Case Based Reasoning for Chemical Engineering Design

Stéphane Négny, Jean-Marc Le Lann

Case-based reasoning for chemical engineering design

Negny Stéphane, Le Lann Jean Marc*

INPT-LGC ENSIACET CNRS 5503, Département PSI, 118 Route de Narbonne, 31077 Toulouse Cedex 04, France

A B S T R A C T

With current industrial environment (competition, lower profit margin, reduced time to market, decreased product life cycle, environmental constraints, sustainable development, reactivity, innovation...), we must decrease the time for design of new products or processes. While the design activity is marked out by several steps, this article proposed a decision support tool for the preliminary design. This tool is based on the case-based reasoning (CBR) method. This method has demonstrated its effectiveness in other domains (medical, architecture...) and more recently in chemical engineering. This method, coming from Artificial Intelligence, is based on the reusing of earlier experiences to solve new problems. The goal of this article is to show the utility of such a method for unit operation (for example) pre-design but also to propose several evolutions for CBR through a domain as complex as the chemical engineering is (because of its interactions, non-linearity, intensification problems...). During the pre-design step, some parameters like operating conditions are not precisely known but we have an interval of possible values, worse we only have a partial description of the problem. To take into account this imprecision in the problem description, the CBR method is coupled with the fuzzy sets theory. After a mere presentation of the CBR method, a practical implementation is described with the choice and the pre-design of packing for separation columns.

Keywords:
Design
Case-based reasoning
Decision support system

1. Introduction

The activity of industrial design of products, processes (or services) is complex because, usually, it includes a lot of interacting components: from a few tens to several thousands. Moreover, the complexity is increased by some additional constraints: technical, social, safety, environmental.... Finally, the design problem can be over constrained and some constraints must be released to reach a satisfactory solution. The design activity is also decisive in the life cycle of an object (here an object is a product or a process).

The design part of an object includes several steps. The design process starts with the formulation of requirements and ends with the realisation of an object which satisfies all these requirements or the majority of them. Among the different design steps for a product, we can identify: the requirements analysis, the preliminary design, the detailed design, the modelling, the simulation, the optimization, tests on a prototype and the fabrication. It is important to notice that it is not a purely linear process, some loops can exist: for example during the simulation step, some results can bring to re-examine the detailed design. Another example is that the final version of the object is often free because the design specifications can evolve progressively during the different steps. Consequently, during all the design steps some choices, decisions are made. Questioning them, induces an increase (in terms of cost and time) of the design activity. It is therefore essential to make wise choices to avoid or to limit iterations and to converge quickly until a satisfactory solution.

Moreover, the evolution of the industrial surrounding world context has also as consequences to reduce, the design time because of: decreased life cycle, reactivity, innovation, competition.... One way, to take into account the acceleration of the design cycle is the fusion of different design steps, for

* Corresponding author.
E-mail address: stephane.negny@ensiacet.fr (N. Stéphane).

example simulation and optimization. But with this approach, the possible actions are very limited and specific to the object to design. Another approach is to exploit the experiences gained during earlier design because they allow to reduce the delay of design since some choices are no longer to make or to question. In this context, some firms want to have methods and tools to support design exploiting past knowledge. A design support system, needs the representation of the knowledge within a firm (or a profession) in order to exploit it and to facilitate the development of new objects. Various techniques coming from Artificial Intelligence has been developed to represent, to capitalize and to exploit knowledge for the problem of support design. Case-based reasoning (CBR) is one of them.

In the whole chaining steps of the process design, CBR has been widely used (in every technical domains) as a decision support system. In the majority of cases, CBR systems are limited to products design where one or two tens of components interact. The CBR method is based on analogical reasoning inside a specific domain (technical or not). This method is a Knowledge Management one, used to capitalize, to store and to reuse knowledge and earlier experiences. CBR has recently appeared in chemical engineering with applications in: process design by reusing flowsheets (Surma and Braunschweig, 1996), synthesis of process separation (Pajula et al., 2001), reactive distillation (Avramenko et al., 2004), mixing equipment selection (Kraslawski et al., 1995), minimisation of environmental impact (King et al., 1999), and generation of process alternatives (Lopez-Arevalo et al., 2007).

Currently in chemical engineering, few studies are dedicated and interested in preliminary design but they are numerous referring to the other phases (from detailed design to prototype or experimental tests). However a good preliminary design allows a saving of time thereafter, during the next steps. In these conditions, it would be necessary to develop tools dedicated to the first design steps in order to propose rapidly a preliminary solution with a high quality. The goal of this paper is to propose a design support system based on CBR, more specifically dedicated to the preliminary design. The main characteristic of the preliminary design is that some relevant features are not precisely known for the description of the new object. To take into account the imprecise values, we couple the step of the description of a new problem in the CBR method with the fuzzy sets theory.

This article is composed of seven main parts. After the introduction, the second part is dedicated to a general presentation of the CBR method. The parts 3, 4 and 5 detail the crucial points of this approach: knowledge modelling with case representation, the research of earlier experiences and the way to reuse these past experiences. Before to conclude, two examples on the packing selection and pre-design for separation unit operations are treated with the developed tool.

2. Case-based reasoning

The goal of CBR is to propose a solution to an initial problem (target problem) in a specific domain starting from the adaptation of solutions of previously solved problems stored in a memory (source problems). CBR is a method for reasoning and learning with support of past experiences. Basically in CBR, users search to solve a new problem by establishing some common characteristics between this initial problem and some previous solved problems. Then they try to adapt earlier successful resolutions, ways of resolutions or solutions in order to solve the target problem. In CBR the main idea is: similar problems have similar solutions.

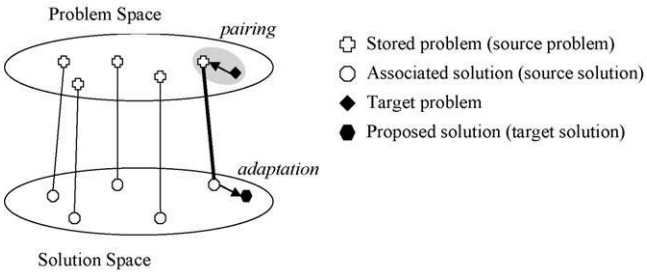


Fig. 1 – Case base representation.

