Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Machine learning approach to design of complex distillation columns

P.K. Mogili & A.K. Sunol
Department of Chemical Engineering, University of South Florida, Tampa FL 33620 USA

Abstract

Automation of the knowledge acquisition process in building knowledge-based systems for process design is addressed through Machine Learning techniques. A hybrid Machine Learning algorithm developed at the University of South Florida is presented as a knowledge acquisition tool for developing knowledge-based systems. The learning algorithm addresses the knowledge acquisition problem by developing and maintaining the knowledge base through inductive learning from the examples.

The learning algorithm named as Symbolic-Connectionist net (SC-net), overcomes the problems associated with neural and symbolic learning systems by integrating the symbolic information into a neural network representation. The learning system allows for knowledge extraction and background knowledge encoding in the form of rules. Fuzzy logic has been made use of in dealing with uncertainty in the learning domain. The description language for the learning system consists of continuous and discrete variables along with relational and fuzzy comparators.

The applicability of the learning system for process design is illustrated through a complex column sequencing example. The performance of the learning system is discussed in terms of the knowledge extracted from example cases and its classification accuracy on the test cases.

1 Introduction

The development of knowledge-based systems in the field of process design and analysis has been limited by the knowledge acquisition bottleneck. The domain knowledge is mostly in the form of heuristics, case histories, expert knowledge and domain texts resulting in a diverse and unstructured body of knowledge. Automation of knowledge acquisition process would not only simplify the knowledge engineer's task but also helps in formalizing knowledge and implementing knowledge bases for large Artificial Intelligence(AI) systems.

In developing AI based systems for process design, automation of process synthesis plays a crucial role, as this step involves most complex and creative tasks. Different types of problems involved in the chemical process synthesis have been addressed by Hendry [1], Westerberg [2]. In this study we limit ourselves to complex column sequencing which is a subclass of more general separation scheme synthesis problem. Complex column sequencing problem has been addressed by many researchers in the past by algorithmic or optimization approach, Hendry [3], heuristic approach, Tedder [4], evolutionary approach, Stephanopoulas [5]. These approaches are not totally exclusive as some methods combine two or more of them, Seader [6].

In this study we have taken the approach of extracting knowledge in the form of rules from example cases using machine learning techniques, which roughly falls in between the boundaries of heuristic and evolutionary approaches. Machine learning is defined here as "a process in which a computer program improves its performance, acquires knowledge and solves new problems in the specified domain".

Over the years, research in machine learning has mainly focused on three different approaches:

- neural modeling and decision-theoretic techniques
- symbolic concept-oriented techniques
- knowledge-intensive, domain-specific learning

these approaches are discussed in detail elsewhere, Michalski [7].

Neural learning (connectionist) systems, with their massive parallelism and robustness have proven effective in many of the pattern classification problems. But they are unsuitable for knowledge acquisition purposes as they do not allow for knowledge extraction in the form of rules and require very long training times. On the other hand, symbolic learning systems can extract knowledge in the form of rules but they are very sensitive to noise and tend to over generalize the knowledge. The problems associated

with these systems have prompted researchers in machine learning to seek a hybrid approach. One such development in this direction is SC-net, Romaniuk [8]. SC-net is designed specifically for the purpose of developing knowledge bases and is based on a hybrid Symbolic and Connectionist architecture.

2 Learning in SC-net

Different learning steps involved in SC-net are shown in Figure 1. Each time SC-net is presented with a training set consisting of input and corresponding output information it invokes a call to Recruitment of Cells Algorithm (RCA). Its main function is to map the training set into a network representation so that they may be used for further processing.

The network generated consists of Input Cells, Information Collector (IC) Cells , Negative Collector(NC) Cells , Positive Collector(PC) Cells, Unknown(UK) Cells, and Output Cells as shown in Figure 2. The input cells represent the input information presented in the training set, the IC cells combine the input information into an intermediate form. The NC and PC cells collect negative and positive evidence present in the training set for a conclusion respectively. These cells are connected to every output and IC cell. The UK cell acts as a threshold on the NC and PC cells, letting them propagate an activation which is representative of the type of evidence present(positive, negative and unknown). The output cells represent the output information from the training set.

