Multi-criteria diagnosis of control knowledge for cartographic generalisation

Patrick Taillandier, Franck Taillandier

Multi-Criteria Diagnosis of Control Knowledge for Cartographic Generalisation

Patrick Taillandier[1, 2], Franck Taillandier[3]

[1] *MTG Lab., UMR-IDEES 6228, 1, rue Thomas Becket, 76821 Mont-Saint-Aignan, France*

[2] *COGIT IGN, 2/4 avenue Pasteur,94165 Saint-Mandé, France*

Phone: + 84 (0) 4 38696017/18/19/20   Fax: + 84 (0) 4 38694509

E-mail address: patrick.taillandier@gmail.com

[3] *Université de Bordeaux 1- I2M (UMR CNRS), 351 cours de la Libération, Bat A11, 33405 Talence cedex, France*

Phone: (33) 05 40 00 6426   Fax: (33) 05 40 00 64 39

E-mail address: franck.taillandier@u-bordeaux1.fr

Abstract— The development of interactive map websites increases the need of efficient automatic cartographic generalisation. The generalisation process, which aims at decreasing the level of details of geographic data in order to produce a map at a given scale, is extremely complex. A classical method for automating the generalisation process consists in using a heuristic tree-search strategy. This type of strategy requires having high quality control knowledge (heuristics) to guide the search for the optimal solution. Unfortunately, this control knowledge is rarely perfect and its evaluation is often difficult. Yet, this evaluation can be very useful to manage knowledge and to determine when to revise it. The objective of our work is to offer an automatic method for evaluating the quality of control knowledge for cartographic generalisation based on a heuristic tree-search strategy. Our diagnosis method consists in analysing the system's execution logs, and in using a multi-criteria analysis method for evaluating the knowledge global quality. We present an industrial application as a case study using this method for building block generalisation and this experiment shows promising results.

(S) Multiple criteria analysis; (S) Knowledge-based systems; Control knowledge quality diagnosis; Heuristic tree-search strategy; Cartographic generalisation

1. Introduction

The cartographic generalisation is a process which aims at decreasing the level of details of geographic data in order to produce a map at a given scale, i.e. to ensure the readability of the map while keeping essential information of the initial detailed data. The cartographic generalisation is not a simple map reduction; it requires applying numerous operations such as object scaling, displacements and eliminations. Figure 1 gives an example of cartographic generalisation.

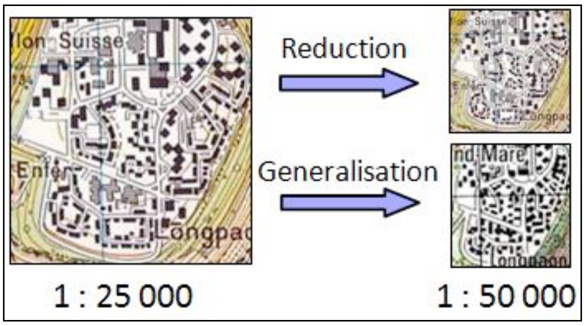


Figure 1. Cartographic Generalisation

Nowadays automating cartographic generalisation process is a key issue. In fact, the multiplication of web sites allowing creating one's own map increases the needs of reliable and effective automatic generalisation processes. Moreover, mapping agencies tend more and more to automate their map production lines in order to limit the production costs. Unfortunately, automating the generalisation process is a complex problem that is far from being solved.

A classical generalisation automation approach consists in using a local, step-by-step and knowledge-based method (Brassel & Weibel, 1988): each vector object (characterized by a series of coordinates that define the object geometry) of the initial dataset is transformed by applying a sequence of generalisation algorithms performing atomic geometric transformations. The algorithmic sequence is not predefined but built on the fly for each object according to control knowledge (heuristics), depending on its characteristics and the expected effects of the algorithms on it. This approach requires to manage a knowledge base. In particular, it requires to adapt knowledge when new elements, such as new generalisation algorithms, are integrated into the generalisation system or when the user requirements (the map specifications) change. Unfortunately, revising knowledge is a complex process, which usually requires domain experts. Thus, it is important to trigger a revision process only when necessary. However, giving a full diagnosis of the knowledge quality when several elements of knowledge are used by the generalisation system is difficult.

This paper deals with the problem of the automatic diagnosis of the control knowledge quality. To face this problem, we propose an approach based on the on-line analysis of generalised objects and on the use of a multi-criteria decision making process to evaluate the global knowledge quality from the generalised objects. In this context, we propose to use the classic ELECTRE TRI Method (Yu, 1992).

In Section 2, we introduce the general context in which our work takes place and the related issues. We also provide a brief state of the art of heuristic evaluation approaches. Section 3 is devoted to the presentation of our approach. Section 4 describes a real case study that we carried out as well as its results. Section 5 concludes and states future works.