This approach, based on human reasoning (it imitates human reasoning), comes from Schank (1982) research on dynamic memory which underlines: the importance of past experiences in the resolution of problems, and the dynamic change of the memory (continuously changing according to the new problems or situations faced or experienced). In this context, the central notion of this method is a case. A case represents an earlier experience with the description of the problem and its associated solution and eventually some results and comments like success or failure of the solution, advises of implementation. A case is a contextualised piece of knowledge representing an experience. Many cases are gathered and stored in a memory, in order to build a case base in a specific domain. Consequently this case base is composed of two spaces as illustrated in Fig. 1: the problem space and the solution one.

In practice, the target problem is compared to other problems (source problems) stored in the case base and the most similar problem is extracted with its associated solution (source solution). Then the user adapts this source solution to propose a solution to the initial problem (target solution). When the facing problem is solved, it is stored (or not) in the case base memory. But before to solve a problem, an important and primordial preliminary step is necessary: the elaboration of a case and the case base structuring (indexation). Of course, the main goal of the representation of a case is to traduce this one in a relevant way: choice of the main characteristics to describe a problem and its associated solution. Expert's knowledge is often needed. But case representation must take into account additional constraints. This representation has to allow the manipulation of cases in the modules of a CBR system (tool built on CBR method). It has also to be relevant for the adaptation, for example.

Various formalisms are available to represent cases, but the most commonly used is the feature-vector representation. Moreover it is the most suitable to the purpose of this article. This formalism represents a case as a vector of feature-value pairs, for the problem and solution descriptions. The features or attributes represent the main and the most relevant characteristics to describe cases. For each feature, an associated value characterize it with different types of data; numeric, semantic. For examples, for chemical engineering problems concerning unit operations, mixture and operating conditions like pressure are relevant features to describe a part of a problem:

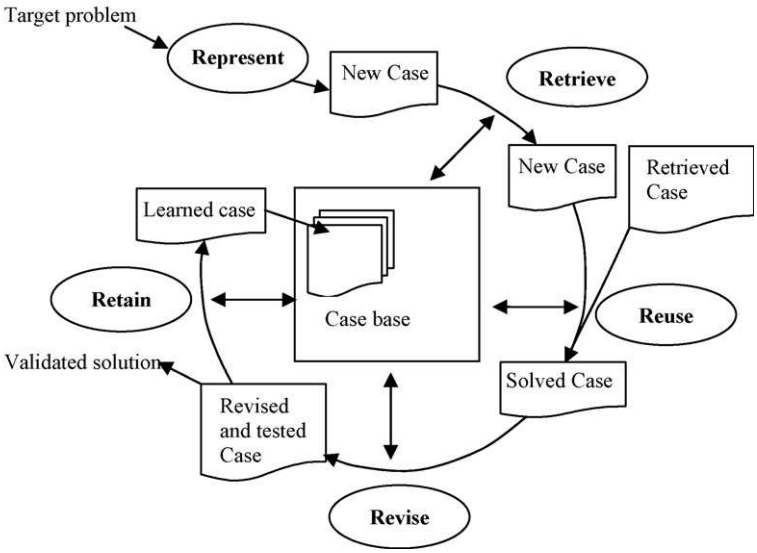


Fig. 2 – The CBR process cycle.

Feature	Value
Pressure	1 atm
Compounds	Ethanol, water, acetic acid

When the user faces a new problem, the initial step before applying the CBR process is to collect data to define the current problem, and to fill each feature of the problem description, and then the CBR cycle can start. Finally, the aim of the CBR is to propose a value for each features of the target solution, for this new problem. At first the target solution is an initial guess for a precise solution. It is a starting point to built a specific solution after its improvement, tests, validation, optimization.

To find a target solution, numerous CBR model exists, however one of the more used is the cyclic process developed by Aamodt and Plaza (1994), known as the R^4 model (but sometimes it is extended to the R^5 model if the preliminary step, i.e. case representation, is included) (Fig. 2).

After the description of the target problem, the cycle is started:

- *Retrieve*: In this step of the CBR cycle, we research in the case base, one or various previous cases which are similar to the target problem. Here, the central issue is the similarity measurement in order to find the most useful and helpful case to solve the target problem. The similarity between two cases is measured by a function which compares the values of each feature between the target problem and the source one. The similarity function calculation depends on the type of features value: words, numerical values, diagrams, plans…. If various similar cases are found, the global similarity function ranks them. To decrease the research time, we adopt a case base indexation to filter and select the most relevant cases before measuring the global similarity on this subset of cases, part 4.
- *Reuse*: The goal of this step is to propose a solution to the target problem, derived from the solution(s) of the retrieved case(s), part 5. This solution is used as a starting point to determine a specific solution for the target problem. Reusing previous cases solutions can be as trivial as applying the source solution without modification (for example when the retrieved case is sufficiently similar). However most of the time, there is a gap between target problem and the retrieved one, therefore the retrieved solution does not exactly correspond to the target problem. The source solution often needs some adjustments to be adapted to the initial problem. This adaptation becomes complex when the differences between both problems are important. This adaptation often needs additional knowledge modelised by rules, equations, correlations….
- *Revise*: The previous adapted solution is used as the starting point for the target problem. This solution is tested to verify its adequacy (by simulation, experimental validation for example). After the tests, the solution may need some adjustments to fit more specifically the target problem. Consequently, the user revises the solution generated in the previous step to withdraw the discrepancies between the desired and the adapted solution.
- *Retain*: After its resolution, the target problem and its associated solution form a new case. If it is relevant, the CBR system may learn this new case by its incorporation into the case base. This step extends the cover of space problems, increasing the CBR effectiveness by enlarging experiences retrained. If a new case is too similar to another one in the case base, it is not stored because it increases the case base without bringing added value.

The description of the CBR cycle clearly highlights the main difficulties during the implementation of such a system: case elaboration, research of a similar case and the reuse of the solution(s) retrieved. These difficulties are detailed in the next parts.

The CBR systems are naturally implemented for knowledge management in several firms (but outside of the chemical engineering domain) thanks to its way of reasoning. CBR has the huge advantage to build learning systems because it has an evolutionary memory with the retain step. Consequently, CBR systems reduce the resolution time.

While CBR is an interesting method for the purpose of this article, it has two main drawbacks, because it is based on earlier experiences. The first one appears, when during

the retrieval step no sufficiently similar cases are extracted or worse when there is no similar case. The user must research a solution with other methods.

