Similarity concept for case-based design in process engineering

Yuri Avramenko, Andrzej Kraslawski *
Department of Chemical Technology, Lappeenranta University of Technology, P.O. Box 20, FI-53851, Lappeenranta, Finland

Received 31 March 2003; received in revised form 31 July 2005; accepted 17 October 2005
Available online 28 November 2005

Abstract

The design process requires significant engineering experience, intuition and creativity. To facilitate the preliminary design, the authors have proposed an approach based on the re-use of design experience. The design problems in chemical engineering are often quite difficult to represent as a well-structured list of features of one-type data. Many problems in chemical engineering are very large and complex, the problem description is often incomplete and uncertain, and sometimes it changes dynamically. In order to cope with design cases that have different structure representation the general similarity concept has been developed. The paper surveys the existing mathematical methods for similarity measurements and extends them to cope with different data formats. The main idea of this paper is an integration of representation and retrieval of the cases into one step. The proposed concept is illustrated by the examples of the model selection for synthesis of distillation systems and formulation of a fats and oils product.

© 2005 Elsevier Ltd. All rights reserved.
PACS: 07.05.K

Keywords: Case-based reasoning; Similarity measures; Design support

1. Introduction

Over the last few years, one of the methods of artificial intelligence, case-based reasoning (CBR) has attracted a general interest. The main idea of CBR is based on the assumption that the similar problems have the similar solutions. The practice shows that often it is more efficient to solve a problem by starting with a solution of a previous, similar problem than to generate the entire solution from scratch.

The central notion of CBR is a case. A case is problem-solving episode of experience that is represented as a pair: problem and its solution. Many cases are collected in a set to build a case library (case base).

In solving a current problem, a CBR system retrieves a similar, past problem and its solution using a set of rules for measuring similarity between actual problem and those stored in case base. Usually it is unlikely that an exact match will occur, therefore the retrieved solution must be adapted. The adaptation rules, based on the problem domain theory, are applied to adjust for any differences between the current case and the retrieved one. Finally, the CBR stores the approved solution to the current case, and it can then be used in solving future problems.

There are several advantages of using CBR. Instead of relying on general knowledge of a problem domain, CBR employs the specific problem characteristics. CBR is beneficial when the problems are not completely understood so that a reliable model cannot be built. Moreover, the problem may not be completely defined before starting to search for possible solutions. However, if the case base does not have sufficiently similar cases, then the retrieved solution may be inappropriate. In addition, the library of cases must roughly cover the set of problems that may arise in the considered domain of application to achieve the acceptable solution.

Due to the above-mentioned properties, CBR systems have attracted a great attention in legal and medical domains, especially as diagnostic and care systems, as well as in finance and insurance for customer support and credit assessment (Allen, 1994). In addition CBR has a diversity of applications in intelligent Web-based sales service (Watson & Gardingen, 1999; Wilke, Lenz, & Wess, 1998), in building and mechanical design (Rivard & Fenves, 2000; Mileman, Knight, Petridis, Cowell, & Ewer, 2000), in material science (Amen & Vomacka, 0098-1354/$ – see front matter © 2005 Elsevier Ltd. All rights reserved.
In support to the complex fault finding and troubleshooting (Aha, Breslow, & Munoz-Avila, 1999) as well as in planning and real-time scheduling tasks (Bonzano, Cunningham, & Smyth, 1997; Coelho & dos Santos, 1999). In diagnostic and care systems actual problem is matched against previous cases in order to find appropriate class or group of problems and to give solution corresponding to this class. The cases are already grouped into typical classes and during the retrieval, the standard classification task is performed. In many other cases CBR systems deal with the structured queries of numeric data and the retrieval procedure consists of the simple calculations.

However, the chemical engineering problems, especially design tasks, are often quite difficult to represent as a well-structured list of features or two or data types. The representation of design cases requires various models because design content involves topological, geometric, and physical properties and relations between them (Maher, Balbas Balachandran, & Zhang, 1995). Many problems in chemical engineering are very large and complex, the problem description is often incomplete and uncertain, and sometimes it changes dynamically.