2. Cartographic generalisation context

2.1. The AGENT generalisation model

The last twenty years have seen the development of numerous generalisation models. The role of such models is to provide a framework to perform complete generalisation of a geographic dataset. In this paper, we investigate generalisation models based on local, step by step, approaches (e.g. (Brassel & Weibel, 1988)) and especially the AGENT model (Ruas & Duchêne, 2007). This model is well-suited for generalisations and it requires discrete operations such as generalisations from large to small scale. Moreover, this model has already been used by many National Mapping Agencies such as IGN (French's NMA) and Ordnance Survey (Great Britain's NMA) for their map production lines.

In the AGENT model, geographic objects are modelled as agents. An agent is a computational entity provided with a goal and it can act autonomously in order to reach this goal thanks to capacities of perception, deliberation, action, and possibly communication with other agents (Weiss, 1999). The geographic agents manage their own generalisation by choosing and applying generalisation algorithms (actions) to themselves. They can as well compute a degree of happiness that will represent the cartographic quality of their current generalisation. The degree of happiness characterises the satisfaction of cartographic constraints (map specifications) by the geographic agent generalisation. For example for a building, a cartographic constraint can be its minimal size in order to be readable. The satisfaction of each constraint is characterised by an integer ranging from 1 to 10 and 10 means that the constraint is perfectly satisfied. The degree of happiness of an agent corresponds to the average of the constraint satisfaction values. Thus, the definition of the happiness function is defined by: *Happiness: states→[1,10]*.

Following the formalism proposed by Russel and Norvig (2003), the agent generalisation problem with the AGENT model is defined by:

- *States*: the different possible generalisations (geometries) of the agent. Each state is evaluated by the happiness function (real between 1 and 10).

- *Initial state*: the initial geometry of the agent.
- *Successor function*: application of a generalisation algorithm (action). The definition domain of the successor function is then defined by: *Successor: states→ states*. Each action ("generalisation algorithm") has its own successor function.

Answering this problem means to find, by testing the least number of generalisation actions, the state that maximises the agent's happiness.

The AGENT model proposes using an explicit search-tree to solve this problem. In this search-tree, each node represents a state of the state space; the root of the tree corresponds to the initial state and the transition from one node to another corresponds to the application of a generalisation action. Several strategies, such as the *Breadth-first search,* the *Depth-first Search* or the *Iterative Deepening Search*), can be used to find an optimal solution (i.e., the node that maximises the happiness function). The strategy used by the AGENT model is the *Greedy best-first search* (Russel & Norvig, 2003, p.95). To guide the search-tree building, this strategy uses heuristics (usually called *procedural knowledge* or *control knowledge* in the Cartographic community). This strategy does not guarantee the optimality of the generalisation process but allows to reduce its time-complexity.

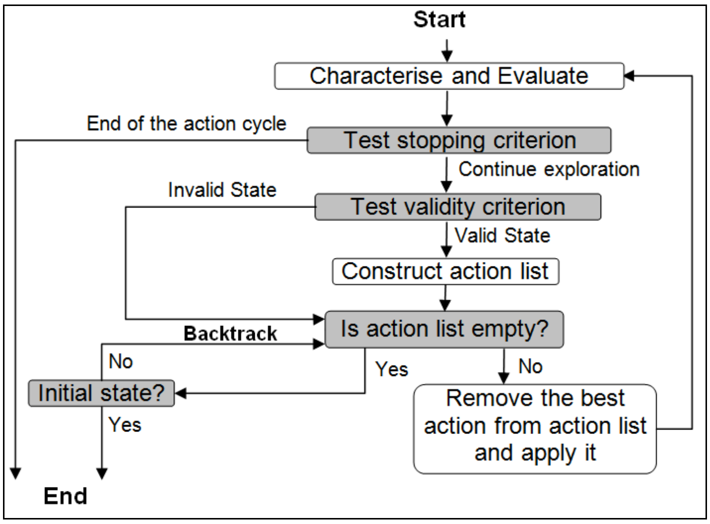


Figure 2. Action cycle

In order to build the search-tree, the agent carries out the action cycle presented Figure 2. The action cycle begins by characterising the current state, i.e. computing the agent happiness for this state. Then, the agent tests if its current state is good enough and, if necessary, continues expanding the tree. If the agent decides to continue the search process, it tests if the current state is valid or not. An invalid state is defined as a state that is unlikely to have a better state among its descendants. If the state is invalid, the agent backtracks to the previous state; otherwise, the agent constructs a list of generalisation actions to be applied. This list of actions is kept in memory and will be reused provided the agent backtracks to this state. If the action list is empty and the current state is not the initial state, the agent backtracks to its previous state (and tests the emptiness of the action list of this previous state); otherwise, the agent chooses the best action, removes it from the action list, applies it and loops to the first step. The action cycle ends either when the stopping criterion is valid or when all actions have been applied for all valid states (i.e. the action lists of all valid states are empty).

