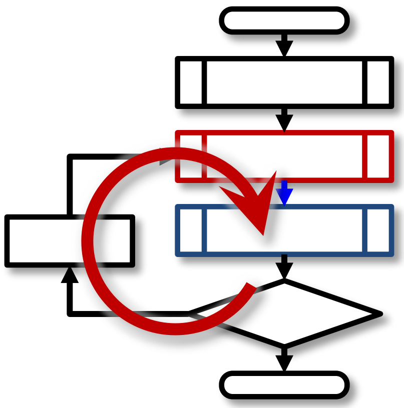
Model created by Andrew Lee at the National Energy and Technology Laboratory

- **Model outputs (13 total)**
  - **Geometry required (2)**
  - **Operating condition required (1)**
  - **Gas mole fractions (2)**
  - **Solid compositions (2)**
  - **Flow rates (2)**
  - **Outlet temperatures (3)**
  - **Design constraint (1)**

ADAPTIVE SAMPLING

Progression of mean error through the algorithm

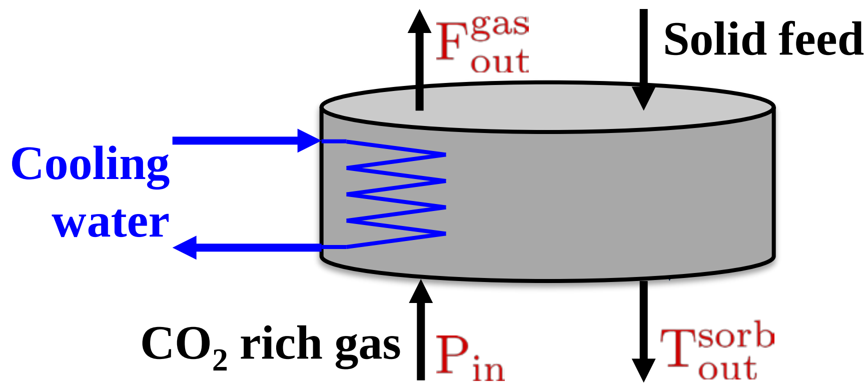




Initial data set: 137 pts

Final data set: 261

EXAMPLE MODELS



$$P_{in} = 1.0\,P_{out} + 0.0231\,L_b - 0.0187\,\ln(0.167\,L_b) - 0.00626\,\ln(0.667\,v_{gi}) - \frac{51.1\,xHCO3^{ads}_{in}}{F^{gas}_{in}}$$

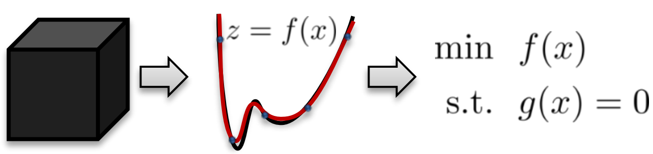
$$T^{sorb}_{out} = 1.0\,T^{gas}_{in} - \frac{\left(1.77\cdot 10^{-10}\right)\,NX^2}{\gamma^2} - \frac{3.46}{NX\,T^{gas}_{in}\,T^{sorb}_{in}} + \frac{1.17\cdot 10^4}{F^{sorb}\,NX\,xH2O^{ads}_{in}}$$

$$F^{gas}_{out} = 0.797\,F^{gas}_{in} - \frac{9.75\,T^{sorb}_{in}}{\gamma} - 0.77\,F^{gas}_{in}\,xCO2^{gas}_{in} + 0.00465\,F^{gas}_{in}\,T^{sorb}_{in} - 0.0181\,F^{gas}_{in}\,T^{sorb}_{in}\,xH2O^{gas}_{in}$$

CONCLUSIONS

- **The algorithm we developed is able to model black-box functions for use in optimization such that the models are**
  - ✓ **Accurate**
  - ✓ **Tractable in an optimization framework (low-complexity models)**
  - ✓ **Generated from a minimal number of function evaluations**

- **Surrogate models can then be incorporated within an optimization framework with complex objective functions and additional constraints**



- **ALAMO site: archimedes.cheme.cmu.edu/?q=alamo**