Recently, CBR has been applied in chemical engineering for quality design (Suh, Jhee, Ko, & Lee, 1998), thermal analysis support (Nakayama & Tanaka, 1999), troubleshooting plant problems (Chaput, 1999), process control and plant supervision (Roda, Poch, Sanchez-Matre, Cortes, & Lafuente, 1999; Sanchez, Cortes, Roda, Poch, & Lafuente, 1997), ecological tasks (King, Banaras-Alcantara, & Manan, 1999), and supporting design in process engineering, more specifically equipment selection (Kraslawski, Lysov, Kudra, Borowiak, & Nystrom, 1999; Kraslawski, Pedrycz, & Nystrom, 1999a), and process synthesis and flowsheet design (Pajula, Seuraran, & Hurme, 2001; Surma & Braunischweig, 1996). In several of these works the problems are described as a simple set of features, but with either the ability of fuzzy representation (Kraslawski et al., 1999a), taxonomy representation (Pajula et al., 2001), or complex case library organization (Roda et al., 1999). Whereas in (Surma & Braunischweig, 1996; Nakayama & Tanaka, 1999) the structures of case description are very complex and contain many types of data and graphical representation. There have been used some techniques to cope with the case complexity. From the cited examples, it can be seen the laborious efforts in representation and management of complex cases.

In order to describe the CBR process, there have been proposed several general models. Kolodner (1993) considers CBR as a process containing the following steps: case retrieval as a primary step, proposing area of solutions by extracting them from some retrieved cases. Next, adaptation, the process of fixing a past solution to fit a new situation, criticism of received new solution, its evaluation based on external feedback, and finally storing of verified solution of the current problem in the case library. Allen's model (Allen, 1994) includes five steps: presentation of the current problem, retrieval of the closest-matching cases stored in a case-base, its adaptation for generating a solution for the current problem, validation of the solution through feedback, and updating of the case base with validated solution.

Aamodt and Plaza (1994) also introduce a model, which consists of the following phases: retrieve the most similar cases, reuse the cases, revise the proposed solution, and retain the new solution as a part of a new case.

However, they assume that the case base is ready "at once" for case retrieval, and ignore the fact that case base building is also an important CBR task. Finnie and Sun (2003) in the recent work have considered the process of preparation of case base. They have proposed the new model of CBR process composed of the following main tasks: repartition, retrieval, reuse, revision and retaining. They present the concept of a similarity relation (Plaze, Esteva, Garcia, Goda, & de Mantaras, 1996) and apply it for case base building. The proposed approach is general one, and no implementation of similarity relations with the real data is proposed. Moreover, the authors assume that the case data are already structured according to some representation. However, in many real world problems, especially those of chemical engineering, the data format is not uniform and information is distributed among many sources located in the different databases. The attempt to integrate the data analysis techniques into CBR system has been recently done in Lau, Wong, Hui, and Pun (2002).

The main idea of this paper is the integration of the representation and retrieval of the cases. The cases are usually stored in databases in free, unstructured format, and they can be even distributed among many databases. The structure of representation can be dynamically changed in CBR process and may involve different data format for representation of the value of the features. Use of the proposed general similarity concept creates the possibility to compare cases that have different structure representation in the case base. Moreover this representation can be even built dynamically before retrieval.

The paper presents the general similarity concept, surveys the existing mathematical methods for similarity measurements and extends them to cope with different data format.

The proposed concept is illustrated in CBR system for the selection of a model for synthesis of distillation systems.

2. Retrieval method and similarity measures

In CBR systems, the quality of the results mainly depends on the similarity measure that is used to retrieve the similar cases. It is so as the retrieved cases constitute the starting point for finding the solution of the actual problem. During the retrieval procedure, the current problem is matched against the problems stored in the case base. Matching is the process of comparing two cases to each other and determining their degree of similarity. If the case is represented as a set of features and their values, the similarity measure between two cases can be determined by the following operations (Kolodner, 1993):

- find the corresponding features in the compared cases;
- for every feature, compute the degree of similarity between the corresponding features of the cases;
- multiply the obtained values by the corresponding coefficient reflecting the importance of the feature (weight of importance) and sum them to get overall similarity value.
2.1. Quantitative distance

Most of engineering parameters is of quantitative nature. The value of the parameters can be expressed as a number, a vector, a set or an ordered sequence. The computation of similarity between numeric values is based on nearest-neighbour algorithm (Voss, 1995). The nearest-neighbour method relies on a metric. Distance is one measure of vector similarity. The larger the distance between two vectors, the smaller their degree of similarity.

Euclidian distance for vectors \(a\) and \(b\) in \(n\)-dimensional space is:

\[
d(a, b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}
\]

(1)

More general metric for \(n\)-dimensional vectors is the Minkowski metric that is also referred as the \(L_2\) norm (Schalkoff, 1992):

\[
L_k(a, b) = \left( \sum_{i=1}^{n} |a_i - b_i|^k \right)^{1/k}
\]

(2)

when \(k = 2\) the norm (2) transforms to the Euclidean distance. The norm with \(k = 1\) is called the city block (or Manhattan) distance. It is the shortest path between a and \(b\), each segment of which is parallel to the coordinate axis.

