ORIGINAL ARTICLE

Model reduction based global optimisation for large-scale steady state nonlinear systems

Min Tao | Panagiotis Petsagkourakis | Jie Li | Constantinos Theodoropoulos

Department of Chemical Engineering and Analytical Science, The University of Manchester, M13 9PL, UK

[1]Department of Chemical Engineering and Analytical Science, The University of Manchester, M13 9PL, UK

Correspondence
Constantinos Theodoropoulos, Department of Chemical Engineering and Analytical Science, The University of Manchester, M13 9PL, UK
Email: k.theodoropoulos@manchester.ac.uk

Funding information
The University of Manchester and China Scholarship Council joint scholarship (file no. 201706250031)

Many engineering processes can be accurately modeled using partial differential equations (PDEs), but high dimensionality and non-convexity of the resulting systems pose limitations on their efficient optimisation. In this work, a model reduction methodology combining principal component analysis (PCA) and artificial neural networks (ANNs) is employed to construct a reduced surrogate model, which is then utilized by advanced deterministic global optimisation algorithms to compute global optimal solutions with theoretical guarantees. However, the optimisation framework is still time-consuming due to the high non-convexity of the activation functions inside the reduced ANN structure. To further enhance the capability of our optimisation framework, two alternative strategies have been proposed. The first one is a piecewise-affine reformulation while the second one is based on deep rectifier neural networks with ReLU activation function. The performance of the two improved frameworks is demonstrated through two illustrative case studies.

KEYWORDS
Model reduction, Distributed parameter systems, Piecewise affine reformulation, Deep rectifier neural networks, Data-driven methodology, Global optimisation

1 | INTRODUCTION

Partial differential equation (PDE)-based process models, also termed distributed-parameter systems, have wide applicability in industrial engineering areas [1], such as chemical [2], biochemical [3], and mechanical engineering [4] and aerodynamics [5]. However, complex PDEs are inherently high-dimensional and non-convex, including multiple local optima, hence resulting in intensive computational costs when the computation of global optima is sought. Moreover, most of the generic commercial PDE simulators [6, 7] are essentially black-box and offer no optimisation options. Even if complex model codes are accessible in open-source software (e.g. [8]), the cost of direct optimisation is often unacceptable. To date, performing optimisation tasks efficiently for large-scale complex systems, is still a challenge in engineering design.

A promising way to deal with high dimensionality is to use projective model order reduction methods, which reduce the complexity of detailed models but preserve their main input-output features [9]. The popular principal component analysis (PCA) strategy, an efficient dimensionality reduction technique in data science [10], also termed as Karhunen-Loeve decomposition or proper orthogonal decomposition (POD), is usually combined with projection and/or surrogate model approaches to construct reduced models. POD together with Galerkin projection is capable of producing high-fidelity low-dimensional models for optimisation tasks [11]. Similarly, the combination of POD and ANN can construct reduced surrogate models for black-box large-scale dynamic systems, resulting in efficient optimisation and control strategies [12]. In addition, PCA and kriging models have been utilized to efficiently replace complex process models [13].

Furthermore, *equation-free* methodologies offer another effective model reduction approach for large-scale black-box systems, for optimisation and control purposes. Exploiting the dominant eigendirections of the outputs of complex black-box system models, or direct historical system data, low-dimensional reduced Jacobian and Hessian matrixes can be computed. An equation-free based reduced SQP method was proposed exploiting the computation of low-dimensional Jacobians and Hessians, to accelerate the optimisation procedure for large-scale steady state nonlinear systems [14]. An aggregation function was subsequently applied to address general nonlinear inequality constraints, extending the scope and capability of equation-free reduced SQP methods [15]. Furthermore, equation-free based dynamic optimisation and control methods have also been constructed [16, 17]. An extensive discussion about model reduction based optimisation methodologies can be found in [18].

To address non-convexity in complex nonlinear optimization problems, both stochastic and deterministic algorithms can be utilized. Stochastic search methods, such as simulated annealing [19] and genetic algorithms [20], can globally explore the feasible solution space avoiding local optima. However, such stochastic search algorithms are slow for large-scale problems and offer no theoretical guarantees on the global optimality of the computed solutions. Deterministic global optimization methods are capable of computing global optima utilizing branch-and-bound techniques [21], but they are often computationally intensive for large-scale systems due to the need for multiple evaluations of the lower bounds of the optimization problems. The aim of this work is to construct an efficient model reduction-based deterministic global optimization framework for large-scale steady-state input/output (black-box) systems. Often a single model reduction technique cannot easily deal with the complexities of large-scale nonlinear systems. For example, although optimal principal component regressions (PCRs)[22] are popular to deal with high dimensional input-output data, the linear or low-complex models are not accurate enough to replace high nonlinear complex system models. POD on the other hand, is a very powerful method, but projecting the original system onto the global POD modes is not always easy and requires full knowledge of the full-scale system model. Meanwhile, ANN models can capture highly nonlinear behaviours but usually require large-scale ANN structures (increasing number of neurons and layers) due to the high dimensionality of the original systems. Combining model reduction techniques, e.g. principal component

analysis (PCA) with artificial neural networks (ANNs)[23], can produce accurate reduced surrogate models. Then such reduced ANN models could be explicitly utilized by global general-purpose optimization solvers.

Nevertheless, performing global optimization tasks with general ANN models is still time consuming (even for reduced ANNs), hence most existing research focuses on local optimization and/or small-scale problems. Surrogate ANN models have been used to replace superstructure process models and were optimized locally [24, 25]. Small-scale ANN models (1 hidden layer, 3 neurons) were constructed and optimized globally by the advanced global solver BARON [26]. Larger ANN models are more expensive as high non-convexity often requires the repeated use of branch-and-bound algorithms. A reduced space-based global optimization method, recently proposed by Schweidtmann and Mitsos [27], projected the iteration space of non-convex variables onto the subspace of dependent variables, resulting in small-size sub-problems and, consequently, in significant computational savings.

In this work, two strategies are adopted to construct efficient reduced models in the PCA-ANN global optimization framework. The first is a piecewise affine (PWA) reformulation technique while the second is the use of a deep rectifier neural network. It should be noted that this work extends previous preliminary findings of the authors [28].

The rest of the paper is organized as follows. In Section 2, the basic PCA-ANN global optimization framework is proposed and the detailed theoretical basis and implementation are provided. In Section 3, sampling and data collection is briefly introduced. In Section 4, the PWA-based reformulation is outlined and illustrated with an example. In Section 4, the deep rectifier ANN-based improvement is employed in the optimization framework and validated using a large-scale combustion case study. In Section 5, conclusions and further applications are discussed.

2 | PROBLEM FORMULATION

In this work, a model reduction-based optimization framework is presented to deal with large-scale nonlinear steady-state systems focusing on the optimization of spatially distributed processes, described by sets of steady-state dissipative PDEs:

$$\frac{\partial \boldsymbol{y}}{\partial t} = D\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}, \boldsymbol{d}\} + R(\boldsymbol{d}, \boldsymbol{y}) \tag{1}$$

Here $t \in \mathbb{R}$ denotes time, $x \in \mathbb{R}^{N_x}$, N_x the spatial dimensions, N_x=1,2, or 3. $D \in \mathbb{R}$ is the dissipative spatial differential operator, $\boldsymbol{d} \in \mathbb{R}^{N_d}$ the parameter variables and $\boldsymbol{y} \in \mathbb{R}^{N_y}$ a set of state variables, $R(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_y}$ are the nonlinear terms. Considering steady state analysis and assuming that $\boldsymbol{y}(t, x) \rightarrow \boldsymbol{y}(x)$, and $\partial \boldsymbol{y}/\partial t = 0$, the above equations become:

$$0 = D\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}, \boldsymbol{d}\} + R(\boldsymbol{d}, \boldsymbol{y}) \tag{2}$$

Therefore, the general optimization problems for steady state PDE-based systems can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{d}} \quad & G(\boldsymbol{d}, \boldsymbol{y}) \\ s.t. 0 = & D\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}, \boldsymbol{d}\} + R(\boldsymbol{d}, \boldsymbol{y}) \\ & A\{\frac{\partial \boldsymbol{y}}{\partial x}, \frac{\partial^2 \boldsymbol{y}}{\partial x^2}, ..., \frac{\partial^n \boldsymbol{y}}{\partial x^n}\}\Big|_{x=\Omega} = h_{bds}(\boldsymbol{d}, \boldsymbol{y}) \\ & \boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{y}) \leq 0 \end{aligned} \tag{3}$$

Where $G(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}$ is the objective function. The equality constraints are the system PDEs with corresponding boundary conditions. $h_{bds}(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_y}$ are the right hand sides of the boundary conditions, A is the operator of the boundary condition equations, Ω are the boundaries and $\boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{y}) : \mathbb{R}^{N_d} \times \mathbb{R}^{N_y} \rightarrow \mathbb{R}^{N_y}$ denote other general constraints, e.g. bounds, for state variables $\boldsymbol{y}$ and design parameter variables $\boldsymbol{d}$.

If we use black-box inputs-outputs to replace the explicit system equations in the above formulation, then the problem can be transformed into the following:

$$\begin{aligned} &\min_{\boldsymbol{d}} \quad G(\boldsymbol{d}, \boldsymbol{y'}) \\ &s.t. \boldsymbol{y'} = F(\boldsymbol{d}) \\ &\boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{y'}) \leq 0 \end{aligned} \tag{4}$$

where $\boldsymbol{y'}$ are the spatially discrete state variables and F the black-box nonlinear operator.

