Machine Learning Representations for Optimization of Process Systems

Logan Watts
University of Arkansas, Fayetteville

Machine Learning Representations for Optimization of Process Systems

Logan Watts with Advisor: Dr. Carl Laird

April 23, 2021

Abstract

The optimal operation of chemical processes provides the foundation for optimization problems to determine the most effective way to operate or design a given process. Chemical processes can be represented as nonlinear systems of equations with decision variables, resulting in a problem that can be solved through nonlinear solvers. The downfalls of nonlinear solvers create the need for improved methods of finding globally optimal solutions to the design or operation of a chemical process. The project will seek to evaluate the use of artificial neural networks to approximate nonlinear systems of equations for the purpose of optimizing chemical processes. The super critical carbon dioxide (sCO2) Brayton recompression cycle was selected as a surrogate chemical process. The process was approximated by a neural network with a rectifying linear activation unit (ReLU). The sCO2 power generation cycle involves discrete decisions and is nonlinear, however the mixed integer nonlinear programming problem can be approximated as a mixed integer linear programming (MILP) form because of the ReLU formulation of the neural network. The MILP formulation for the optimization of the ReLU approximation successfully modeled the locally optimal solution of the original nonlinear model, supporting the use of neural network approximations for complex chemical processes as well as the MILP approximation of the mixed integer nonlinear problem.

1 Introduction

Numerical optimization has proven to be an important tool for improving the design and operation of chemical processes. Focusing on steady-state optimization, chemical processes can often be represented by nonlinear systems of equations with design and operations decision variables that are continuous or discrete. While several nonlinear optimization packages exist that can efficiently find locally optimal solutions for large nonlinear systems (e.g., [10]), it can be very challenging to efficiently solve nonlinear problems with discrete decisions, or even continuous problems to global optimality. In this research project, the use of neural network approximations are investigated in the context of optimization of chemical processes. The use of neural network approximations brings two key advantages in this application. First, there are very effective tools for training the neural network surrogate (e.g., PyTorch, TensorFlow). Second, it has recently been shown that neural networks with rectifying linear activation units (i.e., ReLU networks) are piecewise linear functions that can be represented within mixed-integer linear programming (MILP) problems[9, 1]. Powerful solvers exist that can efficiently find globally optimal solutions of MILP problems. Furthermore, if we want to solve problems with nonlinear chemical processes and discrete decisions, this class of mixed-integer *nonlinear* programming problems is now also representable (approximately) as a mixed-integer linear programming problem. In this work, the applicability of ReLU networks as surrogates for optimization of chemical processes is investigated. Specifically, this work seeks to model a super-critical CO2 power generation cycle, approximate that system with an appropriately sized ReLU neural network, and verify the accuracy of an MILP formulation for optimization of the ReLU approximation against the (locally) optimal solution of the original nonlinear model.

Power generation provides a good case study for problems requiring optimization for the design of novel, efficient power generation processes. Before discussing any power generation cycles, it is important to note the differences between direct and indirect power generation cycles. Direct-fired power generation cycles involve the combustion gases themselves being used to drive the turbine. In contrast, indirect-fired cycle use heat from a boiler to drive a separate working fluid through the turbine cycle. The combustion gases and the working fluid do not mix in indirect cycles. Indirect power generation cycles can use heat from a wide variety of sources, which leads to the efficiency of the cycles generally increasing with increased amounts of present heat [11]. Heat that would normally be wasted as a byproduct of high thermal heat processes can be repurposed for use in power generation cycles to reduce process costs. As compared to similar power generation cycles, namely the Rankine cycle, efficiency improvements between 2 and 6 percent are possible when using the sCO2 indirect Brayton cycles [11]. Significant improvements in efficiency can yield increased process cost reductions for processes that would otherwise waste the heat. Some additional benefits of using sCO2 as opposed to steam power generation cycles are, “the potential for higher efficiency, high-power density (more compact) equipment, reduced capital costs, and emissions reductions compared to steam based power cycles” [11]. The advantages of indirect sCO2 cycles over steam-based power cycles provide the basis for further research into the sCO2 cycles.

The indirect sCO2 Brayton cycle has already shown promise for improvements in efficiency over Rankine cycles and steam-based power cycles. formulation of an indirect sCO2 Brayton cycle as an optimization problem involves the consideration of both the power generating components of the cycle along with the input power requirements of the cycle to maximize the efficiency of the process. The specific implications of efficiency optimization will be discussed along with the methods used to solve the associated optimization problem.