The cells in the network model the min, max and sum operators of fuzzy logic, Zadeh [9]. Every cell contains a bias value CB_i, which indicates the type of fuzzy operator the cell models and its value lies between -1 and $+1$. A min cell has a negative bias value, negate cell has bias value zero and a max cell has a positive bias value. The absolute value of CB_i represents an upper threshold on the cell activation CA_i. The cells in turn are connected by links which carry link weights. If cell C_i (with CA_i) and cell C_j (with CA_j) are connected then the weight of the connecting link is CW_{ij}. Every cell C_i with a cell activation CA_i (except for input cells) computes its new cell activation CA_i according to the formula

$$CA_i' = \begin{cases} |CB_i| * \min\limits_{\substack{j \\ j \neq i}} \{CA_j * CW_{i,j}\} & \text{if } C_i \text{ is a min cell} \\ |CB_i| * \max\limits_{\substack{j \\ j \neq i}} \{CA_j * CW_{i,j}\} & \text{if } C_i \text{ is a max cell} \\ |CB_i| * \sum\limits_{\substack{j \\ j \neq i}}^{j \neq i} \{CA_j * CW_{i,j}\} & \text{if } C_i \text{ is a linear threshold cell} \\ CA_{i_{positive}} + CA_{i_{negative}} - \frac{1}{2} & \text{if } C_i \text{ is a final output cell} \\ 1 - CA_j * CW_{i,j} & \text{if } C_i \text{ is a negate cell} \end{cases}$$

