# Diagnosis in systems based on an informed tree search strategy : application to cartographic generalisation

Patrick Taillandier

# Diagnosis in systems based on an informed tree search strategy: application to cartographic generalisation

Patrick Taillandier
COGIT IGN
2/4 avenue Pasteur
94165 Saint-Mandé Cedex - France
Patrick.taillandier@gmail.com

## ABSTRACT

Many real world problems can be expressed as optimisation problems. Solving such problems means to find, among all possible solutions, the one that maximises an evaluation function. One approach to solve it is to use an informed search strategy. The principle of this kind of strategy is to use problem-specific knowledge beyond the definition of the problem itself to find solutions more efficiently than with an uninformed strategy. This strategy demands to define problem-specific knowledge (heuristics). The efficiency and the effectiveness of systems based on such strategies directly depend on the utilised knowledge quality. Unfortunately, acquiring and maintaining such knowledge can be fastidious. The objective of the work presented in this paper is to propose an automatic knowledge quality diagnosis approach for systems based on an informed tree search strategy. Our approach consists in analysing the system's execution logs and in using multi-criteria decision making techniques in order to determine if the knowledge needs to be revised. We present an experiment we carried out in an industrial application domain where informed search strategies are often used: cartographic generalisation.

## Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control methods and Search – *Graph and tree search strategies, Heuristic methods*

## General Terms

Algorithms, Experimentation, Theory.

## Keywords

Knowledge quality diagnosis, Multi-criteria decision making, Problem Solving, Informed Tree Search Strategy, Cartographic Generalisation.

## 1. INTRODUCTION

Problem-solving is one of the central topics of artificial intelligence. Among solving approaches, some are based on an informed search strategy. The principle of this kind of strategy is to use problem-specific knowledge (heuristics) beyond the definition of the problem itself to find solutions more efficiently than with an uninformed strategy. The efficiency of systems based on such strategies directly depends on the utilised knowledge quality. Unfortunately, it is usually very difficult to acquire expert knowledge. Eward Feigenbaum formulated this problem in 1977 as the knowledge acquisition bottleneck problem [4]. Indeed, the expert knowledge is rarely formalised and its translation into a formalism usable by computers is very complex. An example of domain where such difficulties were already detected [16] and which is of particular interest to us, is cartographic generalisation. The acquisition problem has for consequence that most of time knowledge is good but not perfect. Another problem concerns the evolution of the knowledge when integrating new elements (e.g. new actions) in the system. Thus, it will be sometimes necessary to revise the knowledge. In this paper, the problem that interests us is when triggering the revision process, i.e. how to diagnose that the knowledge needs to be revised. To face this problem, we propose an approach of automatic diagnosis based on the analysis of the execution logs.

In section 2, we introduce the general context in which our work takes place and the difficulties we must face. Section 3 is devoted to the presentation of our approach. Section 4 describes an application of our approach to cartographic generalisation. In this context, we present a real case study that we carried out as well as its results. Section 5 concludes and presents the perspectives for this work.

## 2. CONTEXT

### 2.1 Description of the considered optimisation problems

In this paper, we are interested in a family of optimisation problems, which consists in finding, by action application, the state of an entity that maximises an evaluation function.
Let $P$ be an optimisation problem that is characterised by:

- An entity class $E_P$
- $\{action\}_P$: a set of actions that can be applied on an entity belonging to $E_P$. The result of the application of an action is supposed non-predicable.
- $Q_P$: a function that defines the state quality of an entity belonging to $E_P$

An instance $p$ of $P$ is defined by an entity $e_p$ of class $E_P$, which is characterised by its initial state. Solving p consists in finding the state $s$ of $e_p$ that optimises $Q_P$, by applying actions from $\{action\}_P$ to the initial state of $e_p$.


*CSTST 2008*, October 27-31, 2008, Cergy-Pontoise, France.

Let us consider the following example: Let *Probot* be an optimisation problem where a robot, considering its initial position in a maze, seeks to find the exit.

- $E_{Probot}$: a kind of robot. A robot of the kind $E_P$ is characterised by its initial position in the maze.
- $\{action\}_{Probot}$: *{move forward, turn left, turn right}*
- $Q_{Probot}$: distance separating the robot from the exit of a maze

An instance of *Probot* is: Let $e_{probot}$ be a robot of the kind $E_{Probot}$ with an initial position in the maze. Its goal is to find the exit or at least to reach the closest possible position to the exit.