Modeling the indirect Brayton SCO2 cycle results in a system of nonlinear equations that can be difficult to solve. While numerical packages exist for local optimization of these systems, it can be difficult to find globally optimal solutions or solve problems with both these nonlinear models and discrete decisions.

Recently, ReLU neural networks have been proposed as surrogates for nonlinear systems [3]. The ReLU networks can be represented in mixed-integer linear programming (MILP) formulations, and it is possible to find globally optimal solutions with modern MILP solvers. The ReLU representation approach is particularly beneficial if the computational burden of the data generation is less than the optimization itself. An example of the computational burden of the data generation being less than the optimization occurs if approximations can be constructed for unit operations and applied to the combinatorial problems of process flowsheet design. Another example occurs if multiple instances of the approximation are needed in the optimization, as is the case for optimization under uncertainty. A third example can be found when the offline computational burden can outweigh the benefits of rapid, reliable online optimization.

A fundamental understanding of artifical neural networks (ANNs) is worth discussing to provide a baseline understanding. Neural networks are composed of layers of nodes. Input data is fed to all the first layer of nodes. The connections between the nodes are assigned weights that determine the impact of the output of a particular node on the input to the connected nodes. All the data fed to a particular node is summed with respect to the weight assigned to the data, and a bias is added to the total. A nonlinear activation function is then used to compute the output of a node, and then the output for a node becomes the input of another node. Several forms of these activation functions exist, including the piecewise linear ReLU activation function. Overall, the layers of nodes take the input data and produce an output layer of data that represents the results of the neural network. The training of a neural network involves the network iterating to determine the appropriate weights and biases that best represent the expected output data as defined by the studied system. After a neural network is trained, the network with "trained" weights and biases can be used to evaluate new input data to yield predictions of output data or be reformulated for use as a surrogate in optimization.

The importance of neural networks lies in the ability of ANN's to approximate a wide range of nonlinear processes. Although ANN's can be used in applications from game design to chemical process modeling, the use of neural networks in power generation process optimization provides the focal point of the study. This study brings together a suite of open-source tools. The IDAES computational framework (developed as part of the DOE funded Institute for the Design of Advanced Energy Systems) is used to model the super-critical CO2 process. This framework is an open-source Python-based package for equation-oriented modeling and optimization of chemical process systems. The neural network models are trained with PyTorch, and ReLU reformulations are developed and solved within Pyomo, an open-source optimization modeling platform.

The structure of this report is as follows. Section 2 provides background information on the super-critical CO2 power generation process and the use of surrogates for chemical process systems. Section 3 describes the methods used, including the data generation with the IDAES framework, training with PyTorch, and optimization using IDAES and the Pyomo optimization platform. The Methods section also discussed the reformulation of the ReLU ANN as a MILP. Section 4 presents the numerical results of the study, comparing the optimal solutions from the ReLU approximation with the local solutions from optimization of the nonlinear IDAES model. Section 5 closes with a summary and conclusions.

2 Background

The background involved in the performed study can be broken into two portions. The sCO2 Brayton cycle subsection will detail the initial simulations performed to validate the IDAES model, and outline the optimization problem solved later in the paper. The surrogate modeling subsection with detail the use of ANNs in the modeling of chemical processes along with the connections between optimization and ANNs.

2.1 sCO2 Brayton Recompression Cycle Flowsheet Background

The IDAES computational framework was used to generate data for training the neural network. However, prior to use, the IDAES model was compared with an ASPEN Plus flowsheet model for the cycle. In the ASPEN Plus model the chemical input for the process was carbon dioxide. The property package used for analysis was REFPROP. The process flow diagram displayed as Figure 1 shows the material streams within the sCO2 system as well as two work streams connecting heaters.