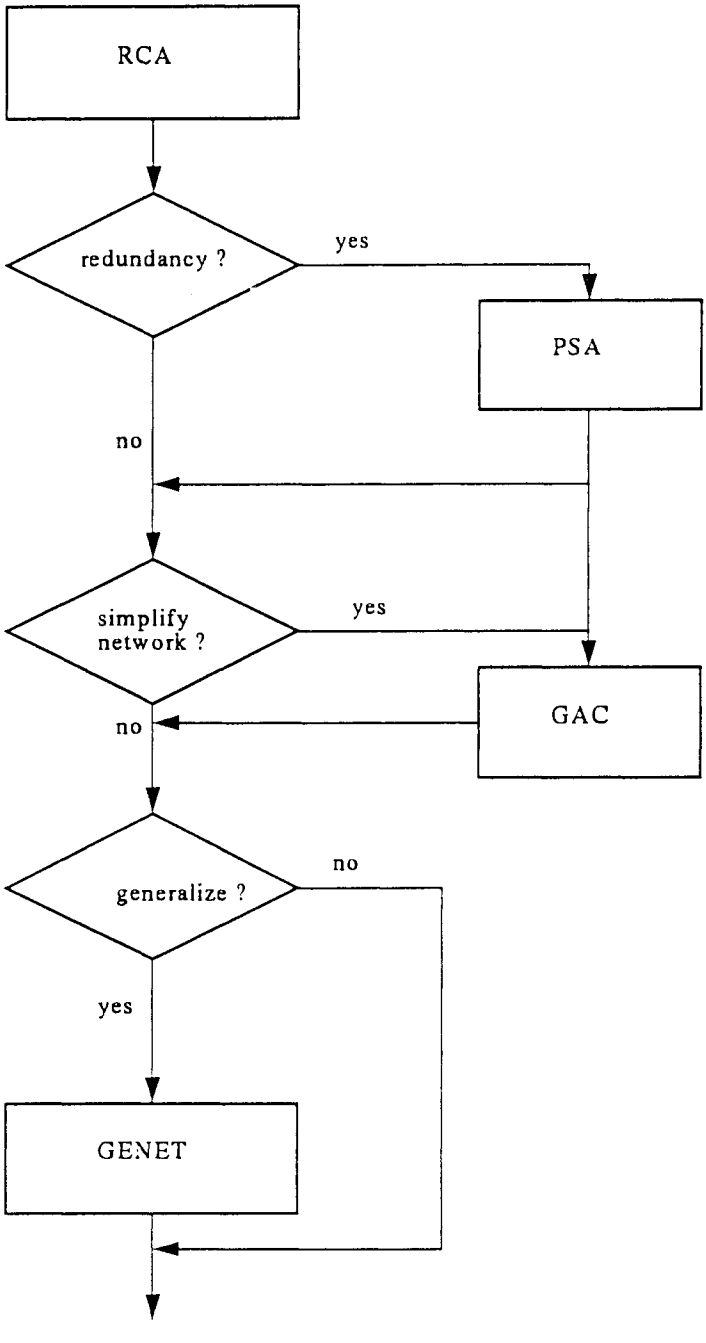

Figure 1. Learning in SC-net

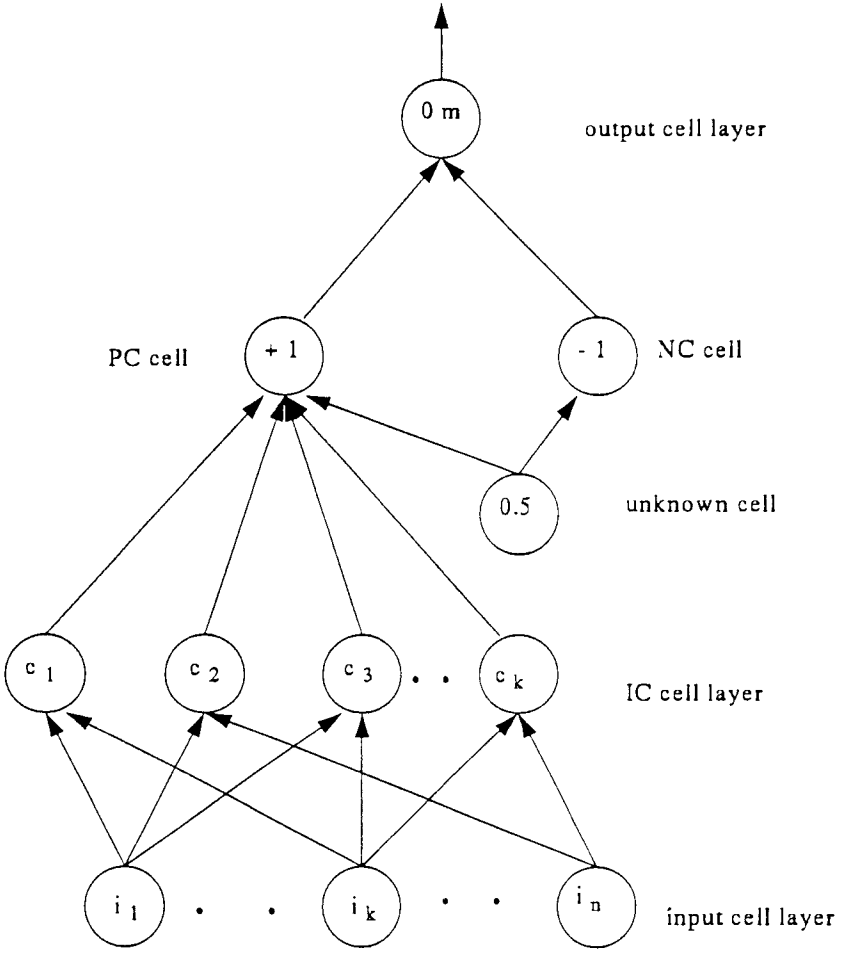

Figure 2. Typical SC-net structure

A cell activation value of 1.0 indicates the complete presence of a fact and/or a conclusion, whereas a value of 0.0 indicates complete absence and finally a value of 0.5 indicates unknown. The last two formulas(final output cell and negate cell) are special cases of the third formula(linear threshold cell), leaving only three basic operators: min, max and sum.

Every training instance, consisting of input cell activations and the expected output cell activations is presented to the network for a single feed forward pass. After the pass has been completed, the actual and the expected activation for every output are compared. Three possible conditions may result from this comparison :

1. The difference is below some epsilon($\epsilon = 0.02$). Then the example is already learned and no modifications are made to the network.

2. The difference is within 5ϵ, then the example is similar to at least one previously seen and stored instance. The cell bias value is adjusted to all the IC cells connected to PC cell if the expected output cell activation is above 0.5 and to all the IC cells connected to NC cell if the expected output cell activation is below 0.5. This procedure is continued until the difference falls below ϵ.

3. The difference is above 5ϵ, this means that the example is new and the network is modified by recruiting a new cell to include this example. The new cell is then connected to appropriate input cells.

The network generated by the above procedure is next presented to the Pre-Selection of Attributes(PSA) algorithm for introducing redundancy in to the network. Since the RCA generated network aims toward a minimal representation of the input domain, it is possible that the network identifies the wrong input features as important. This is especially true in the case of insufficient number of training instances. PSA algorithm addresses this problem by selecting all the input attributes that represent a concept. In many empirical studies, Romaniuk [8], it was found that PSA can improve the classification accuracy in pattern-matching type of problems.

