
This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

A case-based reasoning approach for low volume, high added value electronics

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© IEEE

VERSION

VoR (Version of Record)

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Segura-Velandia, Diana M., Andrew A. West, Paul P. Conway, and David C. Whalley. 2019. "A Case-based Reasoning Approach for Low Volume, High Added Value Electronics". figshare.
<https://hdl.handle.net/2134/4199>.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

A Case-Based Reasoning Approach for Low Volume, High Added Value Electronics

Diana M. Segura-Velandia, Andrew A. West, Paul P. Conway, David C. Whalley
Wolfson School of Mechanical and Manufacturing Engineering
Loughborough University, Leics, LE11 3TU, United Kingdom
Tel.00+440519227677
{d.segura@lboro.ac.uk}

Abstract

This paper will report on the application of the Case-Based Reasoning (CBR) approach [1] to develop a defect prediction system to support the development of new printed circuit assembly (PCA) products. Using a CBR system, past PCA design specifications and manufacturing experiences including defect and yield results can be effectively stored and re-applied for future problem solving. For example, the CBR can then be used at design stage to amend designs or define process options to optimise the product yield and service reliability. A case study using a case-base provided by a PCA manufacturer is presented.

1. Background

The Printed Circuit Assembly (PCA) manufacturing process in low volume, high added value electronics still require a level of manual intervention during product manufacture that can lead to poor first time yield and increased production costs. The reasons are not fully understood but explained by failures at the component-level and most importantly and less studied, failures that stem from non-component causes (i.e. *system-level*) such as defects in design and manufacturing. Historically, these factors have not been incorporated in prediction models, mainly due to the fact that system-failure causes are not driven by well-characterised deterministic processes and also there are not enough structural data for any confidence in statistical process control (SPC) driven models. Instead skilled engineers attempt to predict and generate a mapping between the possible defects and product reliability. This approach requires strong engineering experience and it is time consuming.

In this paper, the application of the CBR methodology to support the problems arising in the design of low volume highly complex electronics is studied. Specifically the CBR methodology circumvents the lack of understanding and quantitative models for predicting defect at both the component and system level through the representation of expert's PCA specifications, defect and historical reliability data into a CBR framework for problem solving. Given a user query, which is a problem description, the CBR system retrieves similar experiences with the ultimate objective of predicting what designs, if any, will allow the resulting product to meet the desired performance criteria. Implementation of the CBR system in an industrial setting allows (i) to study how time and hence the cost of the PCA product development process is reduced and (ii) most importantly, enables simulation studies at the production line to optimise yield and reliability.

Several researchers in both academia and industry have made use of the CBR methodology to solve problems present in the development of aluminium alloys [2, 3], steel products [4], pharmaceuticals [5, 6], steel product quality design [7], rubber [8] plastics manufacturing [9] and more recently in electronics design [10-12]. In general, the drivers for developing these knowledge-based systems (KBS) were concerned mainly with:

- The practice of the discipline was chiefly regarded more as an art rather than as a science and therefore the domain was poorly understood;
- There does not exist comprehensive mathematical models to support product design and manufacture;
- Experience of the manufacturing personnel plays a crucial role for successful processes and troubleshooting, however, there is a need to formalise such experiential knowledge for satisfactory future problem solving;
- There is a need for collecting, structuring, and representing manufacturing knowledge from experts using a computational system that enables the sharing and deployment of solutions in a consistent and referable way when appropriate for future problem solving.

These needs have some resemblance to the needs experienced in the PCA domain. The advantages of CBR in problem solving that have been claimed by several researchers (see for example [13 p.48-50], [14], [15 ch.1 p2-4], [16 p234]) include:

Case-based reasoning allows:

1. *Avoiding repeating mistakes made in the past.* By definition CBR systems must learn. By recording both successful and unsuccessful experiences a CBR can be used to predict potential failures in the future.
2. *Learning over time.* If the solution cases proposed by CBR systems are tested in the real world, knowledge about the correctness of the proposed solution is obtained. Therefore, new cases can be added to the case-base and used in future problem solving while also anticipating and preventing errors.
3. *Reasoning in a domain with a small body of knowledge.* For domain problems in which a few cases are available (i.e. low volume high complex electronics), a CBR system can build its knowledge incrementally as the cases are added and hence increase its efficiency over time.
4. *Reasoning with imprecise data.* Soft computing tools are being used to support CBR tasks to cope with uncertain and imprecise data [2, 3]. Although CBR retrieves cases that might not be identical to the current problem the technique offers an alternative solution with various degrees of correctness (depending on the similarity measure used).