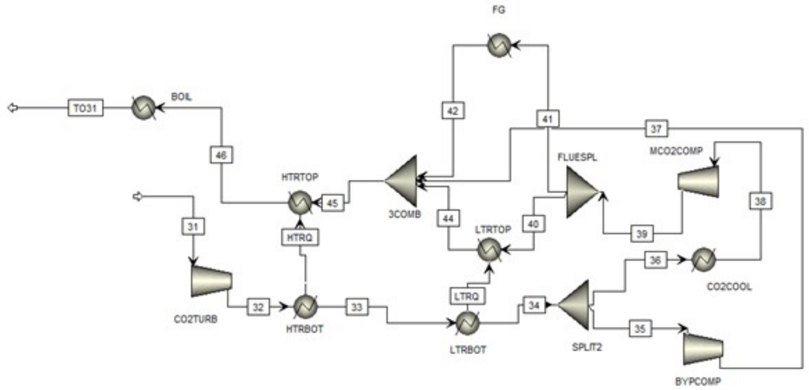

Figure 1: SCO2 Flowsheet simulated in ASPENPlus

The power generated within the cycle is generated at the CO2 Turbine from the 760°C CO2 coming from the boiler. The outlet of the turbine is then used in a high temperature recuperator (HTR) and a low temperature recuperator (LTR). The stream of CO2 is then split into a portion that goes through a bypass compressor and a portion that goes through a cooler. The portion that goes through the cooler then proceeds into the main compressor. The stream from the main compressor flows into a splitter that takes splits the stream to a flue gas cooler and to the LTR. The streams from the LTR, the flue gas cooler, and the bypass compressor meet in a mixer and proceed into the HTR. The flue gas cooler contacts a stream of flue gas that comes from the combustion used to heat the boiler. The heat coming off the flue gas is used to aid in the heating of the sCO2. The HTR feeds directly into the boiler. The boiler is modeled by a heater that heats the HTR stream to the inlet temperature of the turbine. Heat streams titled HTRQ and LTRQ connect HTRBOT with HTRTOP and LTRBOT with LTRTOP, respectively.

The basic premise of the process is that the sCO2 powers the turbine after being heated within the oxy-fired CFB. The compressors prepare the sCO2 to return to the boiler with an appropriate pressure, and the HTR and LTR attempt to salvage heat from the sCO2 exiting the turbine to heat the pressurized stream returning to the boiler. Within the simulation, net power was defined as the sum of the power generated by the turbine subtracting the power used by both compressors. Equation 1 shows the equation that represents net power where the indicated horsepower for each of the blocks is represented by the title of the block.

$$NetPower = -(CO2TURB) - MAINCOMP - BYCOMP \tag{1}$$

The turbine pressure ratio is the next variable considered in the optimization, and Equation 2 represents the turbine pressure ratio as defined by the turbine inlet and outlet pressure.

$$TurbinePressureRatio = \frac{(TurbineOutletPressure)}{(TurbineInletPressure)} \tag{2}$$

In the optimization of this process, a constraint is defined for the net power output, and the optimization seeks to find the turbine pressure ratio and bypass split fraction that minimize the required boiler duty. While other degrees of freedom could be considered, this produced a reasonable case study for comparison.

2.2 Surrogate Modeling of Chemical Process Systems

The optimization portion of the study performed sought to evaluate the feasibility of using a neural network with varied nodes and layers to approximate the process system. Important considerations for the project were to know if ANNs could accurately represent the process over the range of interest, to find if other studies had performed optimization problems using neural networks on process systems, and evaluating the position of this study in the field of ANNs and optimization for chemical engineering applications. Several studies have been performed that confirm the validity of modeling complex thermodynamic data with ANN's. ANN's have successfully modeled refrigerant thermodynamic properties, a power cycle, an absorption process, and a hydroformylation process that will be discussed. Some of the studies have additionally performed optimization surrounding the associated ANN and found successful results.

The optimization of a cryogenic natural gas liquid (NGL) recovery unit was performed using recurring neural networks (RNNs) by Zhu et al. [12]. RNNs are a type of ANN that contains a looping constraint within the hidden layer of the ANN. The study found that the RNN was successfully able to represent complex thermodynamic data of the NGL despite the large amount of chemicals involved in the process.Thus, the findings of the project support the use of ANNs to model chemical data. The objective of the optimization performed was to maximize the profits obtained from the recovery unit. The nature of the optimization problem solved by Zhu et al. [12] creates additional complexities when considering costs along with operating conditions of the system. As a result, the NGL recovery optimization is comparably more complex than the optimization of the sCO2 power generation cycle, which supports the feasibility of solving the sCO2 power generation cycle with ANNs and optimization.

