Refinery Scheduling with Varying Crude: A Deep Belief Network Classification and Multimodel Approach

Xiaoyong Gao, Chao Shang, and Yongheng Jiang
Dept. of Automation, Tsinghua University, Beijing 100084, China

Dexian Huang
Dept. of Automation, Tsinghua University, Beijing 100084, China

Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China

Tao Chen
Dept. of Chemical and Process Engineering, University of Surrey, Guildford GU2 7XH, U.K

DOI 10.1002/aic.14455
Published online in Wiley Online Library (wileyonlinelibrary.com)

In model-based refinery scheduling, the varying composition of the crude being refined is a major challenge, especially for those reaction processes. A classification based, multimodel approach is proposed to handle the frequently varying crude. The idea is to build a scheduling model for each type of feed crude, and the type can be determined using an online classifier. The recently emerged deep belief network is introduced to develop the classifier, which provides more accurate classification than the traditional neural network. The proposed method is demonstrated through modeling a fluidized catalytic cracking unit (the mostly affected by varying crude), and then the scheduling of a refinery that was carefully simulated to mimic the actual operation of a refinery in northern China. The results reveal that the multimodel approach is effective in handling varying crude. © 2014 American Institute of Chemical Engineers *AIChE J*, 00: 000–000, 2014

Keywords: deep belief network, refinery scheduling, multimodel, varying crude

Introduction

Refinery scheduling optimization has attracted increasing interests in both academic and industrial communities. One of the challenges faced by refineries is the varying composition of the crude oil being refined. This can happen because, first the crude can come from different sources (oil fields), and second the crude from the same source may also undergo slow change over time. The first situation is the primary cause of variability particularly in the context of short-term scheduling (time horizon of several days), during which the gradual change of the same crude is insignificant. From the practical perspective, it is not always possible to have a constant supply of crude from a single source; this is especially the case in a number of large scale refineries in northern China, where crude oil from up to 20 sources is routinely blended and processed. Therefore, in this study, we aim to address the impact of blended crude oil from different sources on refinery scheduling.

To further precisely represent the problem, variability in crude has different impact on different processing units in refinery. For distillation units involving no chemical reaction, the boiling point-based swing cut model[1,2] can adequately handle the variability. However, the secondary processing units (SPUs), which convert heavy fractions into lighter ones through various cracking reactions, are largely influenced by the chemical composition of the crude oil. For example, two feeds with similar boiling points may exhibit dramatic differences in the cracking performance and product yields.[3] One useful approach is to adjust the blending ratio of different crudes so as to minimize the variability in both supply and composition.[4–7] However, these optimization-based methods do not always reduce the variability to a level, at which the influence on processing units is sufficiently reduced. For the purpose of scheduling, accurate modeling of the impact of crude variability on SPUs is an open problem.

In the literature of scheduling and planning, a variety of studies have been reported to handle the model formulation problem. Pinto et al. stressed the need to address the influence of feed properties and operation conditions on the outlet properties, yield, and operation cost of processing units[8]; however, how to model such influence is still underexplored. Li et al. proposed a fractions transfer ratios model for planning,[1] but the model only considers the influence of operation conditions and cannot reflect the changeover of feed. Göthe-Lundgren et al. took a multimodel approach to handling the presence of multiple operating modes, one model representing the yield and product quality under each mode.[9] Similar method has been adopted in other studies[10,11]; but none of these considered the impact of varying crude. Shah

Correspondence concerning this article should be addressed to T. Chen at t.chen@surrey.ac.uk.

et al. formulated the units' yield as a variable within a predefined range and took blend component properties as constant,[12,13] which is suitable for fixed, but not varying, crude oil. Li et al. considered crude oil fluctuation in refinery planning[14]; however, modeling the impact of varying crudes was not in the scope.

For the purpose of scheduling, the above reviewed methods do not handle well the large, frequent fluctuation of feed due to blending of different sources of crude. For SPUs, it is desired to develop the unit's scheduling model that depends on the crude oil being processed. The SPUs' scheduling model is generally different for different crudes. One modeling option is to rely on the composition hydrocarbon groups in the feed crude. It is known that for SPUs, different hydrocarbon groups exhibit different cracking performances, resulting in quite distinctive outflow yields and properties.[3] If the hydrocarbon composition of feed crude can be obtained, an empirical or even mechanistic model can be developed to reflect this relationship. Unfortunately, hydrocarbon analysis is not routinely performed in refineries. Moreover, even when these data are available, the nonlinear model, be it empirical or mechanistic, is too complex to be used in scheduling optimization. An alternative approach is to build a specific SPUs scheduling model for each "type" of crude oil without detailed hydrocarbon analysis data. The "type" is defined with respect to the yield and outlet properties of the SPUs and it can be related to the blending ratios of difference sources of crude. These data are readily available in refineries. The blended feed under each type is assumed to give the same yield and outlet properties, which is a reasonable approximation if the types are carefully designed. When used in scheduling, such simple, yet type-specific models result in mixed integer linear programming problems, which are substantially easier to solve than using nonlinear models. To our best knowledge, this approach has not been explored in refinery scheduling.

In more detail, a multimodel approach to SPU scheduling, integrating online feed crude oil classification, is proposed in this article. For each type of feed oil, a set of specific models are developed to describe the unit yield, outlet materials properties, and operating cost. A classifier is developed from historical operation data, and is used to determine the type, and thus the correct type-specific model, for a certain blending ratio of different crude sources. The type-specific model is then used to formulate the scheduling problem, which will subsequently be solved to determine the optimal schedule. It should be noted that the unit yield and outlet properties depend on not only the feed type but also the operating mode of the refinery. Therefore, the model is in fact type/mode-specific. For brevity, we still use the term "type-specific model" unless it causes confusion. For the purpose of feed type classification, we will explore both traditional neural networks (NNs) and the state-of-the-art deep belief network (DBN).[15,16] DBN is a novel pattern classification method recently developed in the machine learning field, and it often provides more accurate classification results.