In general, the unavailability of system equations inside commercial software prohibits the use of direct model-based optimization techniques. Even in the case that large-scale system equations are available, the optimization problem can not be efficiently handled by global optimization algorithms [29]. In this work, this barrier is overcome by employing a double model reduction process through a combination of PCA and ANN and reformulation techniques to generate an accurate reduced model, which is then utilized by a general purpose global optimization solver. In the following sections, we are discussing the basic components of our PCA-ANN-global optimization methodology.

2.1 | Sampling and data collection

To build accurate surrogate models, suitable sampling methods are needed to collect highly representative samples. Inefficient sampling strategies, including too few samples and/or unrepresentative sampling, would result in inaccurate reduced models, resulting in inaccurate optimal solutions. While provably representative sampling is still an open problem, there are several popular sampling techniques such as Latin Hypercube (LHC)[30], Hammersley sequences [31], number-theoretic methods [32] and D-optimal designs [33]. A detailed discussion about sampling methods can be found in [34]. Among these sampling techniques, LHC can produce samples in the whole design space and maximize the difference among the generated samples [35]. Specifically, the sample domain is divided into many sub-intervals, where sample points are generated randomly in order to represent the whole sub-domain. Although LHC may be more time-consuming for high-dimensional problems compared with other sampling methods, it is more likely to generate "enough" representative samples for a general complex systems. Moreover, the sampling process takes place offline and does not affect the computational efficiency of the online optimization computations. Hence LHS is our sampling method of choice here.

We collect samples across the space of design parameters $\boldsymbol{d}$ and corresponding input-output data sets ($\boldsymbol{D} \in \mathbb{R}^{N_d \times N}$, $\boldsymbol{Y} \in \mathbb{R}^{m \times N}$) , where $m \in \mathbb{N}$ is the number of discrete interval points, which for distributed parameter systems tends to be be a large number, and $N \in \mathbb{N}$ is the number of samples. The obtained data sets ($\boldsymbol{D}$, $\boldsymbol{Y}$) are then used to construct accurate reduced surrogate models through the combination of PCA and ANN.

2.2 | Principal Component Analysis (PCA)

Due to high dimensionality of spatially discrete output data $\boldsymbol{Y}$, directly constructing surrogate ANN models would result in large ANN structures. Here, the popular PCA method. is first employed to build a reduced model from output

data $\boldsymbol{Y}$.

A sampling method (here LHC) is firstly employed to construct a data ensemble $\boldsymbol{Y}$ over a finite spatial interval $\Omega' \in \mathbb{R}$. PCA then calclulates a "small" set of principal components (PCs) $P = (p_1, p_2, ..., p_k)$, $k \in \mathbb{N}$ being the number of PCs, by projecting the data sample **Y** onto the subspace of the, k, principal components $\mathbb{P}$.

$$\boldsymbol{U} = \boldsymbol{PY} \tag{5}$$

Here $\boldsymbol{U} \in \mathbb{R}^{k \times N}$ is the projection of the original data $\boldsymbol{Y}$ onto $\mathbb{P}$ and $\boldsymbol{P} \in \mathbb{R}^{k \times m}$ is the orthogonal projector. In the PCA method the matrix $\mathbb{P}$ is constructed through the covariance matrix, $\boldsymbol{C_y} \in \mathbb{R}^{m \times m}$ of the output data $\boldsymbol{Y}$:

$$\boldsymbol{C_y} = \frac{1}{m-1} \boldsymbol{Y Y^T} \tag{6}$$

Here we seek to minimise covariance between data and maximise variance i.e. minimise the off-diagonal elements of $\boldsymbol{C_y}$, while maximising its diagonal elements. This is equivalent to performing singular value decomposition (SVD) on $\boldsymbol{C_y}$:

$$\boldsymbol{C_y} = \boldsymbol{Z^T Z} = (\frac{1}{\sqrt{m-1}} \boldsymbol{Y}^T)^T (\frac{1}{\sqrt{m-1}} \boldsymbol{Y}^T) = \boldsymbol{V D V^T} \tag{7}$$

where $D \in \mathbb{R}^{m \times m}$ is a diagonal matrix whose diagonal elements are the eigenvalues of $Z^T Z$ and V is the orthogonal matrix whose columns are the eigenvectors of $Z^T Z$, which as can be easily shown are equivalent to the principal components of $\boldsymbol{Y}$. In fact we can keep the first k PCs corresponding to the k dominant eigenvalues of $\boldsymbol{C_y}$, where usually $k << m$, hence $\boldsymbol{V} \in \mathbb{R}^{m \times k}$ and D now contains only the k most dominant eigenvalues of the system, $D \in \mathbb{R}^{k \times k}$. We can then set $\boldsymbol{P} = \boldsymbol{V}^T$ and perform data reduction through the projection in Eq. 5. The original data sample, $\boldsymbol{Y}$ can be reconstructed from the projected data:

$$\boldsymbol{Y} = \boldsymbol{P^T U} \tag{8}$$

More details about the theory and application of PCA can be found in [36, 37, 38, 39, 40].

2.3 | Artificial Neural Networks (ANNs)

We employ ANNs are on the reduced models (Eq.5) from the PCA step. ANN-based models are chosen due to both successful practices and proven theoretical supports that a *shallow* feed-forward neural network with one single layer is sufficient to represent any smooth function [41]. Furthermore, advanced optimization algorithms have been developed to handle the manipulated variables for ANN structures, such as Levenberg-Marquardt backpropagation [42] and Bayesian regularization backpropagation [43]. Fig.1 shows a conventional feed forward neural network with a hyperbolic tangent activation function $tanh(\cdot)$.

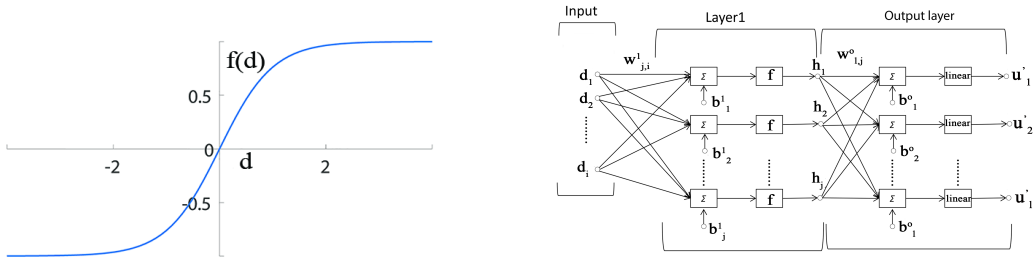


FIGURE 1 Feed-forward neural network with hyperbolic tangent activation function

Shallow ANNs, as the one displayed in Fig.1, are implemented in our basic PCA-ANN global optimization framework. The feed-forward ANN contains three main components: The input layer, the hidden layer (only one in a shallow ANN) and the output layer, which sequentially perform transformations on the input variables. The input variables, $\boldsymbol{d} = (d_1, d_2, ..., d_{N_d})$, are first linearly transformed and then non-linearly activated through the hidden layer, and further forced by linear transformation and sequential activation in the output layer, to finally formulate the output variables $\boldsymbol{u'} = (u'_1, u'_2, ..., u'_k), \in \mathbb{R}^k$. The mathematical description is given in eq. (9) :

$$\begin{aligned} h_j &= f\Big(\sum_{i=1}^{N_d} w^1_{j,i} d_i + b^1_j\Big), \quad \forall j \in \{1, 2, ..., n\} \\ u'_l &= \sum_{j=1}^{n} w^o_{l,j} h_j + b^o_l, \qquad \forall l \in \{1, 2, ..., k\} \end{aligned} \tag{9}$$

Here $h_j \in \mathbb{R}$ is the output value from the hidden layer with $n \in \mathbb{N}$ neurons, $j = 1, \ldots, n$ and $f \in \mathbb{R}$ is the activation function. Each neuron j contains two parameters: weights $w^1_{j,i} \in \mathbb{R}$ and biases $b^1_j \in \mathbb{R}$ which perform linear transformations. Similarly, $u'_l \in \mathbb{R}$ is the final value from the output layer with k neurons, $l = 1, \ldots, k$, including weights $w^o_{l,j} \in \mathbb{R}$ and biases $b^o_l \in \mathbb{R}$. Three activation functions, the sigmoidal, the hyperbolic tangent and the linear function, are widely used in neural networks. In this work, the hyperbolic tangent function f was utilized to convert the output value into the range [0,1] in the hidden layer while the linear function was applied in the output layer. The configured feed-forward neural network was subsequently trained through the back-propagation algorithm using the reduced low-dimensional data sets $(\boldsymbol{D}, \boldsymbol{U})$ from the PCA step. The detailed sampling and training methods used are given below, where illustrative examples are discussed.

