A Classification Framework for Process Operation Optimization and its Application in a Triazophos Plant

G. Rong,[+] H. Gu, B. Jin, and J. Shao
State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027, P. R. China

Original scientific paper
Received: March 12, 2007
Accepted: August 19, 2007

Knowledge-based operation optimization methods may suffer from difficulties in modeling the chemical processes and solving the mathematical equations. In this paper, a data-based classification method for operation optimization is introduced. In contrast with other fields, chemical process is characterized by time delay and interaction between upstream and downstream units. By rebuilding historical data and constructing a group of multiple classifiers, both of the characterized problems are overcome. Some qualitative operational advice may extract from the group of multiple classifiers. As a result, the operation of chemical processes may achieve to a reachable optimal state using rolling optimization strategy by updating the classifiers. In addition, some special data-preprocessing techniques are considered to improve the efficiency of the classification. This classification framework customized for chemical process helps a Triazophos plant to improve the productivity of Triazophos from 93.3 % to 95.8 % after implementation of the proposed method for more than one year.

Key words:
Operation optimization, process operational data, data preprocessing, data mining, classification

Introduction

Process operation optimization intends to maximize the objective function by adjusting operating conditions under guidance. Mathematical programming methods are known to solve process operation optimization problems, but may fail when they are applied to large and complex chemical processes. Modeling the chemical processes may have the following significant limitations: not all processes are understood in basic engineering and scientific principles; some product properties may not be adequately described and measured; the number of skilled model builders is limited, and the cost associated with building such models is thus quite high. In contrast with mathematical programming methods, data mining which is also popularly referred to as Knowledge Discovery in Databases (KDD) is data driven. It constructs and trains some patterns from historical data, in order to reflect salient attributes and behaviors of the phenomena. When the chemical plant runs under the guidance of the steady state optimizer, the process variables always fluctuate more or less, and the suggested operating conditions are not exactly the optimal, because there are no modeling methods which can present the whole 'fact' of a plant. The object of this work is thus to provide a method capable of extracting some patterns from the historical data which reflect the hidden relations between the maximal (minimal) output variables (e.g. productivity of main chemical product) and the operating conditions. In other words, data mining used herein is the complement technology of mathematical programming, in order to make the output variables approach the unknown optimal state.

Recently, data mining has been applied to chemical industries for quality design,[1] process modeling,[2] fault diagnosis,[3] planning and real-time scheduling tasks, supply chain management and process optimization.[4] In contrast to the specific solution of mathematical programming methods, data mining for process optimization finds the qualitative relations between operational conditions and objective function. Both classification[4] and association rules[5] methods are able to finish this task in simulation processes. However, these methods[4,5] suffer failure when applied to practical processes, especially in the large-scale processes comprised by a series of upstream and downstream processes, because there are some specific problems in chemical processes like time delay and interaction between upstream and downstream units. Time delay means that the real impact factor to $OVs[t]$ (the value of **o**bjective **v**ariable**s** that need to be optimized at time t) is $OCs[t-\tau]$ (the value of **o**perating **c**ondition**s** at time $t-\tau$, where τ is the time constant) instead of

[+] Corresponding author.
Fax: 86-571-8666-7616; Tel: 86-571-8666-7032
E-mail: grong@iipc.zju.edu.cn; haijie.gu@gmail.com

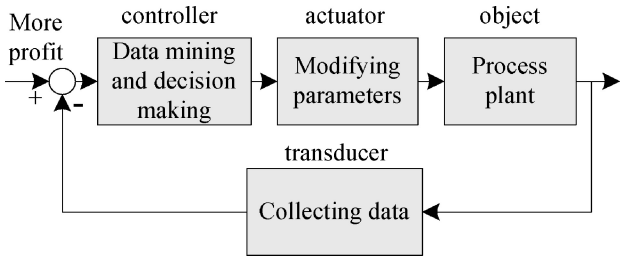
$OCs[t]$. $OVs[t]$ may be even impacted by a group of records of operating conditions within a period of time, e.g. data between $OCs[t-\tau-\varepsilon_1]$ and $OCs[t-\tau-\varepsilon_2]$, taking the upstream-downstream relations of processes into consideration. Dubitable or even wrong answers may be resulted in when general data mining methods are directly introduced into chemical industries regardless of these specific problems.

In contrast with the quantitative optimization methods used in chemical engineering, the main disadvantage of data mining methods is that it usually gives only qualitative guidance. Therefore, we utilize a rolling optimization strategy to fill the gap in this paper. The operating conditions are modified a little in the direction of guidance at each period. As a result, the states of objective variables are improved step by step. Besides the rolling optimization strategy, a customized classification framework for operation optimization is proposed. Under this framework, sample time based records are transformed into event-based data to overcome the time delay, and time-related multiple classifiers are constructed to reduce the influence of upstream-downstream relationship. Decision tree is selected as the type of classifier in this work, so the proposed method involving multiple decision trees is called decision forest. This framework needs the participation of process engineers. Its performance is shown by its application in an industrial Triazophos plant.

This work aims at presenting the whole steps to analyze the operation data and to optimize the operating conditions using classification technology. In the remainder of this paper, the rolling optimization strategy and customized classification framework are proposed, the flowsheet of Triazophos plant is described, the general and specific data preprocessing methods are shown, our decision forest algorithm of the framework is proposed, the performance of the proposed framework is analyzed, and conclusions are made orderly.