The rest of the article is organized as follows. We first introduce a simulated refinery with varying feed crude as the platform for demonstrating the proposed method. The simulation is carefully designed to emulate the actual operation of a refinery in northern China. Subsequently, the overall multimodel approach is presented, followed by the development of crude type classifier using either NN or DBN. The effectiveness of the proposed methodology is demonstrated through a case study of short-term scheduling. Finally, we conclude the article with remarks.

A Simulated Refinery with Varying Feed Crudes

A simulated refinery, mimicking an actual refinery in northern China, is used to demonstrate the method developed in this article, and the flow sheet is depicted in Figure 1. The simulation is carried out in the Petro-SIM environment, a well-known platform for reactor simulation.[17] The main processing units simulated with Petro-SIM include one crude (atmospheric) distillation unit, one vacuum distillation unit (VDU), one fluidized catalytic cracking unit (FCCU), and one main fractionator for FCCU. Besides these units simulated in Petro-SIM, some hydro-upgrading processing units (HUPUs) are considered in this case study, including a straight run gasoline catalytic reformer (SRG Catalytic Reformer in the figure), a diesel hydrotreater, an FCC light gasoline etherification unit and an FCC heavy gasoline desulfurization. Two product oil blenders, gasoline blender and diesel blender, produce five different grades of gasoline and two grades of diesel. To avoid confusion, the units simulated with Petro-SIM are highlighted in red color, as shown in Figure 1. FCCU is the unit for which multimodeling is needed, because its reaction is strongly affected by the varying crude.

Seven crude oils with different composition of hydrocarbons, which are determined according to the actual operation of the refinery, are first defined in the "oil environment" of Petro-SIM. By changing the blending ratios of these crudes, different types of crude oil are obtained, and then passed through the various units to produce the fuels. To closely reflect the real operation of the plant, four operating modes are considered: gasoline-maximizing mode for the reactor and gasoline-maximizing mode for the fractionator (abbreviated as G&G), gasoline-maximizing mode for the reactor and diesel-maximizing mode for the fractionator (G&D), diesel-maximizing mode for the reactor and gasoline-maximizing mode for the fractionator (D&G), diesel-maximizing mode for the reactor and diesel-maximizing mode for the fractionator (D&D). A total of 6364 blending ratios, using random sampling within the range seen in the refinery, are simulated to obtain the data, which will be used to develop and validate the crude type classifier and type-specific scheduling model. These data are also used to explore and define the distinctive feed types, in conjunction with the reaction mechanism of the considered SPUs (i.e., FCCU here). It is known that for FCCU, feed can be broadly classified, according to the hydrocarbon components, as paraffins, olefins, naphthenes, and aromatics.[3] Although in the scenarios considered in this study, no hydrocarbon analysis can be carried out, these four types of feed undergo quite different reaction pathways and result in distinctive outlet yields and material properties. Therefore, four crude types are considered, and they can be determined from assessing the FCCU's outlet. If other SPUs are considered, the same principle could be applied to determine appropriate crude types.

The 6364 samples are labeled (i.e., the type determined) using a semisupervised learning method.[18] Specifically, a small fraction of the samples (415 or 6.5% of the entire dataset) are manually selected because they gave very distinctive

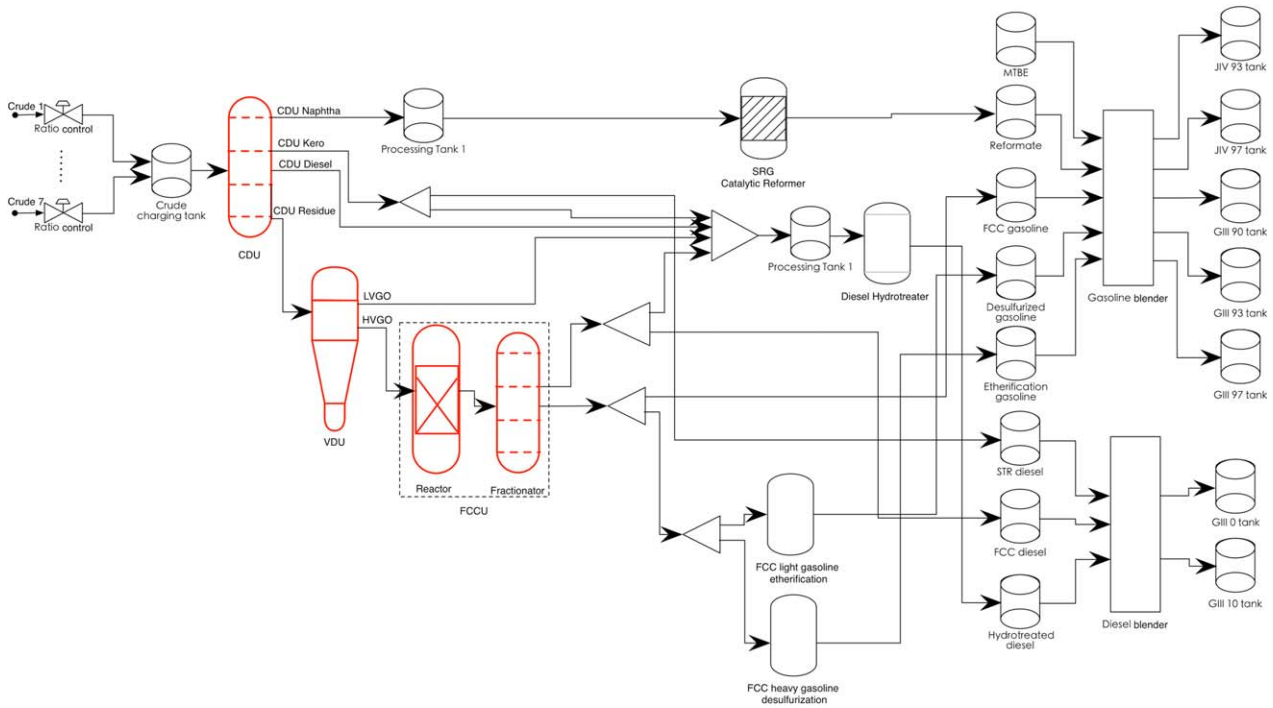

Figure 1. The flow sheet of the investigated refinery.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

