Optimization-based Design of Crude Oil Distillation Units using Surrogate Column Models and a Support Vector Machine

Dauda Ibrahim1,*, Megan Jobson1, Jie Li1, Gonzalo Guillén-Gosálbez2

1Centre for Process Integration, School of Chemical Engineering and Analytical Science, University of Manchester, Manchester M13 9PL, UK

2Department of Chemical Engineering, Centre for Process Systems Engineering, Imperial College, South Kensington Campus, London SW7 2AZ, UK

Abstract

This paper presents a novel optimization-based approach for the design of heat-integrated crude oil distillation units, which are widely used in refineries. The methodology presented combines, within a unified framework, surrogate distillation column models based on artificial neural networks, feasibility constraints constructed using a support vector machine, and pinch analysis to maximize heat recovery, in order to optimize the distillation column configuration and its operating conditions. The inputs to the surrogate column model are given by the column structure and operating conditions, while the outputs are related to the column performance. The support vector machine classifier filters infeasible design alternatives from the search space, thus reducing computational time, and ultimately improves the quality of the final solution. The overall optimization problem takes the form of a mixed-integer nonlinear program, which is solved by a genetic algorithm that seeks the design and operating variables values that minimize the total annualized cost. The capabilities of the proposed approach are illustrated using an industrially–relevant case study. Numerical results show that promising design alternatives can be obtained using the proposed method. The approach can help engineers to design and operate petroleum

* Corresponding author
E-mail address: dauda.ibrahim@manchester.ac.uk; didauda@yahoo.com
refineries optimally, where these are expected to continue to play a major role in the energy mix for some years.

Keywords: Process design, atmospheric distillation unit, heat integration, genetic algorithm, artificial neural network

1 Introduction

1.1 Crude oil distillation: technical background

Distillation is the most widely used separation technique in the chemical and petroleum industries. Petroleum refining starts with crude oil distillation, in which the entire crude oil feedstock undergoes initial separation to produce intermediate products, such as light naphtha, heavy naphtha, light distillate, heavy distillate, and atmospheric residue (see Figure 1). These cuts are further enhanced and blended into marketable products (e.g. gasoline, kerosene, jet fuel, diesel, bunker fuel and fuel oil, etc.) that are supplied to the global energy market.

Crude oil distillation is a capital- and energy-intensive process and is the largest consumer of energy in petroleum refineries (Gu et al., 2014). The energy consumed in crude oil distillation is equivalent to 1 to 2% of the total crude oil being processed (Szklo and Schaeffer, 2007). Consequently, it contributes significantly to the overall refinery CO₂ emissions. Heat integration is typically applied to enhance energy efficiency in crude oil distillation by recovering heat from ‘hot’ streams requiring cooling to ‘cold’ streams requiring heating, thus reducing demand for fired heating, and thus both greenhouse gas emissions and operating cost.
Figure 1 illustrates a typical crude oil distillation system, which comprises a crude oil distillation unit, a heat recovery network (preheat train) and a fired heater (furnace) where the crude oil feedstock is heated and partially vaporized. The crude oil distillation unit is equipped with side-strippers and pump-arounds. Side-strippers are utilized to remove light components from side draws using stripping steam or reboilers, while pump-around loops provide internal reflux and create heat recovery opportunities by cooling and returning liquid streams withdrawn from the column. The crude oil distillation unit is strongly interlinked with the associated heat recovery network through pump-arounds, the condenser and product coolers. Changes in the design and operation of the crude oil distillation unit, therefore, affect the design and operation of the heat exchanger network and fired heater.

A new ('grassroots') design of a crude oil distillation unit aims to determine the optimal values of the design degrees of freedom, which include structural variables and operating conditions, while taking into account the complex interactions between the column and the
heat recovery network. The structural variables include the locations of the feed tray and of pump-around and side-stripper draw streams, and the number of trays in each section of the column. Operating conditions include the feed inlet temperature, pump-around duties and temperature drop, stripping steam flow rates and reflux ratio. The large numbers of degrees of freedom, complex interactions between individual units, and the need to design the column while accounting for heat recovery makes the design of crude oil distillation units highly challenging.

1.2 Crude oil distillation column modelling

Distillation column models may be employed to optimize both the column structure and its operating conditions with respect to various performance criteria. These models should be sufficiently realistic to provide meaningful solutions and robust enough to converge over a wide optimization search space.

Existing design methods (Bagajewicz and Ji, 2001; Liebmann et al., 1998) applied a sequential approach that combines rigorous simulation models embedded in commercial process simulators, and pinch analysis to support the search for an energy efficient column structure and its operating conditions. However, these approaches require trial and error before arriving at the final design, and the trade-off between capital and energy cost has not been considered. To overcome these limitations, an approach that combines rigorous tray-by-tray modelling of the distillation column and pinch analysis in an optimization framework has been developed (Ibrahim et al., 2017a). In this approach, a superstructure of the crude oil distillation column (comprising many design alternatives) is built in Aspen HYSYS, while the optimization is carried out using an external solver in MatLab. The approach takes advantage of the physical property and thermodynamic models, as well as the crude oil characterization and column hydraulic models available in the process simulator, to generate an accurate estimate of the crude oil distillation unit performance. Optimization variables include numbers of trays in column sections and operating conditions.
Despite the fact that rigorous models are versatile and produce accurate estimates of the distillation column performance, their downside is that they require good initial points to converge to a feasible solution. Furthermore, from the optimization point of view, incorporating a rigorous tray-by-tray column model in an optimization framework so as to design the complex crude oil distillation unit is computationally intensive. To overcome these limitations, this work proposes a new strategy for the design and optimization of crude oil distillation units using surrogate column models. As will be seen in Section 4 of this article, the surrogate column model developed herein leads to significant reduction in computational time without sacrificing the model accuracy.

1.3 Surrogate modelling of crude oil distillation columns

In recent years, various regression and fitting techniques have been used to create surrogate models using different mathematical techniques, including polynomial regressions, artificial neural networks, and support vector regressions. Liau et al. (2004) and Motlaghi et al. (2008) applied artificial neural networks to build surrogate models of the crude oil distillation column using data collected from an existing plant. The models were used to perform operational optimization. Liau et al. (2004) focused on improving the yield of kerosene, diesel and atmospheric gas oil, while Motlaghi et al. (2008) optimized the flow rate of products according to their market values.