Three studies were performed around refrigerant thermodynamic data modeling through ANNs. Studies by Sözen et al. [8], Chouai et al. [2], and Laugier and Richon [4] found ANNs to be suitable surrogate models for thermodynamic data. Sözen et al. [8] determined ANNs to have distinct advantages of "speed, simplicity, and capacity to learn from examples" when "compared to classical methods" such as equations of state [8]. ANNs are uniquely advantageous compared to equations of state because ANNs do not rely on constrictive functions. ANNs instead rely on being trained on example datapoints and can then accurately portray thermodynamic data for chemicals. The second study of refrigerants conducted by Chouai et al. [2] found the optimal number of nodes and layers through trial and error. The discerning of the optimal number of nodes and layers is especially important for optimization problems which may scale poorly with large ANNs, but is also beneficial for determining the accuracy at which an ANN can model chemical process data. Additionally, the study was able to determine derived properties, such as enthalpy and entropy, from numerical derivatives involving the compressibility factor, temperature, and pressure. The ANN accurately determined derived properties from input data supports thermodynamic data modeling through neural networks. Laugier and Richon [4] were also able to confirm that ANNs can act as accurate surrogate models for thermodynamic data, noting that "the herein numerical approach seems to be a convenient tool for modelling and predicting purposes" associated with thermodynamic property modeling. The literature of all three confirms that ANNs are capable of modeling thermodynamic data as a replacement for equations of state.

Şencan et al. [13] provides a source of dispute for using equations instead of ANNs for accurate property determination. The study focuses on absorption systems and the related modeling through ANNs. Şencan et al. [13] argue that the equations derived from the data may make modeling by ANN unnecessary. However, the large amount of equations required to determine properties creates complexities with optimization problem feasibility. ANNs provide several benefits as previously discussed, and this study will focus on the use of ANNs within an optimization context. Although Şencan et al. [13] mentions that modeling though ANNs may be unnecessary, the study found that ANNs were able to accurately portray thermodynamic data.

The case study of the hydroformylation of n-dodecane performed by Nentwich and Engell [6] yields an example of flowsheet superstructure optimization through surrogate ANN modeling. The study notes that classical methods of equations of state create systems of equations in thermodynamic models that can greatly slow down computations for process systems. The literature also notes that “by the use of the surrogate models, the good predictive properties of advanced thermodynamic models can be exploited in flowsheet optimization of multicomponent multiphase processes without excessive computation times for the optimization runs” Nentwich and Engell [6]. The goal of this study is also flowsheet optimization. Avoiding excessive computation times is a fundamental part of the study. The feasibility of solving optimization problems with large amounts of equations is further refuted. ANNs provide an exceedingly beneficial method of modeling the thermodynamic data for use in optimization computations.

Research performed by Schweidtmann et al. [7] provides an example of neural networks being used to model pertinent chemical data for use in deterministic global process optimization, or DGPO. The chemical data generated through ANNs for the DGPO was used to optimize a simple Rankine cycle for power generation to maximize the net power generated by the system. Three formulations were used for chemical data for the optimization problem. Thermodynamic properties were obtained through either an artificial neural network as functions of any two intensive properties, through the Helmholtz equation of state with additional equality constraints and optimization variables, or through artificial neural networks with the constraint of using the same inputs, constraints, and optimization variables as the Helmholtz equation of state. The primary difference between the research performed in the study and the research performed by Schweidtmann et al. [7] was the variation of the number of nodes and layers to achieve ideal MSE. While the study by Schweidtmann et al. [7] varied the number of nodes without experimentation into layers, this study sought to determine the ideal number of nodes and layers to achieve a specified MSE. Furthermore, while Schweidtmann et al. [7] utilized classical nonlinear activation fucntions, this study investigates the use of ReLU networks which are MILP-representable.

3 Methods

The goal of this work is to demonstrate optimization of chemical process systems using an artificial neural network surrogate. The case study considered is an optimization problem for a super-critical CO2 power generation process. The flowsheet is discussed in the previous section. The goal of the optimization formulation is to seek the turbine pressure ratio and bypass split fraction that minimize the required boiler duty for a particular desired net power output. The optimization formulation considered in the case study can be described as follows:

$$\begin{aligned} \min_{d,r,b,p,x} \; & d && (3)\\ \text{s.t.} \quad & f(d,p,r,b,x) = 0 && (4)\\ & p = p^{spec} && (5)\\ & r^L \leq r \leq r^U && (6)\\ & b^L \leq b \leq b^U && (7)\\ & x^L \leq x \leq x^U && (8) \end{aligned}$$

where d is the required boiler duty, p is the net power produced, r is the turbine pressure ratio, b is the bypass split fraction, and x are the remaining state variables in the problem. The equations $f(\cdot)$ represent the model of the SCO2 flowsheet.