There are many ways to solve such optimisation problems of this kind. In this paper, we are interested in systems that solve optimisation problems by exploring a state tree by means of an informed strategy. Such systems are often used for real world problems because of their efficiency. In section 2.2, we present the generic system for which our revision approach is dedicated. Our diagnosis approach could be used for other kinds of systems with some adaptations.

## 2.2 Description of the considered systems

We propose a generic system that solved optimisation problems such as the ones presented section 2.1. The system is based on informed depth-first exploration of state trees. The passage from a state to another corresponds to the application of an action. Figure 1 presents the action cycle.

It begins with the characterisation of the current state of the entity and its evaluation using the function $Q_P$. Then, the system tests if the current state is good enough or if it is necessary to continue the exploration of others states. If the system decides to continue the exploration, it tests if the current state is valid or not. If not, the entity backtracks to its previous state, otherwise, the system constructs a list of actions to apply. If the action list is empty the entity backtracks to its previous state, otherwise the system chooses the best action, and applies it. Then it goes back to the first step. The action cycle ends when the stopping criterion is checked or when all actions have been applied for all valid states.

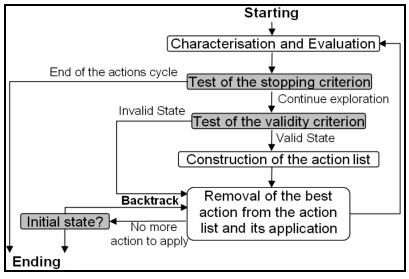



**Figure 1.Action cycle**

This generic system uses three kinds of procedural knowledge:

- *Action application knowledge* determines, for a current state, which will be the action list, i.e. the actions proposed for the state and their application order.
- *Validity criterion* determines, according to all previously visited states, if the current state is valid or not.
- *Ending cycle criterion* determines, according to all previously visited states, if the system action cycle has to continue the exploration or not.

## 2.3 Difficulties of the knowledge quality diagnosis

The knowledge quality diagnosis problem consists in determining if the knowledge needs to be revised, and if so, which part of them seems to be good and which part seems to be defective.

A first difficulty of the knowledge quality diagnosis concerns the dependency existing between the different pieces of knowledge: it is sometimes not possible to determine if a piece of knowledge is really defective or if it is another piece of knowledge that is defective and which influences the application results of the first piece of knowledge. Figure 2 shows an example of knowledge dependency: we solved the same problem instance with two different validity criteria:

- $Crit_1$ allows state quality deterioration. The left tree was built with this criterion.
- $Crit_2$ does not allow state quality deterioration. The right tree was built with this criterion.

For the left tree ($Crit_1$), the best action to apply at *state 1*, is *Act 1*, whereas for the right tree ($Crit_2$), the best action at *state 1*, becomes *Act 2*.

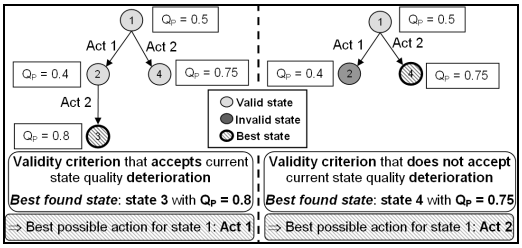



**Figure 2.Knowledge dependency problems**

Another difficulty concerns the kind of information that can be extracted from the study of a state tree. For ending cycle criterion and validity criterion, it is possible to extract information concerning the false positive errors (when the action cycle should not have continued the exploration; when a state should have not been valid) but not concerning the false negative errors (when the action cycle should have continued the exploration; when a state should have been valid). Indeed, it is not possible to know if it would have been possible to find a better state if the ending cycle criterion continued the exploration or if a state was valid.

A last difficulty concerns the resolved problem instances that it is possible to analysis to diagnose the knowledge quality. These problem instances are determined by the user utilisation of the system and not according to the knowledge quality diagnosis needs. If just few instances are taken into account for the diagnosis process, there are chances that these instances are not representative of the whole possible instance set of the problem and then are not reliable to be used for the knowledge quality diagnosis.

# 3. PROPOSED APPROACH

## 3.1 General approach

Our goal is to diagnose automatically the knowledge quality of a system working by informed tree search exploration. Our

diagnosis approach, presented in figure 3, is based on the analysis of the execution logs.