In this article, the CBR method is applied for design. In this context, it is an interesting method for routine design where the problematic situations vary in a small interval. Consequently, the second limit concerns innovation, which is an important challenge for firms. Indeed, with its focus on past experiences inside a specific domain, CBR is an effective approach for relatively restricted targeted field. Moreover for inventive or innovative designs, the problem must find a radically different solution that the CBR cannot reach alone (in some cases it can produce incremental innovation but not rupture one). These two drawbacks have been eliminated by the coupling between CBR and other methods dealing with the generation of inventive ideas, like the TRIZ theory (Cortes Robles, 2006).

3. Case representation

The implementation of the R^4 cyclic process is realised for a chemical engineering unit operation. However, the version of the tool is sufficiently general to treat various other problems. In this article, the main goal, is to demonstrate the contributions of the CBR method in the chemical engineering domain, this is why the example presented concerned the pre-design of column packing for separation. The objective is to determine the packing and to propose a first estimation of the main geometrical characteristics for a target problem. As mentioned before, problems and solutions are represented by a vector of feature-value pairs. Of course, the chosen features to describe the problems and the solutions are different by their number and by the information contained. To determine the relevant features, an analyse of the literature and documentation of the packing suppliers has been done. Finally, this analyse has allowed to fill the case base with more than 200 different cases (at the start, but it is still growing with additional new cases).

For the problem description, the first column of Table 1 sums up the features to fill in order to describe the target problem. The features representing the solutions are also in Table 1, but in the second column. The whole solutions are described with the same global structure (features of Table 1) but some differences can be seen when the features are detailed; for example geometrical characteristics depend on the type of packing. Indeed, the latter are different by their number and size characteristics to design a structured packing or a random one, part 7. Even in the category of random packing, there are differences.

Table 1 – Features to describe a case (problem and solution)

Problem features	Solution features
Compounds	Type of packing
Pressure	Material
Temperature	Specific area
Inlet flow rate	Geometrical characteristics
Reflux*	

* Particular feature.

Here the case base is dedicated to the design of packing for separation unit operation; it can be used for distillation or absorption. Consequently, the reflux which is an important feature for distillation, is not necessary for absorption. This is the first reason to consider it as a particular feature in Table 1. Moreover, even in the subset of cases concerning distillation, this information was not always available for the source cases coming from the literature. Therefore, the case base contains some source cases without value for the feature Reflux; less than 5% of the 200 first source cases. This feature cannot be deleted because it is important for new problem's description and for the next step in the CBR cycle (i.e. the retrieved step).

To increase its effectiveness, the retrieval consists in two steps: to search a subset of relevant source cases and then to calculate the similarity for all the source cases in this subset. In order to select the subset of relevant cases, a decision tree is used to classify the source cases in subsets (detailed in the next part). In this decision tree, the Reflux is a feature that leads to this classification between distillation and absorption cases. If the Reflux value is zero, the system researches source cases in the absorption subset and in the distillation one if the value is greater than zero (Fig. 3).

In the distillation subset, during the similarity measurement if the user gives a value to the Reflux in its target problem, the source cases without this information cannot be reached, because with the way to estimate the global similarity these cases would have a small value for the global similarity and consequently a low rank in the ranking. Therefore, we include the option *IGNORE* for this feature in order to reach these source cases. With that option, the Reflux is taken into account for the research of the relevant subset but it is not for the global similarity measurement. More generally, this option is extended to all the features because for more complicated application than the presented one, the user can have to describe a target problem with a missing value for one feature. Not to penalize the global similarity measurement because of this missing value, the *IGNORE* option can be useful too. This option allows to research similar source cases even for partial description of the target problem.

4. Retrieval

4.1. *Decision tree*

The number of cases in the case base is going to grow because of the Retain step or memorization of new industrial or literature cases. Without case base organization, the cost to estimate the global similarity between the target problem and all the source cases in the memory becomes prohibitive. As explained before, in order to decrease the research time and to increase the effectiveness of the retrieval, the latter is decomposed in two steps. The first one consists in selecting a subset of relevant source cases. The second one is dedicated to the similarity measurement and the ranking of source cases included in the subset. To select the subset of the more relevant cases for one research, we index the case base to constrain the research space to the nearest source cases. The organization of the memory is based on the decision tree approach. In this approach, the case base is successively restricted thanks to decision sequences. All the cases of the base are gathered at a root node. Starting from this node, intermediate nodes are generated to restrict the number of cases by an evaluation on a discriminate feature. And the end of

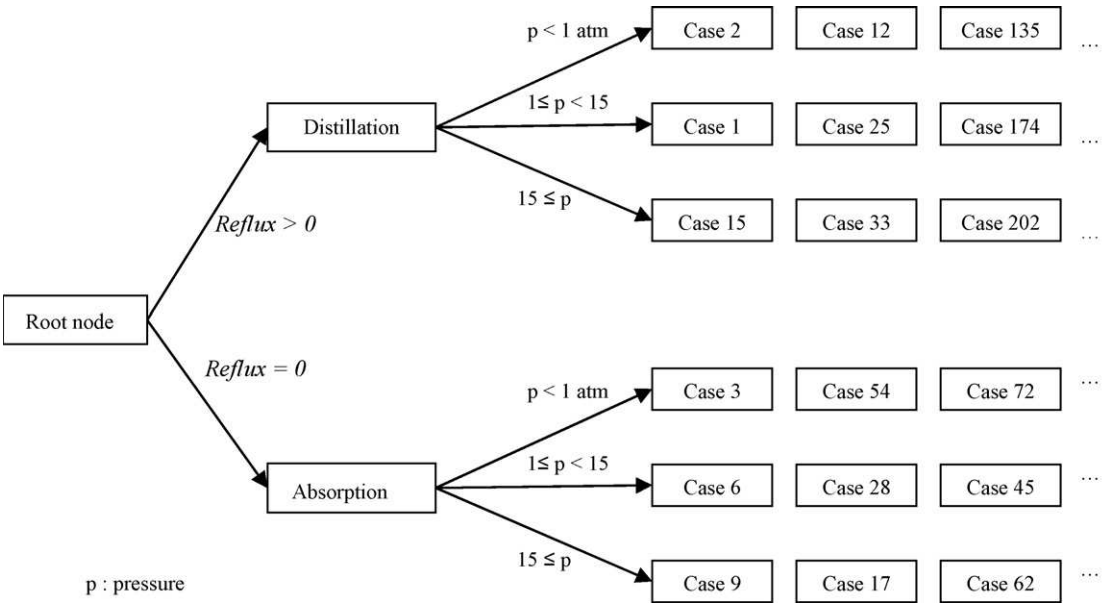


Fig. 3 – Mere example of a decision tree.

the tree, at final nodes called the leaves, there are the source cases. Finally in this approach, leaves represent the classification and branches represent conjunction of features that lead to these classification (Fig. 3).