Yao and Chu (2012) developed a surrogate model of the crude oil distillation unit using the concept of support vector regressions. The model was implemented in a framework to optimize profit by varying operating variables. Gueddar and Dua (2012) applied artificial neural networks to construct a reduced model of the crude oil distillation unit that is suitable for refinery-wide optimization. The model inputs include crude oil properties (e.g. true boiling point) and flow rate, while the outputs consist of product yield and their properties. A deterministic optimization is used to search for the best inputs that improves energy efficiency. López C. et al. (2013) applied a polynomial function to build models of a crude oil distillation system. The models, together with energy balances representing the heat exchanger network, were implemented in a framework to maximize net profit.
Ochoa-Estopier and Jobson (2015a,b) applied data from multiple rigorous simulations to build a surrogate model of the crude oil distillation system using artificial neural networks. The column models, together with a heat exchanger network model, were implemented in an operational optimization framework to improve net profit while fulfilling practical constraints. More recently, Osuolale and Zhang (2017) applied bootstrap artificial neural networks to build a model of a crude oil distillation system, consisting of pre-fractionator, atmospheric column, and vacuum column. Sequential quadratic programming is used to optimize exergy efficiency by varying relevant operating conditions, where heat recovery opportunities are not explicitly considered.

1.4 Scope and objectives

The surrogate models for crude oil distillation columns presented so far assume that the design variables are fixed. Furthermore, these methodologies typically deal with continuous rather than discrete variables. This work extends the use of surrogate models to account for discrete design variables representing the number of trays in each column section (which cannot be handled via the approaches mentioned above). This work uses artificial neural networks, as these have been shown to provide simple, accurate and robust simulation models.

This work proposes a new methodology for the design of heat-integrated crude oil distillation units that implement a surrogate model based on artificial neural networks. Unlike previous work applying surrogate models to represent a crude oil distillation unit (Ochoa-Estopier and Jobson, 2015a,b), the proposed approach takes into account both discrete and continuous variables. The artificial neural network is constructed using ‘samples‘ that are results of multiple rigorous simulations.

Due to the highly combinatorial nature of the data sampling, not all samples within the search space are likely to lead to a converged simulation (i.e. a ‘feasible’ design) using the rigorous model. In addition, numerical issues, e.g. poor initialization, can lead to non-convergence of rigorous simulations. In this work, a feasibility constraint is constructed using a support vector machine (Vapnik, 1995): the resulting classification model helps to
restrict the search space to the region of feasible designs, thus avoiding the need to search in the region of infeasible designs during optimization. Restricting the search in this way increases the likelihood that the optimal design will be feasible when modelled rigorously. Jobson et al. (2017) and Ibrahim et al. (2017b) provide fuller discussions of feasibility constraints in the context of surrogate modelling of crude oil distillation systems.

In this work, the artificial neural network column model and feasibility constraint constructed using a support vector machine are implemented in an optimization framework, in which a genetic algorithm is used to search for column structural variables and operating conditions to minimize total annualized cost. Heat integration is taken into account using pinch analysis, without considering the detailed heat exchanger network structure. Figure 2 summarizes the overall design framework.

![Figure 2 – Overview of design framework](image-url)
The remainder of this paper is organized as follows. Section 2 presents an overview of the modeling tools adopted in this work, including the artificial neural network and the support vector machine. In Section 3, a detailed stepwise approach for modeling the complex crude oil distillation unit is developed. Section 4 presents the proposed mathematical formulation of the design problem together with a solution procedure. Section 5 illustrates and reviews the capabilities of the novel design methodology using a case study. Lastly, conclusions are presented in Section 6.

2 General background on surrogate modeling and support vector machine

The modeling tools applied to model the crude oil distillation column include an artificial neural network and a support vector machine. As mentioned in Section 1, the artificial neural network is used to build a surrogate model of the crude oil distillation column that is subsequently implemented in an optimization framework, while the support vector machine filters out infeasible design alternatives during optimization, thus increasing the likelihood that the design alternatives proposed by the genetic algorithm will be feasible. A detailed description of the two modeling tools follows.

2.1 Artificial neural networks

Artificial neural networks (ANNs) are computational data-driven modeling systems that can represent complex nonlinear relationships between process input-output data points. The network consists of highly interconnected simple elements called neurons. The neurons are arranged in at least two or more layers as shown in Figure 3.

Various types of artificial neural network architectures (Basheer and Hajmeer, 2000) have been applied to model process systems of different levels of complexity (Henao and Maravelias, 2010; Gueddar and Dua, 2011; Nuchitprasittichai and Cremaschi, 2013; Ochoa-Estopier and Jobson, 2015). A multi-layer feedforward network is the most widely accepted architecture due to its mathematical simplicity. This feature makes it suitable for implementation in an optimization framework, where the reduction of the mathematical complexity is an important factor (Nuchitprasittichai and Cremaschi, 2012). Figure 3 shows
a typical multi-layer feed forward network comprising two layers, namely, a hidden layer and an output layer. The hidden layer may consist of one or more sub-layers; one sub-layer is commonplace. The literature (Henao and Maravelias, 2010) has shown that multi-layer feedforward networks with one hidden layer are capable of approximating arbitrary multivariate functions with a finite number of discontinuities; these are features of the problem of interest.

**Figure 3 – Schematic representation of a multi-layer feed forward neural network (adapted from Beale et al. (2015))**

In Figure 3, the superscript and subscript denote the indices of the network layer and neuron respectively. Mathematically, the multi-layer feedforward network in Figure 3 can be formulated as (Beale et al., 2015)

$$ y = f^2(W^2 f^1(W^1 y^0 + b^1) + b^2) $$

(1)

where $y^0$ and $y$ represent the vectors of inputs (independent) and outputs (dependent) variables; $b$ denotes the vector of biases and $W$ denotes the matrix of weights. $f^1$ and $f^2$ are transfer functions of the hidden and output layers, respectively. In this case, the hidden and
output layers are represented by a sigmoid function and a linear function, respectively (Beale et al., 2015).

The multi-layer feedforward network is trained to input-output data points using training algorithms. Different training algorithms are available (Beale et al., 2015). Backpropagation is the most commonly used method, which is the one used in this work. Backpropagation consists of two steps. First the prediction error, e.g., mean square error, is computed using fixed values of weights and biases. Second, the weights and biases are adjusted to minimize the prediction error. The objective function used in most training algorithm can be defined as (Beale et al., 2015):

\[ MSE = \sum_{i=1}^{N} \frac{(t_i - y_i)^2}{N} \]  

(2)

where \( MSE \) denotes the mean square error, \( N \) is the total number of sample points, \( t \) and \( y \) denote the target output and predicted output, respectively. Section 3.2 presents further detail on the development of an artificial neural network model for a crude oil distillation column.