Rolling optimization strategy and customized classification framework

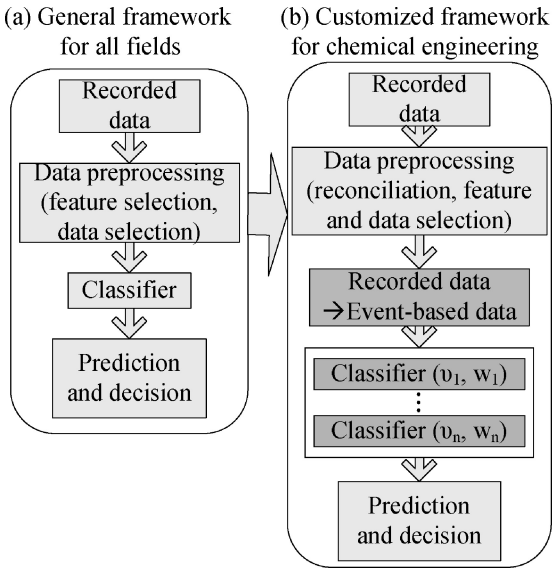
As a complement to mathematical programming, the goal of data mining is to approach the reachable optimal objective according to historical operation data. Initially, the operating conditions are set to be the optimal values calculated by the steady state optimizer. However, the engineers may adjust the operating conditions to better ones according to their experiences when the production process does not perform in accordance with the description and prediction of the optimizer. As a result, redundant process data reflecting knowledge and experience of many process experts were recorded. The classification technology is utilized to extract the historical optimal operating conditions that lead to maximal objective function. Then the operating conditions will be adjusted according to the rules or suggestions generated from the classification results. After a time, the process data has refreshed and the data mining task will be restarted again. After that, a new cycle will be executed including data mining, operation adjusting and data collecting. This rolling optimization strategy is just like a typical control system, shown in Fig. 1. It should be noted that it is not an online optimizing system, since it needs a long time to collect enough process data.



F i g . 1 – *Rolling process operation optimization strategy using data mining technology*

The object of operation optimization is always to maximize the productivity and profit or to minimize the consumption of energy and material. The objective function may be a hybrid function involving these objects. In order to make readers who are not familiar with data mining understand easily, some representations should be illustrated: operating conditions such as flow rates, temperatures and so on, which influence the objective variables, are called *attributes* to a classifier in classification field, while the variables involved with the objective function such as productivity, profit and the amount of pollution emission, are called *classes* to a classifier in classification field. The adjusting direction of operating conditions may be discovered from the qualitative rules between *attributes* and *classes* in the classification system.

A general classification framework for most engineering fields such as retail, finance and telecom is shown in Fig. 2(a). In such typical fields, the concerned data are transaction records and one record represents one event. In chemical processes, however, one event means the whole process of adding, flowing, blending and reacting of the same batch of material, so the delay between these actions or operations should be removed from the database. In addition, upstream-downstream relation of units in a chemical plant is also a characteristic trouble in chemical engineering, which means that



F i g . 2 – *Customized classification framework for chemical engineering*

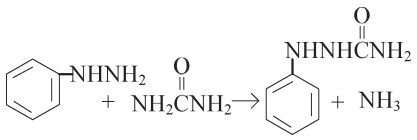
the current state of downstream unit is influenced by the state of upstream unit in an early period. In contrast, the *attributes* are independent in other typical fields. In this paper, a customized classification framework for chemical processes improved from the general classification framework is proposed, as shown in Fig. 2(b). The delay factor is overcome by transformation from recorded data to event-based data and the upstream-downstream relation is weakened by adopting multiple classifiers explained in the section of *data preprocessing* and *algorithm of decision forest*. The multiple classifiers strategy in this paper is time related and weighted, while the original one is sampling based, seeing Section *algorithm of decision forest*.

Processes description of Triazophos plant

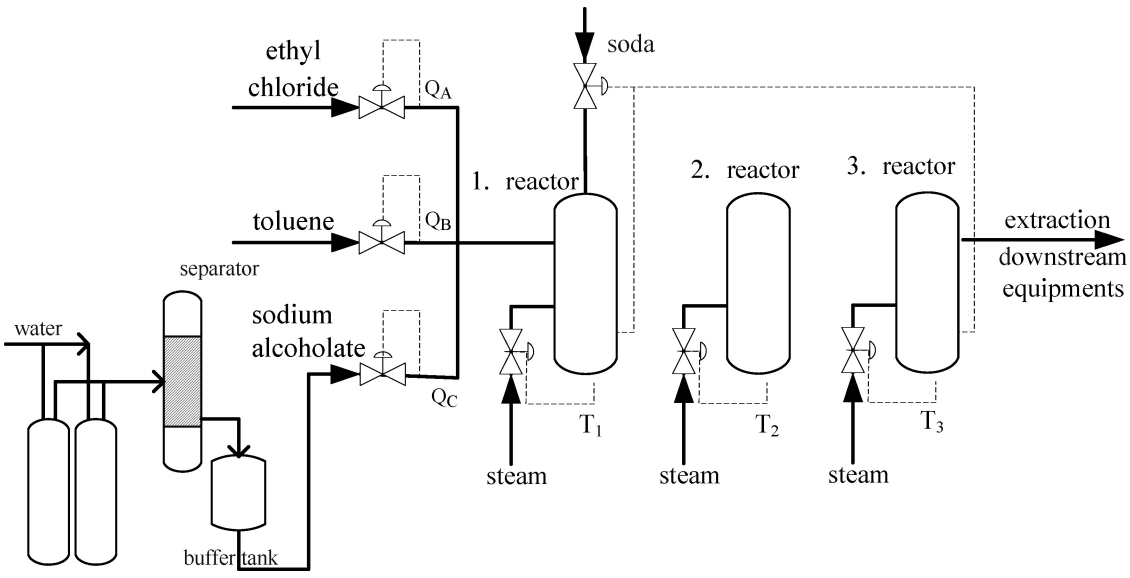
O,O-diethyl-O-(1-phenyl-1*H*-1,2,4-triazol-3-yl) phosphorothioate is the chemical name of Triazophos, which is an effective pesticide. Its formula is $C_{12}H_{16}N_3O_3PS$. It is widely used to destroy pests in rice, trees and vegetables.

The manufacturing process of Triazophos mainly includes four units: synthesis, extraction, isolation and recycle. Triazophos is made in the synthesis unit, and separated from other byproducts in the extraction and isolation units. In the recycle unit most of the solvent is recycled. Therefore, the synthesis unit dominates the quality and quantity of the Triazophos plant, and is the key unit for optimizing the process operation to improve the productivity of Triazophos.