3.1 Model Validation and Data Generation

The model used for data generation was developed as part of the IDAES project (Institute for the Design of Advanced Energy Systems). The IDAES computational framework is an open-source, equation-oriented modeling package based on Python and built upon the Pyomo optimization package.

Prior to data generation the IDAES SCO2 flowsheet was validated against an ASPEN Plus flowsheet developed as part of this project, using specifications from the NETL baseline report for SCO2 power generation [11].

To generate data, simulations were performed over a sweep of values for the net power output, and the turbine pressure ratio. For each simulation, the IDAES model was solved using IPOPT[10]. In addition to these input values, for each of the simulations performed, the required boiler duty and bypass split fraction were recorded. A total of 750 simulations was performed, varying the net power from 600 MW to 650 MW, and the turbine pressure ratio was varied from 0.3 to 0.65. The results of these simulations formed the training and testing data for the ANN. Regarding the implementation, the IDAES framework is completely based on Python, and a Python script was written to loop over all the simulations and record the results to a `csv` file through the Pandas package in Python.

3.2 Network Training

Pytorch is one of several machine learning libraries available to develop neural networks from training data. The two main choices are PyTorch and TensorFlow. PyTorch was selected primarily for the ease of use and effectiveness when training. Another benefit of Pytorch is the presence of large amounts of online resources that guide the development of code. Pytorch is also a Python based machine learning package and it was straighforward to script the training procedure and extract the network parameters for constructing the optimization formulation.

As seen in Section 4, several network sizes were explored to obtain a reasonable trade off between the accuracy of the approximation and the size of the neural network (which impacts the performance of the optimization). For the activation functions, rectifying linear units (ReLU) were selected since transformations exist to represent these networks within mixed-integer linear programming problems. Effective training was found by first using the *RMS* method, followed by the *ADAM* method for a total of $20,000$ iterations. The loss function used was the mean-squared error (MSE), and 70% of the data was used for training with the remaining 30% used for validation. The MSE of both the training and validation data was recorded, and at the end of the training, the maximum absolute error and the maximum relative absolute error were computed and recorded. After training, the structure, weights, and biases of the network can be extracted for use in the optimization formulation.

3.3 Mixed-Integer Representation of the Neural Network Surrogate

Integer programming represents a form of mathematical programming that performs optimization and restricts the variables to hold discrete, integer values. Mixed-integer programming specifically refers to problems that can contain both discrete and continuous variables. Robust mixed-integer programming solvers exist that allow for optimization of problems formatted as mixed-integer equations.

In the study performed, the neural networks used to represent the complex nonlinear sCO2 system can be mathematically represented by the rectifying linear units (ReLU) format. Once the neural network is trained, we can formulate the surrogate as a set of mixed-integer constraints that can be embedded in another optimization problem. The original optimization formulation described in equations (3-8) is modified to include the mixed-integer transformation of the ReLU ANN in place of the nonlinear equations representing the SCO2 flowsheet as follows:

$$\min_{d,r,b,p,x} \; d \tag{9}$$

$$\text{s.t.} \quad (d,b) = ANN(p,r) \tag{10}$$

$$p = p^{spec} \tag{11}$$

$$r^L \leq r \leq r^U \tag{12}$$

$$b^L \leq b \leq b^U \tag{13}$$

$$x^L \leq x \leq x^U \tag{14}$$

The ReLU ANN can then be reformulated into a mixed-integer formulation in similar form to the reformulations detailed by Tjeng et al. [9]. Tjeng provides a detailed example used in the development of the coded ReLU ANN formulation. The structure of the neural network in conjunction with ReLU activation functions provides a means for the ANN to be used to create a mixed-integer programming (MIP) representation that details the forward computation used in the neural network. The neural network is then MIP-representable and can be further used in optimization problems to make process decisions for the SCO2 process.

4 Results

The results of the analysis performed to optimize the sCO2 Brayton Recompression Cycle are defined by both the neural network training as well as the optimization results obtained using the trained neural network within the mixed integer programming formulation of the associated optimization problem.

4.1 Neural Network Size Determination

It is important to determine an appropriate structure for the neural network (i.e., the number of layers and nodes per layer). The network should be large enough to adequately approximate the system, but if the network is too large, then it is subject to over-fitting. Furthermore, smaller network sizes are important to reduce the computational burden of the MILP representation in the optimization problem.