2.2 Support vector machine

A support vector machine (SVM) is a widely used statistical technique for regression analysis (Ławryńczuk, 2016; Zaidi, 2012) and data classification (De Boves Harrington, 2015; Oliynyk et al., 2016). Vapnik (1995) provides a detailed description of the fundamentals of support vector machines. Here, the focus is on application of support vector machines for binary classification of samples (corresponding to column designs) as either feasible or infeasible. The main idea of support vector machines is to define an optimal hyperplane that separates two or more classes of data points. For a given data set consisting of two separable classes, the optimal hyperplane is the one with the largest distance to the nearest data point, where distance is the Euclidian distance in the \( n \)-dimensional space.

Figure 4 illustrates a hyperplane separating two classes of data: feasible samples denoted by squares and infeasible samples denoted by circles. In this work, samples leading to a
converged simulation using the rigorous crude distillation column model built in Aspen HYSYS are termed feasible, while those leading to un converged simulations are termed infeasible.

Figure 4 – Schematic representation of a support vector machine showing an optimal hyperplane for: (a) perfectly separable data sets, and (b) non-separable data sets in a two-dimensional space (adapted from Mahe et al. (2005)). Eq. (4) shows the detailed formulation for finding the optimal hyperplane.

In Figure 4, the optimal hyperplane can be determined as follows: Let \((x_i, y_i)\), where \(i = 1, ..., n\), represent the set of training sample points, where \(x \in \mathbb{R}^n\) are decision variables (e.g. feed inlet temperature, pump-around duties and temperature drops, stripping steam flow rates and reflux ratio) and the corresponding sample class \(y \in \{+1, -1\}\) represents whether the decision variables values lead to a feasible or infeasible solution, where +1 denotes a feasible sample and −1 an infeasible sample. The support vector machine classifier can be defined by (Vapnik, 1995)

\[
y(x) = sign(w \cdot x + b)
\]  

where \(w \in \mathbb{R}^n\) is the normal vector of the classification hyper plane, while \(b \in \mathbb{R}\) is the bias.
Given a new instance $x$, Eq. (3) allows classification of the sample as either feasible (positive sign) or infeasible. Before Eq. (3) is applied to classify new samples, the weight vector ($w$) and bias ($b$) need to be fitted to the training sample points. This process is called ‘learning,’ and is usually carried out by solving Problem P1 using optimization (Vapnik, 1995). The procedure seeks the hyperplane with the largest separation margin between the two classes of data.

$$[P1] \min_{w,b,\xi} \frac{1}{2}w^T w + C \sum_{i=1}^{n} \xi_i$$

subject to

$$y_i(w^T \cdot x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1, ..., n$$

where $n$ represents the number of training data points; $\xi_i$ are slack variables to account for misclassification of data points (see Figure 4b); $C$ is a scalar penalty constant to scale constraint violations.

Eq. (3) represents a linear formulation of the support vector machine, which is mostly applied to linearly separable data points (Vapnik, 1995). Other formulations for non-linearly separable data points are also available, such as polynomial functions, radial base functions and multi-layer perceptron (Vapnik, 1995).

Section 3.3 presents the detailed application of the support vector machine, Eq. (3), to construct a feasibility constraint to be implemented in an optimization framework with the aim of filtering out infeasible designs during the optimization task.

### 3 Modeling and solution procedure applied to a crude oil distillation column

The proposed surrogate modeling method comprises three main steps, namely, data generation (known as sampling), regression for surrogate column modeling using the concept of artificial neural networks, and construction of feasibility constraints using a support vector machine.
3.1 Data generation

Data generation is an essential step in the construction of surrogate models, since the performance of the surrogate model strongly depends on the quality of the data used in the training stage. As discussed in the introduction, this work makes use of multiple rigorous simulations to generate sample points used for training the surrogate model.

Data generation using multiple rigorous simulation consists of three major steps: selection of relevant inputs (independent variables) and outputs (dependent variables), generation of random sample points for each input variable, and lastly, for each set of input data, a rigorous simulation of the crude oil distillation is carried out to obtain the corresponding set of output variables.

In this work, the input variables selected include variables that can be adjusted to improve the crude distillation column performance. These variables include both structural (number of trays in each column section) and operational (feed inlet temperature, pump-around duties, and temperature drops, stripping steam flow rates and reflux ratio) degrees of freedom. The output variables selected include those that represent product quality (e.g., ASTM D86 T5 and T95 boiling temperatures of each product), product flow rates, supply and target temperatures of streams requiring heating and cooling and corresponding enthalpy changes, and diameters of each column section. These variables allow calculation of the objective function and checking whether the product constraints are satisfied.

Next, a sampling technique is applied to generate random samples for each input variable. Several sampling techniques are available, for example, Monte Carlo sampling, Hammersley sequence and Latin hypercube sampling (Subramanyan et al., 2011). Without loss of generality, Latin hypercube sampling is used in this work, since it has been shown to produce accurate statistical estimates of a probability distribution (Subramanyan et al., 2011). Latin hypercube sampling divides the input variable space into intervals, where samples are created randomly from each interval. The sampling method thus guarantees well-distributed samples.
After generating the samples, multiple rigorous simulations are carried out by simulating each such sample of the independent variables using a rigorous distillation column model implemented in a process simulation package. In this work, the multiple rigorous simulations are facilitated via an interface between Aspen HYSYS v8.6 and MatLab R2015a (AspenTech, 2011). Results of the simulations of the crude oil distillation in Aspen HYSYS v8.6 are recorded for each sample set. Samples are labeled as feasible if the simulation converges and infeasible if it does not. It is desirable to remove infeasible samples from the solution space to prevent the optimization algorithm from searching for and converging to infeasible solutions. Section 3.2 describes how an artificial neural network model representing the distillation column is constructed using the converged samples and Sections 3.3 explains how the whole sample set is used to construct a support vector machine.

3.2 Creating a surrogate distillation column model

This work uses an artificial neural network to create a surrogate model of the crude oil distillation column. The network correlates input and output data representing independent and dependent variables of the crude oil distillation column. A major advantage of artificial neural networks over other statistical techniques is the ability to correlate multiple inputs to multiple outputs, leading to compact models that can be implemented in an optimization environment with ease. To take advantage of this important feature, the artificial neural networks representing the crude distillation column are built by correlating all the independent variables to groups of specific dependent variables. In this work, the crude oil distillation unit consists of 18 independent variables and 46 dependent variables. The dependent variables are divided into seven groups: ASTM T5 and T95 boiling temperatures, product flow rates, supply and target temperatures, enthalpy changes and diameters of each column section. Then seven ANN models are constructed to correlate the independent variables to each group of dependent variables.