Each time an optimisation problem instance is solved, the diagnosis module analyses, during an *evaluation phase*, the successes and the failures of each piece of knowledge. Then, it tests if the number of problem instances resolved since the last decision making (*Nb_instances*), is high enough to trigger a new decision making process. This test allows facing partly the last difficulty presented section 2.3. In fact, making a decision on the knowledge quality, only when enough instances are solved, allows avoiding particular cases which could lead to make a wrong decision. The number of resolved problem instances needed to trigger the decision making process (NB_INSTANCES_MIN) depends of the optimisation problem, of the system used to solve it and of the user needs. The higher NB_INSTANCES_MIN is, the more reliable the decision making process is, but the time necessary to solve a problem instance and the number of available instances can compel the user to choose a low value for it.

If the number of instances is high enough, the diagnostic module triggers a d*ecision making phase* which consists in making a decision concerning the need of revising the knowledge and in giving, if necessary, a precise analysis of the pieces of knowledge that need to be revised in priority.

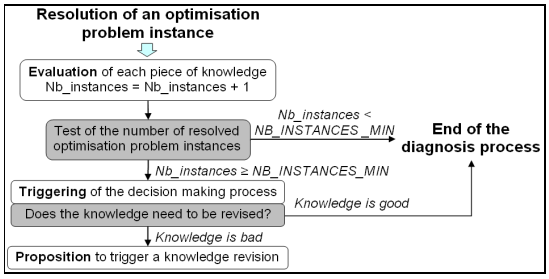



**Figure 3.General diagnosis approach**

## 3.2 Evaluation phase

Each time an optimisation problem is solved (and a state tree is built), two kinds of information are extracted from the state tree analysis: local information concerning each piece of evaluated knowledge and a global evaluation concerning the effectiveness of the system.

### *3.2.1 Local Evaluation*

The evaluation of each piece of knowledge is expressed by four values: the number of false negatives ($nb_{FN}$), the number of false positive ($nb_{FP}$), the number of true positives ($nb_{TP}$) and the number of true negatives ($nb_{TN}$). The computation of these values depends of the nature of the concerned piece of knowledge. As mentioned in section 2.3, for some pieces of knowledge, the number of false negatives is not relevant.

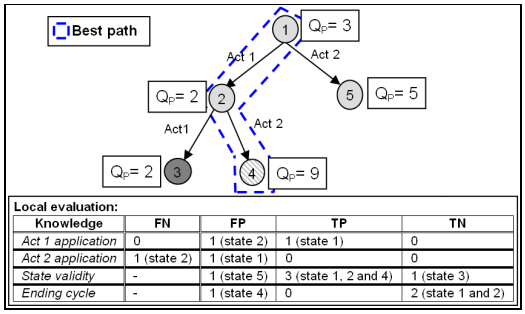
We define the notion of best path. A best path is a sequence of at least two states, which has the root of a tree (or of a sub-tree) for initial state and the best state of this tree (or sub-tree) for final state. The study of the success and failure of the different pieces of knowledge is performed by best path analysis.

For the validity criterion, a false positive is a state which does not belong to a best state, whereas its predecessor belongs to it. A true positive is a valid state which belongs to the best state. A true negative is an invalid state which does not belongs to a best state whereas its predecessor belongs to it.

Concerning the ending cycle criterion, a false positive is a case where the criterion proposed to continue the exploration whereas the best state of the tree was already found. A true positive is a state which belongs to a best state and is not the best state. A true negative is a case where the criterion proposed not to continue the exploration just after visiting the best state of the tree.

For action application knowledge, a false positive is a case, where, from a state belonging to a best path, the application of the action led to a state that does not belong to it. A false negative is a case where, from a state belonging to the best path, the application of the action led to another state of the best path but where the action was not applied in priority. A true positive is a case where, from a state belonging to the best path, the action was applied in priority and led to another state of it. At last, a true negative is a case where from a state belonging to a best path, the action was not proposed.

Figure 4 gives an example of results for the local analysis for a state tree.