An example of a search-tree built with the AGENT model is presented Figure 3.

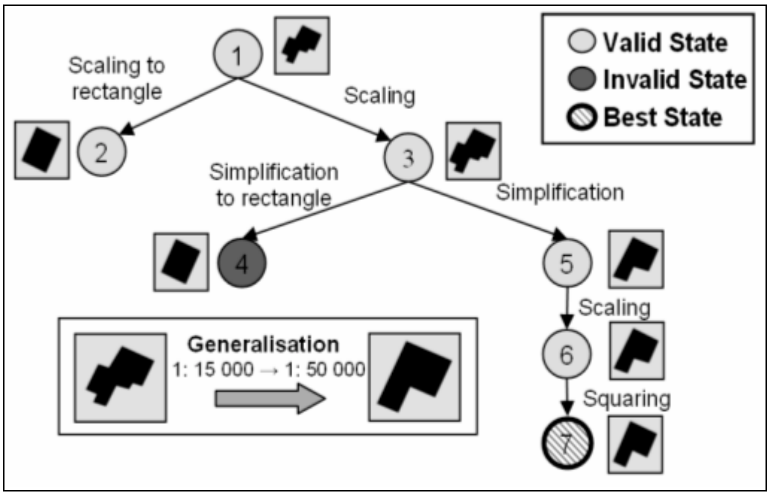


Figure 3. Example of built search-tree

The agent uses three types of control knowledge during the action cycle:

- *Action application knowledge* builds, for the current state of the agent, the action list, i.e. the actions proposed for the state and their application order.
- *Validity criterion* determines, according to all previously visited states (i.e. nodes), if the current state is valid or not.
- *Ending cycle criterion* determines, according to all previously visited states, if the search process has to be continued or not.

In this paper, we call *element of knowledge* each independent knowledge base. For example, the *validity criterion* is an element of knowledge, as well as the *ending cycle criterion*. Concerning the *action application knowledge*, an element of knowledge can be defined for each action.

2.2. Control knowledge quality

The performances of systems based on a *heuristic* tree-search strategy are directly linked to their knowledge quality. The system performances can be expressed in terms of *efficiency* and *effectiveness*. The *effectiveness* refers to the quality of the results obtained by the system, i.e. the quality (in our case, the happiness) of the best found state. The *efficiency* (also called *heuristic power*) (Nilsson, 1980) concerns the time-consuming aspect of the problem instance resolutions, i.e. the system speed to carry out the tree search building. Good knowledge enables the system to be both effective and efficient, i.e. building a small tree that would contain a state of high quality.

2.3. Knowledge quality diagnosis issues

The diagnosis of the system control knowledge quality involves the evaluation of, on one hand, each element of knowledge and, on the other hand, the global quality of the knowledge. The second evaluation is particularly important in the context of map production where the users of generalisation systems are often technicians that have good skills in cartography but no particular knowledge regarding the generalisation system. A simple global evaluation of knowledge is more likely to be understood by them than a complex evaluation of each element of knowledge.

The diagnosis of the control knowledge requires facing three types of difficulties.

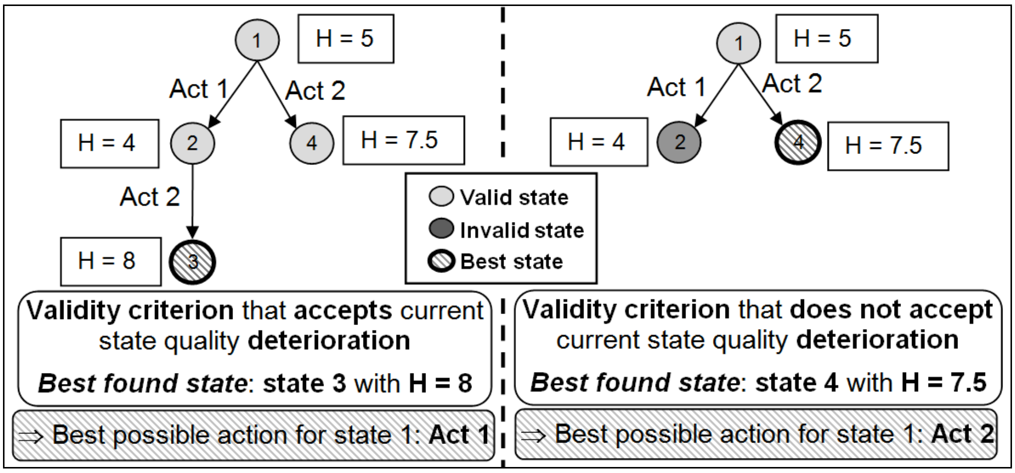


Figure 4. Knowledge dependency problems. *H* represents the happiness of the agent for a given state.