5. *Providing a means for explanation.* A CBR user finds it easier to explain how a solution was arrived at by using the similarities between the cases and the reasoning involved in adaptation.

2. General Architecture of the CBR System

A CBR system consists essentially of two main components that are activated to perform specific tasks of the case-based problem solving process i.e. a *Case-Based (CB) module*, which is explained in this paper, and an *Adaptation module*. The CB module allows the user to describe new and past cases (case description) and it is also responsible for the retrieval process (case retrieval).

As the user decides to solve a problem using the assistance of a CBR system, a query problem is presented. The retrieval process then is used to select from the system's case-base only those cases for which the current problem has been previously modelled (e.g. through the use of an indexing system). For each of these past cases, the retrieval mechanism then computes similarity ratings between particular attributes (e.g. by using a similarity measure). The user is then presented with a list of past cases graded with respect to the design description as well as the problem definition. This preliminary list of retrieved cases can be further manipulated by an Adaptation module (e.g. human expertise) which is responsible for (i) the final selection of the best retrieved case and (ii) its adaptation to meet required constraints.

2.1. Case-based module

The description of what constitutes a case and how cases are retrieved is presented in this section.

A "case" in CBR represents a previous experience or problem [1] and it is composed of a *problem description*, a *problem solution*, and an *outcome*, which is a measure of success of the applied solution in real conditions. A *PCA case* can be represented as a set of attributes that correspond to (i) PCA design and specification features (e.g. materials, processing methods) (ii) the reliability of the product and (iii) its associated defects (see Figure 1).

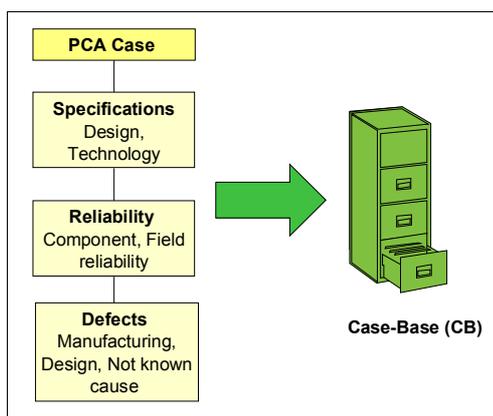


Figure 1. A Printed Circuit Assembly Case
A Printed Circuit Assembly Case (PCA) case is composed of a problem description which specifies the type of board design, its expected reliability and the solution description which specifies the defects and problems experienced during its manufacturing.

Choosing the appropriate cases for a CB library is crucial for the use of CBR systems. This is mainly due to the (i) the vast amount of attributes that can be included when representing a case which influence search time and (ii) how well distributed the solution space is. In addition, in a CB, cases might share common attributes, so there may be overlap between them (Figure 2). In particular for the manufacturing of PCA the attributes that can be included to represent a *case* are endless.

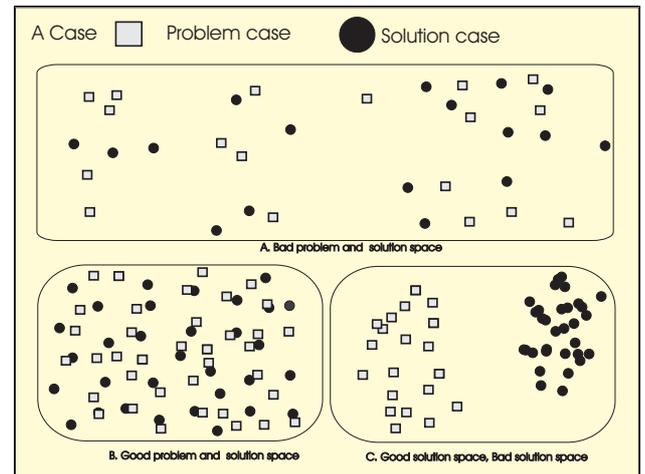


Figure 2. Selection of a case-base. When selecting a case-base, the problem space should be composed of cases that map continually the domain of cluster they belong to. Ideally, it is preferred a selection of cases or set that belong to a specific group while maintaining a degree of mixing, such as in B or less ideally C. Set A (top) is not a good representation of a problem or solution space because cases are too disperse.