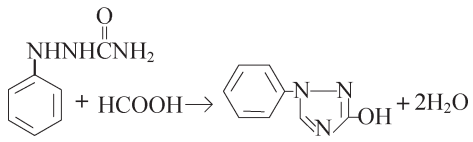
Fig. 3 shows the synthesis unit flow diagram. Ethyl chloride, toluene and sodium alcoholate are the three main raw materials to produce Triazophos, in which sodium alcoholate is made by the upstream processes. The synthesis unit comprises three main reactions, which are executed in 1., 2. and 3. reactors respectively. In 1. reactor, phenyl hydrazine reacts with urea to produce 1-phenylsemicarbazide. Its reaction equation is shown as follows:



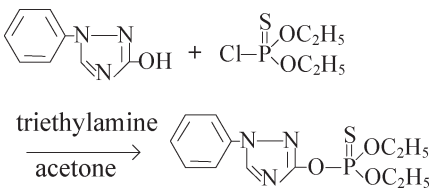
In 2. reactor, 1-phenyl-3-hydroxy-1,2,4-triazole is generated from 1-phenylsemicarbazide and formic acid. Its reaction equation is shown as follows:



F i g . 3 – *Simplified synthesis unit flow diagram in the Triazophos plant*



In 3. reactor, 1-phenyl-3-hydroxy-1,2,4-triazole reacts with O,O-diethylthiophosphoryl chloride to produce Triazophos with the help of triethylamine and acetone. Its reaction equation can be written as:



OVs are supposed to be the objective variables which need to be optimized and *OCs* are supposed to be the operating conditions which are manipulated variables and used to realize the optimization target.

The performance of each reactor relies a lot on its temperature. T_1, T_2 and T_3 shown in Fig. 3 denote the temperatures of three reactors respectively. According to process engineers' experiences, these three variables should be added to the set of operating conditions *OCs* due to their importance.

Additionally, the productivity of Triazophos also relies on the utilization factor of raw materials. Therefore, the flow rate of each kind of feedstock and the percentage of each kind of feedstock out of the flows of all the raw materials are very important for the sake of operation optimization. In Fig. 3, Q_A, Q_B and Q_C represent the flow rate of materials ethyl chloride, toluene and sodium alcoholate respectively. The proportion of different feedstock will be called P_F in the following discussion. According to the guidance of process experts, T_1, T_2, T_3, Q_A, Q_B, Q_C and P_F are the major variables which need to be optimized in order to improve the productivity of Triazophos. In other words, all of these variables should be selected into operating conditions *OCs*.

The objective variable of operation optimization selected in this paper is the productivity (conversion rate of key materials) of Triazophos, denoted by P_T. The target is to find better settings of operating conditions *OCs* in order to achieve higher P_T.

Data preprocessing

The effect of data mining technologies quite relies on the work of data preprocessing. In this section, basic and general work of data preprocessing is introduced first. Subsequently, two customized preprocessing steps are proposed to overcome two special problems, time delay and upstream-downstream relations.

Basic data preprocessing

The database of Triazophos plant is huge and there are many variables. However not every variable is closely related to the objective variable. In addition, not all of the instruments are always working correctly in the plant and some values may be wrong. So firstly, the original database should be made compact and precise through data preprocessing.

The basic preprocessing tasks include feature selection (also called variable selection) and data selection (including missing data and outliers detection).[2] In addition, due to errors of instruments there is a special preprocessing step called data reconciliation, which should be first taken into consideration in chemical engineering.

Data reconciliation. Data reconciliation is a method to improve the data quality in chemical processes. Gross error, which may be caused by malfunctioning instruments, measurement biases and process leaks, dose a lot of harm to the data quality. Literature 7 reviewed many methods about detecting gross errors. Fortunately, if the variable having a gross error is not the member of *OCs* or *OVs* selected by *feature selection*, it will not influence the accuracy of data mining. Otherwise, the data of selected variables having gross errors should be removed or adjusted.

Missing data processing. There are many methods to process missing data.[17(Chapter 3)] Missing value can be filled by process engineers, or replaced by a global constant or the mean attribute, or even the most probable value using regression or a Bayesian formalism. However, we choose to ignore the records with missing values directly, which is the simplest and most secure way when the percentage of such records is low.

Outlier detection. According to data distribution analysis[17(Chapter 8)] and expert suggestions, normal range of operation values can be evaluated approximately, which can be used to exclude noisy or invalid data further.

Feature selection. There may be thousands of sensors in a plant, so feature selection is necessary. Feature selection can be divided into two steps, selecting by expert experience and machine learning methods respectively.

According to the process engineers' analysis, most of the control variables that influence the final productivity lie in the stage of synthesis, shown in Fig. 3. P_T is the only variable selected into the set of objective variables *OVs*. It should be noted that variable P_T is also called *class* in the classification field as mentioned above. In addition, the operating conditions *OCs*, which are also called *attribute* in the classification field, are selected by process experts, including flow rate of material A (Q_A), flow rate of material B (Q_B), flow rate of material C (Q_C), heating temperature 1 (T_1), heating temperature 2 (T_2), heating temperature 3 (T_3), and proportion of material A to C (P_{AC}).

Table 1 – *Features selected into classification from Triazophos plant*

Classifier items	Selected features
OCs (attributes)	Q_A, Q_B, Q_C, T_1, T_2, T_3
OVs (classes)	P_T

The second step is to detect whether there is a smaller set of attributes whose information is the same with that of the full attribute set. A preferred set of attributes is the one that is highly correlated with the objective variable but has low inter-correlation. A data mining tool *Weka*[16(Part II)] is introduced here, which provides a function called *select attributes* to select a better feature set. The only thing you need to do is to choose *attribute evaluator* and *select method.*[16(Chapter 7)] As used herein, we choose *ClassifierSubsetEval* on *attribute evaluator* plus *BestFirst* on *select method*. The result shows that there is a better and smaller subset comprising Q_A, Q_B, Q_C, T_1, T_2, T_3. Indeed, P_{AC} is too stable to devote to classification. The final selected features are shown in Table 1.

Transformation to event-based data

$Q_A[t]$, $Q_B[t]$, $Q_C[t]$, $T_1[t]$, $T_2[t]$, $T_3[t]$ represent the sensors' instant value at sample time t, and are called *recorded data*. The delay between feeding and changes of T_1, T_2, T_3 for the same batch of material are supposed to be τ_1, τ_2, τ_3 respectively. It means the liquids that are fed to the system at time point t, will be heated at time $(t+\tau_1)$ in 1. reactor, at $(t+\tau_2)$ in 2. reactor and at $(t+\tau_3)$ in 3. reactor. Therefore, $Q_A[t]$, $Q_B[t]$, $Q_C[t]$, $T_1[t+\tau_1]$, $T_2[t+\tau_2]$ and $T_3[t+\tau_3]$ represent a series of events happening to the same batch of material, which are called *event-based data* herein. This group of valuables is denoted by $OCs[t]$ which stands for a series of operating conditions for the material fed to the system at time t. In addition, there is also a period between obtaining the productivity of Triazophos P_T and executing corresponding *OCs* for the same batch of material. Assuming that the period is n, then the real record of P_T related to operating conditions $OCs[t]$ in the original database is $P_T[t+n]$.