The first one concerns the dependency that might exist between the different elements of knowledge: sometimes, it is not possible to determine if an element of knowledge is really defective or if another element of knowledge is defective which affects the results of the application of the first element of knowledge. Figure 4 shows an example of knowledge dependency: we generalised the same object with two different validity criteria:

- $Crit_1$ allows state quality deterioration (decrease of the agent happiness). 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*.

The second type of issues concerns information that can be extracted from the study of a search-tree. For example, whereas it is possible to extract information concerning the false positive errors of the validity criterion (when a state should not have been considered valid), this remains impossible regarding the false negative errors (when a state should have been considered valid). In fact, when a state has been considered non-valid, it is impossible to know whether a better state would have been found provided the state had been considered valid (since the exploration from this state has been stopped).

The last type of issues refers to the generalised objects used to diagnose the knowledge quality. These objects are determined by the user practice of the system and not according to the knowledge quality diagnosis requirements. These objects could be non-representative of the whole problem instances and thus not reliable to be used for the diagnosis.

In the next section, we propose an on-line diagnosis approach which copes with these issues.

2.4. Related works

Evaluating control knowledge (or heuristics) is a classic problem in Artificial Intelligence. Numerous works propose to evaluate heuristic performance. If most of them focus on an experimental comparison between different heuristics or metaheuristics (e.g. (Stützle*, et al.*, 2000) (Izakian*, et al.*, 2009)), a few propose heuristic evaluation measures.

Thus, Doran and Michie (1966) propose the *penetrance* measure that evaluates how a search focuses towards the best state rather than visiting useless states. In the same logic, Nilsson (1980) proposes a similar measure, the *effective branching factor*, which is the constant number of successors of each state (node) of the tree. Some other interesting studies concern the impact of the inaccuracy of a heuristic on search complexity (Dinh*, et al.*, 2007). In our context, where the goal is to give a diagnosis for each element of knowledge and not only for the whole knowledge, these measures cannot be used directly. Moreover, these measures only concern the search efficiency and not its effectiveness.

Another interesting work concerns the speedup learning. Speedup learning tries to improve the problem solving system efficiency with experience (Mitchell*, et al.*, 1986). Systems based on speedup learning learn a control knowledge (usually in the form of rules) from the analysis of solved problems. The problem of speedup learning systems comes out from the number of rules that are learnt: this number can be very high and the required time to evaluate each rule at each state can be more time-consuming than the search process. This problem is referred to as the utility problem. Minton (1990) proposes to overcome this problem by estimating empirically the utility of each rule and by removing the useless ones. The utility of a rule is computed from the average savings resulting from that rule (the number of nodes that are not explored when the rule is applied), the fraction of times the rule is applicable and the average cost of matching the rule. The idea of rule utility is very interesting but not relevant in our context. In fact, first, in the AGENT model, not all the elements of knowledge are represented by rules. Second, we are more interested in the accuracy of the element of knowledge rather than in their utility. The time for matching a rule is indeed not relevant for the AGENT model as the application of a generalisation action is far more time-consuming than matching rules.

A last domain of research that can be related to our work is the reinforcement learning (Watkins & Dayan, 1992). The goal of reinforcement learning is to compute an optimal policy in order to choose the best action to apply for each state. If the general principle of using rewards to estimate the accuracy of elements of knowledge could be used in our context, reinforcement learning diverges from our problem as in reinforcement learning the choice of an action is only dependent of the current state. In the AGENT model, some of the elements of knowledge (validity and ending cycle criterion) depend also on previously visited states. Moreover, the use of reinforcement learning requires adapting the formalism used to represent the knowledge, which is not possible for the AGENT model.

3. Proposed Approach

3.1. General approach

Our goal is to automatically diagnose the knowledge quality of generalisation systems based on a heuristic search-tree strategy. In this context, we propose an approach based on the on-line analysis of generalised objects and on the use of a multi-criteria decision making process to evaluate the global quality of the knowledge from the generalised objects (Figure 5).