The block city and the Euclidean equations are wide-spreaded as the similarity measures in CBR applications. The block city is used for numeric values, for example, in (Amen & Vomacka, 2001; Pagula et al., 2001), or with modifications of the criteria in (Roda et al., 1999; Sanchez et al., 1997). The Euclidean metric is employed as a similarity function for numeric features, for example, in (Avramenko, Nystrom, & Kraslawski, 2002).

2.1.1. Numeric vectors

Distance is one measure of vector similarity. The larger the distance between two vectors, the smaller their degree of similarity.

There are several approaches to compute the degree of similarity of the features based on distance on a quantitative scale, a position in a hierarchical structure, comparison of qualitative definitions (this is determination of similarity between values expressed on qualitative scale like ‘high’, ‘low’, etc.) and structural comparison.

2.1.2. Sets

The similarity measures of the set rely on a metric function as well.

The distance between sets \(A\) and \(B\) is defined usually by Tanimoto metric (Duda, Hart, & Stork, 1998):

\[
d(A, B) = 1 - \frac{|A \cap B|}{|A| + |B| - |A \cap B|}
\]

(3)

The metric is mostly used for solving problems where the elements of the set are equally important, and there is no natural notion of graded similarity. This metric (with additional coefficients, specific for the task studied) is used in reference Surma and Braunschweig (1996) to compute the aggregation similarity between the components of flowsheet, and to determine the connection similarity that focuses on the stream connections between its components.

In Nakayama and Tanaka (1999), the authors used the sets distance measure to determine the similarity between the cases for the feature ‘phase’ that is represented as a list of terms such as glass, gum or crystal.

Another related measure is the Levenshtien distance (Schalkoff, 1992):

\[
d(A, B) = \max(|A|, |B|) - |A \cap B|
\]

(4)

The equation was applied in reference Avramenko et al. (2002) to determine the similarity between the features expressed as textual variables, which is represented as a list of keywords. The most significant domain notions were collected in one set, specific for each textual feature type, and hence each context of feature is a subset of those full set of keywords.

2.1.3. Textual strings and sequences

A string in general can be represented as a set of words or symbols. Very often the ordering of the elements is important, and the string must be considered as a sequence of symbols. For such sequences, element-by-element comparison is performed and similarity measure is based on computation of a Hamming distance. The Hamming distance of two sequences of equal length is calculated by counting the character positions in which they differ.

Another similarity metric relies on determination of so-called edit distance between two strings.

The edit distance between strings \(A\) and \(B\) determines how many fundamental operations are required to transform \(A\) into \(B\) (Duda et al., 1998). These fundamental operations are:

- substitutions: a symbol in \(A\) is replaced by the corresponding symbol from \(B\);
- insertions: a symbol in \(B\) is inserted into \(A\), thereby the length of \(A\) is increased by 1;
- deletions: a symbol in \(A\) is deleted, thereby the length of \(A\) is decreased by 1.

The value of the edit distance is calculated as the minimum number of basic operations needed to transform \(A\) into \(B\). Sometimes the feature has a unique value and no partial match is allowed. That is impossible to determine the grade of similarity for such values. In this case, only an exact match is considered and the value of the similarity measure is either 1, if one feature
Another way of measuring degree of similarity is determination of the distance between two values on a qualitative scale (Kolodner, 1993). The distance between the values belonging to the same qualitative category is considered equal to 0. Otherwise, the distance between two qualitative values is determined by the number of other categories separated the values from each other on the qualitative scale. The more qualitative categories separate two qualitative values, the lower the similarity measure. An integer value can be assigned to each qualitative category and the similarity measure between two qualitative variables can be computed by determination of distance between these integer numbers.

The use of qualitative comparison for the calculation of the degree of similarity is advantageous when small differences in features values are irrelevant to the degree of match. However, a problem may arise when similarity measure is inaccurate at the edges of the ranges.

The qualitative scales are widely used in many CBR applications. In Mejasson et al. (2001), the qualitative scale is used to represent the operating parameters, such as temperature etc. The authors have used five-degree scale: very low, low, medium, high and very high. The maximum distance of 1 is between the highest and the lowest categories (Table 1). There are two qualitative factors describing the value and reliability of the design case in application (Pajula et al., 2001). These are technical maturity and performance (goodness) factor. The latter describes the proven efficiency of the design and has six gradation levels (it is numbered from 0 to 5 in order to compute a similarity measure).

Qualitative variables are used also in the reference Chaput (1999) to describe the troubleshooting plant situations.

### 3. General similarity concept

In the framework of the concept an entity from the real world (that can be a substance, a phenomena or a process) is considered only in form of its description – information content that characterizes an entity by a finite set of its properties and their relations. We do not consider entities themselves – process, phenomena – but their descriptions represented by symbols, numbers, sets, graphs, that is their information content. Further the term ‘entity’ implies exactly information description of an entity.