This work uses the Artificial Neural Network Tool Box in MatLab R2015a to construct, validate and test the surrogate crude oil distillation column model. A multi-layer feed-forward network with one input layer, one output layer, and one hidden layer is
The hidden layer contains 10 neurons, while the number of neurons in the output layer depends on output variables associated with the network (for example, the network constructed for product flow rates consists of 18 inputs and five outputs, representing the flow rates of light naphtha, heavy naphtha, light distillate, heavy distillate and residue). The former uses a sigmoid function, and the latter uses a linear transfer function, in line with common practice (Beale et al., 2015). The feasible sample points generated from the multiple rigorous simulations are randomly divided into a training set (70%), a validation set (15%) and a testing set (15%). The ANN model is first constructed using the points in the training set. The points in the validation set are employed during training to avoid model overfitting. The testing set is used for checking the performance of the model. The coefficient of determination is used to assess the performance of the seven networks. The coefficient of determination is a dimensionless quantity, where the typical range is 0 to 1, and it indicates the fraction of the variability of the dependent variable that is explained by the surrogate model (Diamond and Jefferies, 2001). The larger the coefficient of determination (close to one), the better the fitting and vice versa. In building the artificial neural network, the size of the hidden layer and number of neurons can be varied until the desired performance is achieved. While a methodology to optimize the number of layers and neurons has been developed by Dua (2010), implementing this methodology can be time consuming and challenging. In this work, several neural networks were constructed; trial and error was used to identify how many layers and neurons are needed for a good performance for optimization-based design of the crude oil distillation unit.

3.3 Feasibility constraints

This work uses feasibility constraints formulated as a support vector machine to enhance the optimization task. The support vector machine constructed here has a third-order polynomial function with outputs of +1 and -1. Positive one represents converged samples (feasible designs), and negative one denotes unconverged samples (infeasible designs). All the samples generated via multiple rigorous simulations, consisting of converged and unconverged samples, are used to build the support vector machine. The data set is split into a training set (75%) and a validation set (25%). The function fitcsvm, which is implemented in MatLab 2015a, is applied to train and validate the support vector machine.
4 Framework for the design of crude oil distillation units

In this section, the artificial neural networks representing the crude oil distillation unit and the feasibility constraint developed using support vector machine are implemented a framework to design the unit. First, the mathematical formulation of the optimization problem is presented, followed by the approach proposed to solve the optimization problem.

4.1 Mathematical formulation

The design of crude oil distillation column based on surrogate models (artificial neural network and support vector machine) can be formulated as a mixed integer nonlinear programming problem (P1) as follows:

\[
\begin{align*}
\text{min} & \quad f(x_D, x_S, x_O) \\
\text{s.t.} & \quad h(x_D, x_S, x_O) = 0 \\
& \quad g(x_D, x_S, x_O) \leq 0 \\
& \quad x_D \in X_D, x_S \in X_S, x_O \in X_O
\end{align*}
\]  

where \( f \) is the objective function, \( h \) is the set of equality constraints represented by a surrogate model, \( g \) is the set of inequality constraints; while \( X_D, X_S \) and \( X_O \) are the feasible sets of the variables, namely: \( x_D, x_S \) and \( x_O \), which represent dependent, structural and operational variables, respectively. As will be discussed later in more detail, this model is solved using a genetic algorithm.

In this work, the surrogate model comprises seven artificial neural networks. Each neural network predicts a set of specific dependent variables of the crude oil distillation column. In addition to the artificial neural network, a support vector machine is also trained to predict the feasibility of the crude oil distillation column. Hence, the equality constraints in Problem P2 can be more specifically represented as follows
\[ h_1: \quad [T5_i] = ANN_1(x_s, x_o) \]
\[ h_2: \quad [T95_i] = ANN_2(x_s, x_o) \]
\[ h_3: \quad [F_i] = ANN_3(x_s, x_o) \]
\[ h_4: \quad [D_j] = ANN_4(x_s, x_o) \]
\[ h_5: \quad [E_k] = ANN_5(x_s, x_o) \]
\[ h_6: \quad [T5_l] = ANN_6(x_s, x_o) \]
\[ h_7: \quad [TT_l] = ANN_7(x_s, x_o) \]
\[ h_8: \quad [CC] = SVM(x_s, x_o) \]
\[ h_9: \quad CC = 1 \]

where \( T5 \) and \( T95 \) represent the ASTM D86 boiling temperatures of product \( i \) at 5\% and 95\% vaporization, \( F \) is the flow rate of product \( i \), \( D \) is the diameter of section \( j \), \( E \) is the enthalpy change of stream \( k \), \( TS \) and \( TT \) are the supply and target temperatures of stream \( l \), \( CC \) represents the predicted convergence criterion (+1 for converged column and -1 otherwise), \( x_s \) and \( x_o \) are structural and operating variables (independent variables) and lastly, \( ANN \) and \( SVM \) represent artificial neural network and support vector machine functions respectively.

Inequality constraints in Problem P2 define bounds imposed on independent variables and product quality specifications. These can be represented as follows:

\[ g_1: \quad lb_i \leq N_i \leq ub_i \quad i = 1, 2, 3, ..., N_{section} \]
\[ g_2: \quad lb_j \leq Q_{PA,j} \leq ub_j \quad j = 1, 2, 3 \]
\[ g_3: \quad lb_j \leq \Delta T_{PA,j} \leq ub_j \quad j = 1, 2, 3 \]
\[ g_4: \quad lb_k \leq F_{s,k} \leq ub_k \quad k = 1, 2 \]
\[ g_5: \quad lb \leq R \leq ub \]
\[ g_6: \quad lb \leq T_F \leq ub \]
\[ g_7: \quad lb_l \leq T5_l \leq ub_l \quad l = 1, 2, ..., N_{product} \]
\[ g_8: \quad lb_l \leq T95_l \leq ub_l \quad l = 1, 2, ..., N_{product} \]

where \( N_i \) is the number of active trays in column section \( i \); \( Q_{PA,j} \) and \( \Delta T_{PA,j} \) are the duty and temperature drop of pump-around \( j \); \( F_{s,k} \) is the flow rate of stream \( k \); \( R \) is the overhead reflux ratio; \( T_F \) is the feed inlet temperature; \( T5_l \) and \( T95_l \) are the boiling temperatures of product \( l \) at 5\% and 95\% vaporization.
The objective function \( f(x_D, x_S, x_O) \) employed in this work is the total annualized cost, since the aim is to identify the design alternative that minimizes both capital expenditures and operating costs of the crude oil distillation unit. Other types of objective function such as net profit, net present value, energy cost, and CO₂ emissions could be used, depending on the purpose of the design.