This step seems simple but it is very important. The optimization is fruitless if the recorded data are directly used, because data sampled at the same time do not represent the operating conditions for the same batch of material. The time constant τ_1, τ_2, τ_3 and n can be obtained from the system designer, process engineer or evaluated as follows:

$$\tau = Cap/Aver(Q_A + Q_B + Q_C) \tag{1}$$

where *Cap* represents the total capacity from feedstock to the corresponding reactors. For example, the *Cap* value is the sum of the capacities of feeding pipelines, 1. reactor and 2. reactor when evaluate τ_2. Additionally, $Aver(Q_A + Q_B + Q_C)$ denotes the average flow rate of the three kinds of feedstock.

Overcome upstream-downstream relations

Consider the following situation: a former batch of material fed at time t is still reacting in 1. reactor while the batch of material fed at time $(t+\varepsilon)$ enters into 1. reactor. It means that the operating condition $T_1[t+\tau_1+\varepsilon]$ relates not only to feedstock at time $(t+\varepsilon)$ purely, but also to all of the feedstock from t to $(t+\varepsilon)$. To some extent, such phenomena widely exist in continuous processes in which a variable of one time point is influenced by another variable in a time interval.

Consider another fact: the delays, like τ_2, may be up to several hours, whereas the sample time of DCS (Distributed Control System) may be only 1 minute. This may generate a big shift when recorded data is transformed into event-based data, even though only a tiny error happened to the evaluation of the delays.

Since the phenomena mentioned in above cases are caused by the relations of upstream/downstream processes and units, we call these upstream-downstream relations. These phenomena break the application worth of data mining optimization methods. In order to overcome these problems, a method integrating fuzzy concept and multiple classifiers is introduced here. Taking these phenomena into consideration, $P_T[t+n]$ is not decided only by single group of $OCs[t]$, but by all the operating conditions within time period $[t-r,\ t+r]$. As used herein, $2r$ is supposed to be the valid range, which may be calculated as follows:

$$2r = AverCap/Aver(Q_A + Q_B + Q_C) \qquad (2)$$

where *AverCap* is the average capacity of all reactors. It should be emphasized that even within the range the impact factors of the operating conditions at different time are different more or less. The principle to allot the weights to different time will be discussed in detail in the next section.

Algorithm of decision forest

In this paper, decision tree is chosen as the type of classifier. We use the idea of multiple classifiers to overcome the upstream-downstream relations. An algorithm called decision forest is proposed subsequently, which means multiple decision trees.

Decision tree

Decision tree is a kind of classical classification method in data mining. It is chosen as the classifier algorithm prototype in the framework of Fig. 2(b), because the importance of each attribute may be shown on decision tree directly.

Building a decision tree is a procedure in which the data sets are divided recursively. At the beginning, all the training data are in one set. Then the algorithm chooses the best attribute to split according to some criteria. Then the records are partitioned into two or more sets according to the values on this split attribute. Obviously, the data sets after splitting are purer than their parents. The operations are repeated until the requirement is satisfied or no more partition can be done. So the key point of decision tree is how to choose the splitting attribute and when to stop. Early works on decision tree try to give answers to these two questions. The representatives included CART,[9] ID3[10] and C4.5,[11] whose split criteria were maximum information gain (maximum entropy reduction), maximum information gain ratio and GINI Index respectively.

However, the early approaches are all memory-based ones, in which the whole data set must be kept in memory, but nowadays databases are becoming larger and larger. Therefore, the latest study began to concentrate on the scalability of algorithms. A series of scalable algorithms were proposed, such as SLIQ,[6] SPRINT[12] and RainForest.[13] These works made it possible to apply decision trees in the real world. Any of these algorithms may be chosen as a part of the decision forest algorithm described later.

Multiple classifier systems

The idea of multiple classifier systems (MCS) is developed to reduce the risk of the single classifier. MCS consist of an ensemble of different classifiers and a decision function for combining classifier outputs.[15] Therefore, MCS involves two main phases: the design of the classifier ensemble and the design of the combination function. Different classifiers are generated on different subsets from the whole original database. Bagging and boosting are the two main data sampling techniques.[8] There are many combination functions of arbitrary complexity,[14] from simple ones like the majority voting rule to complex ones like 'trainable' functions, which are all available. In this paper, the majority-voting rule is adopted to coordinate the multiple decision trees.

The total number of records in the original data set is supposed to be *N*. 'Bagging' picks up records randomly from the data set for *N* times, one record a time. Therefore, in the generated data set by bagging, there may be some repeated items, while others never appear. 'Boosting' does not do sampling on the data set. It gives different voting weights to different records according to some rules. This operation is done to the whole original data set several times. And every time we obtain a new training data set. Therefore, one record in different training sets may have different voting weights. Literature 8 pointed out that the precondition to show the advantages of bagging and boosting is the instability of the learning machine on the data set.

Decision forest algorithm

The algorithm called decision forest proposed in this paper is explained according to Triazophos processes, but it can be extended to other processes. Decision forest means a group of decision trees. The main processes of decision forest algorithm are as follows: first, preprocessing the original data according to the basic and customized data preprocessing methods introduced in the last section; second, constructing a series of decision trees based on different training datasets, wherein the differences lie on the setting value of parameter *n*, the time constant between $P_T[t+n]$ and $OCs[t]$; third, arranging weights to each decision tree by the majority voting rule; fourth, analyzing the results of decision forest.

It can be noted that the training datasets of the classifier ensemble here are generated according to practical meanings and following a rule, instead of 'bagging' or 'boosting' explained above, which are both based on statistics and with less actual meanings.