To determine an appropriate network structure, several networks were trained on the data, and compared for size and quality of fit. As indicated above in Section 3, PyTorch was used for all training. Table 1 shows the fit achieved for different neural network structures. The table shows the number of nodes, number of layers, the total nodes, and the mean squared error (MSE) of both the training data and the validation data. The network size that was selected to be used in the optimization was the network with 3 layers with 20 nodes per layer as it is a relatively small network and has low MSE for both the training and testing data.

4.2 Optimization and MIP Results

In this section, the MIP representation is compared with the original IDAES model on an set of optimization problems. The optimization case study seeks to minimize the boiler duty for a particular, specification of net power produced with different bounds on the bypass split fraction. The turbine pressure ratio is bounded to be between 0.3 and 1.0. The lower bound of the bypass split fraction is set to 0.001, 0.01, 0.05, and 0.1 for problem A, B, C, and D respectively, while the upper bound is set to 1.0 for all problems. A value of 642 MW was selected as not to exactly match one of the sample boundaries used when generating

Table 1: Contains Neural Network Training Results Data

# Layers	# Nodes per layer	# Nodes	MSE (training)	MSE (validation)
1	10	10	2.86E-01	2.89E-01
1	20	20	3.26E-02	3.22E-02
1	40	40	3.26E-02	3.22E-02
1	5	5	3.38E-02	3.38E-02
2	10	20	2.03E-02	2.30E-02
2	20	40	2.79E-04	2.54E-04
2	40	80	5.49E-05	7.90E-05
2	5	10	5.27E-03	5.97E-03
3	10	30	1.57E-02	1.67E-02
3	20	60	1.30E-04	1.39E-04
3	40	120	7.12E-06	1.21E-05
3	5	15	3.53E-04	3.94E-04
4	10	40	1.55E-02	1.66E-02
4	20	80	2.78E-05	3.86E-05
4	40	160	7.49E-06	1.37E-05
4	5	20	1.86E-02	1.82E-02

the training data. The *Nonlinear Formulation* uses the existing process model in the IDAES framework, and was solved using Ipopt [10]. The *Mixed-Integer Formulation* uses the MIP formulation of the neural network model with 3 layers of 20 nodes each, and was solved using GLPK [5]. The same objective function and constraints were used for both models. Table 2 shows a comparison of the solutions for both models. As we can see, the MIP formulation shows excellent agreement with the nonlinear model.

Table 2: Comparison of Optimizations with MIP and Nonlinear Formulations

		MIP Formulation			Nonlinear Formulation		
Problem	Net Power (1e9 W)	Turbine P_R	Bypass Frac	Boiler Duty (1e9 W)	Turbine P_R	Bypass Frac	Boiler Duty (1e9 W)
A	0.642	0.5195	0.0010	0.9843	0.5159	0.0010	0.9835
B	0.642	0.4657	0.0100	1.0100	0.4661	0.0100	1.0098
C	0.642	0.3979	0.0500	1.0859	0.3989	0.0500	1.0843
D	0.642	0.3613	0.1000	1.1304	0.3615	0.1000	1.1305

However, the MIP formulation brings some advantages over the nonlinear formulation. The MIP formulation does not require initialization, and it provides a globally optimal solution, albeit for the approximated neural network model. The nonlinear model, on the other hand, provides a guarantee of local optimality only. As such, it may become trapped in locally optimal solutions. To make this point clear, the nonlinear formulations are solved again, with different initialization. First, each of the problems is initialized by fixing the turbine pressure ratio to a value of 0.305 and solving the problem. Then the turbine pressure ratio is released (*unfixed*), and the optimization problem is solved from this initialization. The results in table 3 show that the nonlinear formulation is indeed trapped in a local solution, solving to the lower bound of the turbine pressure ratio for all 4 problems.

Table 3: Locally Optimal Solutions from Nonlinear Formulation with Poor Initialization

Problem	Net Power (1e9 W)	Turbine P_R	Bypass Frac	Boiler Duty (1e9 W)
A	0.642	0.3	0.198	1.146
B	0.642	0.3	0.198	1.146
C	0.642	0.3	0.198	1.146
D	0.642	0.3	0.198	1.146

5 Conclusion

The research performed demonstrated that neural network surrogates can be used to represent complex chemical processes in an optimization context. The neural network surrogate obtained accurate results compared to the nonlinear solution, and the neural network surrogate was able to overcome local solutions in the optimization problem. The presence of locally optimal solutions in the optimization problem was verified through the nonlinear formulation of the process with poor initialization. The research performed provides one use of the trained ANN to optimize the singular power generation cycle, but real benefit would be realized if the computational effort for training was significantly outweighed by the burden of the optimization. For example, if neural networks were trained for individual unit operations while the overall optimization sought to solve a combinatorial problem like flowsheet synthesis, or if the neural network surrogate was used within an optimization under uncertainty problem where the model was included for each scenario.