Each time an object is generalised, the diagnosis module analyses, during an *analysis phase*, the successes and failures of each element of knowledge. Afterwards it checks whether the number of objects generalised since the last diagnosis (*Number_objects*) is high enough to make a new diagnosis. This test partially addresses the last difficulty presented in Section 2.3. In fact, making a decision on knowledge quality, only when enough objects are generalised, allows avoiding particular cases that could lead to a wrong decision. The number of generalised objects needed to trigger the decision making process (*NUMBER_OBJECTS_MIN*) depends on the type of objects considered (roads, buildings, building blocks…) and on the user requirements. The higher *NUMBER_OBJECTS_MIN* is, the more reliable the decision making process is, but the required time to generalise an object and the number of available objects can compel the user to specify a low value for it. Thus, the choice of *NUMBER_OBJECTS_MIN* is a matter of compromise. In the context of generalisation systems based on the AGENT model, 30 seems to be a minimum number to ensure the diagnosis relevance; this number was obtained after testing a large set of sample cases.

If the number of generalised objects is high enough, the diagnostic module triggers a *diagnosis phase* which consists in evaluating each element of knowledge and using a multi-criteria decision making method to evaluate the global knowledge quality.

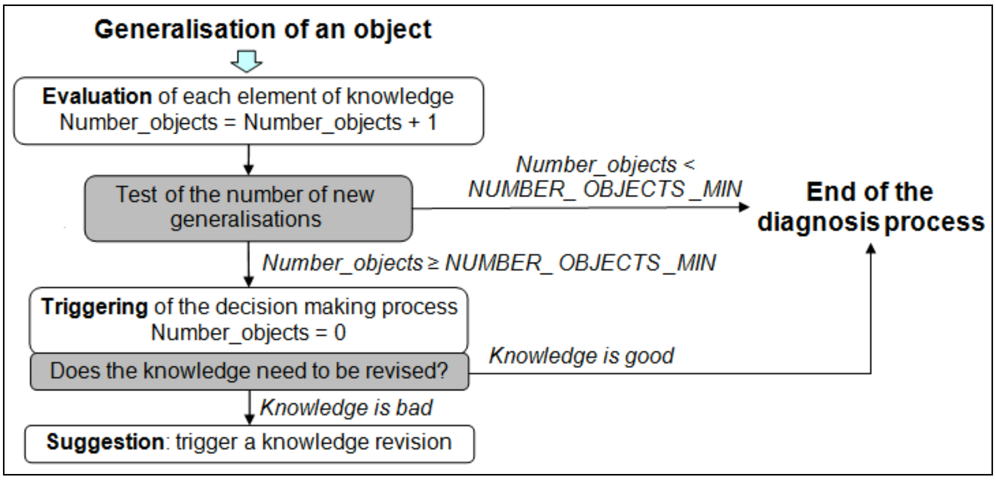


Figure 5. General diagnosis approach

3.2. Analysis phase

Evaluating the global knowledge quality requires, in a first step, to extract information about the relevance of each element of knowledge. The choice of the method used to evaluate this relevance is a key point of the decision-making process. This choice can indeed have a great impact on the results. In general, as mentioned in Section 2.2, there are two main criteria to assess the quality of a generalisation system: the *efficiency* (its speed to generalise an object, i.e. its capability to guide directly toward the best state) and the *effectiveness* (cartographic quality of the generalisation, i.e. the quality of the best state found). Thus, we propose to evaluate the knowledge according to these two aspects. Each time an object is generalised (and a search tree is built), two types of information are extracted from its analysis: information concerning the successes and failures of each element of knowledge (system *efficiency*) and information concerning the performance of the system in terms of cartographic quality of the result (system *effectiveness*).

3.2.1. Knowledge efficiency

In order to determine the system *efficiency*, we propose an approach based on the analysis of the best paths. A best path is a sequence of at least two states, which has the root of a tree (or of a sub-tree) as initial state and the best state (the state that maximize the happiness of the agent) of this tree (or sub-tree) as final state. Once the best path set is computed for a search tree, the computation of the successes and failures of the different elements of knowledge consists in analysing these best paths.

We propose to represent the quality of each element of knowledge through four measures: the number of false negatives (nb_{FN}), the number of false positives (nb_{FP}), the number of true negatives (nb_{TN}) and the number of true positives (nb_{TP}). The way these measure values are computed depends on the nature of the concerned element of knowledge: validity criterion, ending cycle criterion and action application.

For the validity criterion:

- A true positive is a valid state which belongs to the best path.
- A true negative is an invalid state which does not belong to a best path whereas its predecessor belongs to it.
- A false positive is a valid state which does not belong to a best path, whereas its predecessor belongs to it.
- The number of false negatives is not relevant for this criterion. No false negative is considered.