As discussed in Section *data preprocessing*, all the operating conditions in period $[t-r,\ t+r]$ are influencing the objective valuable $P_T[t+n]$ in the Triazophos plant. Therefore, the idea of MCS is adopted to make the mining results more rational by integrating the influences of different operating conditions within the range. In this paper, the method to generate subsets for multiple classifiers is different from traditional sampling technologies. Assuming $(2m+1)$ classifiers are used to make decision, *classifier* $(v_i,\ w_i)$ denotes the *i-th* classifier $(i \in N)$, where v_i represents the time shift in $[-r,\ r]$ and w_i is the weight of this classifier. *Classifier* $(v_i,\ w_i)$ is built on the new records comprising *attributes* $OCs[t+v_i]$ and *class* $P_T[t+n]$. It is assumed that the sample time of DCS is denoted by *samp* and the number of classifiers is $(2m+1)$, then the step length of time shift between neighbors denoted by p is calculated as follows:

$$p = fix(r/(m \cdot samp)) + 1 \tag{3}$$

where $fix(\cdot)$ is the rounding function. Assuming *classifier* $(v_i,\ w_i)$ arranges from left to right on $[-r,\ r]$ uniformly, then v_i is calculated as follows:

$$v_i = \begin{cases} -r, & i = 1 \\ (i-m-1)\cdot p, & 2 \le i \le 2m; \\ r, & i = 2m+1; \end{cases} \tag{4}$$

Obviously, the closer to the center of range $[-r,\ r]$ v_i get, the more the corresponding operating conditions $OCs[t+v_1]$ influence on the objective variable $P_T[t+n]$. Therefore, it can be seen that *classifier* $(v_{m+1},\ w_i)$ should have the highest weight while *classifier* $(v_i,\ w_i)$ and *classifier* $(v_{2m+1},\ w_i)$ should have the lowest one. Assuming that the weight descends in exponential function from center to two sides, then

$$w_i = \alpha^{|m+1-i|}, \text{ where } 0 < \alpha < 1 \tag{5}$$

The pseudocode of the algorithm to construct decision forest is shown in Fig. 4, where BuildTree(DB) may be any decision tree algorithm mentioned in the first subsection of this section.

A subsequent problem is how to make a decision on the generated multiple decision trees. The majority-voting rule is selected herein to make a decision from decision forest. Assuming there are q objective valuables in optimization problem, and objective variable i is divided into γ^i discrete values, the total distinct values of all objective valuables is denoted by γ, namely, there are γ classes. φ_j denotes the j-th class out of γ classes. It can be seen that

Algorithm: DecisionForest

Input: Database DB, m, n, r

Output: A *Forest*

Method:

1. Transform recorded data to event-based data, then obtain new records: $\{OCs[t], OVs[t+n]\}$;
2. $Forest = \emptyset$, New_DB=empty;
3. **For** i=1 to (2m+1)
4. Compute v_i and w_i using Eq.(3)-(5);
5. **For** t=1 to (length(DB)-v_i)
6. New_record=$\{OCs[t+v_i], OVs[t+n]\}$;
7. New_DB=New_DB+New_record;
8. **End**;
9. $Classifier(v_i, w_i)$=**BuildTree(New_DB)**;
10. Add $Classifier(v_i, w_i)$ into $Forest$;
11. New_DB=empty;
12. **End**;
13. **Return** $Forest$;

Fig. 4 – *Algorithm to construct decision forest*

$$\gamma = \prod_{i=1}^{q} \gamma^i \tag{6}$$

To instance Z,

$$assign\ \ Z \to \varphi_j \quad if$$

$$\sum_{i=1}^{2m+1} \Delta_{ji} = \max_{k=1}^{\gamma} \sum_{i=1}^{2m+1} \Delta_{ki}, \quad where \tag{7}$$

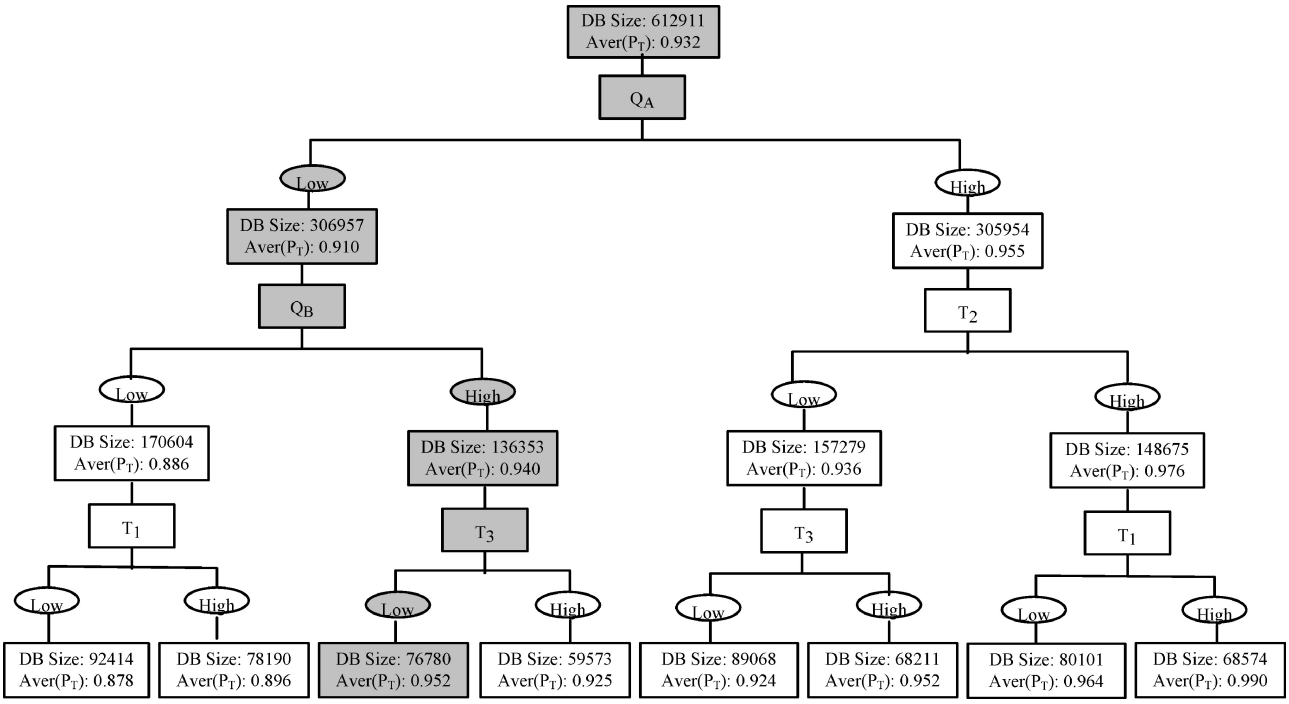
$$\Delta_{ki} = \begin{cases} 1, if\ \ Z \to \varphi_k\ \ on\ Classifier(v_i, w_i); \\ 0, if\ \ Z \not\to \varphi_k\ \ on\ Classifier(v_i, w_i); \end{cases} \tag{8}$$

It means that for a given group of operating conditions the future state of objective variables is predicted to belong to *class j* if most of the decision trees arrange it to *class j*.