The network topology in SC-net is based on the training examples, unlike the user specified ones in other connectionist approaches. The cells are recruited at a single level, the cell growth proceeds horizontally making it an ideal candidate for parallelization. But under worst case conditions, the network growth is linear, resulting in a recruitment of one to three new cells for each example. This problem can be controlled by including some guiding knowledge in the training set. The learning system allows for injection of background knowledge in the form of *if...then* rules. The learning system makes use of the fuzzy logic language hedges(*low, medium,high*),

quantifiers(*never,sometimes,always*) and operators(*max, min*) in developing a more natural language like description language. As a next step in the learning process the PSA modified network is presented to the Global Attribute Covering (GAC) algorithm, whose main function is to minimize the network growth.

Given the RCA (with/without PSA) generated network as its input GAC algorithm generates (except for contradictions and inconsistencies in the examples) an equivalent network, minimized by both in the number of cells and links. As a first step in GAC, all links to IC cells are disconnected, by making $CW_{ij} = 0$, this forces all IC cells to propagate an activation of 1 (firing state) regardless of the inputs presented to the network during the training phase. Since all IC cells fire they will also activate the corresponding output cells they are connected to. Since it is known when a concept should be activated (output cell activation above some threshold value) for a given input set, one can easily determine those IC cells that fired but should not have according to the given input assignment. Whenever an IC cell is incorrectly activated it will be entered into a conflict list. Further all links entering the IC cell as inputs will be considered as potential inhibitors for network minimization. By reconnecting specific links, IC cells can be prevented from firing for the same inputs. A badness measure associated with each link is used to prune the links, the details of the algorithm may be found in, Romaniuk [8].

The GAC generated network is next generalized through the use of the GEneralization of NETwork (GENET) learning component. In symbolic terms, generalization is accomplished by replacing single conditions by a disjunction of similar conditions, which allows extraction of rules covering a larger domain.

Once the learning system completes all the above steps the network generated is capable of classifying new example cases in the specified domain and allows for knowledge extraction in the form of rules. The knowledge can be extracted from the trained network by making use of the symbolic information collected by IC cells. As these cells combine the input concepts and connect to an output cell through a NC or a PC cell depending on the type of evidence present, it is easy to determine which of the input concepts are contributing to a particular output concept. The rules are generated by taking the conjunction of the input concepts to the IC cells as the premise and the output cell concepts as the conclusion.

The learning system is truly hybrid in nature because it has the virtues of highly parallel and uniform knowledge representation, ability to deal with continuous variables, fault tolerance and noise resistance from the connectionist approach. From the symbolic side it has the advantages of knowledge extraction, powerful symbolic description language and background knowledge injection to support knowledge refinement. The next

section describes the application of the learning system as a knowledge acquisition tool for a process design problem.

3 Example

The learning system's knowledge acquisition capability and classification accuracy is illustrated through the selection of a complex column sequence example. The learning system is expected to select a type of sequence for a given set of specifications and to extract the knowledge in the form of rules, after learning the concepts from examples. In this study, we have considered the well known complex column sequences like Direct/Indirect, Sidestream Stripper/Rectifier, Petlyuk columns and Sidestream columns as shown in Figure 3.

Initially, the learning program is presented with example cases which consist of the information required to select a complex column sequence. These examples are generated through the use of ASPEN [10] simulator, which selects the best possible sequence for a given input specification based on Total Annualized Cost, Douglas [11] for each possible sequence. In generating the input information we have made use of the design heuristics proposed, Tedder [4], Glinos [12] in the literature.

The training examples generated from the ASPEN simulator are presented to the SC-net program as a training set. Each example in the training set has input information and output information. The input information consists of relative volatilities, product fractions in the feed and product purity. The output information consists of optimal design sequence for the given input information. A sample training set is shown in Table-I, the following notation is used in the table :

- *Input Information*

  α_{AB} relative volatility of component A w.r.t. B.
  α_{BC} relative volatility of component B w.r.t. C.
  MP percentage of the middle product in the feed.
  OH percentage of the overhead product in the feed.
  BP percentage of the bottoms product in the feed.
  PP Product purity desired.

- *Output Information*

  DESIGN The type of sequence (D1,D2,D3,D4,D5,D6,D7).

The input variables can take on values like close, wide, low, high etc., as shown in Table-I. These values are represented as linguistic variables for generating knowledge which is general and easily understandable/modifiable by a domain expert to incorporate into a knowledge base.