The total annualized cost is the sum of the annualized capital cost \( (ACC) \) and the total operating costs \( (OC) \) of the crude oil distillation unit. The annualized capital cost is the sum of the installed cost of the column shells \( (S_c) \) and of the trays within the column, \( (T_c) \). The column shell and tray costs are estimated using the correlations proposed by Guthrie (1969) (see Appendix A: Supplementary data). The annualization factor described by Smith (2005) is applied to split the total column cost across the entire plant life at a specific interest rate.

\[
ACC = (S_c + T_c) \cdot A_f
\]

\[
A_f = \frac{i(1 + i)^t}{(1 + i)^t - 1}
\]

where \( i \) is the interest rate and \( t \) is the plant life.

The operating cost of the crude oil distillation unit is dominated by the costs of the fired heating, typically using fuel oil or natural gas; of steam for stripping indirect heating and of cold utilities. In this work, the utility demand is estimated using pinch analysis, i.e., using composite curves to determine the minimum utility demand (Smith, 2005). The pinch calculation is carried out using an open source MatLab code (Morandin, 2014). In this way, heat recovery is incorporated during the column optimization without the need to design explicitly the associated heat exchanger network (HEN). Hence, the operating cost \( (OC) \) is evaluated using Eq. (8):

\[
OC = \sum_{i=1}^{n} ST_i \cdot C_{ST,i} + HU \cdot C_{HU} + CU \cdot C_{CU} \quad n = 2
\]

where \( C_{ST}, C_{HU} \) and \( C_{CU} \) are the unit costs of stripping steam and of the hot and cold utilities respectively; \( ST \) is the flow rate of steam; \( HU \) and \( CU \) are the minimum demand for hot and cold utilities, respectively, while \( n \) represents the number of stripping steam streams associated with the column.
4.2 Optimization framework

In Problem P2, the inequality constraints, \(g_1\) to \(g_6\) are bounds on structural and operational variables, while \(g_7\) and \(g_8\) represent constraints on product quality in terms of ASTM D86 boiling temperature (T5 and T95). For the sake of simplicity and to facilitate the search for the optimal solution, the latter constraints are included into the objective function as a penalty function (Edgar et al., 2001); the new formulation, P3, is:

\[
(P3) \quad \min_{x_D, x_S, x_O} f(x_D, x_S, x_O) + \left[ \Pi \sum_{i=1}^{n} \max(0, (g_i)) \right]^2
\]

\[s.t. \quad h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, g_1, g_2, g_3, g_4, g_5, g_6\]

where \(g_i\) denotes the inequality constraints \(g_7\) and \(g_8\); \(\Pi\) is a scalar parameter that scales the magnitude of the violation of constraints, and hence ensures that the product quality specifications are maintained during the optimization. The magnitude of the scalar parameter may impact on the final optimal solution. A small value may allow some constraints to be violated, leading to infeasible solutions, while large values impose constraints very stringently. Computational experience suggests that a value of a similar magnitude as the objective function yields good results. In this work, \(\Pi\) is \(10^6\).

Figure 5 presents the strategy proposed to optimize Problem P3. The proposed approach integrates the artificial neural networks, the support vector machine, the heat recovery model (pinch analysis) and the cost models into a unified framework. The framework applies a stochastic optimization algorithm (i.e., a genetic algorithm) to search for the best column structure and operating conditions, i.e., that minimize the sum of annualized capital costs and annual operating costs.

The surrogate models employed in this work are easier to optimize than rigorous models (e.g., as used by Ibrahim et al., 2017a). Nevertheless, the sigmoid function used in this work to construct the surrogate model and the objective function (including non-linear capital cost correlations) together give rise to a nonlinear, non-convex optimization problem. Gradient-based searches are unlikely to locate the global optimum. Therefore, a stochastic
optimization method, namely a genetic algorithm (GA), is used. The GA can search the solution space more effectively, even though it still cannot guarantee convergence to the global optimum.

**Figure 5 – Optimization framework for the design of crude oil distillation unit**

The framework shown in Figure 5 is implemented in MatLab. A genetic algorithm available in the global optimization toolbox in MatLab (`ga`) is employed to carry out the search for cost-effective design alternative. In each iteration, the genetic algorithm proposes a column structure and its operating conditions; the proposed inputs are then checked by the support vector machine to confirm whether the design is likely to be feasible. Inputs identified as feasible are then used by the artificial neural network model to simulate the proposed design solution and predict its performance. Among the model outputs are process stream
data used in the heat recovery model to determine utility targets and other information required to calculate the value of the objective function. Inequality constraints are applied to check whether product qualities and flow rates specifications are met. In cases that the support vector machine identifies solutions generated by the optimisation algorithm as likely to be infeasible, the corresponding designs are not simulated by the artificial neural network model. Instead, a penalty is added to the objective function. Attributing a poor performance to potentially infeasible solutions helps to train the genetic algorithm not to propose similar design alternatives again. In this way, the search focuses on the feasible region, increasing the likelihood that the optimization will converge to a feasible design option. Furthermore, the computational time is reduced, since potentially infeasible solutions are removed from the search space. The next section illustrates how the methodology is applied to a relevant case study.

5 Case study

5.1 Problem description

This section demonstrates the capabilities of the novel design approach via the design of a crude oil distillation system. The crude oil distillation unit that separates 100,000 bbl day\(^{-1}\) (662.4 m\(^3\) h\(^{-1}\)) of Venezuelan Tia Juana light crude oil (Watkins, 1979) into five products, namely, light naphtha (LN), heavy naphtha (HN), light distillate (LD), heavy distillate (HD) and residue (RES). The initial design of the feasible column is obtained from Chen (2008); the unit comprises a main column with three pump-arounds and three side-strippers. The main column has five sections with 5, 9, 10, 8 and 9 sieve trays respectively. The HD, LD and HN side-strippers have 5, 7 and 6 sieve trays, respectively (Chen, 2008), presented also in Ibrahim et al. (2017a).

Given data include operating conditions, product quality metrics (in terms of ASTM 5 % and 95 % boiling temperatures) and product flow rates. The column operating pressure is taken to be uniform and equal to 2.5 bar. The economic calculation assumes an interest rate of 5%, a plant life of 20 years and 8700 operating hours per year. The costs of utilities, which include stripping steam (260 °C, 4.5 bar), fired heating (1500–800 °C) and cooling water (supplied at 10°C and returned at 40°C) are $0.14 \text{ kmol}^{-1}, $150 \text{ kW}^{-1} \text{ a}^{-1} \text{ and 5.25 $ kW}^{-1} \text{ a}^{-1}$
respectively. A minimum approach temperature of 30°C in all streams is used to calculate the minimum utility requirements. Details of the crude oil assay, initial operating conditions, product quality specifications and flow rates (initial values) of the base case are presented in Tables S1 to S5 of the supplementary material.

### 5.2 Surrogate crude oil distillation column model

The surrogate modeling approach proposed in Section 3 is applied to model the crude oil distillation unit. First, a rigorous simulation model of the crude oil distillation unit is implemented in Aspen HYSYS using Peng-Robinson as thermodynamic property package. Sensitivity analyses are carried out to identify suitable bounds for each independent variable, as shown in Table 1.