**Local evaluation:**

| Knowledge | FN | FP | TP | TN |
|---|---|---|---|---|
| *Act 1 application* | 0 | 1 (state 2) | 1 (state 1) | 0 |
| *Act 2 application* | 1 (state 2) | 1 (state 1) | 0 | 0 |
| *State validity* | - | 1 (state 5) | 3 (state 1, 2 and 4) | 1 (state 3) |
| *Ending cycle* | - | 1 (state 4) | 0 | 2 (state 1 and 2) |

**Figure 4.Example of results for a local analysis**

### *3.2.2 Global Evaluation*

As mentioned in section 2.3, if it is possible to determine the quantity of useless states, it is not possible to know if it would have been possible to find a better state if more states were visited. The local analysis of each piece of knowledge can then carry information on the efficiency of the system (its performance in terms of execution time), it can not bring information on its effectiveness (its performance in terms of best state found quality). Thus it is important to have a new criterion that allows determining if the effectiveness of the system is good enough. We proposed to memorise the quality values, $Q_P$(best state(p)), of each resolved problem instance *p*.

## 3.3 Decision making phase

The decision making stage consists in determining if the knowledge needs to be revised or not. The decision has to be made according to the different criteria: the quality of the different pieces of knowledge and the performance of the system.

### *3.3.1 Decision criteria*

In order to make the decision, we proposed several criteria.

The first one concerns the quality of the different pieces of knowledge. The knowledge quality is represented by a *mark* which is defined between 0 and 1. A *mark* of 0 means that the piece of knowledge seems to be very defective and a mark of 1

that the piece of knowledge seems to be perfect. The *mark* for a piece of knowledge *K* and for a resolved problem instance sample $P_n$, depends of the results obtained for each instance of $P_n$ during the evaluation phase (section 3.2.1):

$$Mark(K, P_n) = \frac{nb_{TP} + nb_{TN}}{nb_{TP} + nb_{TN} + nb_{FP} + nb_{FN}}$$

Another criterion is the system effectiveness. Our approach demands to define a function *Effectiveness*($P_n$), which gives a mark to the system effectiveness for the resolution of a set of problem instances $P_n$. The mark depends of the quality values obtained for each instance of $P_n$ (see section 3.2.2). This mark is a floating point value between 0 and 1. An example of the *Effectiveness*($P_n$) function is presented in section 4.3.

The mark given to the different values will give indication to the user on the knowledge that needs to be revised. If it is the effectiveness mark which is low, the user would have to define knowledge allowing to visit more states (by proposing more actions by state for example).

### 3.3.2 Multi-criteria decision making

In the literature, numerous multi-criteria decision making approaches were proposed.

Among them, several approaches aim at aggregating all criteria in a single criterion (utility function) which is then used to make the decision [7, 8]. Another approach consists in comparing the different possible decisions per pair by the mean of outranking relations [10, 14, 17]. A last approach, which is highly interactive, consists in devising a preliminary solution and in comparing it with other possible solutions to determine the best one [2, 6].

Our decision making problems consists in choosing, between a set of two categories {"*revise the knowledge*", "*do not revise the knowledge*"}, the one to assign to the current situation. We are then in the case of a *sorting* problem. As pointed out in [5], the ELECTRE TRI method [17] is particularly relevant for our problem. Indeed, we dispose of more than three criteria which are very heterogeneous (it is difficult to directly compare the effectiveness criterion to the mark of a piece of knowledge). Moreover, we do not want the loss for a given criterion to be compensated by the gain of another. At last, we want to integrate in our decision making the fact that, for certain criteria, a small value difference is not significant, while the accumulation of several small differences may become significant.

### 3.3.3 Application of ELECTRE TRI method

We propose to use the ELECTRE TRI method to solve our decision making problem. In this method, a situation is called an *action*. The principle of the decision making is to compare the *current action* to a *reference action*, which translates the limit between the category, "*revise the knowledge*" and the category, "*do not revise the knowledge*". The definition of the *reference action* depends of the knowledge concerned by the diagnosis. We proposed in section 4.3, a value for it.

The ELECTRE TRI method demands to define different parameters for each criterion:

- The *weight* of the criterion.
- The *preference threshold*: It translates the threshold from which the difference of two of the criterion values allows to prefer one action over another.
- The *indifference threshold*: It translates the threshold from which the difference of two of the criterion values is considered significant.
- The veto *threshold*: It translates the threshold from which the difference of two of the criterion values disqualifies the action with the smaller value.

A last parameter to define is the *λ-cutting level* of the fuzzy relation. It defines the reference threshold for the action comparison.

The first step of the ELECTRI TRI method consists in computing the concordance, $c_j(aSb)$, and the discordance, $d_j(aSb)$, of each criterion $j \in \{\text{criterion}\}$. We noted $value(a_j)$, the value of the criterion $j$ for the action $a$. Figure 5 illustrates how to compute these values.