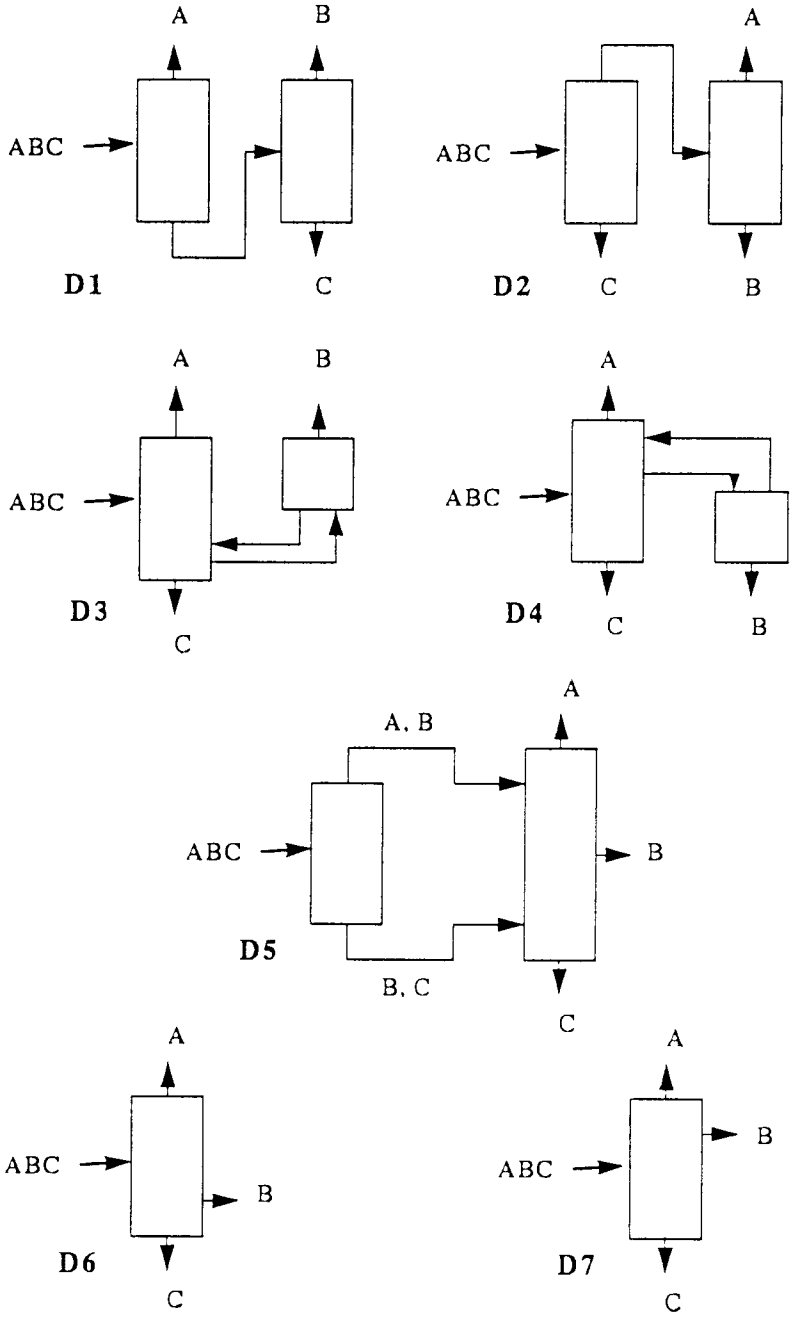

Figure 3. Complex Column Sequences

Table I Train set for Column Sequencing

α_{AB}	α_{BC}	MP	OH	BP	PP	DESIGN
close	near	medium	low	very_low	high	D1
close	close	medium	medium	very_low	high	D1
wide	close	medium	high	medium	high	D1
near	near	medium	very_low	high	high	D2
wide	close	medium	medium	high	high	D2
close	close	low	medium	medium	medium	D3
wide	close	medium	high	low	medium	D3
near	close	medium	low	high	medium	D4
wide	close	medium	low	high	medium	D4
close	near	very_high	very_low	very_low	high	D5
near	wide	high	low	low	high	D5
wide	close	high	medium	low	high	D5
close	close	high	medium	very_low	low	D6
wide	wide	high	low	very_low	low	D6
near	close	high	very_low	low	low	D7
wide	near	high	very_low	medium	low	D7

These variables are modeled using pi-shaped fuzzy numbers, Zimmermann [13]. Figure 4 shows their range and membership values. The qualitative information such as close-boiling mixtures or low middle-product can easily be represented as α[close], MP[low].