2.2. Case representation

The *problem description* part of a case for the PCA domain consisted of a set of design specification features. A simple design specification set consisted of those attributes that are taken into account when a new sale order is produced. For example, these attributes may include the types of component to be placed in the board, soldering methods, non-standard processes (e.g. lead-free components, over-sized connectors, manual fitting of special components), and reliability factors associated with the board.

The *solution description* part of a case is the set of the defects observed and recorded with the solution elements which can also include the cost associated in repairing such defect. Defects during manufacturing may include (e.g. dry joint, excess solder, insufficient solder, poor wetting, disturbed joint, misplaced, tombstone, orientation, unsoldered, missing, lifted leg, shorts, solder balls, spikes, cracked joint, insulation in joints, etc).

Most of the attributes that describe the problem and solution part of a PCA case are quantitative and can be represented as in a *spreadsheet format* [17-20] i.e. cases represented as records, (or rows), and each attribute variable is represented in an associated column. This format is also referred to as the *attribute-value* representation or *feature-valued vector*. In most cases, attribute names are omitted as

the context unambiguously determines which entry corresponds to which attribute (see Figure 3 and Table I).

Case	Feature vector				
	1	2	k	...	p - attributes
(c)					
1					
2					
i	c_1^i	c_2^i	c_k^i	...	c_p^i
...					
N	c_1^N	c_2^N	c_k^N	...	c_p^N

Figure 3. Feature vector representation of cases.

A case is composed of p -attributes where each k -attribute corresponds to either a problem or solution attribute. N is the total number of cases in the case-base.

For a particular case i the feature case vector \mathbf{c} is:

$$\mathbf{c}^{(i)} = \{c_1^i, c_2^i, \dots, c_k^i\} \quad (1)$$

The set of all $\mathbf{c}^{(i)}$'s constitutes the case-base (CB):

$$CB = \{\mathbf{c}^{(i)}, \dots, \mathbf{c}^{(N)}\} \quad (2)$$

Where a case \mathbf{c} is composed of p -attributes where each k -attribute corresponds to either a problem or solution attribute. N is the total number of cases in the case-base.

Table I. A simplified PCA case

Case s	Input Specification Design Attributes (numerical only)				Output Defects			
	Case Problem				Case Solution			
C_i	C_p 1	C_p 2	C_p 3	C_p k	C_s 1	C_s 2	C_s 3	C_s k
C_1	19	11	35	7.45	A1	A2	A3	A4
C_2	19	12	35	7.2	A5	A6	A7	A8
C_3	27	16	27	0.75	A9	A2	A1 0	A8
C_4	27	10	27	0.5	A11	A6	A1 2	A9
...								
C_{24}	27	15	35	0.5	A11	A2	A3	A9

Where

C_p : Case problem k^{th} attribute

C_s : Case solution k^{th} attribute

A1	Missing component	A3	Shorts
A2	Lifted Leg	A4	Incorrect
A7	Copper Tops	A9	Lead Lengths
A8	Preforming	A10	Insulation in Joints
A5	Damage		
A6	Flow Through		
A11	Blow Holes		
A12	Adhesive		

2.3. Retrieval Component

In a CBR system, case retrieval is the process of obtaining from the case-base library those stored cases that are most similar to a query case [1]. This process is illustrated in Figure 4. It can be observed from this figure that several subtasks have to be fulfilled for retrieval. First, an efficient search in the case library has to be performed to find the appropriate case(s). Second, the problem of matching or similarity-assessment is the recognition that a case(s) is applicable to a new situation. Third, ranking heuristics are used to choose those cases partly matching a query that can best address the reasoner's purpose.