#### Table 1 – Bounds on trays in column sections and operating conditions

<table>
<thead>
<tr>
<th>Variables</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>Initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of trays</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section 1</td>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Section 2</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Section 3</td>
<td>8</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>Section 4</td>
<td>6</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Section 5</td>
<td>7</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Section 6</td>
<td>3</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Section 7</td>
<td>5</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Section 8</td>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td><strong>Operating condition</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA 1 duty (MW)</td>
<td>8.40</td>
<td>14.00</td>
<td>11.2</td>
</tr>
<tr>
<td>PA 2 duty (MW)</td>
<td>13.42</td>
<td>22.36</td>
<td>17.89</td>
</tr>
<tr>
<td>PA 3 duty (MW)</td>
<td>9.63</td>
<td>16.05</td>
<td>12.84</td>
</tr>
<tr>
<td>PA 1 DT (°C)</td>
<td>10</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>PA 2 DT (°C)</td>
<td>40</td>
<td>60</td>
<td>50</td>
</tr>
<tr>
<td>PA 3 DT (°C)</td>
<td>20</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>Main stripping steam (kmol h⁻¹)</td>
<td>900</td>
<td>1500</td>
<td>1200</td>
</tr>
<tr>
<td>HD stripping steam (kmol h⁻¹)</td>
<td>188</td>
<td>313</td>
<td>250</td>
</tr>
<tr>
<td>Feed temperature (°C)</td>
<td>340</td>
<td>375</td>
<td>365</td>
</tr>
<tr>
<td>Reflux ratio</td>
<td>3.17</td>
<td>6.17</td>
<td>4.17</td>
</tr>
</tbody>
</table>

Next, the Latin Hypercube sampling method is applied to generate 7000 samples, each consisting of different combinations of the independent variables within the bounded region. Through an interface established between HYSYS and MatLab, all the samples are
simulated on the rigorous column model developed in HYSYS. Out of these samples, 59% (4130) simulations converged; for the remaining 41% (2870), the simulations did not converge. The sampling is carried out on a HP desktop PC with Intel(R) Core i5 processor running at 3.2 GHz, and 8 GB of RAM. It took around 1.5 h to generate the set of 7000 samples.

Figure 6 shows the performance results of the seven artificial neural networks representing the crude oil distillation units. The coefficient of determination for all artificial neural network models is approximately 0.9999. The values of the coefficient of determination indicate that the models built are able to explain 99.99% of the variance of the outputs, which indicates the goodness of fit.

**Figure 6** Parity plots showing predictions of artificial neural network versus rigorous model.
To gain further confidence in the ANN models, additional statistical tests are carried out to further assess the model accuracy. The performance criteria are average absolute error and average relative error, calculated for temperature related variables and other variables (product flow rate, enthalpy change and diameter in column sections) respectively. These tests are carried out using 100 converged samples generated using the approach presented in Section 3.1. Table 2 shows the average error of the predictions of the artificial neural network column model compared to the rigorous simulation model results. The seven artificial neural network models have 46 outputs in total: five each for T5 and T95 boiling temperatures; five product flow rates; 11 supply temperatures; three target temperatures; eight duties for the condenser, three pump-arounds, two reboilers and five product coolers; the eight diameters, one for each column section.

As shown in Table 2, the largest average absolute error in temperature-related variables is less than 0.5°C and average relative error for all other variables is less than 3%. The largest deviation is observed for the section diameter in the HN and HD side strippers, yet the associated error is small and should have a marginal impact on the column capital cost calculations. These results confirm the effectiveness and accuracy of the artificial neural network column model.
Table 2 – Validation results for the ANN column model

<table>
<thead>
<tr>
<th>Variables</th>
<th>Average absolute error (°C)</th>
<th>Variables</th>
<th>Average relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product quality</strong></td>
<td></td>
<td><strong>Product flow rates</strong></td>
<td></td>
</tr>
<tr>
<td>LN T5%</td>
<td>0.098</td>
<td>LN</td>
<td>0.169</td>
</tr>
<tr>
<td>HN T5%</td>
<td>0.250</td>
<td>HN</td>
<td>0.368</td>
</tr>
<tr>
<td>LD T5%</td>
<td>0.454</td>
<td>LD</td>
<td>0.168</td>
</tr>
<tr>
<td>HD T5%</td>
<td>0.358</td>
<td>HD</td>
<td>0.736</td>
</tr>
<tr>
<td>RES T5%</td>
<td>0.269</td>
<td>RES</td>
<td>0.084</td>
</tr>
<tr>
<td>LN T95%</td>
<td>0.009</td>
<td>Exchanger duties</td>
<td></td>
</tr>
<tr>
<td>HN T95%</td>
<td>0.028</td>
<td>ADU condenser</td>
<td>0.295</td>
</tr>
<tr>
<td>LD T95%</td>
<td>0.013</td>
<td>LN cooler</td>
<td>2.227</td>
</tr>
<tr>
<td>HD T95%</td>
<td>0.002</td>
<td>HN cooler</td>
<td>1.035</td>
</tr>
<tr>
<td>RES T95%</td>
<td>0.027</td>
<td>LD cooler</td>
<td>0.301</td>
</tr>
<tr>
<td><strong>Supply temperature</strong></td>
<td></td>
<td><strong>.Column diameter</strong></td>
<td></td>
</tr>
<tr>
<td>ADU condenser</td>
<td>0.173</td>
<td>HD cooler</td>
<td>1.825</td>
</tr>
<tr>
<td>LN cooler</td>
<td>0.054</td>
<td>RES cooler</td>
<td>0.248</td>
</tr>
<tr>
<td>HN cooler</td>
<td>0.196</td>
<td>HN reboiler</td>
<td>2.122</td>
</tr>
<tr>
<td>LD cooler</td>
<td>0.199</td>
<td>LD reboiler</td>
<td>0.683</td>
</tr>
<tr>
<td>HD cooler</td>
<td>0.732</td>
<td>Fired heater</td>
<td>0.101</td>
</tr>
<tr>
<td>RES cooler</td>
<td>0.423</td>
<td>Main column</td>
<td></td>
</tr>
<tr>
<td>PA1</td>
<td>0.322</td>
<td>Section 1</td>
<td>0.898</td>
</tr>
<tr>
<td>PA2</td>
<td>0.208</td>
<td>Section 2</td>
<td>0.903</td>
</tr>
<tr>
<td>PA3</td>
<td>0.277</td>
<td>Section 3</td>
<td>0.915</td>
</tr>
<tr>
<td>HN reboiler</td>
<td>0.220</td>
<td>Section 4</td>
<td>0.942</td>
</tr>
<tr>
<td>LD reboiler</td>
<td>0.235</td>
<td>Section 5</td>
<td>0.767</td>
</tr>
<tr>
<td><strong>Target temperature</strong></td>
<td></td>
<td>HN side-stripper</td>
<td>2.409</td>
</tr>
<tr>
<td>ADU condenser</td>
<td>0.057</td>
<td>LD side-stripper</td>
<td>0.345</td>
</tr>
<tr>
<td>HN reboiler</td>
<td>0.178</td>
<td>HD side-stripper</td>
<td>3.012</td>
</tr>
<tr>
<td>LD reboiler</td>
<td>0.165</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.3 Feasibility constraint based on support vector machine