The learning algorithm was trained with 64 example cases and the time taken for training is 20 seconds of CPU on a SUN *SPARC* station. Figure 5 shows the network generated by SC-net for this training set, the input cell layer and the interconnection network consists of the network created to represent the fuzzy input variables along with each of their values. The output cells from the interconnection network will be the input variables with their values presented in the example cases. Only a few of them are shown in the Figure 5.

The knowledge extracted in the form of *if...then* rules from the training set is shown in Table-II. Consider for example the rules generated for the sidestream columns (D6 and D7). These two rules identify that sidestream columns are to be chosen when low purity products are desired and also that the location of sidedraw to be below the feed (D6), if very low amounts of bottom product is present in the feed and to be above the feed (D7), if the amount of overhead products present in the feed is very low. The extracted rules are very similar to the design heuristics proposed in the literature, Tedder [4], Glinos [12] by human experts, in terms of their simplicity and accuracy.

To evaluate the classification accuracy, the learning system was tested with 26 new test cases apart from the one used for training. Table-III shows the performance of SC-net. The learning system was able to classify all the cases correctly. This high degree of classification accuracy can be attributed to the general nature of the rules and hybrid nature of the learning system. The system has been found, Romaniuk[8] to have a classification accuracy of the order of 95 to 99 % in many of the standard classification problems.

4 Conclusion

A hybrid learning mechanism has been studied and its applicability in process design is illustrated through the complex column sequencing problem. The learning system was successful in extracting knowledge in the form of natural language like rules. The extracted rules were general in nature and performed well on classification problem. The high degree of classification accuracy of the learning system on the studied test cases make it a suitable system for studying structured selection and fault diagnosis problems. The learning system can be used as a powerful knowledge acquisition tool for building knowledge-based systems for domains with a large body of