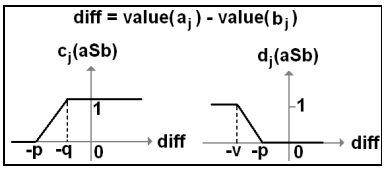



**Figure 5.Concordance and discordance**

The second step consists in computing the *concordance index C(aSb)*:

$$C(aSb) = \frac{\sum_{j \in \{criterion\}} w_j \times c_j(aSb)}{\sum_{j \in \{criterion\}} w_j}$$

Then, it is possible to compute the credibility index $\rho(aSb)$:

$$\rho(aSb) = C(aSb) \times \prod_{j \in \{i \in \{criterion\} / d_i(aSb) > C(aSb)\}} \frac{1 - d_j(aSb)}{1 - c(aSb)}$$

The credibility index is used to establish the relation between the reference action *Ref* and the current action *A* (figure 6).

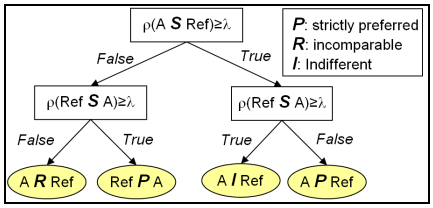



**Figure 6.Conclusion for an action**

There are two procedures to affect a category to an action:

- The *optimistic* procedure: if *Ref* **P** *A*, then *A* is in category "*revise the knowledge*", otherwise *A* is in category "do not *revise the knowledge*".
- The *pessimistic* procedure: if *A* **P** *Ref*, then *A* is in category "do not *revise the knowledge*", otherwise *A* is in category "*revise the knowledge*".

The using of the ELECTRE TRI method implies to defined numerous parameters, and finally to choose between two possible procedures. An important point of the decision making is the robustness of the conclusion [11]. An approach to test the robustness of the conclusion consists in analysing its variation when the different parameters vary (and the procedure). For our application, we propose to apply the ELECTRE TRI method with

several sets of parameters and to proceed by majority vote to determine the final conclusion. The number of vote for each category gives an idea of the robustness of the conclusion. A high percentage of votes (for example 95%) for the category "*revise the knowledge*" means that it is very important to revise the knowledge whereas a low percentage (for example 55%) means it is less urgent. The choice whether to revise the knowledge also depends on the system user constraints. Indeed, if the user has time to trigger a knowledge revision process, he could trigger it even if the percentage of votes for the category "*revise the knowledge*" is close to 50%. On the contrary, for the same percentage of votes, a user that has time constraints could decide not to revise the knowledge.

We note *{K}* the set of knowledge criterion, *{weight(C)}*, the set of weights to test for the criterion *C*, *{Preference(C)}*, the set of *preference thresholds* to test for *C*, *{Indifference(C)}*, the set of *indifference thresholds* to test for *C* and *{veto(C)}* the set of *veto thresholds* to test for *C*. We note *{λ}*, the set of *λ-cutting level* to test. The number of tested parameter sets will be:

$$nb = 2 \times card(\{\lambda\}) \times card(\{W(Ef)\}) \times card(\{P(Ef)\}) \times card(\{I(Ef)\}) \times card(\{V(Ef)\}) \times \prod_{K \in \{K\}} card(\{W(K)\}) \times card(\{P(K)\}) \times card(\{I(K)\}) \times card(\{V(K)\})$$

This number can be very important. It is then important to choose for each parameter set, a reasonable number of values (between 1 and 3).

# 4. APPLICATION TO CARTOGRAPHIC GENERALISATION

## 4.1 Automatic cartographic generalisation

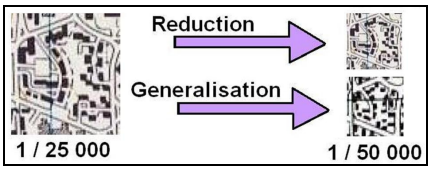



**Figure 7.Cartographic Generalisation**

Cartographic generalisation is a process that aims at decreasing the level of details of geographic data in order to produce a map at a given scale. Figure 7 gives an example of cartographic generalisation. In the figure, cartographic generalisation is not a simple size reduction. The application of numerous operations such as local scaling, displacements or elimination of objects are needed in order to ensure the readability of the map while keeping the essential information of the initial map. The automation of this process is a complex problem, which has been the core of numerous research works in the recent years. Some of these works try to solve it by local, step-by-step and knowledge-based approaches [3]. The difficulty then consists in choosing the best sequence of generalisation operations to apply to the various geographic objects. An approach to solve this problem is to use an informed search. Nowadays, the procedural knowledge used to guide the search is entered "by hand" by generalisation experts. Its tuning is often long and fastidious. Thus, it is interesting to integrate in the system, a module able to diagnose the knowledge quality on-line, and able to point out the pieces of knowledge which pose difficulties.