yield and/or outlet material properties, which match one of the four types of hydrocarbon components.[3] These samples are then labeled as such. For each type, the yield, outlet properties, and operating cost are calculated from the average of the samples in that type. These average values form type-specific models. The remaining 5949 samples are not easily distinguished as one of the crude types by manual inspection, and they are labeled by comparing the Euclidean distance between the data and the type-specific models following an iterative procedure. First, one sample is randomly selected, and it is labeled to the type with the minimal Euclidean distance. This sample is then added to the labeled type, and the type-specific model is updated. This procedure is repeated until all data are labeled.

After this semisupervised labeling process, the data will be subsequently used to develop and validate the crude type classifier and type-specific scheduling model.

Multimodel Approach to Scheduling

A multimodel approach to scheduling, enabled by a crude type classifier, is presented in this section for the purpose of handling varying feed crudes. The scheduling model needs to describe the outlet yield, material properties, and unit operating cost, which are different for different operating modes (i.e., G&G, G&D, D&G, and D&D). The models are, therefore, type/mode-specific. The overall approach is illustrated in Figure 2, where $\mathrm{YLD}_{u,s',m,ct}$ is the yield of outlet material "s'" from unit "u" under the operating mode "m" and crude oil type "ct," $\mathrm{PRO}_{u,s',p,m,ct}$ the corresponding material property with index p, and $\mathrm{OPC}_{u,m,ct}$ the operating cost. These type/mode-specific models are developed from historical data and stored in a database. A crude type classifier will also need to be developed offline from past data. During the actual operation, the crude oil blending ratio is determined according to the refinery plan (outside the scope of this study), and it is then passed through the classifier to determine the type. Based on the crude type and the required operating mode, an appropriate scheduling model is selected from the database, and then used to formulate the scheduling problem to be solved.

We first demonstrate the need of crude type classification. Here, the entire dataset is randomly divided into a training set of 336 "historical" samples, and the remaining 6028 samples are used for validation. For illustration, only one operating mode (i.e., G&G) is considered, and the model outputs are the gasoline yield (YLD), research octane number (RON), and sulfur content (SUL) from the FCCU. The results, in terms of root mean square error (RMSE), are summarized in Table 1. In the table, "Models I to IV" are developed using the training data from the four crude types, and "Crudes 1 to 4" are the unseen validation data from the corresponding crude types.

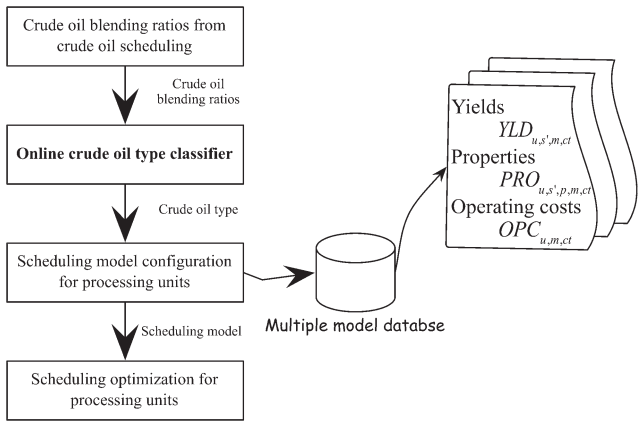

Figure 2. The overall multimodel approach to refinery scheduling with varying crude.

Table 1. RMSE of Type-Specific Models

	Model I			Model II			Model III			Model IV		
Crude Type	YLD	RON	SUL	YLD	RON	SUL	YLD	RON	SUL	YLD	RON	SUL
Crude 1	**0.77**	**1.24**	**2.97**	1.43	38.14	16.98	2.92	23.98	23.83	1.92	16.21	16.69
Crude 2	1.21	22.61	27.35	**0.88**	**1.90**	**2.38**	1.73	13.61	19.42	1.97	20.08	24.71
Crude 3	1.68	18.95	16.37	1.71	8.46	22.03	**0.69**	**1.39**	**1.36**	1.36	17.65	19.22
Crude 4	1.79	15.28	19.71	1.94	25.91	25.69	2.41	29.72	17.81	**0.47**	**1.84**	**2.87**

The bold characters indicate the case that the data are described in the correct model.

The results in Table 1 reveal that the multimodel approach is promising. When the data are described in the correct model, the RMSE is small, whereas if an incorrect model is used, the accuracy is substantially lower. This phenomenon suggests that the data within each type are sufficiently homogeneous, and that between-type variability is very large. Therefore, type-specific modeling is necessary and likely to give improved scheduling results, if the data can be correctly classified. Furthermore, this method maintains a relatively simple model structure and facilitates the subsequent optimization, which is important for practical application.

Motivated by the necessity of multimodeling, we next explore the development of crude type classifier so that the correct type-specific model can be determined for a particular blending ratio.