2.4 | PCA-ANN global optimization framework

To cope with the non-convexity of highly non-linear systems, deterministic optimization methods are considered for the reduced surrogate model from the PCA-ANN reduction. The black-box global optimization problem can be transformed

into the general explicit NLP optimization problems as follows combining Eqs.(4,5,9):

$$\begin{aligned} \min_{\boldsymbol{d}=[d_1,d_2,\ldots,d_{N_d}]} \quad & G(\boldsymbol{d},\boldsymbol{u'}) \\ s.t. h_j &= f(\sum_{i=1}^{N_d} w_{j,i}^1 d_i + b_j^1), \quad \forall j \in \{1,2,\ldots,n\} \\ u'_l &= \sum_{j=1}^{n} w_{l,j}^o h_j + b_l^o, \quad \forall l \in \{1,2,\ldots,k\} \\ \boldsymbol{u'} &= (u'_1, u'_2, \ldots, u'_k), \\ \boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{P^T u'}) &\leq 0 \end{aligned} \tag{10}$$

In this work, $g_{cons}(\boldsymbol{d}, \boldsymbol{P^T u'})$ are the box bound constraints for design variables $\boldsymbol{d}$ and discretised state variables $\boldsymbol{y'}$. The ANN-based nonlinear objection function G, can be reformulated into constraints. The non-convexity of the optimization problems lies on the constraints $h_j = f(\cdot)$ due to the highly non-convex activation function, i.e. the hyperbolic tangent function $tanh(\cdot)$ in the feed-forward ANN structure. General-purpose global optimization commercial software, including ANTIGONE [44], BARON [26] and SCIP [45], are efficient tools for the above problems due to the advanced bound tightening and branching techniques. Nevertheless, these general global solvers can not identify $tanh(\cdot)$ formulation directly, as high performance algorithms need the explicit model equations. Therefore the explicit algebraic form $tanh(z) = (e^z - 1)/(e^z + 1)$ is required [46]. The basic formulation is further transformed into $tanh(z) = -2/(e^z + 1) + 1$ in order to produce a tighter under-estimator for the global solver [27]. The flow chart of the basic PCA-ANN global optimization framework is shown in Fig.2. It will be first tested through a small peaks function with a known global optimum solution, and then applied to large-scale nonlinear systems. It should be noted that the PCA step is not necessary for the small peaks function example with the low-dimensional input-output variables, but is utilized for the two large-scale case studies, the tubular reactor and the combustion process, where also the two improvements of the basic framework constructed are demonstrated.

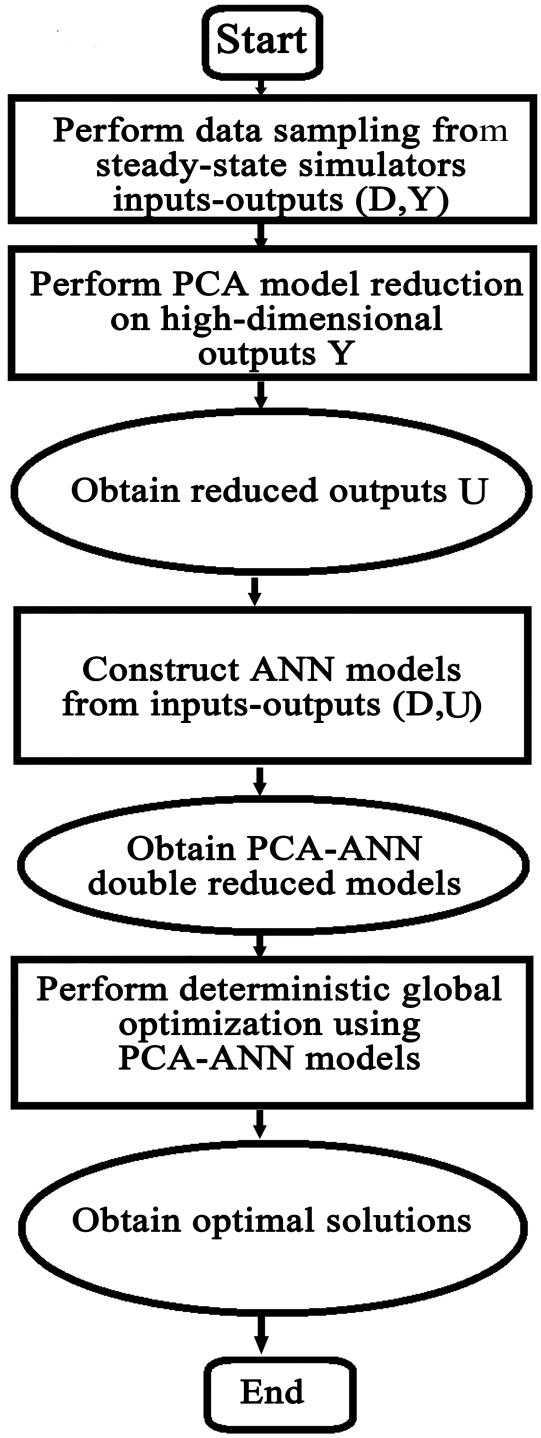


FIGURE 2 Flow chart of the basic PCA-ANN global optimization framework

2.5 | Illustrative example

A two-dimensional multi-modal peaks function is chosen with the following mathematical formulation:

$$g_{peaks} = 3(1-x)^2 exp(-x^2-(y+1)^2) - 10(\frac{x}{5} - x^3 - y^5)exp(-x^2-y^2) - \frac{1}{3}exp(-(x+1)^2 - y^2) \qquad x, y \in [-3, 3] \tag{11}$$

We treated this peaks function as a black-box input/output system and we employed the LHC sampling method to collect snapshots and1600 samples were collected. to avoid over- and under-fitting, the defined domain is randomly divided into a training, a validation and a test set with respective size ratios of 0.7 : 0.15 : 0.15. The MATLAB Neural Network Toolbox was utilized to fit the weights and biases by minimizing the mean squared error (MSE) between the ANN model and the training set using Levenberg-Marquardt algorithm and the early stopping procedure. To obtain a suitable number of neurons in the hidden layer, the training process is repeated using an increasing number of neurons until the MSE for all three sets becomes less than a pre-defined tolerance, here $1x10^{-4}$. Finally, a neural network structure with 52 neurons is chosen, with MSE of 4.78×10^{-5}, 6.20×10^{-5} and 4.45×10^{-5} on the training, validation and test sets respectively. Fig.3 shows the comparison between the original full model (FOM) and reduced ANN model (ROM) with 961 grid points (31 grid points in each direction). In this domain, multiple local optimal solutions exist for the original FOM. The known unique global optimum lies on $(0.228, -1.626)$ with the corresponding function value being -6.551. As it can be seen, the ROM approximates the FOM with excellent accuracy. Here, the general purpose optimization solver BARON 17.4.1 was used to perform global optimization of the ROM, with both absolute and relative tolerances being 0.002, and a time limit of 36000 seconds (10 hours). All runs were performed on a Desktop (Intel® Core(TM) CPU 3.3 GHz, 8 GB memory, 564-bit operating system) running Windows 7.

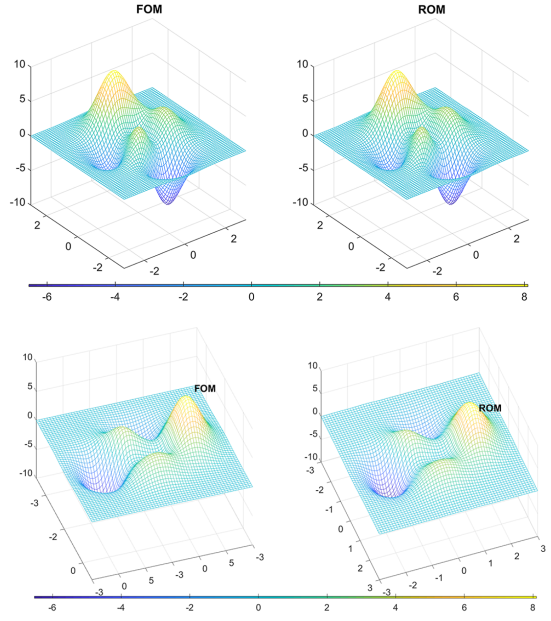


FIGURE 3 Comparison between the full model (FOM) and reduced model (ROM)

The optimal solution (0.228, -1.625) with value -6.555 was computed within 30294.26 CPU seconds. This is close to the known global solution of the FOM. Nevertheless, despite the relative simplicity of the orginal problem, the computational cost is still appreciable mainly due to the high non-convex activation function in the surrogate ANN model. This drawback drives our further improvements for the basic optimization framework. The first one being the piece-wise linear approximation (PWA) of the nonlinear activation funcction.

3 | PIECE-WISE AFFINE BASED FORMULATION

In this section, a piecewise affine (PWA) reformulation is introduced to deal with the non-convex hyperbolic tangent activation function in the reduced surrogate ANN model. Previous research has suggested the PWA technique for the ANN model [47], which has been verified to be efficient [48]. Although these studies provided some computational results, further detailed implement schemes and analysis have not been reported. In this work, the PWA reformulation was utilized to approximate the highly non-convex NLP problem with a MILP problem. The global optimization algorithms for both NLP and MILP problems are based on the branch and bound framework. However, the branching step is performed on continuous variables for the NLP problems and on auxiliary binary variables for the MILP problems through the use of CPLEX 12.7.1. An adaptive procedure to construct PWA models is presented below.

3.1 | Adaptive procedure

The hyperbolic tangent activation function $f(z) = tanh(z)$ is an odd function with central symmetry, which is concave on $(0, +\infty]$ and convex on $[-\infty, 0)$. Therefore the PWA approximation on $[-\infty, 0)$ can be directly computed from the PWA formulation on $(0, +\infty]$. Within the range of $(0, +\infty]$, $tanh(z)$ function first increases and then tends to level off with a slight increase towards the limit value of 1. The adaptive PWA procedure starts from the interval $(0, +\infty]$ and two points, the point of symmetry and one point close to the maximum value (equal to 1). Then a new point is chosen between the two original points so that the error E_{ero} between $f(z)$ and its PWA approximation $f_{PWA}(z)$ (currently consisting of two intervals) is minimised.

$$E_{ero} = \int abs(f(z) - f_{PWA}(z))\mathrm{d}z, \tag{12}$$

Then the segment with the largest error is chosen and a new point is added within to minimise E_{ero} in this segment. This procedure continues iteratively until the error in eq. 12 becomes less than a pre-defined tolerance. Finally the points chosen for the $(0, +\infty]$ interval are mirrored to the $[-\infty, 0)$ interval. The iteration procedure efficiently produces a tight PWA representation of the $tanh(z)$ function. Fig.4 shows the adaptive process, narrowing the interval sizes and reducing the error (red shade) between multiple linear models and $tanh(z)$.