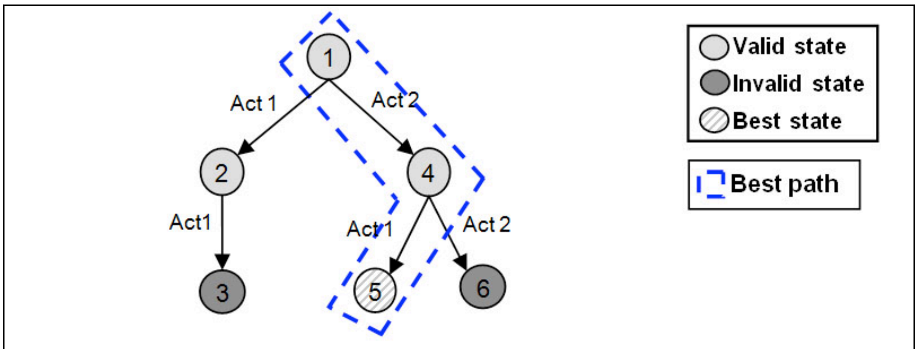
For the ending cycle criterion:

- A true positive is a case where the criterion does not propose to continue the exploration just after having visited the best state of the tree.
- A true negative is a state which belongs to a best path and is not the best state.
- The number of false positives is not relevant for this criterion. No false positive is considered.
- A false negative is a case where the criterion proposes to continue the exploration whereas the best state of the tree has already been found.

For action application knowledge:

- 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 the best path.
- A true negative is a case where, from a state belonging to a best path, the action was not proposed.
- 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.

Figure 6 gives an example of results obtained after having analysed a search tree.



Local evaluation:

Knowledge	FN	FP	TP	TN
Act 1 application	0	1 (state 1)	1 (state 4)	0
Act 2 application	1 (state 4)	1 (state 4)	1 (state 1)	0
State validity	-	1 (state 2)	3 (states 1, 4 and 5)	1 (state 6)
Ending cycle	1 (state 5)	-	0	2 (states 1 and 4)

Figure 6. Example of results for a knowledge successes and failures analysis

To summarize, the higher the numbers of true positive and true negative, the more relevant an element of knowledge is. Actually, these two measures characterise the fact that the element of knowledge allows finding directly the best state. At the opposite, a good element of knowledge has to minimise the number of false positives and false negatives in order to limit the number of useless states (i.e. states that are not necessary to reach the best state).

So, for each element of knowledge, we have four measures that characterise its relevance. We suggest to group these four measures into only one mark defined between 0 and 1 named *Knowledge_Efficiency*. A *Knowledge_Efficiency* of 0 means that the element of knowledge is a priori very defective in terms of efficiency; a *Knowledge_Efficiency* of 1 suggests that the element of knowledge is perfect in terms of efficiency. The *Knowledge_Efficiency* for a given element of knowledge *K* and for a set of generalised objects *Obj* depends on the results obtained for each object during the *analysis phase*:

$$Knowledge\_Efficiency(K, Obj) = \frac{nb_{TP} + nb_{TN}}{nb_{TP} + nb_{TN} + nb_{FP} + nb_{FN}}$$

where nb_{TP} is the number of true positives, nb_{TN} is the number of true negatives, nb_{FP} is the number of false positives and nb_{FN} is the number of false negatives.

As expressed before, the higher the number of true positives and true negatives against the number of false positives and false negatives, the more the element of knowledge will be considered as being good (and thus get a high mark). The choice to use the same weight for each measure was made after testing a large set of example cases.

This evaluation gives an indication to the user concerning the elements of knowledge that have to be revised in priority. If an element of knowledge has a low *Knowledge_efficiency*, it indicates that it fails to relevantly guide the generalisation process.

3.2.2. System effectiveness

The successes and failures of each element of knowledge provide information concerning the *efficiency* of the system, but it cannot provide information concerning the *effectiveness* of the system (the agent happiness of the best state found). In fact, it is not possible to know if it would have been possible to find a better state if more states were visited. Thus, we propose to store the agent happiness of the best found state for each generalised object. This information will be used to evaluate the global quality of the knowledge.

In our application context that concerns the use of the generalisation system for a map production line, we use the following function to evaluate the effectiveness of knowledge:

$$Knowledge\_Effectiveness(Obj) = \left(\frac{FirstQuartile(\{H(obj)\}_{obj \in Obj}) + Average(\{H(obj)\}_{obj \in Obj})}{20} \right)^2$$

where *H(obj)* returns the best happiness found for the generalisation of an object *obj*.

This function allows taking into account the average happiness of the generalised objects and to balance this result by the first quartile happiness value. The interest of this weighting comes from the fact that it is preferable for the mapping agencies to obtain three-quarters of well generalised objects and one quarter of bad-generalised objects (that can be later edited by technicians) rather than obtaining average homogeneous results (which require much more editing). The factor 1/20 is used to normalise the value of this function. Actually, we remind that the happiness of a geographic agent ranges from 1 to 10. So the *Knowledge_Effectiveness* is a floating point value between 0 (the worst effectiveness) and 1 (a perfect effectiveness). We add a power 2 in order to increase the difference between values.