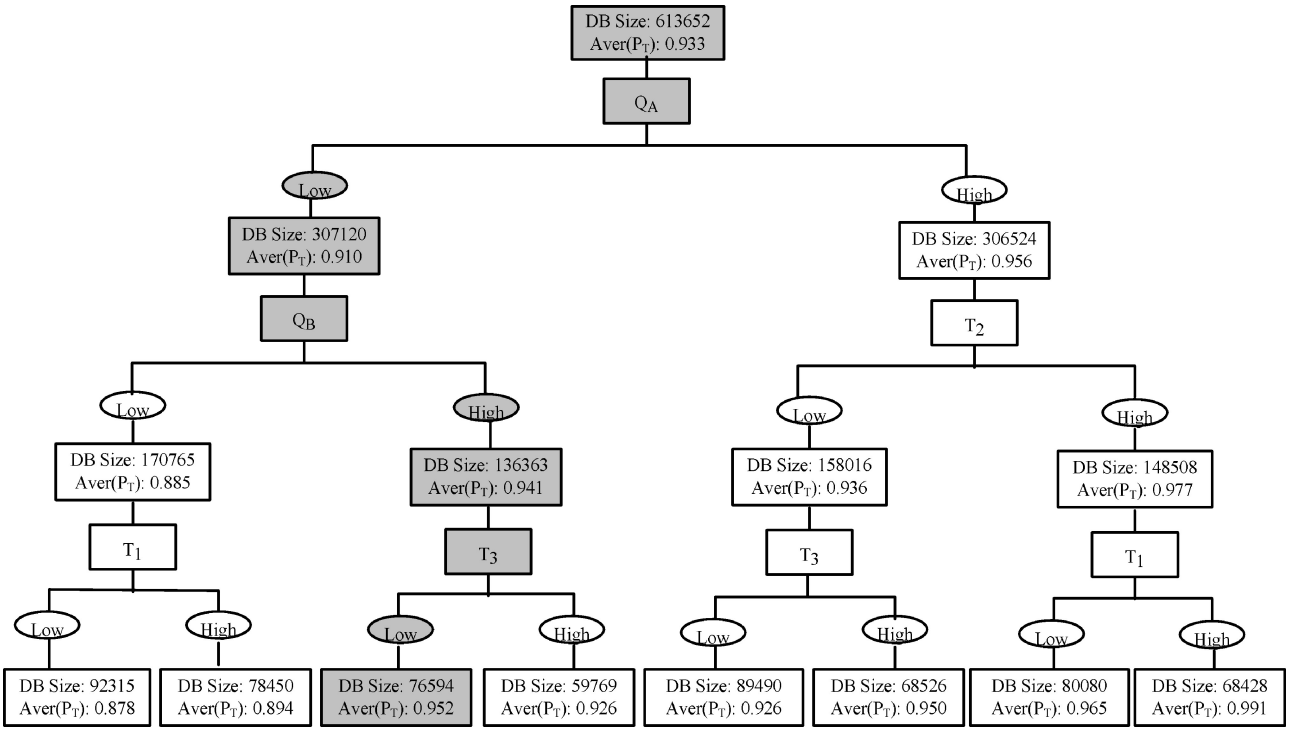
Practice in Triazophos plant

The classification framework proposed in this paper is applied to a Triazophos plant, Zhejiang Xinnong Chemical Co., Ltd., which had a manufacturing capacity of 10,000 tons of raw drugs annually. We preprocessed raw data and constructed decision forest in the means mentioned above, and wherein we select scalable algorithm SLIQ[6] as the algorithm of BuildTree(DB) in Fig. 4. It should be noted that the attributes and classes in SLIQ are discrete value. Therefore the continuous values of *OCs* and *OVs* should be split into discrete intervals. As

used herein, each variable in *OCs* and *OVs* is split into two parts, one part is bigger than average and denoted by 1, another is smaller than average and denoted by 0. Obviously, the objective of optimization is to promote P_T as much as possible, and the classification framework is used to discover the corresponding operating conditions with the high P_T. In this plant, n is approximated to 10 hours and r is around 1 hour. The results of three decision trees are shown in Fig. 6, Fig. 5 and Fig. 7 when $m = 1$, and the time shifts set as $v_1 = -0.5$ h, $v_2 = 0$ h and $v_3 = 0.5$ h respectively. In these figures, a bigger rectangle means a data subset, *DB Size* and *Aver(P_T)* are two features of this subset. *DB Size* in-