Let introduce the basic definitions.

**Definition 1.** An entity $E$ is a three-tuple and is defined as follows:

$$E = (F, V, R)$$

where $F$, a finite set of features of an entity reflecting their nature; $V$, a set of values of features; $R$, a set of relations between features.
Definition 2. The set of features \( F \) and the set of relations \( R \) form the structure \( S \) of entity \( E \):

\[
S = (F, R) \]

The representation of the structure may be extended by including numerical and perhaps symbolic attributes of features of an entity. The attributes may reflect a degree of importance of corresponding feature in the description. In such a case the structure is called weighted and is defined as follows:

\[
S = (F, R, W) \]

where \( W \) is a set of weights corresponding to features of an entity.

The entities that have identical structure are considered to belong to one type. Obviously, the entities belonging to one type have the equal number of elements in the sets and the features in both sets correspond to each other.

If for two entities \( E_1 = (F_1, V_1, R_1) \) and \( E_2 = (F_2, V_2, R_2) \), \( F_1 \subseteq F_2 \) and \( R_1 \subseteq R_2 \) or \( F_2 \subseteq F_1 \) and \( R_2 \subseteq R_1 \), then the entities are denoted as structurally similar, otherwise the entities are structurally dissimilar.

Further only the structurally similar entities are considered. Two entities \( E_1 \) and \( E_2 \) will be described as \( E_1 = (S_1, V_1) \) and \( E_2 = (S_2, V_2) \), where \( S_1 = (F_1, R_1) \) and \( S_2 = (F_2, R_2) \).

Definition 3. Two features of different entities are denoted as corresponding if they are in the same relations with other features in the structures.

Definition 4. Two entities are similar if all or some part of their corresponding features has identical values. Otherwise, when no part of values is matched, the entities are dissimilar.

If all corresponding values of entities \( E_1 \) and \( E_2 \) belonging to one type are matched, i.e. \( S_1 \equiv S_2 \) and \( V_1 \equiv V_2 \), the entities are called identical. Obviously, all identical entities are similar as well when only a part of corresponding values are different the entities are partly similar.

Definition 5. If all or a part of corresponding features of two entities have values that belong to certain classes (specific for each pair of features) of specific classification, then the entities are called conditionally similar. The condition implies a set of rules that divide the features values into the classes.

The determination of a membership of a value to some class is a standard task of pattern recognition, and therefore it can be done by the respective methods. The conditional similarity is approaching the simple similarity with increasing the specificity of classification law, i.e. when the classes becoming narrower and therefore including less items (values of features). The similar entities are also conditionally similar when each feature value creates own specific class.

The introduced notion of similarity is not enough useful in practice, since the entities will be considered as similar when both all values and even only one value of the features are equal. Therefore, it is necessary to define the degree of similarity between two entities.

Definition 6. Degree of similarity is the value expressed how much one entity is similar to another entity and is defined by a ratio of the number of matched features of two entities to overall number of features in the structure. Thus, degree of similarity (denoted as \( \text{sim} \)) is greater than 0 but lower than 1 for partly similar entities; is equal 1 for identical entities; and equals 0 for dissimilar entities.

Hereinafter instead of term ‘degree of similarity’ a shorter term ‘similarity’ is used unless there is a need to distinguish between them.

3.1. Overall and partial similarity

For two entities \( A \) and \( B \) of one type (i.e. \( S_A \equiv S_B \)), defined by sets of values \( V_A = \{a_1, a_2, \ldots, a_n\} \) and \( V_B = \{b_1, b_2, \ldots, b_n\} \), where equal indices determine the corresponding features of the entities, the similarity is defined as

\[
\text{sim}(A, B) = \frac{1}{n} \sum_{i=1}^{n} \text{ident}_i, \quad \text{where } \text{ident}_i = \begin{cases} 1, & a_i = b_i \\ 0, & a_i \neq b_i \end{cases}
\]

(6)

If the entities have the weighted structure, then the similarity is defined by taking into account the values of weights \( w_i \):

\[
\text{sim}(A, B) = \frac{\sum_{i=1}^{n} \text{ident}_i w_i}{\sum_{i=1}^{n} w_i}
\]

(7)

For the conditionally similar entities there is the notion of conditional degree of similarity:

\[
\text{sim}(A, B(C)) = \frac{\sum_{i=1}^{n} \mu_i w_i}{\sum_{i=1}^{n} w_i},
\]

(8)

where \( \mu_i = \begin{cases} 1, & a_i, b_i \in C_k \\ 0, & a_i \in C_k, b_i \notin C_k \end{cases} \)

In the last equation, \( C \) means the condition, \( a \) is the classification law, \( \mu \) determines the membership of values \( a \) and \( b \) of corresponding features to a class. If the bounds of a class cannot be certainly determined, then the function \( \mu(a, b) \) takes the values within the interval \([0, 1]\). If \( b \) exactly belongs to a class, then \( \mu(a, b) \) shows how \( a \) is similar to \( b \). Contrary to overall similarity (also called global) between the entities the similarity between feature values is called partial or local.