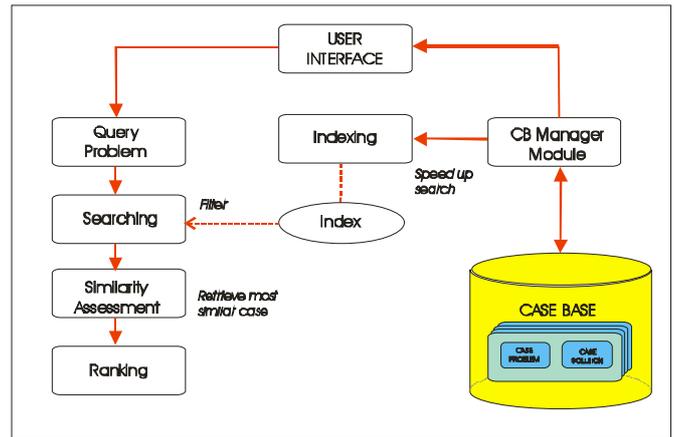


Figure 4. The process of retrieving cases from the case-base

Similarity assessment is performed by using a similarity measure. Similarity measures were developed in the area of cluster analysis where clustering is usually viewed as a process of partitioning data into groups of "similar" objects. In considering retrieval of PCA cases, it is important, at the outset, to recognise that similarity assessment depends greatly on three features: (i) the point of view (or context) the user chooses to compare two cases, (ii) the retrieval algorithm used, which in turn depends upon (iii) the representation format of the cases.

Only a fraction of the distance measures available in the literature are presented in Table II. For the purposes of this paper, only distance functions that handle numerical attributes are described. This is because the features that normally describe a PCA case available for this study are numerical. Some comparison studies exist among similarity measures that use heterogeneous (qualitative and quantitative) attributes [18, 19, 21]. Results obtained in these studies suggest that the performance of a similarity measure depends upon the type of attribute constituting the case and on the relative importance of each attribute (i.e. *attribute weight*).

3. Results

PCA cases come from a subcontractor company that makes avionic equipment for aerospace contractors. The company produce products in lots (or *batches*) in an intermittent and complex way. The process is very difficult to control and is characterised by high levels of work-in-progress inventory. Their equipment is laid out in a functional and flexible way (i.e. is arranged by process of function

performed) which is advantageous for them meeting the needs of batch production of many different boards. This layout tends to be “*process-focused*” since it is not optimised for the products; hence the process is more flexible but less efficient in comparison to “*product-focused*” layouts such as in the production of high volume electronics.

Table II. Selected quantitative distance measures.

Adapted from [22], [23].

x_{ik} denotes the value that the k^{th} quantitative variable takes for the i^{th} object ($i=1, \dots, n; k=1, \dots, p$) and w_k ($k=1, \dots, p$) are non-negative weights associated with the variables. r_k denotes the range of values for the variable k .

Function Name	Equation
Euclidean	$d_{ij} \equiv \left(\sum_{k=1}^p w_k^2 (x_{ik} - x_{jk})^2 \right)^{1/2} \quad (3)$
City	$d_{ij} \equiv \sum_{k=1}^p w_k x_{ik} - x_{jk} \quad (4)$
Canberra	$d_{ij} \equiv \sum_{k=1}^p \frac{ x_{ik} - x_{jk} }{ x_{ik} + x_{jk} } \quad (5)$
Chebyshev	$d_{ij} \equiv \max_{k=1}^p x_{ik} - x_{jk} \quad (6)$
Taxonomic	$d_{ij} \equiv \left(\sum_{k=1}^p w_k^2 \frac{(x_{ik} - x_{jk})^2}{r_k} \right)^{1/2} \quad (7)$
Divergence	$d_{ij} \equiv \left(\sum_{k=1}^p w_k^2 \frac{(x_{ik} - x_{jk})^2}{(x_{ik} + x_{jk})^2} \right)^{1/2} \quad (8)$
D7	$d_{ij} \equiv \sum_{k=1}^p w_k \frac{ x_{ik} - x_{jk} }{ x_{ik} + x_{jk} } \quad (9)$
D8	$d_{ij} \equiv \sum_{k=1}^p w_k \frac{ x_{ik} - x_{jk} }{(x_{ik} + x_{jk})} \quad (10)$

Using each case as a query, the nearest neighbour rule was implemented using the Euclidean distance eq. (3). The pseudo-code for the local-based similarity algorithm is presented in Figure 5.