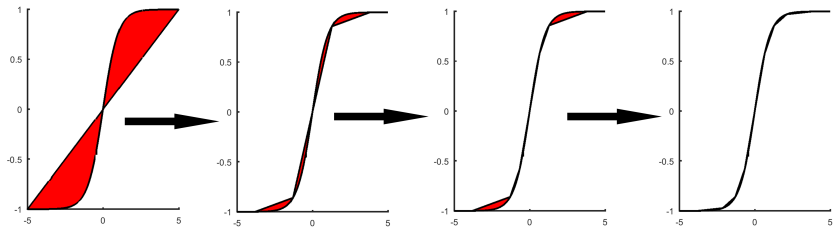


FIGURE 4 Adaptive PWA procedure for the hyperbolic activation function

For $N' + 1$ generated grid points $z_1, z_2, ..., z_{N'+1} \in \mathbb{R}$ and correspondingly N' linear models, the general PWA formulation introducing the auxiliary variables h'_i and λ'_i, is as follows [49]:

$$
\begin{aligned}
f(z) \approx f_{PWA}(z) &= \sum_{i=1}^{N'+1} \lambda'_i f(z_i), \\
z &= \sum_{i=1}^{N'+1} \lambda'_i z_i, \\
\sum_{i=1}^{N'+1} \lambda'_i &= 1, \\
\lambda'_1 &\leq h'_1, \\
\lambda'_i &\leq h'_i + h'_{i-1}, \forall i \in \{2, 3, ..., N'\} \\
\lambda'_{N'+1} &\leq h'_{N'}, \\
\lambda'_i &\geq 0, \forall i \in \{1, 2, ..., N'+1\} \\
\sum_{i=1}^{N'} h'_i &= 1, \\
h'_i &\in \{0, 1\}^{N'}
\end{aligned}
\tag{13}
$$

It should be noted here that the above formulation allows only two adjacent λ_i's to be non-zero.

Substituting the highly non-convex $f(z)$ in the PCA-ANN optimization formulation (Eq.10)with the above PWA

reformulation (Eq.13), the general PCA-ANN-PWA based MILP optimization problem can be obtained:

$$
\begin{aligned}
&\min_{\boldsymbol{d}} \quad G(\boldsymbol{d}, \boldsymbol{u'}) \\
&s.t. z^j = \sum_{i=1}^{N_d} w_{j,i}^1 d_i + b_j^1, && \forall j \in \{1,2,...,n\} \\
&h_j = \sum_{i=1}^{N'+1} \lambda_i^j f(z_i), && \forall j \in \{1,2,...,n\} \\
&z^j = \sum_{i=1}^{N'+1} \lambda_i^j z_i, && \forall j \in \{1,2,...,n\} \\
&\sum_{i=1}^{N'+1} \lambda_i^j = 1, && \forall j \in \{1,2,...,n\} \\
&\lambda_1^j \le h_1^j, && \forall j \in \{1,2,...,n\} \\
&\lambda_i^j \le h_i^j + h_{i-1}^j, && \forall i \in \{2,3,...,N'\}, \forall j \in \{1,2,...,n\} \\
&\lambda_{N'+1}^j \le h_{N'}^j, && \forall j \in \{1,2,...,n\} \\
&\lambda_i^j \ge 0, && \forall i \in \{1,2,...,N'+1\}, \forall j \in \{1,2,...,n\} \\
&\sum_{i=1}^{N'} h_i^j = 1 && \forall j \in \{1,2,...,n\} \\
&h_i^j \in \{0,1\}^{N'} && \forall j \in \{1,2,...,n\} \\
&u_l' = \sum_{j=1}^{n} w_{l,j}^o h_j + b_l^o, && \forall l \in \{1,2,...,k\} \\
&\boldsymbol{u'} = (u_1', u_2', ..., u_k'), \\
&\boldsymbol{g_{cons}}(\boldsymbol{d}, \boldsymbol{P^T u'}) \le 0
\end{aligned}
\tag{14}
$$

3.2 | Illustrative example

To verify the efficiency of the above PWA formulation, global optimization is performed for the surrogate ANN model and the ANN-PWA model of the peaks function (Eq.(11)). Here, the ANN model is the reduced model (No PCA reduction was necessary). Two different PWA models (with 30 and 58linear segments, respectively) following the above adaptive procedure. Tab.1 shows the comparison of optimal results of three reduced models. Almost the same optimal solutions are computed, which are close to the global optimum value of the FOM. The proposed ANN-PWA model with 30 linear segments could requires significanlty less computational time compared to the other two. In fact we can observe a 4-fold reduction compared to the ANN-PWA model with 58 linear segments and a major 30-fold reduction compared to the ANN formulation.

TABLE 1 Comparative results of ANN model and ANN-PWA models

Model	Solver	Optimal value	CPU time (s)	Rel.tolerance
ANN(1 layer, 52 neurons, tanh)	BARON	-6.555	30294.26	0.002
ANN-PWA(30 linear segments)	CPLEX	-6.542	1004.71	0.002
ANN-PWA(58 linear segments)	CPLEX	-6.540	4190.16	0.002

3.3 | Case study

To further investigate the computational efficiency of the PWA method, the PCA-ANN-PWA optimization framework is illustrated using a chemical engineering application: a tubular reactor, where an exothermic reaction takes place [50]. The model of the reactor consists of 2 differential equations in dimensionless form. The mathematical formulation of the optimization problem is as follows :

$$\max_{T_{wi}} C_{exit}$$

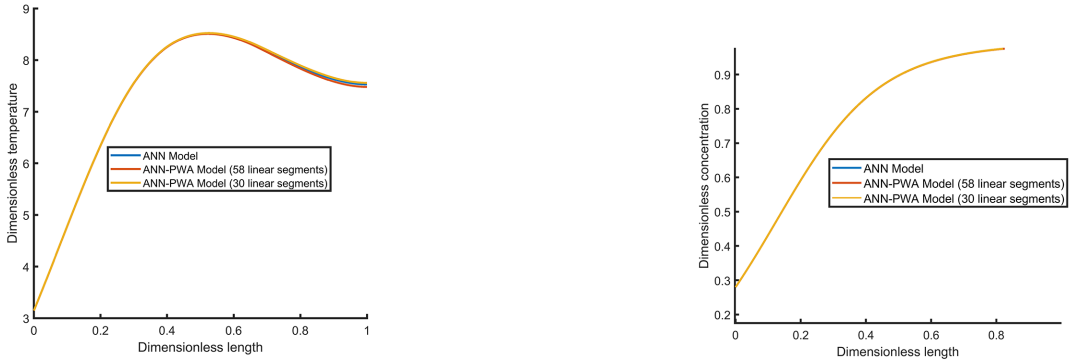
$$
\begin{aligned}
&s.t.\\
&\quad 0 = \frac{1}{Pe_1}\frac{\partial^2 C}{\partial y^2} - \frac{\partial C}{\partial y} + Da(1-C)exp(T/(1+T/\gamma))\\
&\quad 0 = \frac{1}{LePe_2}\frac{\partial^2 T}{\partial y^2} - \frac{1}{Le}\frac{\partial T}{\partial y} - \frac{\beta}{Le}T + BDa(1-C)exp(T/(1+T/\gamma)) + \frac{\beta}{Le}T_w\\
&b.c.\\
&\quad \frac{\partial C}{\partial y} - Pe_1 C = 0, \frac{\partial T}{\partial y} - Pe_2 T = 0, \quad at \quad y = 0\\
&\quad \frac{\partial C}{\partial y} = 0, \frac{\partial T}{\partial y} = 0, \quad at \quad y = 1\\
&\quad\quad 0 \le Twi \le 5\\
&\quad T_w(y) = \sum_{i=1}^{3}(H(y-y_{i-1}) - H(y) - y_i)\,T_{wi}
\end{aligned}
\tag{15}
$$

Here, C and T are the dimensionless concentration and temperature respectively, while C_{exit} is dimensionless output concentration. Da is the Damköhler number, Le is the Lewis number, Pe_1 is the Peclet number for mass transport and Pe_2 for heat transport, β a dimensionless heat transfer coefficient, C is the dimensionless adiabatic temperature rise, γ the dimensionless activation energy and y the dimensionless longitudinal coordinate. The system parameters are Pe_1 = 5, $Pe_2 = 5$, Le = 1, Da = 0.1, β = 1.5, γ = 10, B = 12; T_w is the adiabatic wall temperature and $T_w i$ are the corresponding wall temperatures at the three cooling zones. H is the Heaviside step function.

The resulting discretized 500 algebraic equations comprise our in-house FOM simulator. PCA reduction is performed first to reduce the 500 state variables down to 12. Subsequently ANNs are used to obtain a reduced PCA-ANN model comprising 3 inputs, 12 state variables, and 30 ANN neurons. The optimization results, as displayed in Tab.2 and Fig.5, are computed to compare the optimization performance using the PCA-ANN model and two PCA-ANN-PWA models with 30 and 58 linear segments, respectively. All three computational cases converge to almost the same solutions, with objective function values close to 0.99998, which is the values computed by performing optimization with the FOM. The maximum error is 1.17%, and the optimal solution profiles for concentration and temperature distributions are very close to each other for all models. (Fig.5). Fig.6 compares the computational time required to perform optimization using the PCA- ANN and the PCA-ANN-PWA models with different number of ANN neurons. The limit time (max time for computations to stop) was set to 36000 seconds. The computational time increases rapidly with more neurons for all three surrogate models. The computational cost reaches the limit time for the PCA-ANN model with 40 neurons while the CPU time required for the two PCA-ANN-PWA models is less than 1000 seconds. It can be also seen that the computational time required is significantly less for both PCA-ANN-PWA models, irrespective of the number of ANN neurons, implying the high computational efficiency of the proposed PWA methods.