The notion of similarity is opposite to the definition of difference. Therefore, it is possible to come to the standard task of determination of the distance between two values.

Definition 7. The local similarity \( l \) for two features \( a \) and \( b \) is defined as:

\[
l(a, b) = 1 - d(a, b)
\]

(9)

where \( d(a, b) \) is the distance function, as defined in the subsequent chapter.

The value of distance \( d \) is from the interval from 0 to 1. If the features are completely different then \( d \) equals 1.

The determination of the distance between two values of the different data types has been presented in Section 2. Further only comments related to the concept will be presented.
3.2. Distance measurements

Let \( \Delta \) be the absolute distance between two features having the values \( a \) and \( b \); and \( d \) its relative value, that is the ratio of absolute distance to maximum distance.

3.2.1. Numeric values

For the numbers the distance is determined by the absolute value of difference between numbers:

\[
\Delta = |a - b|, \quad d = \frac{|a - b|}{\text{range}}
\]  

(10)

where range is the range of values of variable \( a \) and \( b \).

For \( n \)-dimensional vectors \( a = (a_1, a_2, \ldots, a_n) \) and \( b = (b_1, b_2, \ldots, b_n) \), the distance is calculated using Eq. (1). If all coordinates of the vectors are equivalent, it means the coordinates are equally important and none can be emphasized, it is necessary to make normalization: the real values of coordinates are converted to relative ones, belonging to interval \([0, 1]\).

The maximum ranges of the values of the coordinate create in space \( \mathbb{R}^n \), an area which during conversion to the relative coordinates, forms \( n \)-dimensions unit cube. The cube contains all vectors from gathered set—a set of all considered vectors. Let transform this area into a new space with basis vectors \( e_1 = (1, 0, \ldots, 0) \), \( e_2 = (0, 1, \ldots, 0) \), \ldots \( e_n = (0,0, \ldots, 1) \). Each basis vector corresponds to maximum change of corresponding coordinate for vectors of the gathered set. It determines the maximum possible distance along one of the coordinate. The maximum distance between two points on the cube is the diagonal of the cube. It is also the sum of basis vectors; thus the relative distance between two vectors is defined as

\[
d = \frac{|a - b|}{|e_1|}
\]

where \( \frac{|e_1|}{|e_2|} \) is the sum of basis vectors; thus the relative distance between two vectors is defined as

\[
\tilde{d} = (1, 0, \ldots, 0)
\]

(11)

The Eq. (11) is suitable for a number as for one-dimensional vector.

3.2.2. Sets

The value of the distance function for the sets is determined by the number of the elements in the sets, which are not common, i.e. the distance for the sets \( a \) and \( b \) that equals to cardinal number of set \((a \cup b) \cap (b \cup a)) \), the set difference of \( a \) from \( b \) and difference of \( b \) from \( a \). Formally as:

\[
\Delta = |(a \cup b) \cap (b \cup a)| - |a \cap b|
\]

\[
d = \frac{|a \cup b| - |a \cap b|}{|a \cup b|} = 1 - \frac{|a \cap b|}{|a \cup b|}
\]

(12)

The last equation is the Tanimoto formula (3).

The Eq. (5) can be obtained if a feature with unique value is represented as a set with only one element. In this case, according to the Eq. (12), only the exact match can get the value that differs from 0.

3.2.3. Members of hierarchical structure

Let the values of feature belong to a given hierarchy. They might be located in different branches of the hierarchical tree with the different depths. A value can be described by the path from the root of hierarchy to the dangling vertex that corresponds to this value. For example of a tree in the Fig. 1, the elements \( a \), \( b \) and \( c \) are described as

\[
a = \{n_1, n_5, n_{10}\}; \quad b = \{n_1, n_{10}\}; \quad c = \{n_2, n_4\}
\]

The distance between the vertexes is defined as the maximum difference of their paths: maximum length of path from first common node where their paths have met, i.e.

\[
\Delta = \max(|a' \backslash b|, |b' \backslash a|)
\]

or in more convenient way

\[
\Delta = \max(|a' \backslash b|, |b' \backslash a|) = \max(|a' \backslash b|, |b' \backslash a|) = \max(|a' \backslash b|, |b' \backslash a|) = \max(|a' \backslash b|, |b' \backslash a|) = \max(|a \cap b|, |b \cap a|) = \max(|a \cap b|, |b \cap a|) = \max(|a \cap b|, |b \cap a|) = \max(|a \cap b|, |b \cap a|)
\]

(13)

It is the distance Levenshtien formula (4). The formula proved to be good for comparison of the members of the hierarchical tree.