In the tool, the decision tree can be automatically built with an algorithm based on the ID3 algorithm (Quinlan, 1979). Nevertheless, the organization of the case base must reflect the point of view of the user, therefore he can generate its own decision tree corresponding to the aim of his retrieval (or to select a previous created tree stored in a tree base).

4.2. *Local similarity measurement*

During the second step of retrieval, the system tries to establish some resembles between the target problem and the source ones. This crucial operation is realised with a similarity operator. This resemblance measurement is achieved on the descriptive features of the problems. The latter contains different types of values: numeric, semantic (chemical compounds name) for the presented example. Because of these different feature types, a similarity is calculated for each feature: local similarity. Then all the local similarities are gathered to evaluate the global similarity between problems.

For numerical values, various distances have been proposed to measure the variation between two values. But the most use ones are the Euclidian and the Manhattan distances which are in fact particular cases of the Minkowski measurement (for two problems X and Y):

$$d(X, Y) = \left(\sum_{i=1}^{L} w_i |x_i - y_i|^p \right)^{1/p}. \quad (1)$$

For $p = 1$ we have the Manhattan distance, $p = 2$ the Euclidian one, $p = \infty$ the Chebychev distance ($\mathrm{Max}|x_i - y_i|$). In formula (1), x_i and y_i, respectively represent the ith features of X and Y, and w_i the associated weight to this feature. However, in this study, this calculation of the global distance is not available because of different types of the features values, consequently a local distance is calculated for each feature. For numerical features, the calculation of the local distance is based on the following equation (derived from formula (1)):

$$d(x_i, y_i) = |x_i - y_i|. \quad (2)$$

However, this way of measuring can distort the results when the features have different sizes for their domains of definition. Therefore, it is necessary to normalize the distance calculation. One solution consists in explicitly express the definition domain and to implement its expression in the calculation. This is done by the Int_i function, which is the difference between the maximum and the minimum values for the feature i.

$$d(x_i, y_i) = \frac{|x_i - y_i|}{\mathrm{Int}_i}. \quad (3)$$

Finally, the local similarity on a numerical feature can be calculated from distance expressed in formula (3), but it must express that the nearest two problems are the more similar they are:

$$\mathrm{sim}(x_i, y_i) = 1 - \frac{|x_i - y_i|}{\mathrm{Int}_i}. \quad (4)$$

Formula (4) is used when you exactly know features value for the target problem. But usually, during chemical engineering preliminary design, we do not know precisely the whole values for the operating parameters (like pressure, temperature...), there are often inaccuracies. In the majority of cases we know an interval of possible values, a value not to exceed.... This idea is implemented with the fuzzy sets theory.

4.3. *Local similarity measurement with fuzzy sets*

In order to soften the problem description, the user gives an estimated value v of a feature, then he can specify an imprecision on this value and a relation. With these additional informations, the domain of acceptable values for a feature

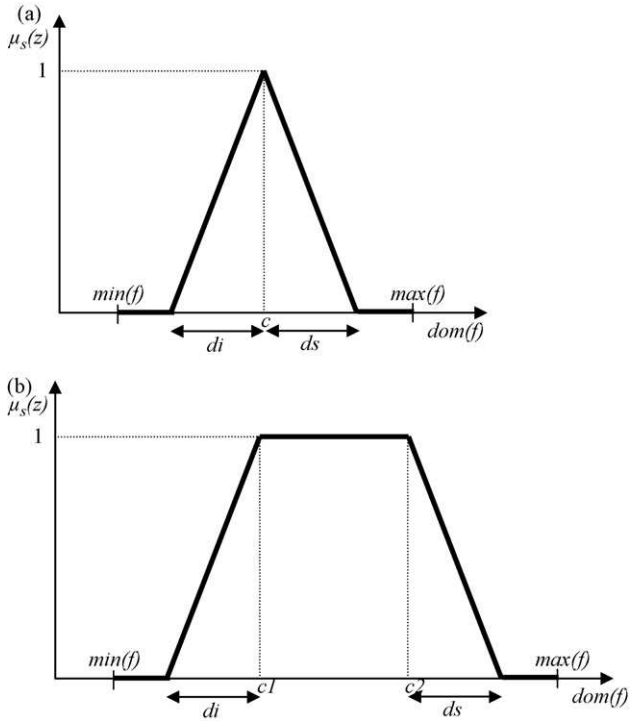


Fig. 4 – (a) Triangular distribution. (b) Trapezoidal distribution.

is created thanks to the fuzzy sets theory developed by Zadeh (1965). A fuzzy set S on a domain D is defined by a characteristic function μ_s, which has values in [0,1]. $\mu_s(z)$ indicates the degree to which z is a possible value in the sub-domain S. When z is a value really in S; $\mu_s(z) = 1$, and when z is outside S; $\mu_s(z) = 0$, but when z is an acceptable value in S; $\mu_s(z)$ is between 0 and 1. To represent the different possible sub-domains, we have considered two representations for the fuzzy sets: triangular or trapezoidal representations (Fig. 4). In these figures, f is a feature, dom(f) is the domain of definition of this feature, min(f) and max(f) represents, respectively the lower and upper domain limits. The functions μ_s are defined with three parameters (di, ds, c) for the triangular representation (5) and four for the trapezoidal one (di, ds, $c1$, $c2$) (6). For the two types of representation, the function $\mu_s(z)$ is built with the following formulas:

$$\begin{cases} \mu_s(z) = 0 & \forall z < c - di \\ \mu_s(z) = \dfrac{1}{di}(z - c + di) & \forall z \in [c - di, c[\\ \mu_s(z) = 1 & \text{if } z = c \\ \mu_s(z) = \dfrac{1}{ds}(-z + c + ds) & \forall z \in]c, c + ds] \\ \mu_s(z) = 0 & \forall z > c + ds \end{cases}$$