TABLE 2 Optimal result comparisons for surrogate models of tubular reactor

Model	Optimal value	True value (full model)	Error
PCA-ANN(1 layer, 52 neurons, tanh)	0.98859	0.99998	1.14%
PCA-ANN-PWA(30 linear segments)	0.98825	0.99998	1.17%
PCA-ANN-PWA(58 linear segments)	0.98847	0.99998	1.15%



(a) Solution profiles for temperature

(b) Solution profiles for concentration

FIGURE 5 Solution profiles for dimensionless temperature and concentration

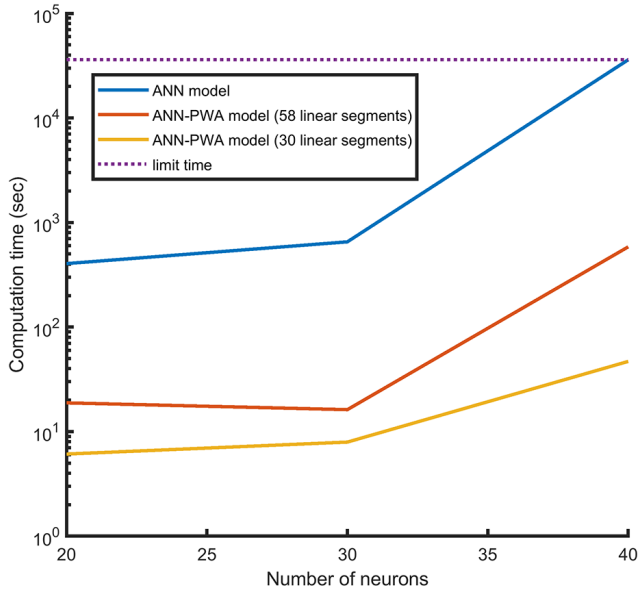


FIGURE 6 Computational time (seconds) for different numbers of neurons

4 | DEEP RECTIFIER NEURAL NETWORK BASED FORMULATION

Or PCA-ANN-PWA global optimization framework worked efficiently for the peaks function and the tubular reactor cases. However, the ANN-PWA models will in general lead to additional approximation error especially for large-scale problems. To preserve the computational accuracy and still use the advanced MILP solver, the continuous piece-wise linear activation function is introduced and directly embedded in the ANN structures. Past efforts in computer science have developed efficient activation functions, such as the sigmoid and the $tanh(\cdot)$ function. The traditional S-shaped sigmoid function can transfer any input signal into the range [0,1] while the zero centered $tanh(\cdot)$ function can map the output values in the interval [-1,1]. Both of them can learn features of high nonlinear functions efficiently. Nevertheless, the high non-convexity of these functions makes ANN training hard to in order to reach a satisfying result. The continuous piece-wise linear functions, including the $ReLU$ function and its variants, have been adopted to deal with this problem. In this work, the widely applied standard $ReLU$ function is utilised. Shallow neural networks require an exponentially larger number of nodes in one layer to successfully represent a complex function, while deep neural networks result in more complex and non-convex error [51]. Low-complexity two- or three-hidden layer NN are, however, enough to capture the low-dimensional nonlinear behaviour of PCA-reduced systems.

Although deep rectifier NN-based MILP problems have been formulated in previous studies [52], the combination of PCA and deep rectifier NN has not been reported. The mathematical equations of deep rectifier NNs are similar to fig. 1 with more hidden layers and activation function $f(z) = max(0, z)$, which can be reformulated into piece-wise

linear function through the big-M method [53]:

$$
\begin{aligned}
z_1^{j_1} &= \sum_{i=1}^{N_d} w_{j_1,i}^1 d_i + b_{j_1}^1, && \forall j_1 \in \{1,2,...,n_1\} \\
z_1^{j_1} &= z'^{j_1}_1 - z''^{j_1}_1, && \forall j_1 \in \{1,2,...,n_1\} \\
h_1^{j_1} &= z'^{j_1}_1 + z''^{j_1}_1, && \forall j_1 \in \{1,2,...,n_1\} \\
z'^{j_1}_1 &\le M_1(1 - bz_1^{j_1}), && \forall j_1 \in \{1,2,...,n_1\} \\
z''^{j_1}_1 &\le M_1 bz_1^{j_1}, && \forall j_1 \in \{1,2,...,n_1\} \\
z_2^{j_2} &= \sum_{j_1=1}^{n_1} w_{j_2,j_1}^2 h_1^{j_1} + b_{j_2}^2, && \forall j_2 \in \{1,2,...,n_2\} \\
z_2^{j_2} &= z'^{j_2}_2 - z''^{j_2}_2, && \forall j_2 \in \{1,2,...,n_2\} \\
h_1^{j_2} &= z'^{j_2}_1 + z''^{j_2}_2, && \forall j_2 \in \{1,2,...,n_2\} \\
z'^{j_2}_2 &\le M_2(1 - bz_2^{j_2}), && \forall j_2 \in \{1,2,...,n_2\} \\
z''^{j_2}_2 &\le M_2 bz_2^{j_2}, && \forall j_2 \in \{1,2,...,n_2\} \\
&\vdots && \vdots \\
z_{II}^{j_{II}} &= \sum_{j_{II-1}=1}^{n_{II-1}} w_{j_{II}j_{II-1}}^{II} h_{II-1}^{j_{II-1}} + b_{j_{II}}^{II}, && \forall j_{II} \in \{1,2,...,n_{II}\} \\
z_{II}^{j_{II}} &= z'^{j_{II}}_{II} - z''^{j_{II}}_{II}, && \forall j_{II} \in \{1,2,...,n_{II}\} \\
h_{II}^{j_{II}} &= z'^{j_{II}}_{II} + z''^{j_{II}}_{II}, && \forall j_{II} \in \{1,2,...,n_{II}\} \\
z'^{j_{II}}_{II} &\le M_{II}(1 - bz_{II}^{j_{II}}), && \forall j_{II} \in \{1,2,...,n_{II}\} \\
z''^{j_{II}}_{II} &\le M_{II} bz_{II}^{j_{II}}, && \forall j_{II} \in \{1,2,...,n_{II}\} \\
z'^{j_i}_i &\ge 0, && \forall i \in \{1,2,...,II\}, \forall j_i \in \{1,2,...,n_i\} \\
z''^{j_i}_i &\ge 0 && \forall i \in \{1,2,...,II\}, \forall j_i \in \{1,2,...,n_i\} \\
bz_i^{j_i} &\in \{0,1\} && \forall i \in \{1,2,...,II\}, \forall j_i \in \{1,2,...,n_i\} \\
u'_l &= \sum_{j_{II}=1}^{n_{II}} w_{l,j_{II}}^o h_{II}^{j_{II}} + b_l^o, && \forall l \in \{1,2,...,k\}
\end{aligned}
\tag{16}
$$

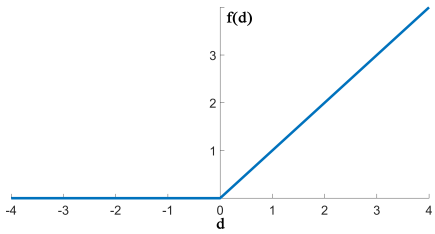
where M_i is the big-M constant, $z'^{j_i}_i$ and $z''^{j_i}_i$ are the auxiliary non-negative variables, $bz_i^{j_i}$ is the auxiliary binary variable and $h_i^{j_i}$ is the output value from the j_ith ReLu based neuron of the ith hidden layer. II is the number o f hidden layers and n_i is number of neurons at ith hidden layer.

Substituting the ANN model equations in the PCA-ANN optimization formulation (Eq.10)with the above PWA

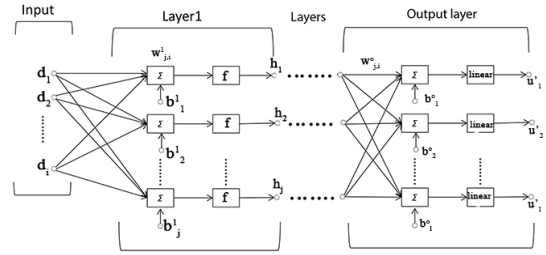
reformulation (Eq.17), the general PCA-DNN(Relu) based MILP optimization formulation can be obtained:

$$\begin{aligned} \min_{\boldsymbol{d}=[d_1,d_2,\ldots,d_{N_d}]} \quad & G(\boldsymbol{d},\boldsymbol{u'}) \\ s.t. & Eq.16; \\ & \boldsymbol{u'} = (u'_1, u'_2, \ldots, u'_k), \\ & \boldsymbol{g}_{cons}(\boldsymbol{d}, \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{u'}) \leq 0 \end{aligned} \tag{17}$$

This way, an improved framework is formulated using a deep neural network (DNN) with rectified linear units (ReLU) as illustrated in Fig.7. This improved framework is first tested with the small peaks function, and then extended to a large-scale combustion process.