3.3. Diagnosis phase

The analysis phase enables storing information concerning the knowledge quality. The diagnosis phase consists in using this information to determine the global quality of the knowledge. The role of this global evaluation is to provide users that have little understanding of the generalisation system with a brief and simple summary of the control knowledge diagnosis.

3.3.1. From analysis criteria to global evaluation

In order to make the diagnosis, we use several criteria. The first type of criteria refers to the capacity of the different elements of knowledge to ensure a high efficiency of the system: *Knowledge_Efficiency*. The second type of criteria concerns the capacity of the knowledge to (globally) ensure a high effectiveness: *Knowledge_Effectiveness*. At the end of the analysis phase, we have one mark between 0 and 1 per element of knowledge for the efficiency and one mark for the effectiveness (also between 0 and 1).

The goal of the diagnosis process is to determine the global knowledge quality level according to the criterion values. We propose to define five quality levels for the knowledge: {very bad, bad, average, good, very good}. The objective is to define to which of these categories the considered knowledge belongs.

The current quality of the knowledge is characterised by a vector of values corresponding to the current criteria values (i.e. the *Knowledge_Efficiency* assigned to each element of knowledge and the *Knowledge_Effectiveness*). We note $V^{current}$ this vector of values.

3.3.2. Multi-criteria decision making methods

In the literature, several approaches and methods were proposed to solve this type of multi-criteria decision making problems. We propose to use ELECTRE TRI-B method (Almeida-Dias*, et al.*, 2010; Yu, 1992). It was used with success to solve numerous problems (Georgopoulou*, et al.*, 2003; Lourenço & Costa, 2004; Raju*, et al.*, 2000). As pointed out in (Figueira*, et al.*, 2005), the ELECTRE TRI method is particularly relevant for our problem. Actually, we have more than three criteria which are very heterogeneous (it is difficult to directly compare the effectiveness criterion to the mark of an element of knowledge). Moreover, we do not want the loss of a given criterion to be compensated by the gain of another. Finally, we want to integrate in our decision making the fact that, for some criteria, a small value difference is not significant, while the addition of several small differences may become significant.

3.3.3. ELECTRE TRI-B Principle

We defined five levels of quality for the knowledge: very bad, bad, average, good, very good. We express each quality level by two vectors of values, one representing its lower bound and the other one, its upper bound. Thus, for each criterion, each quality level is characterised by an interval of possible values for this criterion. We impose that, for each criterion, the intervals associated to the quality levels are disjoint and that they cover the whole possible values of the criterion. We remind that the values of our criteria are real numbers that range from 0 to 1.

We note $V^{x\to x'}$ the vector of values characterising the boundary between the quality level *x* and the quality level *x'*. Let $V_j^{x\to x'}$ be the value of the criterion *j* for the vector of values $V^{x\to x'}$ and let *C* be the set of criteria. The vector of values $V^{x\to x'}$ can be described by:

$$V^{x} = \left\{ V_j^{x\to x'} \right\}_{j\in C}$$

In the context of our five quality levels of knowledge, we define four boundary value vectors:

- $V^{VBd\to Bd}$: the boundary between the quality level “knowledge set of very bad quality” and the quality level “knowledge set of bad quality”
- $V^{Bd\to Av}$: the boundary between the quality level “knowledge set of bad quality” and the quality level “knowledge set of average quality”
- $V^{Av\to Gd}$: the boundary between the quality level “knowledge set of average quality” and the quality level “knowledge set of good quality”
- $V^{Gd\to VGd}$: the boundary between the quality level “knowledge set of good quality” and the quality level “knowledge set of very good quality”

Figure 7 presents the different intervals of values taken by the five defined knowledge quality levels for a criterion *j*.