The maximum distance is determined as the distance between the vertexes that are located in the different branches of the tree that are connected only in the root. Thus the relative distance is determined as

\[
d = \frac{\max(|a' \backslash b|, |b' \backslash a|)}{\max(|a \cap b|, |b \cap a|)} = 1 - \frac{\max(|a \cap b|, |b \cap a|)}{\max(|a \cap b|, |b \cap a|)}
\]

(14)

3.2.4. Qualitative values

The qualitative values can be encoded using integer numbers (see Chapter 2) and then the distance between them is determined using Eq. (10).

4. Applications of the general similarity concept

A process synthesis and product development examples are considered to illustrate the method based on general similarity concept. First example is the conventional process design task of synthesis of distillation system. The second one is a formulation of fats and oils product.
4.1. Selection of model for synthesis of distillation system

The design problem can be stated as follows. An ideal or close to ideal mixture of arbitrary components is to be separated into a number of products of specified compositions by means of distillation. The goal is to get a right model as a starting point for the process of synthesis of distillation column or distillation sequence.

The case library contains detailed examples of separation of ideal mixtures for up to five components. Cases are composed of the features of feed mixture and required products, the operational parameters, the mixed integer nonlinear programming model with superstructure and the optimal flowsheet with detailed specifications.

4.1.1. Representation of cases

Cases are represented as entities with the set of features, their values and relations. The model, consisting of a superstructure, a set of variables and parameters, the mass and enthalpy balances and other constraints, together with flowsheet and its mathematical representation builds the solutions part of a case. The form of model varies with different layouts of distillation system. Therefore, the structure of entities that also involves the model specifications is dependent on certain characteristics of the distillation system.

There have been selected three characteristics affecting the model structure. First one is the type of heat integration. A system can be either with heat integration or without it or thermally coupled. In single column configuration only non-heat integrated structure is possible. The single column configurations and models do not include the mass balances for the connections of distillation columns; and so, these models cannot be used for problems with three or more products. In addition, there is the dissimilarity between the models with single and multiple feeds. Therefore, number of products (with only two options: two or more products) and number of feeds (also only two options: one or more) affect the structure of entities as well. A total number of different structures being determined by combination of these characteristics is twelve (3 × 2 × 2 = 12). However, some of the combinations are not realistic. The different structures presented in the case library are given in Table 2.

The entities with different structures are structurally dissimilar (see Chapter 4) and they are not considered simultaneously during retrieval procedure. Only entities which belong to same type \((E_1 = (S_a, V_a), E_2 = (S_b, V_b), a = b)\) are compared.

The list of features describing the problem is shown in Table 3.

Table 2

<table>
<thead>
<tr>
<th>Name</th>
<th>Heat integration</th>
<th>Number of feeds</th>
<th>Number of products</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>No</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>S2</td>
<td>No</td>
<td>1</td>
<td>More than 2</td>
</tr>
<tr>
<td>S3</td>
<td>No</td>
<td>More than 1</td>
<td>2</td>
</tr>
<tr>
<td>S4</td>
<td>No</td>
<td>More than 1</td>
<td>More than 2</td>
</tr>
<tr>
<td>S5</td>
<td>Normal</td>
<td>1</td>
<td>2 or more</td>
</tr>
<tr>
<td>S6</td>
<td>Thermally coupled</td>
<td>1</td>
<td>More than 2</td>
</tr>
</tbody>
</table>

Table 3

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Components</td>
<td>Set of elements</td>
</tr>
<tr>
<td>Feeds</td>
<td>Set of elements</td>
</tr>
<tr>
<td>Products</td>
<td>Set of elements</td>
</tr>
<tr>
<td>Sharp separation</td>
<td>Logical</td>
</tr>
<tr>
<td>Maximum number of trays per column</td>
<td>Numeric</td>
</tr>
<tr>
<td>Additional constraints</td>
<td>Textual</td>
</tr>
</tbody>
</table>

The advantage of the described concept consists in the state that each feature can be represented as a new entity. This situation leads to a universal way of calculation of similarity between features: the local similarity between features-entities is the overall similarity of all components of those features, and if the components are represented as entities as well they are treated in similar way. In the problem description, each element of set in the list of features ‘Components’, ‘Feeds’ and ‘Products’ is an entity with own structure. The lists of features of those entities are put in Tables 4–6.