(a) ReLU activation function



(b) Deep artificial neural network

FIGURE 7 Deep neural network with rectified linear units

4.1 | Illustrative example

To verify the superiority of the reLu-DNN models, in terms of computational efficiency global optimization is first constructed for four surrogate models (ANN, ANN-PWA, tanh-DNN and relu-DNN model, respectively) for the peaks function previously presented. Tab.3 shows the optimization results. The small-scale tanh-DNN could replace the larger shallow ANN model, resulting in a significant computational saving, more than one order of magnitude. The relu-DNN model requires much more neurons than the tanh-DNN model, due to the lower non-linearity of the $relu$ activation function. Despite the fact that the relu-DNN model is larger, its ptimization cost is much lower, two orders of magnitude less than the cost of the tanh-NN models. The rapid global optimization computations using the relu-NN model is attributed to the advanced MILP solver algorithm utilised. Furthermore, the computation cost using the relu-DNN model is much less than that using the ANN-PWA model with 30 linear segments because of the large(r) number of linear models involved in the PWA formulation. More linear models lead to more binary variables, requiring more branching steps reducing the computational efficiency.

TABLE 3 Comparative optimization results of different ANN models

Model	Solver	Optimal value	CPU time (s)	Rel.tolerance
ANN(1 layer, 52 neurons, $tanh$)	BARON	-6.555	30294.26	0.002
ANN-PWA (30 linear segments)	CPLEX	-6.542	1004.71	0.002
DNN(2 layers, 8-8 neurons, $tanh$)	BARON	-6.558	2579.68	0.002
DNN(2 layers, 40-40 neurons, $relu$)	CPLEX	-6.543	25.93	0.002

4.2 | Case study

To verify the significant advantages of the deep rectifier neural network in our global optimization formulation observed in the previous section, a more challenging combustion process [54, 23] is considered here.

4.2.1 | Process description

A combustion process taking place in a horizontal cylindrical combustor, 1.8m in length and 0.45m in diameter with a fuel nozzle with diameter 0.0045m is considered here. The overall reactions in the combustor are as follows:

- $CH_4 + 2\,O_2 \rightarrow CO_2 + 2\,H_2O$
- $C_2H_4 + 3\,O_2 \rightarrow 2\,CO_2 + 2\,H_2O$
- $C_3H_8 + 5\,O_2 \rightarrow 3\,CO_2 + 4\,H_2O$
- $C_4H_{10} + 8.5\,O_2 \rightarrow 4\,CO_2 + 5\,H_2O$

In addition a complex NO mechanism, comprising thermal NO, prompt NO and N_2O intermediate mechanism is also taken into account. Fuel NO mechanism was ignored due to the small amount of nitrogen in the feed. Thermal efficiency can be improved by increasing combustion temperature, which however, inevitably leads to more pollutant emissions, such as NOx. The NOx production is dominated by the thermal NO mechanism, given below, which is very sensitive to temperature.

- $O + N_2 \rightleftharpoons NO + N$
- $O_2 + N \rightleftharpoons NO + O$
- $N + OH \rightleftharpoons NO + H$

This work focuses on the optimization of inlet operational conditions (shown in Tb. 4) in order to minimize NOx emissions. In addition to chemical reactions, multiple physical phenomena are involved, including complex turbulent flows, heat and mass transfer. Commercial CFD software was used, namely ANSYS/FLUENT, to construct high-fidelity CFD models to calculate velocity, temperature and component fraction fields.

4.2.2 | CFD Model Description

The computation domain for the CFD model consisted of a 2-dimensional axisymmetric depiction of the combustor To ensure that computations are grid independent, numerical experiments using 5481, 6381, 9081 and 14832 computa-

tional cells were performed for the maximum temperature. Finally, 9081 computational cells (9332 nodes) were chosen as solutions did not change with more computational cells/nodes. The renormalisation group (RNG) $k-\epsilon$ turbulence model for fluid flow is employed. The eddy-dissipation model was employed for the species transport equations because the overall reaction rate is controlled by turbulent mixing. To take into account the effects thermal radiation, including absorption and scatting coefficients, a discrete ordinates (DO) radiation model was used.

The second-order upwind scheme was applied for the space derivatives of the advection terms in all transport equations. The SIMPLE algorithm was employed to handle the velocity-pressure coupling in the flow field equations. Convergence criteria required the residual for the energy equation to be below 1×10^{-6} and the residuals for the other model equations below 1×10^{-3}. The mass-weighted-averages of temperature at the exit and the maximum temperature of the entire fluid were also monitored as other convergence criteria.

The base case inlet conditions are given below. For fuel gas, the base value for inlet velocity was 100 m/s and that for inlet temperature was 298K. The inlet composition was as follows: CH_4: 87.8%, C_2H_4: 4.6%, C_3H_8: 1.6%, C_4H_{10}: 0.5%, N_2: 5.5%. For preheated air gas, the inlet velocity was 85 m/s and the inlet temperature 1473 K while the inlet composition was: O_2: 19.5%, N_2: 59.1%, H_2O: 15%, CO_2: 6.4%, NO: 110 ppm. Five independent variables were used to optimize the whole process. The independent variables along with their allowable ranges are listed in Tab.4 below.

TABLE 4 Range of independent variables

Variables	Range	Units
inlet air velocity	[85,125]	m/s
inlet fuel velocity	[80,120]	m/s
oxygen mass fraction (inlet air)	[18.5, 19.5]	%
inlet air temperature	[1450, 1600]	K
inlet fuel temperature	[298, 398]	K

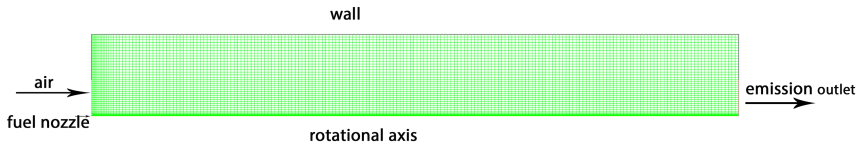


FIGURE 8 Two-dimensional geometry of a single axisymmetric combustor can and its mesh

4.2.3 | Model reduction

Although the high-fidelity CFD model can provide accurate simulation results, its black-box characteristics and overall complexity make further optimization and control tasks coputationally tedious. Reduced surrogate models need to be developed to deal with the challenges arising. The LHC sampling method was utilized to collect 1024 CFD samples. The input variables are the ones listed in Tab.4, while the output results are the physical field data along with the average NOx emission at the outlet surface. Due to the high dimensionality of the FOM, direct mapping of the input-output relationship would result in very large-scale ANN surrogate models, which often exceed the capability of current optimization algorithms. Therefore, the PCA step was first employed and then surrogate ANN models were constructed based on the PCA-reduced models. ANN models were built for the field data, to construct the redcuced PCA-ANN constraints and for the average output NOx emission to formulate the ANN-reduced objective function. The field data

include axial and radial velocity, Static Temperature, N_2, H_2O, O_2, CO_2, C_4H_{10}, CH_4, C_2H_4, C_3H_8, and NO fraction concentrations (12 state variables). It should be noted that the average output NOx emission is only one variable so does not require a PCA reduction step. In this work, PCA was performed separately for each state variable. While some PCA methods compute principal components for all state variables together, we found that working on each state variable we could generate more accurate principal components. The standard criterion, of capturing 99.99% of the total energy, was set. This way, the reduced surrogate models, were built, as displayed in Tab.5.

TABLE 5 Number of PCs and corresponding ANN models

Variables	Number of PCs	DNN (2 layers, tanh) No of neurons	DNN (2 layers, relu) No of neurons
Axial velocity	4	14, 14	14, 14
Radial velocity	9	15, 15	22, 22
Temperature	6	16, 16	16, 16
N_2 concentration fraction	7	19, 19	24, 24
H_2O concentration fraction	8	15, 15	18, 18
O_2 concentration fraction	6	17, 17	20, 20
CO_2 concentration fraction	7	12, 12	14, 14
C_4H_{10} concentration fraction	6	15, 15	17, 17
CH_4 concentration fraction	7	26, 26	28, 28
C_2H_4concentration fraction	6	10, 10	18, 18
C_3H_8concentration fraction	6	18, 18	24, 24
NO concentration fraction	4	12, 12	14, 14
Objective: output NOx emission	-	ANN (1 layer, tanh)	ANN (1 layer, relu)
	-	14	30

4.2.4 | Model validation

Model validation was implemented on the reduced models before the subsequent optimization step, taking into account two aspects, representation ability and prediction ability. The representation ability of the reduced models was tested through the comparison between the FOM and ROMs on the base case inlet conditions. Computational results show only very small differences, especially for N_2, C_4H_{10}, CH_4, C_2H_4, C_3H_8, and NO fraction fields. The above species fraction fields are close to uniform distribution across the combustor, except for the small area near the fuel nozzle. Fig.9(a), 9(b), 9(c), 9(d), 9(e) give the velocity field, temperature field, O_2 , CO_2 and H_2O concentration fraction field of FOM, tanh-ROM and relu-ROM under the inlet being base values. The five contour diagrams illustrate that flow, temperature and mass fraction fields of FOM and ROMs are very close, indicating the strong representation ability of the ROMs. Moreover, the tanh-DNN reduced models show smaller difference than the relu-DNN reduced models, especially for the temperature field, implying the better accuracy of the tanh-DNN models due to the non-linearity of tanh function. Tab.6 shows the comparison of maximum field values between FOM and ROMs and the corresponding errors. The largest error is only 0.56%. To test the ROMs prediction ability, 40 random inlet condition points different than the base case ones were chosen and compered with FOM results. The largest error was less than 5% indicating that the ROMs can be reliably used for further optimisation studies. Fuerhtermore, the ROMs exhibit significant computational savings compared to the full-order CFD models as expected. The average CPU time for the CFD model

(run in ANSYS/FLUENT) is approximately 1560 CPU seconds, while each ROM requires less than 0.1 CPU seconds and can be efficiently used to perform global optimisation studies.