(triangular representation) (5)

Table 3 – Parameter values for the characteristic function in trapezoidal representation

	Between
c1=	$v1$
c2=	$v2$
di=	$\text{Min}(\lambda v1; v1 - \text{min}(f))$
ds=	$\text{Min}(\lambda v2; \text{max}(f) - v2)$

$$\begin{cases} \mu_s(z) = 0 & \forall z < c1 - di \\ \mu_s(z) = \dfrac{1}{di}(z - c1 + di) & \forall z \in [c1 - di, c1[\\ \mu_s(z) = 1 & \forall z \in [c1, c2] \\ \mu_s(z) = \dfrac{1}{ds}(-z + c2 + ds) & \forall z \in]c2, c2 + ds] \\ \mu_s(z) = 0 & \forall z > c2 + ds \end{cases}$$

(trapezoidal representation) . (6)

With di is the distance between the lower limit of the sub-domain S and the central value; ds the distance between the upper limit of the sub-domain S and the central value; c the central value for the triangular representation; $c1$, $c2$ respectively the lower and upper limit between which $\mu_s(z) = 1$

For the relation there are six possible choices: equal (equal to a specified value (SV)), sup (superior to a SV), sup–equ (superior or equal to a SV), inf (inferior to a SV), inf–equ (inferior or equal to a SV), between (between two SV). The triangular representation is automatically used for the first five relations and the trapezoidal representation for the last one (between).

The local similarity of one feature is estimated with μ_s, this function depends of the value v, the relation and the imprecision, all these parameters are given by the user. The imprecision is introduced with a parameter λ: which is the percentage of variation around the specified value, v. For each numerical feature, we have to create the sub-domain S in order to build the function shape μ_s, therefore to find the values for di, ds, c or $c1$ and $c2$. The calculation of these parameters for the possible sub-domains are listed in Table 2 for the triangular distribution (Fig. 5) and Table 3 for the trapezoidal one. We have to notice that if the user chooses the *between* relation for one feature instead of specifying one value v, he has to give two values $v1$ and $v2$ to define the interval.

After the creation of the sub-domain for each feature, the local similarity measurement can be calculated. If v is the value of a feature for the target problem, and z is the value for the same feature for a source problem, the local similarity is $\text{sim}(v, z) = \mu_s(z)$.

With the fuzzy sets theory, we improve the problem description. Nevertheless, for some features the value is precisely known even in the preliminary design and we want to research previous experiences (source cases) having exactly this value. This is implemented with the option *EXACT*, if this

Table 2 – Parameter values for the characteristic function in triangular representation

	Equation	Sup, sup–equ	Inf, inf–equ
c=	v	Max(f)	Min(f)
di=	$\text{Min}(\lambda v; v - \text{min}(f))$	$\text{Min}(\text{max}(f) - (v - \lambda v); \text{max}(f) - \text{min}(f))$	0
ds=	$\text{Min}(\lambda v; \text{max}(f) - v)$	0	$\text{Min}((v + \lambda v) - \text{min}(f); \text{max}(f) - \text{min}(f))$

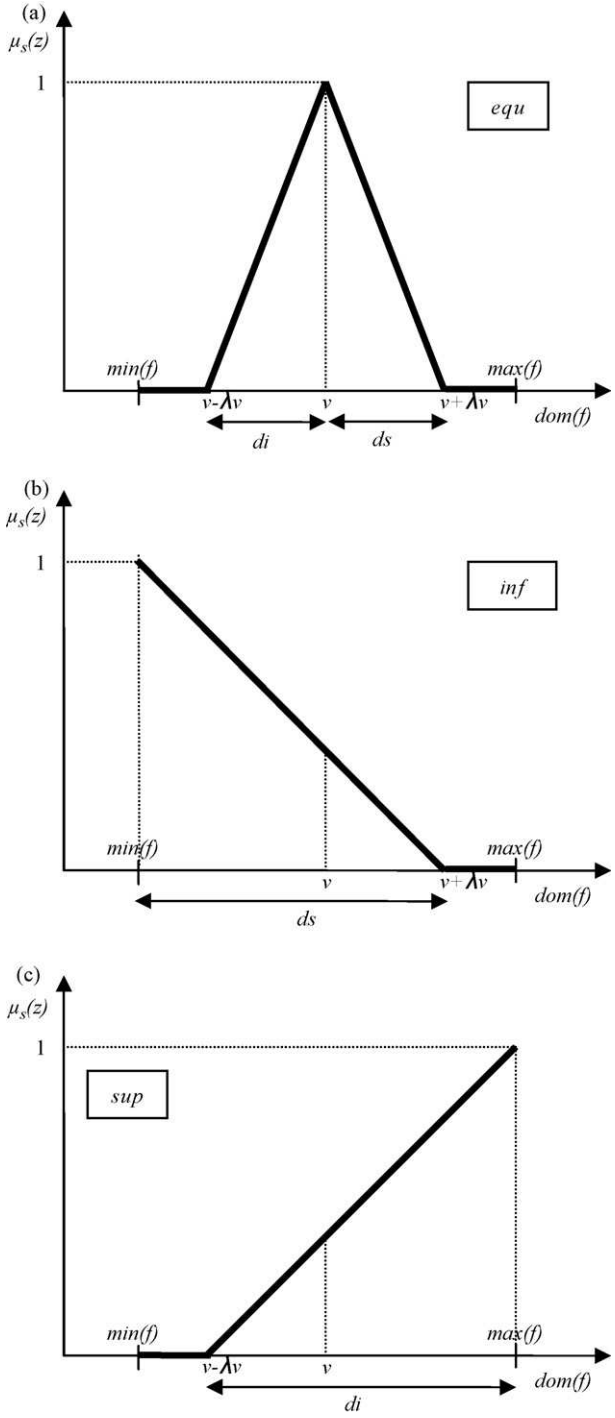


Fig. 5 – (a) Characteristic function for equ relation. (b) Characteristic function for inf, inf–equ relations. (c) Characteristic function for sup, sup–equ relations.

option is activated for a feature, $\mu_s(z)$ only takes two values: if $v = z$ then $\mu_s(z) = 1$ else $\mu_s(z) = 0$.