DBN for Crude Oil Type Classification

In practical refinery operations, the sources of crude oils and their blending ratio are often known, providing the basis for classification. Considering the scenario in which the sources of crude oils do not change for a long time (which is often the case in reality), the blending ratio is used as input for the classifier to determine the type of the blended feed. Classification is a classical topic in the area of statistical pattern recognition and machine learning.[19] Among many classification methods, NN is perhaps the most popular with wide applications.[20] In addition to the traditional NN, we will also explore the use of DBN, a recently emerged machine learning method.[15,16,21] In this section, the DBN method will be briefly presented, and then applied to crude type classification. The results of NN will also be illustrated for comparison.

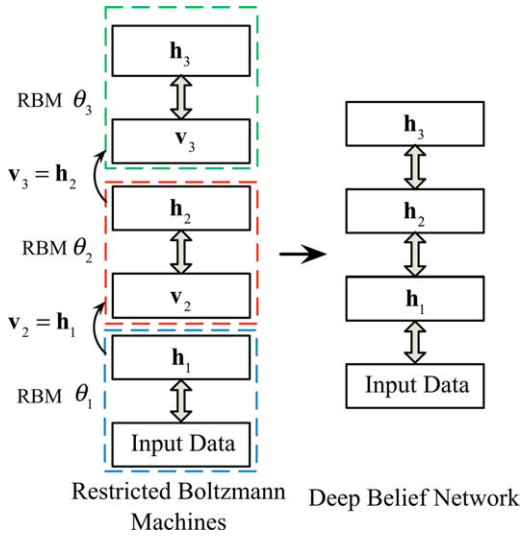

Figure 3. The structure of DBN.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

DBN method for classification

An overview of DBN and its use in classification is presented here, following the work of Hinton et al.[15,16,22] In a nutshell, a DBN consists of multiple layers, which are formed by stacking a series of restricted Boltzmann machines (RBMs) on top of each other. Each RBM is a bidirectional probabilistic model, consisting of an m-dimensional vector ($\boldsymbol{v}$) of a visible layer, and an n-dimensional vector ($\boldsymbol{h}$) of a hidden layer. Suppose that a DBN has L RBMs, each of which has the visible and hidden layers $\left\{\boldsymbol{v}^{(l)}, \boldsymbol{h}^{(l)}\right\}, l=1,2,\ldots,L$ The input data of the DBN (e.g., the blending ratio of the crudes) is given to the visible layer of the first RBM $\boldsymbol{v}^{(1)}$. The second RBM then takes the hidden layer of the first RBM as its visible layer $\boldsymbol{v}^{(2)}=\boldsymbol{h}^{(1)}$, and so on for the remaining RBMs. The structure of a DBN is illustrated in Figure 3.

For the purpose of classification, the training strategy for DBN involves two steps, namely an unsupervised pretraining step without using the class labels, followed by a supervised refining step that includes the class label as an extra output layer. For quite a long time, DBN has been difficult to train due to the presence of many local optima. The two-step has proved effective in addressing this problem: the pretraining can help find a desired region of the model parameters adjacent to a good optimum, and the supervised step further refines this optimal solution. These two steps are discussed in the following.

Unsupervised DBN *P*retraining. . In the pretraining stage, the RBMs need to be specified and their parameters determined. To facilitate the discussion, let us consider the binary–binary RBM, in which both $\boldsymbol{v}$ and $\boldsymbol{h}$ are vectors of binary values. The binary–binary RBM can be readily extended to continuous-valued inputs, so that the model can be used for classifying the crude type using continuous-valued blending ratio. The detailed method for such extension can be found in Bengio et al.,[21] and will not be discussed in this article.

For binary–binary RBM, the model is defined by the following joint distribution of the visible and hidden layers

$$P(\boldsymbol{v},\boldsymbol{h})=\frac{\exp\left\{-\boldsymbol{h}^T\boldsymbol{W}\boldsymbol{v}-\boldsymbol{b}^T\boldsymbol{h}-\boldsymbol{c}^T\boldsymbol{v}\right\}}{Z} \tag{1}$$

where $\boldsymbol{W}\in R^{n\times m}$ is the parameter matrix; $\boldsymbol{b}\in R^{n\times 1}$ and $\boldsymbol{c}\in R^{m\times 1}$ are bias vectors for the hidden and visible layers, respectively. A partition function Z is introduced to ensure that Eq. 1 is a proper distribution. The conditional distributions associated with $\boldsymbol{v}$ and $\boldsymbol{h}$ are given by

$$P(h_j=1|\boldsymbol{v})=\sigma(W_{j\cdot}\boldsymbol{v}+b_j),\ j=1,2,\ldots,n$$
$$P(v_i=1|\boldsymbol{h})=\sigma(W_{\cdot i}^T\boldsymbol{h}+c_i),\ i=1,2,\ldots,m \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(x)=\frac{1}{1+e^{-x}}$, $W_{j\cdot}$ and $W_{\cdot i}$ are jth row and ith column of $\boldsymbol{W}$, respectively. Here, $P(h_j=1|\boldsymbol{v})$ can be seen as the feed-forward pass to obtain the hidden layer from the visible one, and $P(v_i=1|\boldsymbol{h})$ the back-forward pass in which the visible layer $\boldsymbol{v}$ are reconstructed from the hidden layer.