The automation of cartographic generalisation is a particularly interesting industrial application context. First, it is a problem which is far from being solved. Second, it directly concerns many mapping agencies that wish to improve their map production lines. Finally, the problem of on-demand mapping that takes growing place with the multiplication of the possibilities to create one's own map on the web.

## 4.2 The generalisation system

The generalisation system that we use for our experiment is based on the AGENT model [1, 13] and follows the specification that we defined in section 2.2.2. It generalises a geographic object or a group of geographic objects by the mean of an informed tree search strategy. Each state represents the geometric state of the considered geographic objects and is evaluated by a *satisfaction* function, which translates the respect of cartographic constraints by the geographic objects. A cartographic constraint can be, for example, for a building to be big enough to be readable. The satisfaction of a state is ranged between 1 and 10 (10 represents a perfect state and a score lower than 5, a non acceptable state). The action cycle used is the one presented figure 1. Different works proposed off-line approaches to revise automatically the knowledge of this model [12]. Our diagnosis approach completes these approaches by giving a method allowing to trigger the off-line revision process when necessary.

## 4.3 Application of our revision approach

We applied our diagnosis approach to diagnose the knowledge quality of our generalisation system.

We choose for the function *Effectiveness($P_n$)* :

$$Effectiness(P_n) = \frac{1}{10 \times card(P_n)} \sum_{p \in P_n} Satisfaction(p)$$

The function *Satisfaction(p)* returns the best satisfaction found for the generalisation of an object *p*.

We choose as a *reference action* the one defined by:

- For all knowledge criterion: *Mark = 0.5*
- For the effectiveness : Mark = 0.9 (a satisfaction of 9 is considered as a good satisfaction)

We defined different sets of values for the ELECTRE parameters as well:

- For all knowledge criterion: *Weight = {1,2,5}, Preference = {0.1, 0.2, 0.5}, Indifference = {0.01, 0.05, 0.1}, Veto = {1}*
- For the effectiveness criterion: *Weight = {5,10,20}, Preference = {0.03, 0.04, 0.05}, Indifference = {0.01, 0.02, 0.03}, Veto = {0.05, 0.1, 0.2}*
- For the *λ-cutting level*: {0.5, 0.7, 0.9}

The effectiveness criterion is the most important criterion for us. In fact, our priority is to obtain good generalisation results. If this condition is not assured, whatever the knowledge results are, the knowledge has to be revised. The parameter set values were determined empirically by tests on other knowledge bases.

## 4.4 Case study

The real case study that we carried out concerned the generalisation of geographic objects of the kind "building group". The building group generalisation is an interesting case study because it is not yet well managed and because it is very time

consuming. We defined five actions for the building group generalisation as well as four knowledge bases: the first one was defined randomly ($K_{Rand}$), the second was defined by a cartographic expert ($K_{Carto}$), the third one by a AGENT model expert ($K_{Agent}$), and the last one consists in the revised version of $K_{Agent}$ ($K_{Revised}$). $K_{Agent}$ was revised with a methods derived from [15]. $K_{Revised}$ is supposed to be the best knowledge base, then $K_{Agent}$ and $K_{Carto}$ and finally, $K_{Rand}$. We drew randomly 2 sets of 30 building groups that we used to establish the diagnosis: $\{BGS_1, BGS_2\}$.

## 4.5 Results

Table 1 shows that the obtained diagnosis results are consistent. Indeed, the results obtained on the two building group sets are consistent with each other and are consistent with the supposed knowledge base quality. The only knowledge base for which the diagnosis module did not propose to revise is $\boldsymbol{K_{Revised}}$, which is already a revised version of $\boldsymbol{K_{Agent}}$. Concerning the other knowledge bases, the diagnosis proposed to revise all of them, but with different percentages of votes. The percentage of votes represents the reliability of the decision. Thus, for a conclusion "revise the knowledge", the higher the percentage, the more important the knowledge revision is. Concerning the weak percentage of votes for the category "not revise" obtained by $\boldsymbol{K_{Revised}}$, it can be explained by the fact that the measure set used to characterise the geometric state of the building groups is not pertinent enough to define, even after revision, perfect knowledge.