Distance functions were calculated using two sets of feature weights. One set in which all features are equally important (i.e. $k \{w_k=1\}$), and the second set using feature weights recommended by a PCA design expert. In some circumstances, knowledge about the relevance of the attributes describing the cases in a given context is necessary for retrieval. This might be for instance, the case when the user

wants to give relevance to a certain feature(s) over others in retrieval.

```

for each feature k of the query
  Find the corresponding feature in the stored case

  Compare the two values to each other and compute
  the degree of match using different distance functions

for k = 1 to attribute p (k = 1 ... p)
  'Square Euclidean distance' = distance + [Ck - Qk]2
end

Retrieve most similar case to query (i.e. calculate
minimum distance case-query)
for j = 1 to all cases
  min = minimum(distance)
end
end

```

Figure 5. Retrieval algorithm using standard distance functions.

A user-query was first presented to the CBR system, and this retrieves the closest PCA designs to it according to some measure of similarity e.g. Euclidean distance (eq. 3). Preliminary studies indicated that the Euclidean, City-Block, Canberra, Taxonomic and Divergence distance measures have the same tendency i.e. they retrieve similar cases. Therefore, for the 24 cases available for this study, only the Euclidean distance is presented (see Table III).

Table III. PCA Retrieved cases using the Euclidean distance

Sample	Euclidean distance
	case distance
11	24 0.447
	27 0.513
12	13 0.142
	20 0.189
22	18 0.184
	17 0.195
25	17 0.119
	15 0.162

4. Conclusions

Case-based reasoning is a methodology of problem solving that has shown the potential of providing a very effective framework in assisting designers in selecting, applying and adapting PCA designs using historical data and the lesson they have learned through time. The framework is currently being implemented in a contractor’s manufacturing company but is aimed at any electronics manufacturing company seeks to at reduce defects opportunity by using past knowledge that highlights potential problem in processing and design that have an impact on the reliability of the product.

Given a problem description or user query, the system retrieves similar experiences with the ultimate objective of predicting (i) what defects can happen based on that similar designs lead to similar defects and (ii) what *manufacturing conditions* increase yield. The implementation of the CBR system in an industrial setting has the potential of reducing the time and hence the cost of the product development process while at the same time enables carrying out simulation studies. It also serves as an “institutional memory” that supports problem solving.

At its present state of development the described CBR system relies on a relatively limited number of cases. However, it can be readily extended to include new cases as they become available and it is expected that such extension will enhance the usefulness of the presented CBR system. Additionally, the amount of information used to describe a case can be increased and other similarity measures can be used accordingly.

5. Future work

Case-based reasoning integrates many different AI techniques and paradigms at various stages in the cycle such as retrieval and adaptation. By reflecting the way a modelling expert revises his knowledge when dealing with new circumstances, different types of AI techniques may be combined within the CBR framework to match the characteristics of the problem domain and comply with the principles and assumptions of PCA design (e.g. rule-based adaptation, fuzzy retrieval). Currently, the authors are carrying out research into the application of several of these AI techniques. One of them is the use of rules capable of linking the defect opportunity with manufacturing and design attributes. The incorporation of a rule-based system (RBS) into the adaptation process will improve drastically the performance of the design process by enabling it to conclude what rules at both manufacturing and design level are triggered when a defect occurs.

For many real-world applications the variables describing a set of objects are of different types. For instance, in most of the PCA design graphical data is represented by CAD data. This type of representation requires a different retrieval approach (i.e. the assessment of similarity between cases containing graphical data) compared to when numerical attributes are used. Heterogeneous distance functions that can handle mixed variables are also being evaluated as well as algorithms that can compare graphical information.

Acknowledgments

The authors wish to express their gratitude to the industrial collaborators. This work was supported financially by the Innovative Electronics Manufacturing Research Centre (IeMRC).