4.4. *Local similarity for compounds*

Concerning the local similarity for semantic value (compounds name), the classical local measurement consider two values:

$$\mathrm{sim}(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{if } x_i \neq y_i \end{cases}. \tag{7}$$

In our case this local similarity between chemical components, can be improved. It can be refined as (Avramenko et al., 2004) have demonstrated. This approach measures the similarity between compounds basing on their chemical structure. This semantic data can be divided into classes (and sub-classes) and a hierarchical structure is built to describe the relations between classes. The local similarity is described in a tree like structure. The root of the tree represents all the compounds. The first level nodes in the tree corresponds to a basic group of chemical compounds (organic/inorganic). The daughter nodes correspond to classes/subclasses of the chemical substances (hydrocarbons, acids...). The value of the similarity between two compounds depends on the first common level. Each node of the tree has a value. So the local similarity can be numerically estimated: the lower the node is in the tree, the higher the numerical value is (same compounds have a similarity of 1, if the common node is the first level in the tree the local similarity is 0.1) (Fig. 6).

The feature "compounds" describing the mixture to separate, contains several individual substances, consequently the similarity of the whole feature "compounds" has to be calculated. The first step consists in finding the most similar pairs of components comparing the source cases and the target problem mixtures. This is done:

- by building the matrix of binary similarity (between compounds of mixtures);
- by maximizing the sum of the binary similarity in the set of every possible pairing under the constraint that if a compound is embedded in a pair, it cannot be in another one.

Next the local similarity of the feature "compounds" is calculated:

$$\mathrm{sim}(t, s) = \frac{1}{m} \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} x_{ij} b\mathrm{sim}_{ij}. \tag{8}$$

With m the maximum number of components and $b\mathrm{sim}_{ij}$ the value of the binary local similarity and x_{ij} an affectation variable (binary), the whole calculation is detailed in Appendix A.

4.5. *Global similarity*

Finally the global similarity is calculated from all the local similarities:

$$\mathrm{sim}(X, Y) = \frac{\sum_i w_i \mathrm{sim}(x_i, y_i)}{\sum_i w_i}. \tag{9}$$

With the weight w_i, the user can customize the global similarity, giving to one feature more importance than the others. This choice is crucial to have relevant similarity measurement. If the user is not an expert in the domain, the w_i can be estimated automatically. Then for the selection and the ranking of the cases, the k nearest neighbour's algorithm is used.

5. Reuse

As previously mentioned, the reuse of a retrieved case can be as trivial as the reuse of the source solution without modification for the target solution. Of course, most of the time, it is necessary to adjust the source solution in order to elim-

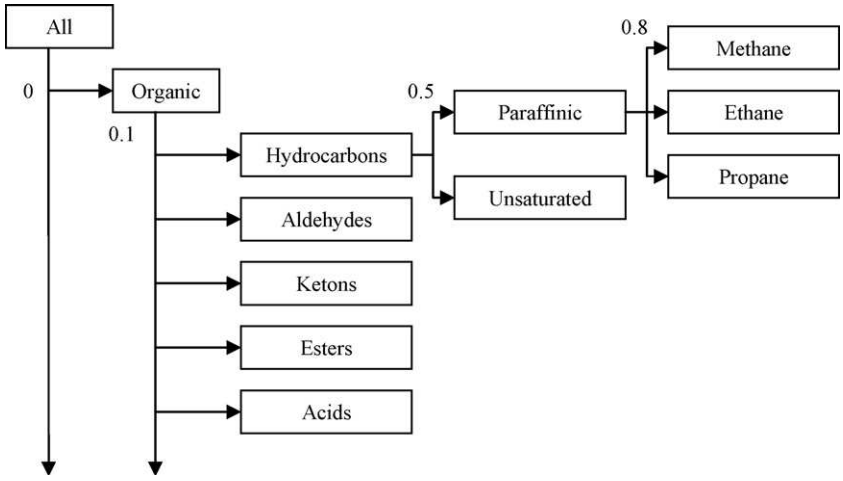


Fig. 6 – Similarity tree for chemical compounds (Avramenko et al., 2004).

inate the discrepancies between the target and the source problems. The adaptation step is an important point in a CBR system, numerous researches are focused on this issue. Various methods to adapt a case exist, but two main categories emerge:

- Methods where additional knowledge on the specific domain are added, with the support of rules, correlations. This can be done here with unit operation design in chemical engineering.
- The adaptation is realised thanks to the different cases available in the case base, without additional knowledge (generic method).

In the second category, source solutions of the most similar problems are used to build and to propose a solution to the target problem. In the tool, several methods are implemented but here only the method proposed by Avramenko et al. (2004) is presented, because it is used in the examples. This adaptation method is based on the main idea that the relative distances between the target problem and the source problems (3 of them, but the user can change this number) in the problem space are transferred in the solution space. This process is valid only for numerical features, which is interesting to estimate the geometrical characteristics of packing. Finally, the adaptation method consists in minimizing the following function:

$$F(\text{sol}) = \sum_j |\text{sim}(S_j, \text{sol}) - \text{sim}(P_j, C)|. \quad (10)$$

With P_j one similar problem, C the target problem, S_j the solution to the source problem P_j, sol the target solution (research solution). Sol is initialised with the solution of the most similar problem. It is important to underline that sol is a first estimation of the solution, and then this solution must be modified after some additional validation tests (simulation, experimentation...). Once solved, this new case is retained in the case base.

It is important to notice that the source case which is the most similar is not always the easier to adapt because of technical constraints, cost. To improve the quality of the proposed solution after the Reuse step, sometime it is would be more interesting to select a source case easy to adapt, than the most similar one. Because, most of the time they are not the same. This is the notion of retrieval guided by adaptation (Smyth and Kean, 1998). With this idea, the retrieve and the reuse steps are linked. A module to take into account this improvement by application of the fuzzy sets theory is still under development.