TABLE 6 Average value comparison of FOM and ROMs

Variables	FOM	relu-ROMs	errors	tanh-ROMs	errors
Velocity(m/s)	29.82089	29.6652	0.40%	29.70166	0.56%
Temperature(K)	1625.259	1621.948	0.03%	1625.702	0.20%
H_2O mass fraction	0.151743	0.1518191	0.02%	0.1517722	0.05%
O_2 mass fraction	0.1906906	0.1908453	0.01%	0.1907176	0.08%
CO_2 mass fraction	0.0663304	0.06627689	0.00%	0.06633245	0.08%

4.2.5 | Global optimisation

In this section, global optimisation is implemented using the validated reduced models. The general mathematical formulation is given in Eq.(17). In the combustion optimisation problem, $\boldsymbol{d}$ are the 5 inlet operation parameters, $\boldsymbol{u'}$ are the 76 reduced state variables . The objective function $G(\boldsymbol{d}, \boldsymbol{u'})$ represents the average outlet NOx emission. The allowable ranges for the input variables are given in Tab.4, while the bounds for the state variables are given in Tab.7. It should be noted that the state variable bounds are implemented through the inverse projection

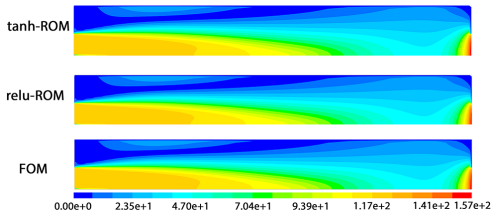
$$lb \leq \boldsymbol{P^T u'} \leq ub \tag{18}$$

where lb and ub denote lower and upper bounds, respectively.

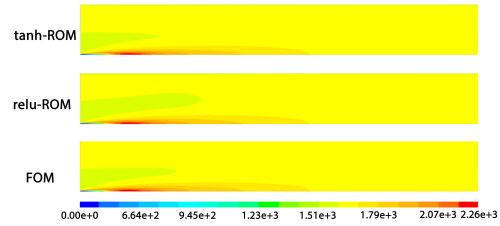
Finally, a MILP problem with 29,903 linear constraints, corresponding to the equality constraints and 488 binary variables corresponding to the total number of ANN neurons is formulated for the relu-based ROM, while a NLP problem with 28247 linear constraints, and 392 nonlinear terms is constructed for the tanh-based ROM. The limit value for the computational time was set to be 100 hours. Both of the relative and absolute tolerances were set to be 0.002.

The NLP problem did not converge to a feasible solution in BARON within the allowable time, probably due to the high non-convex activation function $tanh$ and large number of variables than inhibited the branch-and-bound algorithm.

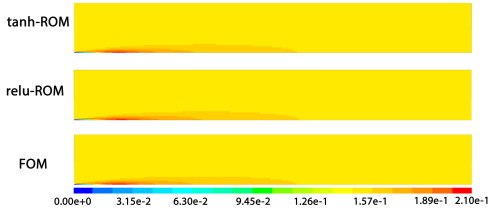
The $relu$-based MILP problem converged in 703.39 s in CPLEX. The computed optimal solution was: NOx emission: 110.17 ppm, air velocity: 95.07 m/s, fuel velocity:119.08m/s, oxygen fraction concentration (air): 18.50 %, air temperature: 1450 K and fuel temperature: 369.83K. To validate the computed optimal solutions, we performed a full CFD simulation in ANSYS/FLUENt using the calculated optimal inlet conditions. The outlet NOx emission was computed to be 113.26 ppm, which was very close to the calculated optimum with an error of approximately 2.73 %, which is small enough for most industrial cases. Fig.10(a), 10(b), 10(c), 10(d) ,10(e), depicts a comparison of the main field state varibales at the optimal; conditions computed by the reduced and the full models, respectively. As it can be observed, the optimal solution computed through the ROM is very close to FOM simulation using the optimnal inlet conditions. Tab.8 gives a comparison of the corresponding max values across the whole domain. The performance of the reduced model is very close to the full model with the biggest error being less than 3%. The computational cost for the relu-based MILP problem is significanlty reduced compared to the NLP problem.



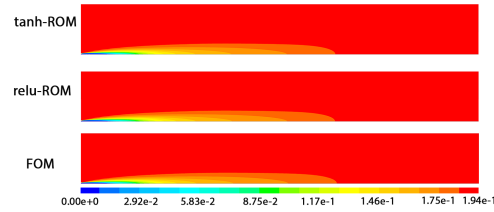
(a) Velocity (m/s) comparison among full model and reduced models



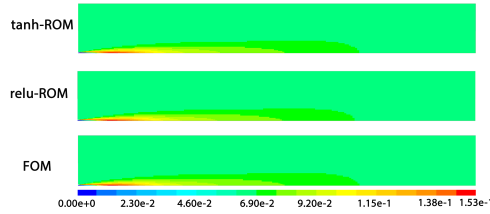
(b) Temperature (K) comparison among full model and reduced models



(c) H_2O mass comparison among full model and reduced models



(d) O_2 mass comparison among full model and reduced models

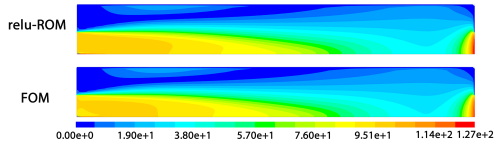


(e) CO_2 mass comparison among full model and reduced models

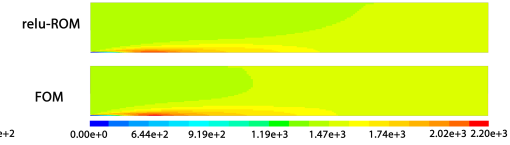
FIGURE 9 Comparison of velocity, temperature and concentration fraction field between FOM and ROMs

TABLE 7 Range of state variables

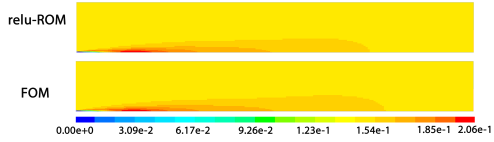
State variables	Range	Units
Axial velocity	[-150,150]	m/s
Radial velocity	[-150,150]	m/s
Temperature	[0, 2200]	K
N_2 concentration fraction	[0,1]	-
H_2O concentration fraction	[0,1]	-
O_2 concentration fraction	[0,1]	-
CO_2 concentration fraction	[0,1]	-
C_4H_{10} concentration fraction	[0,1]	-
CH_4 concentration fraction	[0,1]	-
C_2H_4 concentration fraction	[0,1]	-
C_3H_8 concentration fraction	[0,1]	-
NO concentration fraction	[0,1]	-



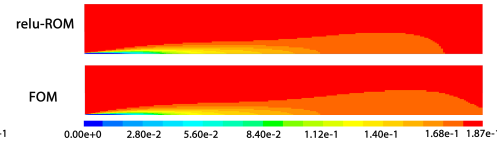
(a) Velocity (m/s) comparison among full model and reduced models under optimal condition



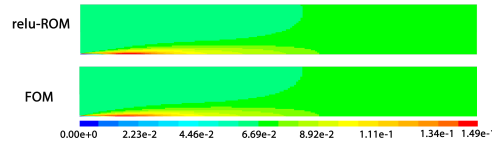
(b) Temperature (K) comparison among full model and reduced models under optimal condition



(c) H_2O mass fraction comparison among full model and reduced models under optimal condition



(d) O_2 mass fraction comparison among full model and reduced models under optimal condition



(e) CO_2 mass fraction comparison among full model and reduced models under optimal condition

FIGURE 10 Comparison of optimal velocity, temperature and fraction concentration field between FOM and ROM

TABLE 8 Average value comparison of FOM and ROMs

Variables	FOM	relu-ROMs	errors
Velocity(m/s)	23.11901	23.73662	2.67%
Temperature(K)	1487.854	1456.745	2.09%
H_2O mass fraction	0.1522371	0.1520701	0.11%
O_2 mass fraction	0.1803872	0.1806236	0.13%
CO_2 mass fraction	0.06699268	0.06689615	0.14%
Output NOx emission (ppm)	113.26	110.17	2.73%

5 | CONCLUSIONS

This paper presents a model-reduction based global optimisation framework for large-scale nonlinear steady-state systems. A double model reduction, comprising principal component analysis and artificial neural networks, were first employed to construct the reduced model, which was then utilized by deterministic global optimization methods. The high non-convexity of the activation function in reduced ANN models affects the computational speed branch-and-bound algorithms. To overcome this barrier, two improvements are proposed. Firstly, a piece-wise affine reformulation to transform the nonlinear branching into binary variables resulting in a MILP problem with higher computational efficiency. Secondly. the implementation of a continuous piece-wise linear activation function-based deep ANN structure to improve computational accuracy. Applications including peaks function, a tubular reactor and a complex large-scale combustion process were employed to illustrate the favorable performance of the improved framework. Nevertheless, it is still a challenge to efficiently compute the global optimum for large-scale optimization problems. Firstly, this work assumes enough representative samples as a basis to construct the reduced order models. Smart sampling methods to achieve optimal trade-off between quality and quantity are important for improving both efficiency and accuracy, as well as verification methods to guarantee the accuracy of the computed solutions[55]. Secondly, global optimizsation even using reduced surrogate models is still computationally expensive. Advanced data techniques and MILP algorithms [56] may further improve computational efficiency of this optimisation framework.