4.1.2. Similarity determination

The overall similarity for the entities has been defined by following equation:

\[
\text{sim}(E_1, E_2) = \frac{\sum_{i=1}^{S} w_i \text{sim}_i}{\sum_{i=1}^{S} w_i}
\]

where \(\text{sim}_i\) is the local similarity and \(w_i\) is the weight of importance of \(i\)-feature.

The similarity between numeric features (such as molar weight, boiling point etc.) is calculated according to Eq. (10).

The similarity of components is based on theirs chemical structure. The hierarchical similarity tree was presented in Fig. 2.

Table 4

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Hierarchical</td>
</tr>
<tr>
<td>Molar weight</td>
<td>Numeric</td>
</tr>
<tr>
<td>Boiling point</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

Table 5

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow rate</td>
<td>Numeric</td>
</tr>
<tr>
<td>Composition</td>
<td>Vector</td>
</tr>
</tbody>
</table>

Table 6

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow rate</td>
<td>Numeric</td>
</tr>
<tr>
<td>Temperature</td>
<td>Numeric</td>
</tr>
<tr>
<td>Pressure</td>
<td>Numeric</td>
</tr>
<tr>
<td>Composition</td>
<td>Vector</td>
</tr>
</tbody>
</table>
where the nodes represent the basic group of chemical compounds. Each compound group has a numerical value of similarity. For two components the similarity value is the value of the closest common node. The more similar the components, the higher the similarity value between them. For the identical components the similarity value is 1.

The components in the sets can be placed in any order. Hence, in order to find two identical components in the different sets there is a need to look over all elements in the sets. More general task is to find the most similar pairs of components belonging to the different sets.

For two sets $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_k\}$, there is a need to find such matching (graph theory notion) $m = ((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_k))$ that $\sum_{(a_i, b_j)} \text{sim}(a_i, b_j)$ is maximum. If $i \neq k$ then only the elements with the maximum pair similarity are included in the matching, the rest of elements (difference between $n$ and $k$) have no match. When the matching is found the elements of one set are rearranged to have the equal order with the most similar element from another set. This order of the most similar pair of components is used in vector similarity measurement of composition value (mole fractions).

The same procedure is applied in the measurement of similarity between ‘Feeds’ as well as between ‘Products’. The similarity between vectors is calculated according to Eq. (11).

The Eq. (5) can be applied for logical variables.

Additional constraint is a textual variable that is represented as a set of important keywords, while the rest of the words are ignored. The similarity values between the sets of keywords can be determined by Eq. (5).

4.1.3. Example: separation of heptane–toluene mixture

The described concept has been used to support of design of distillation system for a heptane–toluene mixture.

The flowrate of the feed (toluene, 0.5; heptane, 0.5) is 100 kmol/h. The goal is to separate the mixture into pure components with 95% purity requirement at the top and at the bottom. It is a sharp separation problem with single column configuration should be used. Therefore the structure of new problem is $S_1$ (not heat-integrated, one feed and two products). There had been identified five cases of such structure in the case base. Four of them showed the purity of the product compositions [0.95, 0] at the top, and [0,0.95] at the bottom. The overall similarity between new problem and identified source cases varied from 0.312 to 0.801. Thus, the most similar case with similarity of 0.801 was benzene–toluene problem (source: Yeomans & Grossmann, 2000). The superstructure and the MINLP model of this most similar case have been used as initial state in adaptation through simulation. The column diameter and the reflux ratio are increased according to the lower relative volatility.

The MINLP model has been solved by GAMS DICOPT++ (Brooke, Kendrick, & Meeraus, 1988) on a Sun Sparc Station in Budapest University of Technology and Economics, on Department of Chemical Technology by Tivadar Farkas and Zoltan Leikes. They found the following column specification: the number of trays of the column, 60; feed location (from bottom) on 31st tray; column diameter 1.51 m, and reflux ratio 4.23.

4.2. Formulation of fats and oils products

Fats and oils are key functional ingredients in a large variety of foods. Successful development of the food products relies on effective use of the different functional properties of the available fats and oils and the manipulation of the fat blend to satisfy the prepared foods requirements.

Application development of fats and oils products begins with identification of the key functional attributes that the final product is expected to provide. The important functional attributes, which is considered for product development, are: lubricity, structure, clarity, consistency, plasticity, emulsion, creaming property, spreadability, aeration, hardness, freeze stability, flavor (odor, taste and mouth feel) and flavor stability. The product functionality can usually be translated into analytical measurements and physical properties. For example, mouth feel and flavor release can be controlled by the melting properties and oxidative stability. The solids fat index (SFI) curve characterizes the consistency and spreadability of the product.