For each RBM, the parameters $\{\boldsymbol{W},\boldsymbol{b},\boldsymbol{c}\}$ can be trained by maximizing the marginal probability of the visible layer $P(\boldsymbol{v})$, using, for example, the gradient ascent optimization algorithm. Given a single input sample $\boldsymbol{v}$, the gradient of the log marginal probability, $\log P(\boldsymbol{v})$, with respect to $\boldsymbol{W}$, is given by[21]

$$\frac{\partial \log P(\boldsymbol{v})}{\partial W_{ji}}=\langle v_i h_j\rangle_{\text{data}}-\langle \hat{v}_i h_j\rangle_{\text{recon}} \tag{3}$$

where $\langle v_i h_j\rangle_{\text{data}}=\sum_{\boldsymbol{h}} v_i h_j P(h_j|\boldsymbol{v})=v_i P(h_j=1|\boldsymbol{v})$ is the expectation of the pairwise product of v_i and h_j conditional on $\boldsymbol{v}$, and $\langle \hat{v}_i h_j\rangle_{\text{recon}}=\sum \hat{v}\sum_{\boldsymbol{h}} \hat{v}_i h_j P(\hat{v}_i,\boldsymbol{h})$ is the expected value of $\hat{v}_i h_j$ when $\{\hat{\boldsymbol{v}},\boldsymbol{h}\}$ follow the distribution given in Eq. 1. The gradient of $\log P(\boldsymbol{v})$ with respect to $\boldsymbol{b}$ and $\boldsymbol{c}$ can be formulated in a similar manner

$$\frac{\partial \log P(\boldsymbol{v})}{\partial b_j}=\langle h_j\rangle_{\text{data}}-\langle h_j\rangle_{\text{recon}},$$
$$\frac{\partial \log P(\boldsymbol{v})}{\partial c_i}=\langle v_i\rangle_{\text{data}}-\langle \hat{v}_i\rangle_{\text{recon}} \tag{4}$$

In the actual implementation, the computation of Eqs. 3 and 4 can be simplified using the contrastive divergence algorithm,[23] a second-order approximation of the gradient of $\log P(\boldsymbol{v})$. After obtaining the model parameters, the hidden layer $\boldsymbol{h}$ can be determined using its conditional expectation $h_j=E(h_j|\boldsymbol{v})=\sigma(W_j\boldsymbol{v}+\boldsymbol{b}_j)$, which will then be used as the visible layer for the next RBM.

The entire DBN is trained by a greedy layer-wise strategy, that is, the individual RBMs are trained one by one, from bottom to top.[17] The basic idea is to add a RBM on the top of an already trained DBN: after an l-layer DBN with l individual RBMs has been trained, the input data are passed through the DBN to collect the outputs at the lth hidden layer (the top of the DBN). Then, the outputs are used as inputs to train the $(l+1)$th RBM, which will be added to the previous model to form a new $(l+1)$-layer DBN. The whole procedure is unsupervised, as the class labels have not been used.

Supervised DBN Refining. After the RBMs are successively pretrained, the DBN needs to be further refined by including the class labels to minimize classification error. This is accomplished by adding an additional layer of weights on top of the pretrained DBN, so that the unsupervised DBN can be linked to the class labels. Suppose that the pretrained DBN has $L-1$ layers, and its top hidden layer is $\boldsymbol{h}^{(L-1)}$. Using the same stacking method, the additional layer takes $\boldsymbol{h}^{(L-1)}$as its visible layer, that is, $\boldsymbol{v}^{(L)}=\boldsymbol{h}^{(L-1)}$; this $\boldsymbol{v}^{(L)}$ passes through the weight $\boldsymbol{W}^{(L)}$ and the bias $\boldsymbol{b}^{(L)}$, and is finally linked with the class probability p_i using the usual soft-max function[15]

$$p_i=\frac{\exp\left\{W_i^{(L)}\boldsymbol{v}^{(L)}+b_i^{(L)}\right\}}{\sum_{i'}\exp\left\{W_{i'}^{(L)}\boldsymbol{v}^{(L)}+b_{i'}^{(L)}\right\}} \tag{5}$$

where $W_i^{(L)}$ denotes the ith row of matrix $\boldsymbol{W}^{(L)}$, and $b_i^{(L)}$ the ith element of vector $\boldsymbol{b}^{(L)}$. Notice that p_i is normalized, that is, $\sum_i p_i=1$, as it represents the probability that the sample belongs to class i. Through Eq. 5, the likelihood function of the data can be formulated, and an error back-propagation method can be used to maximize the likelihood, similar to the traditional NN. Note that $\boldsymbol{v}^{(L)}$is a function of the weight and bias in the $(L-1)$th layer, and in turn a function of all the weight and bias terms throughout the DBN. As a result, by maximizing the likelihood, the model parameters at all layers can be refined.

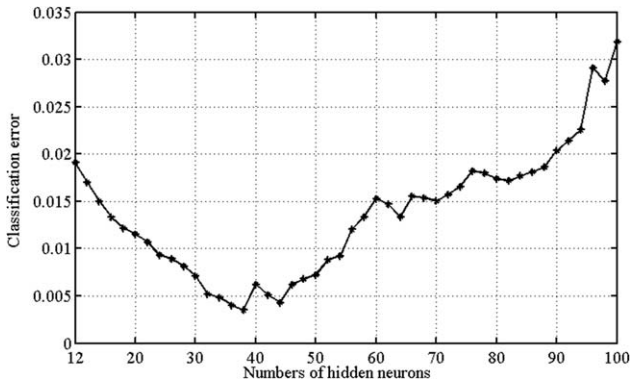

Figure 4. The cross-validation results of NN.

Classification results

To demonstrate the classification results, the data samples are divided in the same way as before: 336 historical samples for developing the classifier, and the remaining 6028 for validation. To develop a DBN, both the number of hidden layers, and the number of neurons in each hidden layer, must be determined. Because training a DBN is computationally slow, we follow the empirical guidelines in the literature[22,24] to use two hidden layers, and selected the following candidates for the number of neurons in the two hidden layers according to experience[24]: {14,7}, {14,9}, {14,11}, {18,7}, {18,9}, {18,11}, {18,17}, {20, 9}, {20,17}, {22, 15}, {24, 7}, {24,9}, {24,17}, {26,9}, {26, 15}, {26,17}. Then, a sixfold crossvalidation is used to find the best option {18,9}. This 7–18-9-4 DBN (input layer: seven nodes for the blending ratio of seven sources of crude; two hidden layers: 18 and 9 neurons, respectively; output layer: four types of crude) is used in subsequent classification. Being an emerging technique, DBN is still under intense research in terms of optimizing the network configuration; nevertheless, it seems that by following reported guidelines, together with modeler's experience, DBN does provide practically excellent accuracy.