6. Example

6.1. *Binary system*

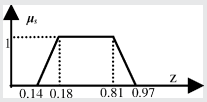
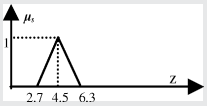
This first mere example is introduced to underline the importance of user knowledge during the CBR process: for weight determination for the global similarity calculation and for the selection of the retrieved cases for adaptation. The mixture to separate is composed of the binary system toluene/methyl chloroform, at atmospheric pressure. After the filling of all the problem features (first column of Table 1), the global similarity measurement is realized thanks to the local similarity, detailed in part 4. For the first run, the weight affected to the features, is the same for all of them. After the retrieval, the three most similar cases are: two cases with Pall Rings respectively of 25 mm (1 in.) and 50 mm (2 in.) and a structured packing. For the next step, i.e. adaptation, the solution with the structured packing is occulted. With this elimination, we can underline that during the resolution process, the user should not use past cases blindly. The user knowledge is important and sometimes mandatory for intervention. After adaptation made with formula (10), the target solution proposed is: metal Pall Rings of 38 mm (1.5 in.) (solution 1 in Table 4).

Following these first results, a second run is realised for a more refine research. For packing selection, the operating conditions are often more important than the compounds in the mixture (but compounds are also important because they can influence the solution: material for example), consequently

Table 4 – Solutions descriptions for the binary mixture

	Solution 1	Solution 2	Real solution
Packing	Pall Rings	Pall Rings	Pall Rings
Material	Metal	Metal	Metal
Specific area (m^2/m^3)	135	201	201
Dimension (mm)	38	25	25
Packing factor (m^{-1})	144	177	177

Table 5 – Multi component description

	Relation	Value(s)	Imprecision	Ignore	μ_s function
Mixture	Equ	Methanol/ethanol/water		Off	
Pressure	Equ	1	EXACT	Off	
Temperature	–	–	–	On	
Inlet flow rate	Between	0.1877 and 0.8123	20%	Off	
Reflux	Equ	4.5	40%	Off	

the weights corresponding to the operating conditions features are increased. Once again the retrieved step gives the following source solutions: 2 cases with metal Pall Rings of 25 mm and one structured packing (if the research is extended to the 6 most similar cases, there is 5 cases with metal Pall Rings respectively of 25 mm (3), 38 mm (1), 50 mm (1)). For this new run, the adaptation step gives metal Pall Rings of 25 mm (solution 2 in Table 4). This result is exactly the packing used in these operating conditions coming from real case presented in Kister (1992).

We can notice that after the first run, the proposed solution is closed to the real one. With this target solution as starting point, there is a small effort to make during the revised step (to have a specific solution to the target problem), thus the problem resolution time decreases.

6.2. *Three component distillation*

This example is presented in order to illustrate several parts of the method. The mixture to separate is a three components distillation methanol/ethanol/water. The column is operated at finite reflux, at atmospheric pressure, with feed flow rate between 0.1877 and 0.8123 mol/s. This distillation corresponds to the work of Mori et al. (2006). The previous operating conditions are used to define our target problem. Moreover, in our problem description we impose that the distillation is at atmospheric pressure to exemplify the option *EXACT*, consequently p is exactly 1. In the description of their operating conditions, the authors do not give the range of temperature, consequently we suppose that it is not known. Of course, this range of temperature can be easily calculated with a thermodynamic analysis of the mixture at atmospheric pressure (because the molar fraction are known). But in order to show how our system treat the partial description of a problem, we do not fill this feature and we use the option *IGNORE*. The first five columns of Table 5 sum up the problem Description.

For the retrieved step, the first work is to build automatically the function μ_s for each numerical feature, except for the temperature because the option *IGNORE* is activated. Therefore, this feature is not included in the global similarity calculation. These functions are represented in the last column of Table 5. Before to calculate the global similarity, the case base is restricted to the subset of the most relevant cases thanks to a decision tree with the following succession of feature evaluation: at the root node the evaluation is on the Reflux, then the pressure, then the inlet flow rate. Here again the temperature is ignored. For each cases in the isolated subset, the global similarity measurement is calculated on four features; compounds, pressure, inlet flow, reflux, with the same weight for each one.

After the retrieved step, the ranking gives three structured packing (and two random packing, which are eliminated): one of one type, and two of another one. The two different Montz-pak B1 are retained for adaptation (they have different geometrical characteristics, specific area and material). Finally, after adaptation, the proposed target solution is the Montz-pak B1 30, (Table 6). The second column of Table 6 gives the characteristics of the structured packing used by Mori et al. (2006). In this example, the tools gives again a good starting point for the resolution of the initial problem. It is to notice that the material of the two retrieved cases selected are: stainless steel (in case 1) and carbon steel (in case 2). Consequently, for adaptation we search in the subset of metal. Then, the choice is oriented to the stainless steel because, under operating conditions in the same magnitude, the mixture of case 1 is most similar to the mixture of the target problem than the one of case 2. Therefore the choice is made with the following assumption: under operating conditions in the same magnitude (which is often the case), the most the mixtures are similar, the most the risk of degradation (corrosion...) is reduced. This way to proceed is just a first approximation, and of course it needs to be improved because this assump-

Table 6 – Solution descriptions for the multi components mixture

	Proposed solution	(Mori et al., 2006) Solution
Type of packing	Structured packing Montz pak B1 300	Structured Packing Montz pak B1 250
Material	Stainless steel	Metal (not specified)
Specific area (m^2/m^3)	350	247
Geometrical characteristics		
Angle	45°	45°
Element height (m)	0.201	0.197
Corrugation height (m)	0.008	0.012
Corrugation base (m)	0.0167	0.0219
Corrugation side length (m)	0.0116	0.016

tion is not completely satisfactory. This adaptation step is still under development and we want to improve it thanks to the constraint satisfaction problem method.

7. Conclusion and perspectives

This article demonstrates the utility of Artificial Intelligence method like CBR for the preliminary design in Chemical Engineering. This method is simple to use because of its affinity with human reasoning. CBR is an effective method to rapidly pre-design some unit operations. However, the proposed solution must be adjusted for the detailed design step. To take into account one specificity of the preliminary design, i.e. imprecise values for the problem description, the CBR system is coupled with the fuzzy sets theory. Even if we improve the problem description in our CBR system, the latter still has the classical drawbacks of this approach: the problem space must be sufficiently covered, the number of cases on the memory must be important to built a target solution with high quality, and an effective method of adaptation.