An important consideration for future work is related to the relatively small size used in this study with only two inputs and two outputs. The scalability of the entire approach should be investigated on larger problems with more inputs and outputs. However, the use of neural network surrogates shows promise for optimization of process energy systems as demonstrated in this work focused on optimization of the indirect super-critical CO2 power generation cycle.

Acknowledgements

The authors would like to thank Michael Bynum for use of the IDAES model of the supercritical CO2 flowsheet model and his advice on selection of optimization variables.

References

[1] Ross Anderson, Joey Huchette, Will Ma, Christian Tjandraatmadja, and Juan Pablo Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, pages 1–37, 2020.

[2] A. Chouai, S. Laugier, and D. Richon. Modeling of thermodynamic properties using neural networks: Application to refrigerants. *Fluid Phase Equilibria*, 199(1):53 – 62, 2002. ISSN 0378-3812. doi: https://doi.org/10.1016/S0378-3812(01)00801-9. URL `http://www.sciencedirect.com/science/article/pii/S0378381201008019`. 2nd international workshop on thermochemical, thermodynamic and transport properties of halogenated hydrocarbons and mixtures.

[3] Bjarne Grimstad and Henrik Andersson. Relu networks as surrogate models in mixed-integer linear programs. *Computers & Chemical Engineering*, 131: 106580, 2019.

[4] S Laugier and D Richon. Use of artificial neural networks for calculating derived thermodynamic quantities from volumetric property data. *Fluid Phase Equilibria*, 210(2):247 – 255, 2003. ISSN 0378-3812. doi: https://doi.org/10.1016/S0378-3812(03)00172-9. URL `http://www.sciencedirect.com/science/article/pii/S0378381203001729`.

[5] A. Makhorin. Glpk (gnu linear programming kit).
Available at `http://www.gnu.org/software/glpk/glpk.html`, 2009.

[6] Corina Nentwich and Sebastian Engell. Application of surrogate models for the optimization and design of chemical processes. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1291–1296. IEEE, 2016.

[7] Artur M. Schweidtmann, Wolfgang R. Huster, Jannik T. Lüthje, and Alexander Mitsos. Deterministic global process optimization: Accurate (single-species) properties via artificial neural networks. *Computers and Chemical Engineering*, 121:67 – 74, 2019. ISSN 0098-1354. doi: https://doi.org/10.1016/j.compchemeng.2018.10.007. URL `http://www.sciencedirect.com/science/article/pii/S009813541830886X`.

[8] Adnan Sözen, Mehmet Özalp, and Erol Arcaklioğlu. Investigation of thermodynamic properties of refrigerant/absorbent couples

using artificial neural networks. *Chemical Engineering and Processing: Process Intensification*, 43(10):1253 – 1264, 2004. ISSN 0255-2701. doi: https://doi.org/10.1016/j.cep.2003.12.008. URL `http://www.sciencedirect.com/science/article/pii/S0255270104000133`.

[9] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming, 2019.

[10] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.

[11] Charles W. White, Walter Shelton, Nathan Weiland, Travis Shultz, John Plunkett, and David Gray. Techno-economic evaluation of utility-scale power plants based on the indirect sco2 brayton cycle - report. Technical Report 0, National Energy Technology Laboratory, 12 2017. URL `https://www.osti.gov/biblio/1490272`.

[12] Wenbo Zhu, Jorge Chebeir, and Jose A Romagnoli. Operation optimization of a cryogenic ngl recovery unit using deep learning based surrogate modeling. *Computers & Chemical Engineering*, 137:106815, 2020.

[13] Arzu Şencan, Kemal A. Yakut, and Soteris A. Kalogirou. Thermodynamic analysis of absorption systems using artificial neural network. *Renewable Energy*, 31(1):29 – 43, 2006. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2005.03.011. URL `http://www.sciencedirect.com/science/article/pii/S0960148105000753`.