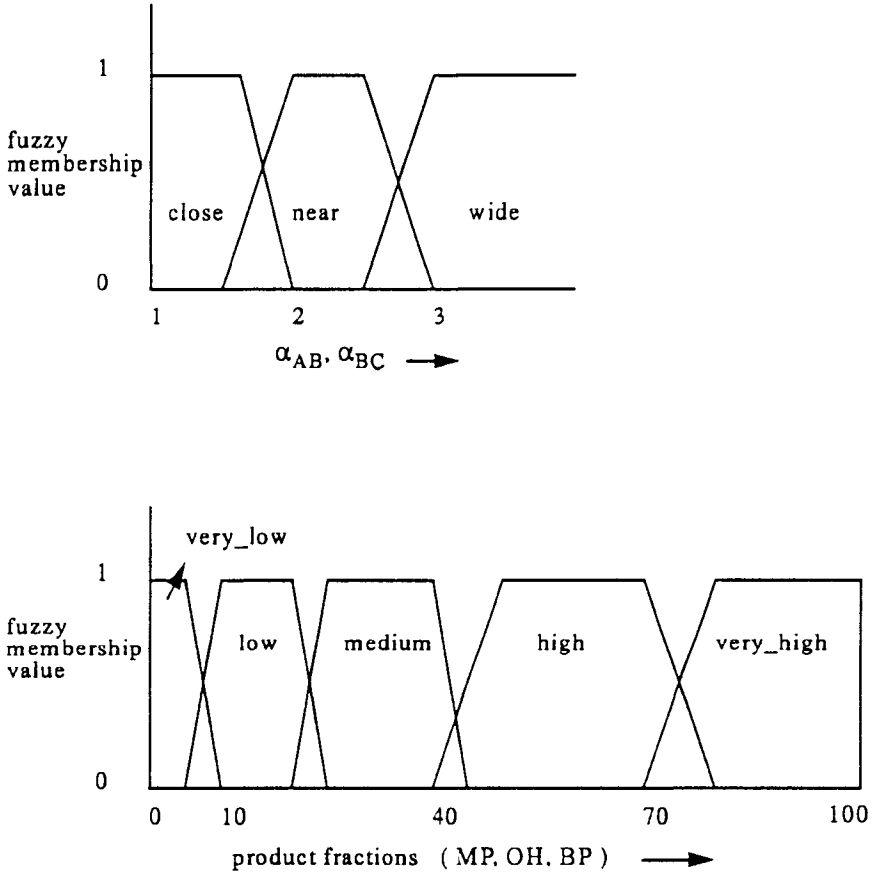

Figure 4. Representation of the input variables

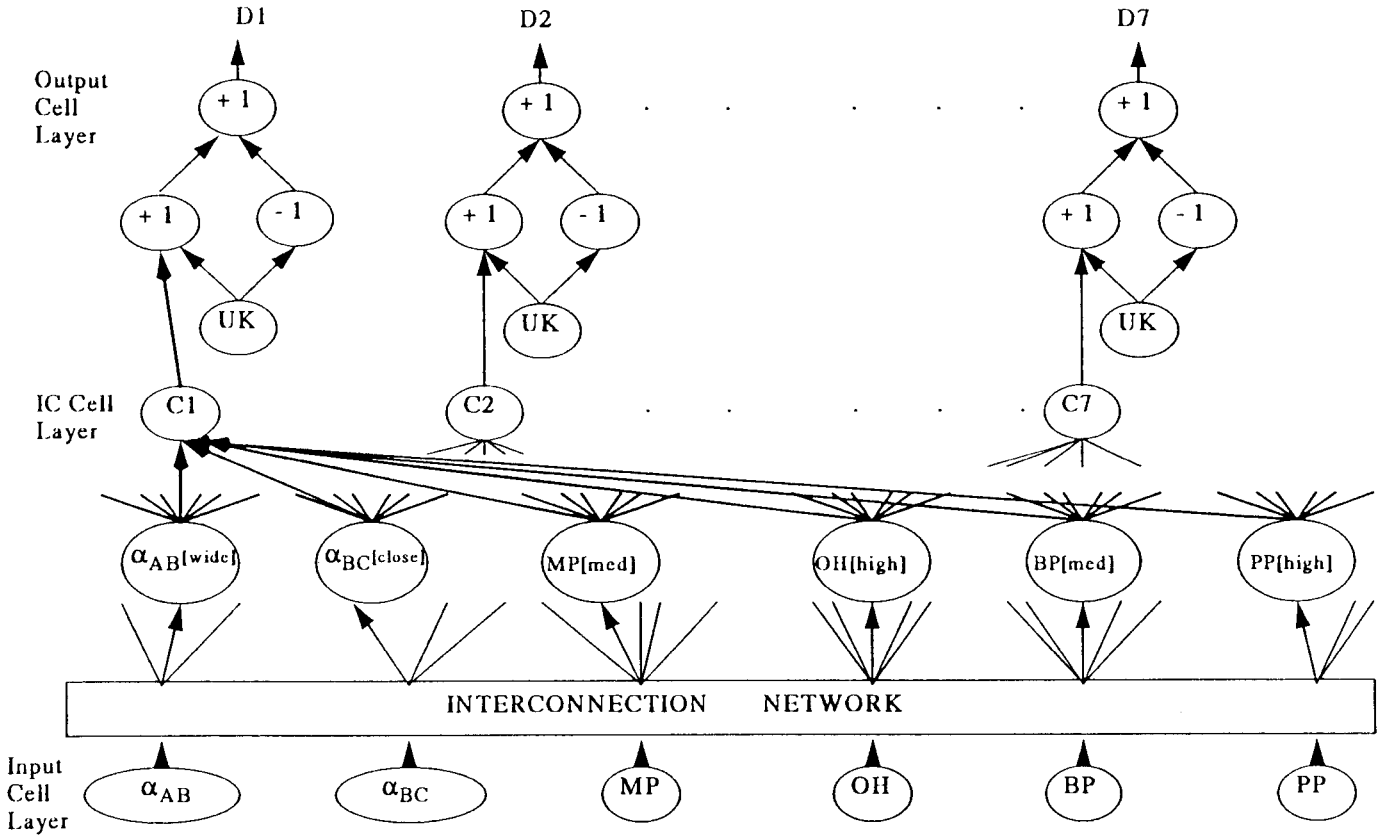

Figure 5. Network generated by SC-net for the example case

Table II Rules generated by SC-net

if and(MP[medium],BP[very_low],PP[high]) then D1 (1.000);

if and(BP[medium],PP[high]) then D1 (1.000);

if and(MP[low],BP[low],PP[high]) then D1 (1.000);

if and(MP[medium],BP[low],PP[high]) then D1 (1.000);

if and(BP[high],PP[high]) then D2 (1.000);

if and(BP[medium],PP[medium]) then D3 (1.000);

if and(α_{BC}[close],BP[low],PP[medium]) then D3 (1.000);

if and(BP[very_low],PP[medium]) then D3 (1.000);

if and(α_{BC}[near],BP[low],PP[medium]) then D3 (1.000);

if and(BP[high],PP[medium]) then D4 (1.000);

if and(α_{AB}[close],MP[medium],BP[low],PP[medium])
then D4 (1.000);

if and(α_{AB}[near],MP[medium],BP[low],PP[medium])
then D4 (1.000);

if and(MP[very_high],PP[high]) then D5 (1.000);

if and(MP[high],PP[high]) then D5 (1.000);

if and(BP[very_low],PP[low]) then D6 (1.000);

if and(OH[very_low],PP[low]) then D7 (1.000);