For the purpose of comparison, a traditional NN is also developed using a single hidden layer with the sigmoidal activation function. The number of hidden neurons is also determined using sixfold crossvalidation. According to the results in Figure 4, 38 hidden neurons are used. For both DBN and traditional NN, an early stopping strategy is used to prevent overfitting. As the training algorithms for both models are not deterministic and give different results each time, we repeat the training 50 times and report the mean performance with standard deviation. The classification errors are summarized in Table 2.

A comparison of the results shows that, first, validation error is significantly higher than that of the training set, which is a common phenomenon. Second, DBN achieves

Table 2. Classification Errors on Both Training and Validation Dataset (Mean ± Standard Deviation %)

		Type 1	Type 2	Type 3	Type 4	All
Training	NN	0.31 ± 0.1 7	0.29 ± 0.32	1.29 ± 1.21	0.53 ± 0.15	0.47 ± 0.34
	DBN	0.27 ± 0.11	0.26 ± 0.08	0.88 ± 0.43	0.48 ± 0.16	0.39 ± 0.21
Validation	NN	3.27 ± 0.36	2.03 ± 0.21	4.19 ± 1.25	3.31 ± 0.68	2.83 ± 0.42
	DBN	2.35 ± 0.21	1.81 ± 0.16	2.96 ± 0.72	1.85 ± 0.61	2.01 ± 0.25

significantly more accurate classification than the traditional NN. The difference in classification error on the validation set (2.83–2.01% = 0.82%) suggests that on average, the DBN correctly classifies 49.4 more samples than the traditional NN out of a total sample size of 6028 (6028×0.82%=49.4), which can have significant impact on the ultimate schedule modeling and optimization.

Furthermore, recall that Table 1 reports the RMSE of type-specific models if all data samples are correctly classified. It would be interesting to examine the accuracy of type-specific models when data are classified using an imperfect classifier. Table 3 reports such results, averaged over 50 repeats of the training process, for the yield of the FCCU. (To save space, the results for other outlet properties are not reported.) It can be seen that, again, DBN classifier gives more accurate type-specific prediction, because its classification is more accurate. In addition, it may be surprising to observe that for crude Type 1 and 2, DBN results in even lower RMSE than perfect classification. However, the results in Table 3 are only for the FCCU's yield, and when considering all other outlet properties, the RMSEs of DBN are actually all higher than perfect classification.

Scheduling Results

The developed multimodeling and crude type classification approach is further demonstrated through optimal scheduling of the entire refinery (cf., Figure 1 for the flow sheet). We follow the state-task network-based discrete time representation[9] to formulate the scheduling problem. As discussed previously, the CDU, VDU, and FCCU are simulated in Petro-SIM, highlighted in red color in Figure 1, to develop data-driven scheduling models. Swing cut model is used for CDU and VDU. The proposed DBN classifier-based multimodel is developed for FCCU, as DBN provides significantly better classification accuracy than the traditional NN. The classifier and the type-specific models are the same as presented in the previous sections, developed using 336 simulated historical samples. Modeling the HUPUs is beyond the scope of this study. For simplicity, the HUPUs' models are taken as the averaged value from operating data for each predefined operating mode.

An assumed 5-day demand based on the real planning results of the investigated refinery is given in Figure 5, where JIV 93, JIV 97, GIII 90, GIII 93, and GIII 97 are different gasoline products available on the Chinese market, and GIII 0 and GIII 10 correspond to two diesel grades. Detailed specifications are provided in Appendix.

Table 3. RMSE of Type-Specific Models when the Choice of Model is Determined by Classification

	Type 1	Type 2	Type 3	Type 4	All
Neural network	0.79	0.90	1.79	0.70	0.98
Deep belief network	0.67	0.79	1.10	0.69	0.87
Perfect classification	0.77	0.88	0.69	0.47	0.75

The results of "perfect classification" are reproduced from Table 1.

Based on the plant monthly planning and crude oil scheduling (neither of these is within the scope of the study and thus taken as is), the future 5-day crude oil blending schedules are given in Figure 6. Crude 1 and Crude 2 are the primary crude oils from a local oil field in northern China, and their supply to this refinery is relatively stable. The other five crude oils are supplied from overseas.

The proposed multimodel approach is applied for scheduling. For comparisons, the FCCU is also described by a single model, where the yield, outlet material properties, and operating cost are taken as the average of the historical data. In both methods, the units other than FCCU are modeled in the same way, and the scheduling problems are solved using the same solver, LINGO 11.[25] The profit for the two methods is compared in Table 4, where the accounted profit (of FCCU only) is obtained by implementing the optimized schedules on Petro-SIM and calculated with the accounting prices provided by the field engineers, such as the prices of cooling water, steam, electricity, inflow materials, outflow components and so on. The computed overall profit is essentially the value of the objective function of the scheduling problem, and it tends to overestimate the actual profit because of model-reality mismatch. The overestimate may be especially substantial for the one-model method for the

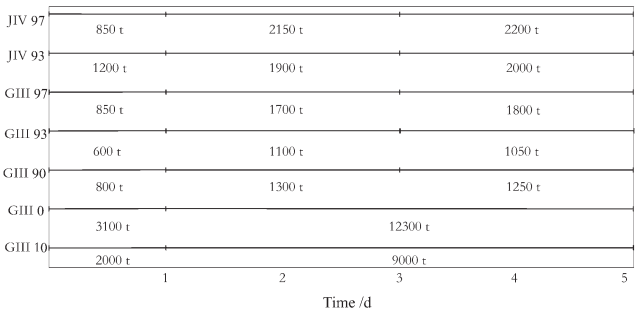

Figure 5. Gantt chart for the future 5-day demand.