References

1. Kolodner, J.L., *Case-based reasoning*. 1993, San Mateo, CA: Morgan Kaufmann Publishers.
2. Farinacci, M.L., et al., *The Development of ALADIN (Aluminum Alloy Design Inventor), an Expert System for Aluminum Alloy Design. Interim Report*. 1986.
3. Hulthage, I., et al., The Architecture of ALADIN, A Knowledge-Based Approach to Alloy Design. *IEEE Expert*, 1990, 5 (4): p. 56-73.
4. Iwata, Y. and N. Obama. QDES: Quality-design expert system for steel products. In Proceedings of the 3rd Innovative. in *The Third Conference on Innovative Applications of Artificial Intelligence (IAAI-91)*. 1991.
5. Craw, S., N. Wiratunga, and R. Rowe, Case-Based Design for Tablet Formulation. *Lecture notes in computer science*, 1998(1488): p. 358.
6. Boswell, R., S. Craw, and R. Rowe. Refinement of a Product Formulation Expert System. in *ECAI-96 Workshop on Validation, Verification and Refinement of KBS*. 1996. Budapest, Hungary.
7. Suh, M.S., et al., A case-based expert system approach for quality design. *Expert Systems with Applications*, 1998, 15(2): p. 181-190.
8. Bandini, S. and S. Manzoni, *Knowledge Management Applications at Pirelli Tires: from P-Race to P-Truck projects. Theme 7: Case Study - Knowledge Management Applications at Pirelli Tires*, in *KnowledgeBoard*. 2003, Phillip Sainter.
9. Malek, M., et al. An Operator Support System Based on Case-Based Reasoning for the Plastic Moulding Injection Process. in *European Workshop in Case-Based Reasoning, EWCBR98*. September 1998. Dublin, Ireland: Springer, Berlin.
10. Cunningham, P., B. Smith, and A. Bonzano, An Incremental Retrieval Mechanism for Case-Based Electronic Fault Diagnosis. *Journal of Knowledge Based Systems*, 2002.
11. Vollrath, I. Reuse of Complex Electronic Designs. Requirements Analysis for a CBR Application. in *Proceedings of the Fourth European Workshop on Case-Based Reasoning*. 1998. Berlin: Springer Verlag.
12. Tsai, C.-Y., C.-C. Chiu, and J.-S. Chen, A Case-based reasoning system for PCB defect prediction. *Expert Systems with Applications*, 2005, 28: p. 813-822.
13. Slade, S., *Case-Based Reasoning: A Research Paradigm*, in *AI Magazine*. 1991. p. 42-55.
14. Ketler, K., Case-based reasoning: An introduction. *Expert Systems with Applications*, 1993, 6(1): p. 3-8.
15. Leake, D.B., *CBR in Context: The Present and Future, in Case-Based Reasoning. Experiences, Lessons & Future Directions*, D.B. Leake, Editor. 1996, AAAI Press/ MIT Press: Cambridge, MA. p. 3-30.
16. Pal, S.K. and S.C.K. Shiu, Case-Based Reasoning: Concepts, Features and Soft Computing. *Applied Intelligence*, 2004, 21(3): p. 233-238.
17. Núñez, H., et al. Classifying Environmental System Situations by means of Case-Based Reasoning: a Comparative Study. in *iEMSS 2002. Integrated Assessment and Decision Support*. 24-27 June 2002. Lugano, Switzerland.
18. Wilson, D.R. and T.R. Martinez, Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, 1997, 6: p. 1-34.

19. Liao, T.W., Z. Zhang, and C. Mount, Similarity Measures for Retrieval in Case-Based Reasoning Systems. *Applied Artificial Intelligence*, 1998, 12(4): p. 267-288.
20. Gower, J.C., A General Coefficient of Similarity and Some of its Properties. *Biometrics*, 1971, 27: p. 857-874.
21. Núñez, H., et al., A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations. *Environmental Modelling & Software*, 2004, 19(9): p. 809-819.
22. Michalski, R., R.E. Stepp, and E. Diday, *A Recent Advance In Data Analysis: Clustering Objects Into Classes Characterised By Conjunctive Concepts*, in Progress in Pattern Recognition, L.N. Kanal and A.R. Kanal, Editors. 1981, Amsterdam; New York, Oxford: North-Holland Pub. Co.
23. Gower, J.C., *Measures of similarity, dissimilarity and distance*, in Encyclopedia of Statistical Sciences, S. Kotz, N.L. Johnson, and C.B. Read, Editors. 1985, John Wiley & Sons: New-York. p. 397-405.