Table III Comparison of SC-net performance on test cases

α_{AB}	α_{BC}	MP	OH	BP	PP	Expected	SC-net
2.45	2.15	40	55	5	high	D1	D1
2.45	2.15	40	5	55	high	D2	D2
2.45	2.15	10	45	45	medium	D3	D3
2.45	2.15	60	15	25	high	D5	D5
2.45	2.15	46	50	4	low	D6	D6
2.45	2.15	46	4	50	low	D7	D7
2.8	1.38	25	20	55	high	D2	D2
2.8	1.38	50	30	20	medium	D3	D3
2.8	1.38	68	20	12	high	D5	D5
2.8	1.38	52	45	3	low	D6	D6
2.8	1.38	75	5	20	low	D7	D7
1.69	1.3	65	17	18	high	D5	D5
3.6	1.8	70	15	15	high	D5	D5
1.69	1.3	80	18	2	low	D6	D6
1.69	1.3	20	78	2	high	D1	D1
3.6	1.8	65	31	4	low	D6	D6
3.6	1.8	20	78	2	medium	D3	D3
1.69	1.3	65	4	31	low	D7	D7
1.69	1.3	20	2	78	high	D2	D2
3.6	1.8	65	4	31	low	D7	D7
3.6	1.8	20	2	78	high	D2	D2
1.69	1.3	25	20	55	high	D2	D2
3.6	1.8	25	20	55	high	D2	D2
1.69	1.3	25	55	20	high	D1	D1
1.69	1.3	10	45	45	medium	D3	D3
3.6	1.8	15	60	25	medium	D3	D3

unstructured knowledge. The background knowledge encoding features, knowledge extraction capabilities and the powerful description language of SC-net can be utilized in building AI systems which can automate the knowledge acquisition process.

References

[1] Hendry, J.E., Rudd, D.F. and J.D. Seader, " Synthesis in the Design of Chemical Processes," *AIChE J.*,**19**, 1-15 (1973).

[2] Westerberg, A.W. "A Review of Process Synthesis," *ACS Symp. Ser.*, **124**, 53-87 ACS, Washington, DC (1980).

[3] Hendry, J.E. and R.R. Hughes, " Generating Separation Process Flowsheets," *Chem Eng. Prog.*,**68**, 71-76 (1972).

[4] Tedder, D.W. and D.L. Rudd, " Parametric Studies in Industrial Distillation, " *AIChE J.*,**24**, 303-334 (1978).

[5] Stephanopoulas, G. and A.W. Westerberg, " Studies in Process Synthesis-II. Evolutionary Synthesis of Optimal Flowsheets," *Chem. Engg. Sci.* , **31**, 195-204 (1976).

[6] Seader, J.D. and A.W. Westerberg," A Combined Heuristic and Evolutionary Strategy for Synthesis of Simple Separation Sequences," *AIChE J.*,**23**, 951-954 (1977).

[7] Michalski, R.S., Carbonell, J.G. and T.M. Mitchell, *MACHINE LEARNING An Artificial Intelligence Approach*, Tioga, Palo Alto, CA (1983).

[8] Romaniuk, S. "Extracting Knowledge from a Hybrid, Symbolic, Connectionist Network ," Ph.D. Thesis, Univ. of South Florida, Tampa, FL (1991).

[9] Zadeh, L.A. "Fuzzy Logic," *IEEE Computer*, **21**, 83-92 (1988).

[10] *ASPEN PLUS* Aspen Technology, Inc., Cambridge, MA (1988).

[11] Douglas, J.M. *Conceptual Design of Chemical Processes*, McGraw-Hill (1988).

[12] Glinos, K. and M.F. Malone, " Optimality Regions For Complex Column Alternatives in Distillation Systems," *Chem. Eng. Res. Des.*,**66**, 229-240 (1988).

[13] Zimmermann, H.J. *Fuzzy Set Theory and Its Applications*, Kluwer Academic, Hingham, MA (1985).