Table 3 presents the validation results of the support vector machine classifier. Note for positive class: true prediction means the SVM classifies the sample as feasible, and it later converges in the simulation model; false means that it is labeled as feasible, but it does not converge in the rigorous model. For negative class: true prediction means the SVM labels the sample as infeasible, and it indeed does not converge in the rigorous model; false prediction means the SVM labels the point as infeasible, yet it converges in the rigorous model.
Table 3 – Validation results for support vector machine.

<table>
<thead>
<tr>
<th>Prediction class</th>
<th>True prediction</th>
<th>False prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive class</td>
<td>966</td>
<td>95</td>
</tr>
<tr>
<td>(Converged samples, +1)</td>
<td>[91.1%]</td>
<td>[8.9%]</td>
</tr>
<tr>
<td>Negative class</td>
<td>396</td>
<td>293</td>
</tr>
<tr>
<td>(Unconverged samples, −1)</td>
<td>[58%]</td>
<td>[42%]</td>
</tr>
</tbody>
</table>

Overall: Correct prediction 78%
Wrong prediction 22%

From the optimization point of view, false positives remove part of the feasible region, therefore rejecting design alternatives that may be feasible (and potentially optimal). False negatives wrongly enlarge the search space by adding infeasible points. False positives need to be minimized, as otherwise there is the danger that the optimization algorithm will converge towards an infeasible solution. On the other hand, low false negatives will lead to larger CPU times, but do not compromise the quality of the final solution.

The support vector machine results are compared with previous work (Ochoa-Estopier and Jobson, 2015), in which an artificial neural network was used to classify sample points. The same data set used in training the support vector machine is used to train the artificial neural network. The procedure for setting up the artificial neural network can be found elsewhere (Beale et al., 2015). Table S10 in the supplementary material shows the performance results obtained from this analysis. From the results, artificial neural network retains slightly (3.4%) more feasible designs than the support vector machine. On the other hand, the support vector machine removes a higher proportion (4.1%) of the infeasible designs than the artificial neural network. This analysis indicates that the support vector machine is slightly advantageous when the emphasis is to increase the likelihood for optimization solutions to be feasible when simulated on a rigorous model.

5.4 Crude oil distillation unit optimization results

The MINLP problem [P3] is solved using MatLab R2015a, employing the genetic algorithm, ‘ga’ implemented in the Global Optimization Toolbox. The initial population contains 100
chromosomes (representing alternative designs), and the maximum number of generations is set to 300. These parameters values were tuned via some preliminary tests.

The genetic algorithm is run ten times in order to confirm that the solutions obtained are of a reasonable quality, as indicated by standard deviation.

To check for the effectiveness of the support vector machine introduced in this work, the optimization is run with and without the support vector machine. As shown in Table S11 (see supplementary material), the order of magnitude of the objective function for the two cases are within similar range, although when the support vector machine is not used, only 70% of the optimized solutions lead to a feasible simulation with the rigorous model, compared to 100% when the support vector machine is used, confirming the need to account for feasibility within an optimization framework for the design of crude oil distillation unit.

Details of the computational results for the multiple runs of the genetic algorithm (including a support vector machine) are summarized in Tables S6 (see supplementary material). The solution with the lowest objective function value and minimum computational time is reported Figure 7 and Tables 4 to 6. The five sections of the main column consist of 6, 10, 11, 8 and 7 trays respectively (counting from the bottom), while HD, LD, and HN side strippers have three trays each. The minimum hot and cold utility demand has decreased by 20% and 28%, respectively, compared with the initial design. These decrease are due to: (i) reduction in the temperature of the feed to the column, which reduces the fired heating duty for a given flow rate and furnace inlet temperature; (ii) redistribution of product flow rates between adjacent streams, i.e., transfer of product from streams with low temperature to those with high temperature, without compromising product quality specifications (see Table 5). The latter strategy improves the total amount of recoverable heat at high temperature (high-quality heat) within the system, thereby decreasing hot utility demand.

The total operating cost, consisting of costs of hot and cold utilities and of stripping steam, amounts to 8.24 $MM a⁻¹ (where MM is million), while the annualized capital cost of the column is 0.21 $MM a⁻¹. Therefore, the column total annualized cost amounts to 8.45 $MM a⁻¹, which is 20% lower than the initial design.
Figure 7 – Optimal configuration of crude oil distillation unit

Table 4 – Crude oil distillation unit operating conditions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Initial design</th>
<th>Optimized design</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA 1 duty (MW)</td>
<td>11.2</td>
<td>11.05</td>
</tr>
<tr>
<td>PA 2 duty (MW)</td>
<td>17.89</td>
<td>14.01</td>
</tr>
<tr>
<td>PA 3 duty (MW)</td>
<td>12.84</td>
<td>14.35</td>
</tr>
<tr>
<td>PA 1 DT (℃)</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>PA 2 DT (℃)</td>
<td>50</td>
<td>56</td>
</tr>
<tr>
<td>PA 3 DT (℃)</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>Main steam (kmol h⁻¹)</td>
<td>1200</td>
<td>900</td>
</tr>
<tr>
<td>HD steam (kmol h⁻¹)</td>
<td>250</td>
<td>189</td>
</tr>
<tr>
<td>Feed temperature (℃)</td>
<td>365</td>
<td>361</td>
</tr>
<tr>
<td>Reflux ratio</td>
<td>4.17</td>
<td>3.33</td>
</tr>
</tbody>
</table>
Table 5 – Product quality and flow rate

<table>
<thead>
<tr>
<th>Products</th>
<th>T5% (°C, ASTM D86)</th>
<th>T95% (°C, ASTM D86)</th>
<th>Flow rates (m³ h⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial design</td>
<td>Optimized design</td>
<td>Initial design</td>
</tr>
<tr>
<td>LN</td>
<td>25.9</td>
<td>24.5</td>
<td>110.6</td>
</tr>
<tr>
<td>HN</td>
<td>138.9</td>
<td>131.1</td>
<td>186.6</td>
</tr>
<tr>
<td>LD</td>
<td>215.9</td>
<td>215.7</td>
<td>301.5</td>
</tr>
<tr>
<td>HD</td>
<td>310.7</td>
<td>306.9</td>
<td>354.4</td>
</tr>
<tr>
<td>RES</td>
<td>361.4</td>
<td>360.4</td>
<td>754.3</td>
</tr>
</tbody>
</table>