CBR seems to be simple in its operation but it is complex to build, more precisely in the retrieved and the adaptation steps which are crucial to elaborate a good target solution. In the examples treated in this paper, the retrieved cases do not need adaptation on the attribute "packing" (which represents the chosen type of packing). However with the presented adaptation method in part 5, only numerical features can be adapted. For non-numerical features some rules are applied (like for the material in the second example). But in some other cases where an adaptation on the feature "packing" is needed, more interesting methods are currently tested: for example with Constraint Satisfaction Problem technique.

This article is focused on retrieval which is an important step, especially when the case base is growing. In this situation, it becomes primordial to refine the research to a subset of relevant cases to reduce the research time avoiding testing the whole cases. A sphere indexing algorithm is implemented in the tool to replace the decision tree but it is still in validation.

Another way to improve this tool is to model and implement knowledge. This will increase the precision of the target solution. For example, the most similar case is not inevitably the most relevant for adaptation. By introducing knowledge in the retrieve step we can also take into account this adaptation problem (knowledge introduction can be useful in numerous other points of the CBR system, like in the adaptation step as presented in the example).

To extend this work, one of the future subjects of research, it is to couple different case bases in order to propose solutions for the design of more complex unit operations, like in process intensification: for example the coupling between a distillation packing case base with another one dedicated to catalytic reaction in order to propose a new solution for reactive distillation. Of course, a specific case base for reactive distillation can be built (this is done by Avramenko et al. (2004)) but the idea is to couple the two case bases in order to propose a solution when the specific reactive distillation case base cannot find similar cases.

Appendix A

The local similarity for mixture is detailed in this part. Consider a target and a source problems with respectively n_t and n_s compounds ($n_t = 3$; $n_s = 4$ in the example). The first step is to build the matrix of binary similarity, which gives the numerical local similarity for the first common node in the tree of substances. For example methanol and ethanol have the nearest common level alcohol $b\text{sim}_{11} = 0.9$, but water and acetic acid have the level organic, $b\text{sim}_{33} = 0.1$. The Table 7 is automatically generated by the tool.

Table 7 – Example of local binary similarities matrix

	Ethanol	Water	Acetic acid	Ethyl acetate
Methanol	$b\text{sim}_{11} = 0.9$	0.1	0.1	0.1
Ethanol	1	0.1	0.1	0.1
Water	0.1	1	0.1	0.1

Since the feature compounds can contain several individual compounds, the local similarity of the whole feature has to be calculated. First the most similar pairs of components in source cases and target problem are found, basing on the matrix of binary similarity. We have to analyse every possible pairs and selected the best ones. For this, we define the binary variable x_{ij} with $i = 1$ to n_t; $j = 1$ to n_s.

If $x_{ij} = 1$, the compounds i (in the target mixture) and j (for the source mixture) are selected to form a pair.

If $x_{ij} = 0$, the pair composed of compounds i and j is not selected.

A compound (of the source or target mixture) embedded in a selected pair, cannot be chosen in another one. This can be traduced by the following constraints:

$$\begin{matrix} \sum_{i=1}^{n_t} x_{ij} \leq 1 \quad \forall j \in [1, n_s] \\ \sum_{j=1}^{n_s} x_{ij} \leq 1 \quad \forall i \in [1, n_t] \end{matrix} . \tag{A.1}$$

We use $\leq$ because the mixtures of the target and source problems do not have the same number of components. To select the best pairs, i.e. to find the value of all x_{ij}, we have to maximize the following objective function:

$$F = \max \left(\sum_{i=1}^{n_t} \sum_{j=1}^{n_s} x_{ij} b\text{sim}_{ij} \right) . \tag{A.2}$$

For the example of Table 7, the selected pairs are x_{13}, x_{21}, x_{32}, and $F = 2.1$. Finally, the local similarity of the feature compounds is:

$$\text{sim}(t, s) = \frac{1}{m} \sum_{i=1}^{n_t} \sum_{j=1}^{n_s} x_{ij} b\text{sim}_{ij} \quad \text{with } m = \max(n_t, n_s). \tag{A.3}$$

REFERENCES

Aamodt, A. and Plaza, E., 1994, Case-based reasoning: foundation issues, methodological variations and system approaches. Artif Intell Commun, 7: 39–59.

Avramenko, Y., Nyström, L. and Kraslawski, A., 2004, Selection of internals for reactive distillation column—case based reasoning approach. Comput Chem Eng, 28(1/2): 37–44.

Cortes Robles, G., 2006, Management de l'innovation technologique et des connaissances: synergie entre la théorie

TRIZ et le Raisonnement à Partir de cas, Ph.D. Thesis, INP Toulouse.
King, J., Banares-Alcantara, R. and Zainuddin, M., 1999, Minimising environmental impact using CBR: an azeotropic distillation case study. Environ Model Softw, 14(5): 359–366.
Kister, H.Z., (1992). *Distillation Design*. (Mc Graw Hill Professional).
Kraslawski, A., Koiranen, T. and Nyström, L., 1995, Case-based reasoning system for mixing equipment selection. Comput Chem Eng, 19S(S1): S821–S826.
Lopez-Arevalo, Banares-Alcantara, R., Aldea, A., Rodriguez-Martinez, A. and Jimenez, L., 2007, Generation of process alternatives using abstract models and case based reasoning. Comput Chem Eng, 31(8): 902–918.
Mori, H., Ibuki, R., Taguchi, K., Futuma, K. and Olujic, Z., 2006, Three-component distillation using structured packing: performance evaluation and model validation. Chem Eng Sci, 61(6): 1760–1766.
Pajula, E., Seuranen, T., Koiranen, T. and Hurme, M., 2001, Synthesis of separation processes by using case-based reasoning. Comput Chem Eng, 25(4–6): 775–783.
Quinlan, R., 1979, Discovering rules by induction from large collections of example, in Expert Systems in Micro-Electronic Age, Michie, D. (ed) (Edinburgh), pp. 168–201.
Schank, R., (1982). *Dynamic Memory: A Theory of Learning in Computers and People*. (Cambridge University Press).
Smyth, B. and Kean, M.T., 1998, Adaptation-guided retrieval: questioning the similarity assumption in reasoning. Artif Intell, 102(2): 249–293.
Surma, J. and Braunschweig, B., 1996, Case base retrieval in process engineering: supporting design by reusing Flowsheets. Eng Appl Artif Intell, 19(4): 385–391.
Zadeh, L.A., 1965, Fuzzy sets. Inform Control, 8: 338–353.