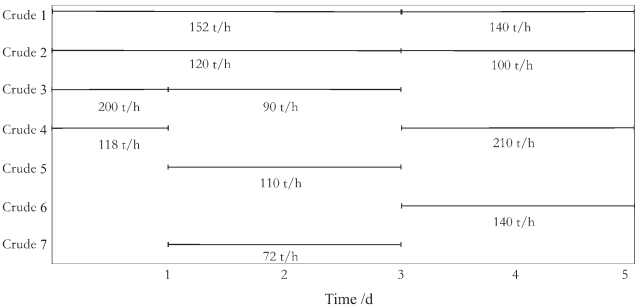

Figure 6. The future five-day crude oil blending schedule.

Table 4. Optimized Profit in Chinese Yuan

	Without Classification	DBN Based Classifier
Accounted profit of FCCU	1,237,060.1	1,460,226.8
Computed overall profit	4,027,552.3	4,572,077.4

FCCU, given its large model-reality mismatch when compared with the multimodel approach. In this sense, the accounted profit is a more accurate indicator for comparison.

Table 4 clearly indicates that the proposed multimodeling approach results in significantly higher profit than using a single model. This result again confirms the need of performing crude oil classification for SPUs' scheduling model, from the profit point of view: even for computed overall profit, the multimodel method results in 13.5% more profit.

As for the accounted profit of the FCCU, the DBN-based multimodel method gives 18.0% more profit than using a single model. The detailed schedule is shown in Figure 7. For the one-model approach, although the calculated schedule is "optimal" in theory, the large model-reality mismatch results in higher energy and utility cost to realize such schedule. For example, in the first day, due to the large modeling error, the one-model approach requests to produce more gasoline and diesel with less feed flow. To realize this request, the reflux rate has to be increased and this leads to a higher temperature set-point and thus higher energy cost. FCCU is among the most important SPUs in China[26] and its energy consumption counts for about 30–50 % of that of the entire refinery[27]; optimal scheduling, when properly formulated and implemented, is potentially a big contributor to reducing the energy usage and improving process efficiency.

Conclusions

A multimodel approach to refinery scheduling, which integrates with deep learning technique for crude oil type classification, is developed in this article. The method is particularly useful for scheduling SPUs in the presence of varying feed crude. The basic idea is to build the scheduling models that are specific to each crude oil type and operating mode, and select the proper model using classification for schedule optimization. Deep learning method used here improves the classifying accuracy for the nonlinear refinery processes. Furthermore, this method provides a good approximation of the units' yield and outlet properties, and meanwhile maintains a simple model structure, which is necessary for optimization. The usefulness of the proposed method is demonstrated on modeling and scheduling of a simulated refinery in terms of handling varying crude.

It should be noted that NN, and its deep learning versions, are certainly not the only options for classification. Other classification methods, ranging from simple linear regression to support vector machines, could also be used. Because the primary purpose of this work is not to compare various classification models, but to propose the use of the multimodeling approach, we refer the readers to the literature for alternative classification models.[19,28,29]

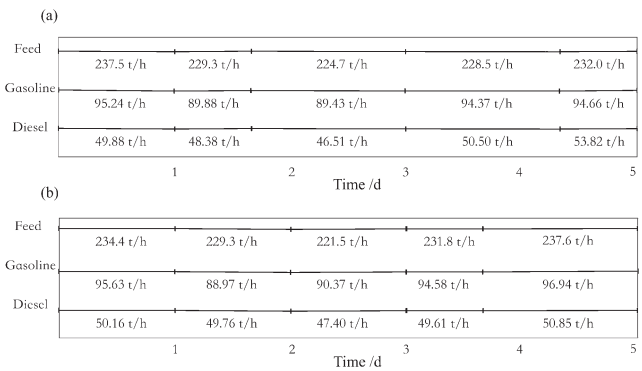

Figure 7. Gantt chart of FCCU by: (a) the proposed multimodel approach; (b) the one-model approach.

Acknowledgments

This research was supported by the National Basic Research Program of China (2012CB720500) and the National Natural Science Foundation of China (No. 21276137, No.61273039). Xiaoyong Gao's visit to the University of Surrey was funded by the Santander Universities through a Santander Postgraduate Research Award.