ACKNOWLEDGEMENTS

The financial support of the University of Manchester and China Scholarship Council joint scholarship (file no. 201706250031) for MT's PhD studies is gratefully acknowledged

CONFLICT OF INTEREST

The authors declare there is no conflict of interest.

REFERENCES

[1] Boukouvala F, Hasan MF, Floudas CA. Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption. Journal of Global Optimization 2017;67(1-2):3–42.

[2] Tao M, Guo K, Huang Z, Liu H, Liu C. A hybrid optimization method to design shapes of three-dimensional flow channels. Chemical Engineering Research and Design 2016;114:190–201.

[3] Park S, Li Y. Integration of biological kinetics and computational fluid dynamics to model the growth of Nannochloropsis salina in an open channel raceway. Biotechnology and bioengineering 2015;112(5):923–933.

[4] Yang GL, Zhou WH, Liu F. Simulation of flow field of high-pressure water-jet from nozzle with FLUENT [J]. Journal of Lanzhou University of Technology 2008;34(2):49–52.

[5] Kleber A. Simulation of air flow around an Opel Astra vehicle with FLUENT. Journal Article, International Technical Development Center Adam Opel AG 2001;.

[6] Multiphysics C. Introduction to COMSOL Multiphysics®. COMSOL Multiphysics, Burlington, MA, accessed Feb 1998;9:2018.

[7] Fluent A. Ansys fluent. Academic Research Release 2015;14.

[8] Jasak H, Jemcov A, Tukovic Z, et al. OpenFOAM: A C++ library for complex physics simulations. In: International workshop on coupled methods in numerical dynamics, vol. 1000 IUC Dubrovnik Croatia; 2007. p. 1–20.

[9] Schilders WH, Van der Vorst HA, Rommes J. Model order reduction: theory, research aspects and applications, vol. 13. Springer; 2008.

[10] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. science 2006;313(5786):504–507.

[11] Theodoropoulou A, Adomaitis RA, Zafiriou E. Model reduction for optimization of rapid thermal chemical vapor deposition systems. IEEE Transactions on Semiconductor Manufacturing 1998;11(1):85–98.

[12] Xie W, Bonis I, Theodoropoulos C. Data-driven model reduction-based nonlinear MPC for large-scale distributed parameter systems. Journal of Process Control 2015;35:50–58.

[13] Malik MR, Isaac BJ, Coussement A, Smith PJ, Parente A. Principal component analysis coupled with nonlinear regression for chemistry reduction. Combustion and flame 2018;187:30–41.

[14] Bonis I, Theodoropoulos C. Model reduction-based optimization using large-scale steady-state simulators. Chemical engineering science 2012;69(1):69–80.

[15] Petsagkourakis P, Bonis I, Theodoropoulos C. Reduced Order Optimization of Large-Scale Nonlinear Systems with Nonlinear Inequality Constraints Using Steady State Simulators. Industrial & Engineering Chemistry Research 2018;57(30):9952–9963.

[16] Bonis I, Xie W, Theodoropoulos C. Multiple model predictive control of dissipative PDE systems. IEEE Transactions on Control Systems Technology 2013;22(3):1206–1214.

[17] Luna-Ortiz E, Theodoropoulos C. An input/output model reduction-based optimization scheme for large-scale systems. Multiscale Modeling & Simulation 2005;4(2):691–708.

[18] Theodoropoulos C. Optimisation and linear control of large scale nonlinear systems: a review and a suite of model reduction-based techniques. In: Coping with Complexity: Model Reduction and Data Analysis Springer; 2011.p. 37–61.

[19] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. science 1983;220(4598):671–680.

[20] Chambers LD. Practical handbook of genetic algorithms: complex coding systems, vol. 3. CRC press; 2019.

[21] Floudas CA, Akrotirianakis IG, Caratzoulas S, Meyer CA, Kallrath J. Global optimization in the 21st century: Advances and challenges. Computers & Chemical Engineering 2005;29(6):1185–1202.

[22] Pires JCM, Martins FG, Sousa S, Alvim-Ferraz M, Pereira M. Selection and validation of parameters in multiple linear and principal component regressions. Environmental Modelling & Software 2008;23(1):50–55.

[23] Lang Yd, Malacina A, Biegler LT, Munteanu S, Madsen JI, Zitney SE. Reduced order model based on principal component analysis for process simulation and optimization. Energy & Fuels 2009;23(3):1695–1706.

[24] Henao CA, Maravelias CT. Surrogate-based superstructure optimization framework. AIChE Journal 2011;57(5):1216–1232.

[25] Fahmi I, Cremaschi S. Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models. Computers & Chemical Engineering 2012;46:105–123.

[26] Tawarmalani M, Sahinidis NV. A polyhedral branch-and-cut approach to global optimization. Mathematical Programming 2005;103(2):225–249.

[27] Schweidtmann AM, Mitsos A. Deterministic global optimization with artificial neural networks embedded. Journal of Optimization Theory and Applications 2019;180(3):925–948.

[28] Tao M, Li J, Theodoropoulos C. Reduced model-based global optimisation of large-scale steady state nonlinear systems. In: Computer Aided Chemical Engineering, vol. 46 Elsevier; 2019.p. 1039–1044.

[29] Houska B, Chachuat B. Global optimization in Hilbert space. Mathematical programming 2019;p. 1–29.

[30] Loh WL, et al. On Latin hypercube sampling. Annals of statistics 1996;24(5):2058–2080.

[31] Faure H. On the star-discrepancy of generalized Hammersley sequences in two dimensions. Monatshefte für Mathematik 1986;101(4):291–300.

[32] Fang KT, Wang Y. Number-theoretic methods in statistics, vol. 51. CRC Press; 1993.

[33] de Aguiar PF, Bourguignon B, Khots M, Massart D, Phan-Than-Luu R. D-optimal designs. Chemometrics and intelligent laboratory systems 1995;30(2):199–210.

[34] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 2000;42(1):55–61.

[35] Stein M. Large sample properties of simulations using Latin hypercube sampling. Technometrics 1987;29(2):143–151.

[36] Richardson M. Principal component analysis. URL: http://people maths ox ac uk/richardsonm/SignalProcPCA pdf (last access: 35 2013) Aleš Hladnik Dr, Ass Prof, Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales hladnik@ ntf uni-lj si 2009;6:16.

[37] Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of educational psychology 1933;24(6):417.

[38] Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. Annual review of fluid mechanics 1993;25(1):539–575.

[39] Jirsa VK, Friedrich R, Haken H, Kelso JS. A theoretical model of phase transitions in the human brain. Biological cybernetics 1994;71(1):27–35.

[40] Park S, Lee JJ, Yun CB, Inman DJ. Electro-mechanical impedance-based wireless structural health monitoring using PCA-data compression and k-means clustering algorithms. Journal of intelligent material systems and structures 2008;19(4):509–520.

[41] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural networks 1989;2(5):359–366.

[42] Yu H, Wilamowski BM. Levenberg-marquardt training. Industrial electronics handbook 2011;5(12):1.

[43] MacKay DJ. A practical Bayesian framework for backpropagation networks. Neural computation 1992;4(3):448–472.

[44] Misener R, Floudas CA. ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. Journal of Global Optimization 2014;59(2-3):503–526.

[45] Rehfeldt D, Koch T. SCIP-Jack—a solver for STP and variants with parallelization extensions: An update. In: Operations Research Proceedings 2017 Springer; 2018.p. 191–196.

[46] Smith JD, Neto AA, Cremaschi S, Crunkleton DW. CFD-based optimization of a flooded bed algae bioreactor. Industrial & Engineering Chemistry Research 2012;52(22):7181–7188.

[47] Henao CA. A superstructure modeling framework for process synthesis using surrogate models. PhD thesis, The University of Wisconsin-Madison; 2012.

[48] Keßler T, Mertens N, Kunde C, Nentwich C, Michaels D, Engell S, et al. Efficient global optimization of a novel hydroformylation process. In: Computer Aided Chemical Engineering, vol. 40 Elsevier; 2017.p. 2113–2118.

[49] Floudas CA. Nonlinear and mixed-integer optimization: fundamentals and applications. Oxford University Press; 1995.

[50] Jensen KF, Ray WH. The bifurcation behavior of tubular reactors. Chemical Engineering Science 1982;37(2):199–222.

[51] Lee JH, Shin J, Realff MJ. Machine learning: Overview of the recent progresses and implications for the process systems engineering field. Computers & Chemical Engineering 2018;114:111–121.

[52] Grimstad B, Andersson H. ReLU networks as surrogate models in mixed-integer linear programs. Computers & Chemical Engineering 2019;131:106580.

[53] Belotti P, Liberti L, Lodi A, Nannicini G, Tramontani A. Disjunctive inequalities: applications and extensions. Wiley Encyclopedia of Operations Research and Management Science 2010;.

[54] Wei Z, Li X, Xu L, Tan C. Optimization of operating parameters for low NO x emission in high-temperature air combustion. Energy & Fuels 2012;26(5):2821–2829.

[55] Botoeva E, Kouvaros P, Kronqvist J, Lomuscio A, Misener R. Efficient verification of relu-based neural networks via dependency analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34; 2020. p. 3291–3299.

[56] Anderson R, Huchette J, Tjandraatmadja C, Vielma JP. Strong convex relaxations and mixed-integer programming formulations for trained neural networks. arXiv preprint arXiv:181101988 2018;.