Table 6 – Utility demand and column cost

<table>
<thead>
<tr>
<th>Variable</th>
<th>Initial design</th>
<th>Optimized design</th>
<th>Unit*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot utility</td>
<td>54.61</td>
<td>44.46</td>
<td>MW</td>
</tr>
<tr>
<td>Cold utility</td>
<td>61.18</td>
<td>46.04</td>
<td>MW</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utility cost</td>
<td>8.51</td>
<td>6.91</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Steam cost</td>
<td>1.77</td>
<td>1.33</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Total operating cost</td>
<td>10.28</td>
<td>8.24</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Annualized capital cost</td>
<td>0.33</td>
<td>0.21</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Total annualized cost</td>
<td>10.61</td>
<td>8.45</td>
<td>$MM a⁻¹</td>
</tr>
</tbody>
</table>

*MM denotes millions of dollars

The minimum energy demand is calculated using pinch analysis, without taking into account details of the heat exchanger network. Nevertheless, the stream information in Table S8 of the supplementary material can be used to design the heat exchanger network for the column. For a more detailed analysis, a heat exchanger network model will be required to replace the grand composite curve. In this way, the column and the heat exchanger network could be designed simultaneously.

The optimization results i.e., the optimal column configuration and operating conditions obtained using the surrogate model, were then used as inputs to a rigorous simulation in Aspen HYSYS, in order to check whether: (i) the simulation will converge; (ii) the ANN results are in good agreement with the converged simulation results; (iii) the performance in the two cases is similar, i.e., the ANN gave good guidance about the economic performance.
Table S7 (see supplementary material), indicates that the optimal results obtained using the surrogate model are in excellent agreement with those generated by the rigorous simulation, confirming the accuracy and effectiveness of the proposed approach.

The optimization results were also compared to results obtained via a direct simulation-optimization approach using a rigorous distillation column model, the results of which are shown in Table 7. The same input data (as presented in Tables S1 to S4) were used in both approaches. Details of the procedure for setting up the problem and generating the results for the simulation-optimization approach can be found elsewhere (Ibrahim et al., 2017). The simulation-optimization is facilitated via an interface established between Aspen HYSYS v8.6 and MatLab, where the rigorous column modeling is carried out in Aspen HYSYS v8.6, and the optimization is performed in MatLab using a genetic algorithm.

### Table 7 – Comparison of surrogate and rigorous model performance

<table>
<thead>
<tr>
<th>Variable</th>
<th>Optimization with the rigorous model</th>
<th>Optimization with the ANN+SVM</th>
<th>Unit*</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Utility requirements</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hot utility</td>
<td>44.32</td>
<td>44.46</td>
<td>MW</td>
</tr>
<tr>
<td>Cold utility</td>
<td>45.9</td>
<td>46.04</td>
<td>MW</td>
</tr>
<tr>
<td><strong>Cost analysis</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utility cost</td>
<td>6.89</td>
<td>6.91</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Steam cost</td>
<td>1.33</td>
<td>1.33</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Total operating cost</td>
<td>8.22</td>
<td>8.24</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Annualized capital cost</td>
<td>0.26</td>
<td>0.21</td>
<td>$MM a⁻¹</td>
</tr>
<tr>
<td>Total annualized cost</td>
<td>8.47</td>
<td>8.45</td>
<td>$MM a⁻¹</td>
</tr>
</tbody>
</table>

*$MM denotes millions of dollars

As can be seen in Table 7, the heating and cooling duties for two different methods are more or less the same, leading to similar utility cost and the same stripping steam cost. A total annualized cost of 8.45 $MM a⁻¹ is obtained using the surrogate model-based approach, which is within 0.3% of that of the simulation-optimization approach, although the two results have the same order of magnitude.

The CPU time for the simulation-optimization approach was 180 CPU minutes compared with about 100 CPU minutes obtained using the surrogate column model. It should be noted
that the CPU time for the surrogate model includes sampling (94 minutes), model construction and validation (5.2 minutes), and column optimization and pinch analysis (1.25 minutes). Overall, significantly less time was required (44%) without affecting the optimality and accuracy of the solution.

It is also worth mentioning that model development needs to be carried out only once, and this enables optimization of the system considering different objectives and constraints. For example, in a design problem where many scenarios might be explored, the proposed approach may significantly out-perform the direct optimization using the rigorous model. As an example, consider the application of a standard sensitivity analysis that aims to understand how the design changes according to variations in a cost parameter. In the proposed approach, the artificial neural network and support vector machine are built only once, and then the optimization is run (which is quite quick) as many times as desired. On the contrary, carrying out the same calculations with the rigorous model would entail much longer times as the time-consuming optimization would need to be repeated many times.

6 Conclusions

Optimization-based design of a crude oil distillation unit using surrogate models naturally leads to a mixed integer nonlinear programming problem that is very difficult to solve. Heat recovery needs to be considered during column design in order to account effectively for the trade-off between capital and operating cost.

This work proposes a novel systematic framework for the design of heat-integrated crude oil distillation unit that combines a surrogate column model (in particular, an artificial neural network), the support vector machine as a feasibility classifier and pinch analysis within an optimization framework that applies a genetic algorithm.

Suitable dependent and independent variables identified via sensitivity analysis were applied to generate sample points. Those samples that resulted in converged rigorous simulations were employed to build a surrogate model of the column using an artificial neural network, while the entire set of samples (whether or not the rigorous simulation
converged) was used to train the support vector machine which aimed to classify the solution space into feasible (i.e. likely to converge) and infeasible designs. The artificial neural network predictions are shown to be in good agreement with the results from the rigorous column model, and the support vector machine is shown to have some ability to remove infeasible design options from the search space. The column design was optimized, considering both structure and operational variables, using a genetic algorithm. The proposed approach was demonstrated to be capable of identifying cost-effective designs in significantly reduced times, compared to direct simulation-optimization using rigorous column models, rather than surrogate column models.

Future work will extend the proposed approach to design of flexible heat-integrated crude oil distillation unit that optimally processes multiple crude oil feedstocks.

Appendix A: Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.cherd.2018.03.006.

Acknowledgement

The authors would like to acknowledge the financial support from Petroleum Technology Development Fund (PTDF), Nigeria, for sponsoring this PhD research.

References