Literature Cited

1. Li WK, Hui CW, Li AX. Integrating CDU, FCC and product blending models into refinery planning. *Comput Chem Eng*. 2005;29:2010–2028.
2. Zhang J, Zhu XX, Towler GP. A level-by-level debottlenecking approach in refinery operation. *Ind Eng Chem Res*. 2001;40:1528–1540.
3. Sadeghbeigi R. *Fluid Catalytic Cracking Handbook: An Expert Guide to the Practical Operation, Design, and Optimization of FCC Units*, 3rd ed. Oxford: Elsevier Inc., 2013.
4. Bai L, Jiang Y, Huang D. A novel two-level optimization framework based on constrained ordinal optimization and evolutionary algorithms for scheduling of multipipeline crude oil blending. *Ind Eng Chem Res*. 2012;51(26):9078–9093.
5. Mouret S, Grossmann I, Pestiaux P. A novel priority-slot based continuous-time formulation for crude oil scheduling problems. *Ind Eng Chem Res*. 2009;48(18):8515–8528.
6. Zhang J, Wen Y, Xu Q. Simultaneous optimization of crude oil blending and purchase planning with delivery uncertainty consideration. *Ind Eng Chem Res*. 2012;51(25):8453–8464.
7. Li J, Misener R, Floudas C. Continuous-time modeling and global optimization approach for scheduling of crude oil operations. *AIChE J*. 2012;58(1):205–226.
8. Pinto JM, Joly M, Moro LFL. Planning and scheduling models for refinery operations. *Comput Chem Eng*. 2000;24(9–10):2259–2276.
9. Göthe-Lundgren M, Lundgren J, Persson J. An optimization model for refinery production scheduling. *Int J Prod Econ*. 2002;78:255–270.
10. Luo C, Rong G. Hierarchical approach for short-term scheduling in refineries. *Ind Eng Chem Res*. 2007;46:3656–3668.
11. Kelly JD. Production modeling for multimodal operations. *Chem Eng Prog*. 2004;100(2):44–54.
12. Shah N, Ierapetritou M. Short-term scheduling of a large-scale oil-refinery operations: incorporating logistics details. *AIChE J*. 2011;57(6):1570–1584.
13. Shah N, Safaridis G, Jia Z, Ierapetritou M. Centralized-decentralized optimization for refinery scheduling. *Comput Chem Eng*. 2009;33(12):2091–2105.
14. Li W, Hui C, Li P, Li A. Refinery planning under uncertainty. *Ind Eng Chem Res*. 2004;43(21):6742–6755.
15. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science*. 2006;313:504–507.
16. Hinton GE, Osindero S, The YW. A fast learning algorithm for deep belief nets. *Neural Comput*. 2006;18(7):1527–1554.
17. KBC Advanced Technologies. Petro-SIM User Guide. KBC Advanced Technologies, KBC Profimatic, 2012. Available at: http://www.kbcat.com/support-and-training/software-technical-support.
18. Ge Z, Huang B, Song Z. Mixture semisupervised principal component regression model and soft sensor application. *AIChE J*. 2014;60(2):533–545.
19. Bishop CM. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
20. Michie D, Spiegelhalter DJ, Taylor CC. Machine Learning, Neural and Statistical Classification. Ellis Horwood: Prentice Hall; 1994.

21. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. *Adv Neural Inform Process Syst*. 2007;19:153–160.
22. Hinton GE. A practical guide to training restricted Boltzmann machines. Technical Report, No. UTML TR 2010-003. Toronto: Department of Computer Science, University of Toronto, 2010.
23. Carreira-Perpinan MA, Hinton GE. On contrastive divergence learning. In: Proceedings of the tenth international workshop on artificial intelligence and statistics. NP: Society for Artificial Intelligence and Statistics, 2005;33–40.
24. Shang C, Yang F, Huang D, Lyu W. Data-driven soft sensor development based on deep learning technique. *J Process Control*. 2014; 24:223–233.
25. Cuiwen C, Xingsheng G, Zhong X. A data-driven rolling-horizon online scheduling model for diesel production of a real-world refinery. *AIChE J*. 2013;59(4):1160–1174.
26. Ma L, Fu F, Li Z, Liu P. Oil development in China: current status and future trends. *Energy Policy*. 2012;45:43–53.
27. Yu L. Calculation and analysis of actual energy consumption for FCCU. *Pet Process Petrochem (In Chinese)*. 2004;35(3):5–8.
28. Balabin RM, Safieva RZ, Lomakina EI. Gasoline classification using near infrared (NIR) spectroscopy data: comparison of multivariate techniques. *Anal Chem Acta*. 2010;671:27–35.
29. Kotsiantis SB, Zaharakis ID, Pintelas PE. Machine learning: a review of classification and combining techniques. *Artif Intell Rev*. 2006;26:159–190.

APPENDIX A : Quality Specification of the Gasoline and Diesel Products

Detailed Quality Specification of Different Grade Gasoline Products and Different Grade Diesel Products are given in Tables A1 and A2, respectively.

Table A1. Detailed Quality Specification of Different Grade Gasoline Products

Items		GIII 90#	GIII 93#	GIII 97#	JIV 93#	JIV 97#
Antiknock quality						
RON	Not less than	90	93	97	93	97
(RON+MON) /2	Not less than	85	88	Report	88	Report
Lead content/(g/L)	Not more than		0.005			0.005
Iron content/ (g/L)	Not more than		0.01			0.01
Manganese content/(g/L)	Not more than		0.016			0.006
Sulfur content/mg/kg	Not more than		50			50
Benzene (by volume)/%	Not more than		1.0			1.0
Aromatics (by volume)/ %	Not more than		40			
Aromatics + Olefins/ %	Not more than					60
Olefin (by volume)/%	Not more than		28			25
Oxygen content (by mass)%	Not more than		2.7			2.7
Reid vapor pressure/kPa						
November 1– April 30			48 – 85			≤88
May 1– October 31			40 – 68			≤65
Distillation range						
10% point/°C	Not more than		70			70
50% point/°C	Not more than		120			120
90% point/°C	Not more than		190			190
End point/°C	Not more than		205			205
Residue (by volume)'%	Not more than		2			2

Table A2. Detailed Quality Specification of Different Grade Diesel Products

Items		GUIII 0#	GUIII −10#
Sulfur content (by mass)/% Acidity	Not more than		0.035
Grade(mg KOH/100 mL)	Not more than		7
Carbon residue (by mass)/%	Not more than		0.3
Freezing point/°C	Not more than	0	−10
−10 Viscosity (20°C)/(mm2/s)		3.0 – 8.0	2.5 – 8.0
Flash point closed Cup)/°C	Not less than		55
Flammability			
Cetane number	Not less than		45
Cetane index	Not less than		43
Distillation range:			
50% point/°C	Not more than		300
90% point/°C	Not more than		355
95% point/°C	Not more than		365

Manuscript received Dec. 22, 2013, and revision received Feb. 27, 2014.