**Table 1. Diagnosis results: category chose and percentage of votes in favour of this category**

| | $BGS_1$ | $BGS_2$ |
|---|---|---|
| $K_{Revised}$ | "not revise" with *51%* | "not revise" with *51%* |
| $K_{Agent}$ | "revise" with *55%* | "revise" with *55%* |
| $K_{Carto}$ | "revise" with *76%* | "revise" with *83%* |
| $K_{Rand}$ | "revise" with *87%* | "revise" with *85%* |

# 5. CONCLUSION

In this paper, we proposed a knowledge quality diagnosis approach based on the use of a multi-criteria decision making method. We validated our approach on a real case study.

A point that deserves more study concerns the elicitations of the parameters values of the multi-criteria decision making method. In fact, the quality of the decision is directly linked to the pertinence of the values chosen for the parameters. Some works deal with the problem of the elicitation of these parameter values [8] by using labeled examples.

In the same way, it will be interesting to integrate, inside the diagnosis module, methods allowing to help with the definition of the effectiveness criteria. In fact, in our experiment, we chose a very simple effectiveness criterion, but a more complex criterion could help to obtain more pertinent conclusions.

# 6. REFERENCES


[1] Barrault, M., Regnauld, N., Duchêne, C., Haire, K., Baejis, C., Demazeau, Y., Hardy, P., Mackaness, W., Ruas, A., and Weibel, R. Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalization. In *ICC*, 2001, 2100–2116.

[2] Benayoun, R., Laritchev, O., de Mongolfier, J., and Tegny, J. Linear programming with multiple objective functions: STEP method (STEM). *Math. Program., 1,* 3, 1971, 366–375.

[3] Brassel, K., and Weibel, R. A review and conceptual framework of automated map generalization. *IJGIS*, 1988.

[4] Feigenbaum, E.A.F. *The art of artificial intelligence 1: Themes and case studies of knowledge engineering*. Technical report STAN-SC-77-621, Stanford University, Department of Computer Science, 1977.

[5] Figueira, J., Mousseau, V., and Roy, B. ELECTRE Methods. *Multiple Criteria Decision Analysis: State of the Art Surveys*, chapter 4, 2005, 134-162.

[6] Geoffrion, A., Dyer, J. and Feinberg, A. An interactive approach for multicriterion optimisation with an application to the operation of an academic department. *Manage. Sci.*, *19*, 4, 1972, 357–368.

[7] Ignizio, J.P. A review of goal programming: a tool for multi objective analysis. *J. Oper. Res. Soc., 29,* 11, 1978.

[8] Jacquet-Lagreze, E., and Siskos, J. Assessing a set of additive utility functions for multicriteria decision making. *The UTA method. Eur. J. Oper. Res., 10,* 2, 1982, 151–164.

[9] Mousseau, V., and Slowinski, R. Inferring an ELECTRE TRI model from assignment examples. *Journal of Global Optimization, 12, 2,* 1998, 157-174.

[10] Roy, B. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision, 31,* 1991.

[11] Roy, B. A missing link in OR-AD: Robustness analysis. *Foundations of Computing and Decision Science, 23,* 3, 1998, 49-73.

[12] Ruas, A., Dyèvre, A., Duchêne, C., and Taillandier, P. Methods for improving and updating the knowledge of a generalization system. In *Autocarto*, 2006.

[13] Ruas, A., and Duchêne, C. A Prototype Generalisation System Based on the Multi-Agent Paradigm. *Generalisation of Geographic Information: Cartographic Modelling and Applications*, 2007.

[14] Siskos, J., Wäscher, G., and Winkels, H. *A bibliography of outranking approaches*. Cahiers du Lamsade, Paris Dauphine, 1983.

[15] Taillandier, P. Automatic Knowledge Revision of a Generalisation System. *Workshop in generalisation and multiple representations*, Moscow, 2007.

[16] Weibel, R., Keller, S., and Reichenbacher, T. Overcoming the Knowledge Acquisition Bottleneck in Map Generalization: the Role of Interactive Systems and Computational Intelligence' knowledge of generalization. In *COSIT*, *13*, 1995.

[17] Yu., W. *Aide multicritère à la décision dans le cadre de la problématique du tri : Concepts, méthodes et applications*. Thèse à Université Paris Dauphine. 1992.