F i g . 5 – *Decision tree of v = 0 h*



F i g . 6 – *Decision tree of v = −0.5 h*

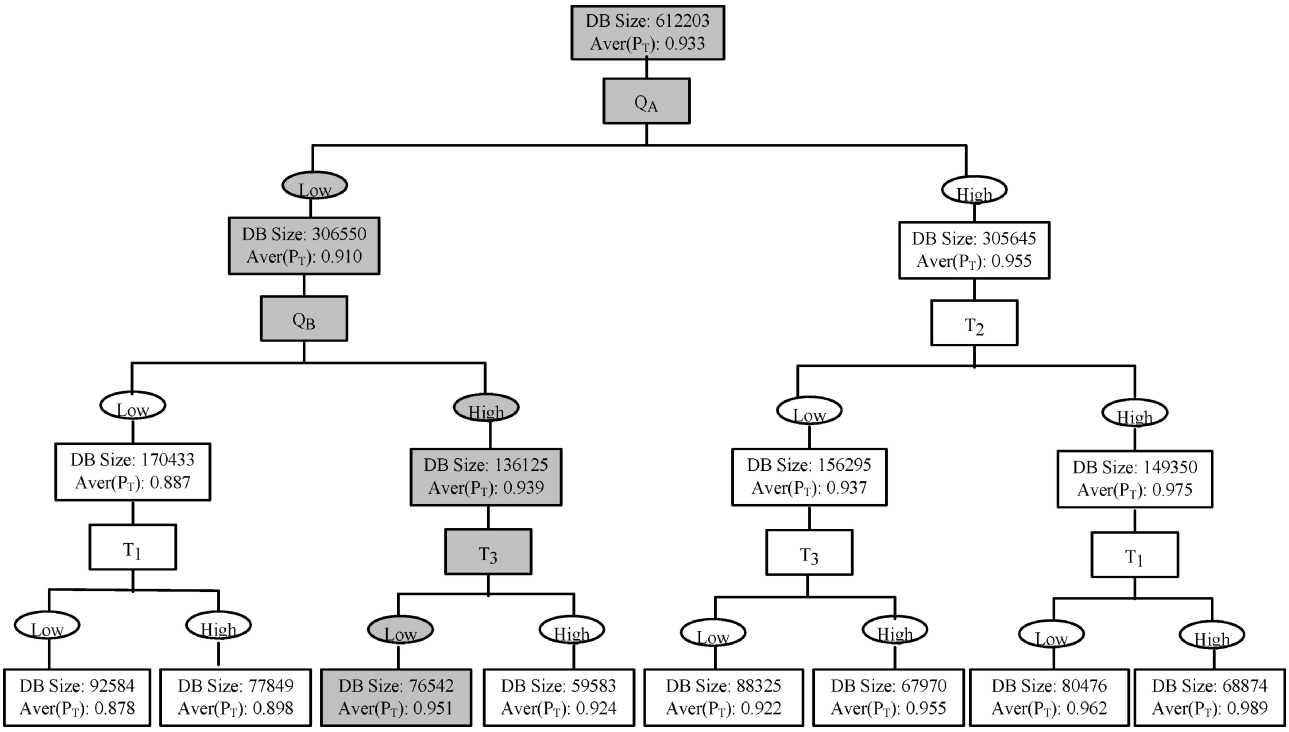


Fig. 7 – *Decision tree of* $v = 0.5$ *h*

dicates how many records are included in this subset, while *Aver(*P_T*)* gives the average conversion rate of raw material in this subset. The smaller rectangle means the selected attribute to be split. The value in ellipses represents that the sub nodes of it all locate in this discrete interval. Taking Fig. 5 as example, the number of records reaches 612,911 in the database of $v_2 = 0$ h and the average P_T equals 0.932, the best splitting variable is Q_A by gini index,[9] then the database are divided into two parts, which are $Q_A = low$ and $Q_A = high$. There are 306,957 records in the subset in which $Q_A = low$, and 305,954 in which $Q_A = high$. Then both the two nodes will be split in the same way, one by one, a tree is built following that way. Every node in the tree can generate a rule except the root node. The rule corresponding to the leaf node is called a complete rule in the tree.

The trees differ a little from each other as shown in the three figures, which demonstrate that the data mining results are very confident under the range of error tolerance of v. In this situation, the result of the majority-voting rule is near that of the single classifier, however, it is still necessary to adopt MCS in order to guarantee the decision security.

Assuming $\alpha = 0.5$, then $w_1 = w_3 = 0.5$ and $w_2 = 1$ following eq. (5). From Figs. 5, 6 and 7, it can be seen that the structures of all the triple trees are the same. However, the average P_T and the frequency of each rule are different from forest and

Table 2 – *Decision rules with high* P_T *in decision forest*

1	Q_A = low, Q_B = high, T_3 = low ⇒ $P_T = 0.952$	*Sup* = 12.5 %
2	Q_A = high, T_2 = low, T_3 = high ⇒ $P_T = 0.952$	*Sup* = 11.1 %
3	Q_A = high, T_2 = high, T_1 = high ⇒ $P_T = 0.990$	*Sup* = 11.2 %
4	Q_A = high, ⇒ $P_T = 0.955$	*Sup* = 49.9 %
5	Q_A = high, T_2 = high ⇒ $P_T = 0.990$	*Sup* = 24.3 %

trees. We can find three useful rules 1, 2 and 3 shown in table 2 for improving P_T, which is generated and combined from triple trees. Beyond the complete rules, there are two other rules A and B. In this table, *Sup* means the percentage of records supporting this rule. The P_T and *Sup* of rule k are calculated as follows:

$$P_T(k) = \frac{\sum_{i=1}^{2m+1} (w_i \sum_{j=1}^{f_{ki}} P_T / All\,\mathrm{Re}\,c(i))}{\sum_{i=1}^{2m+1} w_i}$$

$$Sup(k) = \frac{\sum_{i=1}^{2m+1} (w_i \cdot f_{ki} / All\,\mathrm{Re}\,c(i))}{\sum_{i=1}^{2m+1} w_i} \qquad (9)$$

where f_{ki} is the count of records supporting rule k in tree i, and $All\,\mathrm{Re}\,c(i)$ is the total records of tree i. Rule A is an integrated form of rules 2 and rules

3, while rule 3 is also included in rule B. It can be noted that the average of P_T ascend one by one from rule A to rule B, until rule 3, which are extracted from the root down to the leaf on the same branch. It can also be seen that the closer to the leaf, the lower the *Sup* of nodes will be. However, the support value *Sup* of rule 3 is already high enough. Generally speaking, the complete rule is more effective and worthy of admission than incomplete one, as long as the support value *Sup* of complete one is high enough to be accepted.

Table 3 – *Solutions to improve productivity of Triazophos*

1:	decrease Q_A, increase Q_B, decrease T_3
2:	increase Q_A, decrease T_2, increase T_3
3:	increase Q_A, increase T_2, increase T_1

In rules 1, 2 and 3, the values of *class* P_T are all labeled with 'high'. Accordingly, there are three solutions can be extracted to improve productivity of Triazophos, as shown in table 3.