Next step is the use of historical knowledge to identify and evaluate the physical and chemical properties most likely to produce the intended functionality. The database of physical properties and compositions of nature and genetically modified oils and fat blends has been created based on materials of book (O’Brien, 2004) and other sources.

4.2.1. Representation of cases

A case description contains composition, physical and chemical properties as well analytical test results of fats and oils blend.

The solution part includes composition of a blend, and it is represented by four entities: fatty acids (e.g. stearic, palmic, linolenic etc.), tocopherols (α-,β-,γ-,δ-), tocotrienols (α-,β-,γ-,δ-), and triglycerides (tristaturated, disaturated, monosaturated, trinsaturated). The list of features included into the problem part of a case is given in Table 7.

The oxidative stability is represented by the separate entity because it is complex characteristic and some of its components might not be available but similarity must be calculated over this property. Solid fat index curve is represented by a set, each element of which is a vector in temperature-index space.
4.2.2. Similarity determination

The functionality of the product under development translated to the physical properties and the test results is used to identify the oils with most similar properties. The similarity between numbers is calculated using Eq. (10), between vectors identify the oils with most similar properties. The similarity related to the physical properties and the test results is used to

4.2.3. Example: cookie filler development

The goal is to design specific cookie filler with eating character and flavor stability, high plasticity, with soft and delicate mouth feel. The product must have high oxidative stability.

For a fat blend to be plastic, it must have both a solid and liquid phase. The ratio of these two phases determines its consistency. In addition, a melting point must be lower than body temperature to have good eating characteristics. Thus, the solid fat indices curve must be steep around room temperature. The product requires as low iodine value as possible for oxidative stability. Summarizing the requirements, the product features must satisfy to values presented in the Table 8. The crystal habit and some parameters can be easily shifted from one description to another. Same features can be considered as a solution specification or additional requirements for the problem description. Thus the structure of the entities can be dynamically changed; during case-based reasoning. The proposed general similarity concept is able to cope with the cases that have different structure representation and some features can be excluded or conversely included in the problem description is often incomplete and uncertain, and sometimes it changes dynamically what creates the difficulties during case-based reasoning. The proposed general similarity concept is able to cope with the cases that have different structure representation and some features can be excluded or conversely included in the problem representation to get steeper SFI slope.

The results are given in Table 9.

The needed processing of fats and oils raw materials can be selected during the validation phase after analysing the modified composition. The desired composition and physical properties can be achieved by blending, hydrogenation, and emulsification of palm kernel and coconut oils.

5. Summary

Case-based design is an approach based on the reuse of the past experiences in finding the solutions to the new, similar problems. The approach is beneficial when the problems are not completely understood and a reliable model cannot be built. It is a typical situation for most of design tasks. The chemical engineering design problems are often quite difficult to represent as a well-structured list of features of one data types. The problem description is often incomplete and uncertain, and sometimes it changes dynamically what creates the difficulties during case-based reasoning. The proposed general similarity concept is able to cope with the cases that have different structure representation in the case base and contain the features expressed in the different format, as numbers, vectors, sets, sequences, graphs etc.

The problem and solution descriptions can be quite flexible and some parameters can be easily shifted from one description to another. Same features can be considered as a solution specification or additional requirements for the problem description. Thus the structure of the entities can be dynamically changed; some features can be excluded or conversely included in the structure and the retrieval method will not change.

The concept has been illustrated with application to process synthesis (the selection of a model for synthesis of distillation systems) and product development (fats and oils blend formulation) problems.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iodine value</td>
<td>20</td>
</tr>
<tr>
<td>AOM test (h)</td>
<td>72</td>
</tr>
<tr>
<td>Melting point</td>
<td>25</td>
</tr>
<tr>
<td>Crystal habit β'</td>
<td></td>
</tr>
<tr>
<td>Solid fat indices (10°C, 60%), (20°C, 30%), (25°C, 10%), (25°C, 9%)</td>
<td></td>
</tr>
</tbody>
</table>
Based on the described similarity concept the integrated case-based design supporting system can be built. The information describing the design cases is stored in free format in several different databases. The structure of case can be dynamically changed according to the query requirements. Based on this query, the design data is processed and the representation structure of the cases is built. The case base is created and then CBR is employed to find a solution of actual problem introduced to the system. In such a situation the cases representation and retrieval phases of CBR process are integrated into one step. The development of such system is the subject of future work.

References


Willke, W., Lena, M., & Weiss, S. (1998). Intelligent sales support with CBR. In M. Lenz, et al. (Eds.): Case based reasoning technology, from foundations to applications, LNCS 1400 (pp. 91–113). Berlin: Springer.