Therefore, there are three different solutions to be considered, and they cannot be integrated as a single solution because directions of the same variable sometimes are different, like T_2 in solution 2 and solution 3. We should choose the best solution among the three through cooperation with process engineers. Firstly, solution 1 is the best when considering cost. It costs no more than current solution, because in this solution the only thing that should be increased is material B, and B is a recyclable material in Triazophos plant. Theoretically, solution 2 can achieve the same conversion rate as solution 1, but it needs to increase feed of material A and temperature of 3. reactor, which makes the solution cost more. Although solution 3 can achieve the highest conversion rate in theory, it is also the most costly one. It needs to increase the amount of steams of 1. and 2. reactors to heighten their temperatures, and increase the feed of material A at the same time. These costs may be very high, and they may be beyond the benefit from the improvement of P_T. So solution 1 is the most competitive one, which is effective and economic.

According to the selected solution, process engineers modified settings of some operation variables to the average of the records related with rule 1. Repeat processes of producing, collecting data, classifying and modifying parameters as Fig. 1 month by month, then the settings move to the right direction step by step.

Table 4 – *Change of operating variables over one year*

	Q_A	Q_B	Q_C	T_1	T_2	T_3	P_T
Previous	489	929	1426	65	70	72	0.933
Current	369	1002	1097	62	68	70	0.958

The variables that have been modified over one year were listed in Table 4. Compared to the values of last year, Q_A was decreased by 24 %, Q_B increased by 8 %, and Q_C decreased by 23 %, while the proportions (P_F) keep the same. T_1, T_2 and T_3 were all decreased a little. As a result, the conversion rate of the expensive material sodium benzoxazolate has been improved a lot, from last year's 93 % to today's 95 %, and the productivity of Triazophos ascend from 93.3 % to 95.8 %. This case demonstrated the efficiency of the classification framework for Triazophos plant optimization.

Conclusions

The design values are usually not the real optimal ones that a real plant can achieve, due to incomplete modeling and assumptions during mathematical programming. A closed-loop data mining strategy looking like feedback control systems is proposed in this paper, as a complement for mathematical programming, driving the settings of operating conditions to optimal step by step in practical Triazophos plant. A customized classification framework for process operation optimization is also proposed, which is the core of the rolling optimization strategy. Chooses process variables into two parts, one is objective variables, another is operating conditions, and the purpose is to find in which conditions the objective variables can reach the state we expected. As you know, time delay and upstream-downstream relations generally exist in chemical industries, which are ignored in previous studies of data mining application to process optimization. Before classifying, a lot of work involving with data preprocessing have to do first. Especially, recorded data must transform into event-based data to make the data in a new record be interrelated in order to overcome time delay. One record of objective variables is actually influenced by a group of records of operating conditions instead of by one record of that when take upstream-downstream relations into consideration. Therefore, we construct a group of decision trees, so called decision forest, to make a decision by changing the time shift between objective variables and operating conditions. Training datasets for every decision tree are organized in a meaningful and time-related way,

which differ from common sampling technology like bagging and boosting. The stepwise closed-loop strategy and customized classification framework have been applied to a Triazophos plant, and the practical results over one year demonstrate the efficiency of the proposed method.

ACKNOWLEDGEMENT

This work is supported by National Natural Science Foundation of China (60421002) and The National High Technology Research and Development Program of China (863 Program) (2007AA04Z191).

Assistance of engineers of Zhejiang Xinnong Chemical Co., Ltd is also gratefully acknowledged.

Nomenclature

OCs – set of operating conditions
OVs – set of objective valuables
P_F – proportion of different fluids
Q_A – volume flow rate of ethyl chloride, L h^{-1}
Q_B – volume flow rate of toluene, L h^{-1}
Q_C – volume flow rate of sodium benzoxazolate, L h^{-1}
T_1 – temperature of 1. reactor, °C
T_2 – temperature of 2. reactor, °C
T_3 – temperature of 3. reactor, °C
P_T – productivity of Triazophos
P_{AC} – proportion of ethyl chloride to sodium benzoxazolate
n – producing time from raw material to product
r – time range influencing current point
w – weight of multiple classifiers
Sup – support of a rule
p – time interval between neighboring classifiers

Greeks

τ – time delay
υ – offset on time dimension of multiple classifiers
φ – class label

References

1. *Suh, M. S., Jhee, W. C., Ko, Y. K., Lee, A.,* Expert Syst. Appl. **15** (1998) 181.
2. *Papadokonstantakis, S., Machefer, S., Schnitzlein, K., Lygeros, A. I.,* Comput. Chem. Eng. **29** (2005) 1647.
3. *Chiang, L. H., Kotanchek, M. E., Kordon, A. K.,* Comput. Chem. Eng. **28** (2004) 1389.
4. *Yamashita, Y.,* Comput. Chem. Eng. **24** (2000) 471.
5. *Zhang, Y., Su, H. Y., Chu, J.,* Chin. J. Chem. Eng. **13** (2005) 751.
6. *Mehta, M., Agrawal, R., Rissanen, J.*, Lecture Notes In Computer Science **1057** (1996) 18.
7. *Crowe, C. M.,* J. Process Control. **6** (1996) 89.
8. *Quinlan, J. R.,* Proceedings of AAAI Conference on Artificial Intelligence, Portland, 1996, pp 725.
9. *Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.,* Classification and Regression Trees. Belmont, CA: Wadsworth, 1984.
10. *Quilan, J. R.,* Mach. Learn. **1** (1986) 81.
11. Quilan, J. R., C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann, 1993.
12. *Shafer, J., Agralwal, R., Mehta, M.*, Proceedings of the 22nd VLDB Conference Bombay, India, 1996, pp 544-555.
13. *Gehrke, J., Ramakrishnan, R., Ganti, V.,* Data Min. Knowl. Discov. **4** (2000) 127.
14. *Kittler, J., Hatef, M., Duin, R. P. W., Matas, J.,* IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998) 226.
15. *Roli, F., Giacinto, G., Vernazza, G.,* Lecture Notes In Computer Science **2096** (2001) 78.
16. *Witten, I. H., Frank, E.*, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann, San Francisco, 2005. (http://www.cs.waikato.ac.nz/~ml/weka/index.html)
17. *Han, J., Kamber, M.,* Data Mining: Concepts and Techniques. Academic Press, Morgan Kaufmann Publishers, San Francisco, 2000.