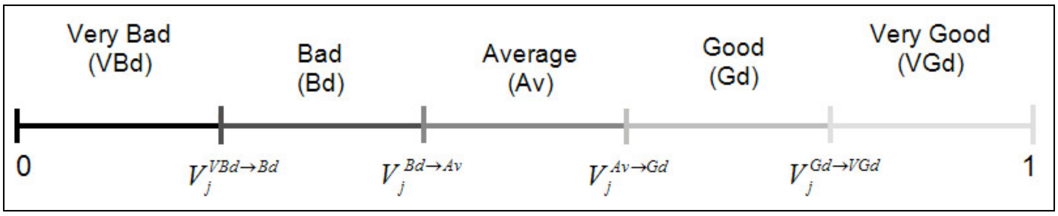


Figure 7. Knowledge quality level for a criterion *j*

The principle of ELECTRE TRI-B method is to compare the vector of criterion values representing the current knowledge set to each boundary vector. In fact, this method tries to define the relation existing between the current vector and each boundary vector. In particular, this method analyses the possible outranking relation (noted $\boldsymbol{S}$) existing between the current vector and each boundary vector. The notation $V^{current}\boldsymbol{S}V^{x\_x'}$ means that the current vector outranks (i.e. is at least as good as) the boundary vector describing the boundary between the knowledge quality levels x and x', which means that the quality of the knowledge set is at least at the level x'. There are two other kinds of relations: the incomparability (noted $\boldsymbol{R}$) and the Indifference (noted $\boldsymbol{I}$). The notation $V^{current}\ \boldsymbol{R}\ V^{x\_x'}$ means that none of the two vectors outranks the other. The notation $V^{current}\ \boldsymbol{I}\ V^{x\_x'}$ means that the vector $V^{current}$ is as good as the vector V^{x_x}; this relation links two vectors which are outranking each other.

By these relations, it is possible to determine in which of the five categories the current knowledge belongs.

3.3.4. ELECTRE TRI-B parameters

The use of the ELECTRE TRI-B requires defining several parameters for each criterion:

- The *weight* of the criterion: importance of the criterion in the knowledge quality level assignment.
- The *preference threshold*: represents the threshold from which the difference between two criterion values allows to prefer one vector over another.
- The *indifference threshold*: represents the threshold from which the difference between two criterion values is considered significant.
- The *veto threshold*: represents for the criterion j the smallest difference $(V_j - V'_j)$ incompatible with the assertion $V\boldsymbol{S}V'$ (Mousseau*, et al.*, 2000).

The last parameter to define is the *λ-cutting level* of the fuzzy relation, which defines the reference threshold for the vector comparison. The higher this threshold, the more the establishment of the relation $V_1 \boldsymbol{S} V_2$ requires unanimity from the criteria concerning the fact that the vector V_1 is higher than to the vector V_2.

These different parameters have a great influence on the result. The definition of their values is the source of an important literature (e.g. (Dias & Mousseau, 2006; Mousseau & Dias, 2004; Rogers & Bruen, 1998). Section 3.3.6 will come back on this point.

3.3.5. ELECTRE TRI-B process

The ELECTRE TRI-B is divided into 5 steps that are described hereafter:

Step 1 - concordances and discordances calculation: The concordance $c_j(a,b)$ represents the degree of certitude on the criterion j that the assertion "the vector a is at least as good as the vector b" is true. The discordance $d_j(a,b)$ represents the degree of certitude on the criterion j that this assertion is false.

Step 2 - global concordance indexes calculation: The global concordance indexes represent the mean concordance obtained for the whole criterion set weighted by the criterion weight. It allows estimating the part of the criteria for which a vector is at least as good as another one.

Step 3 - credibility degree calculation: The credibility degree represents the degree with which a vector is at least as good as another one. It corresponds to the concordance index weakened by a possible effect of a veto.

Step 4 – establish outranking relation through the cutting level: By comparing the credibility degrees with the λ-cutting level, we established the relation (***S***, ***R*** or ***I***; see 3.3.3) between the current vector $V^{current}$ and each boundary vector $V^{x \to x'}$.

Step 5 – determine quality level to the current knowledge set: This last step consists in assigning a quality level to the current knowledge set according to the outranking relations. There are two possible procedures to assign a quality level to the current knowledge set: optimistic and pessimistic procedures.

At the end of the fifth step, a level of quality (or two if we use both *optimistic* and *pessimistic* procedures) is assigned to the current knowledge.

3.3.6. Decision robustness analysis

A key issue of the decision making is the robustness of the decision. The robustness is defined as "capacity for withstanding "fuzzy approximations" and/or "zones of ignorance" in order to prevent undesirable impacts, notably the degradation of the properties to be maintained" (Roy, 2010).

Robustness analysis proposes the use of a set of acceptable parameter values as input to analyse the corresponding result. It allows to measure the capacity of the result to support a change in the parameters. It also allows moderating a result in case of important variation. Traditionally, robustness was viewed as the last phase of the ELECTRE process, but several authors suggested to integrate it in the parameters choice phase (Dias*, et al.*, 2002).

For our application, we propose to make a decision with several sets of parameters and to proceed by a majority vote to determine the final decision. The percentage of votes for each knowledge quality level gives an idea of the robustness of the decision. A high percentage for the votes for a quality level means that the decision is reliable. If the percentage is low, it is important to analyse the percentage of votes obtained by the other quality levels in order to get a better evaluation of the knowledge quality. This method allows limiting the influence of parameters, and thus to have more relevant results.

4. Application

4.1. Case study context

The real case study that we carried out concerns the generalisation of building blocks. The building block generalisation is an interesting case study because it is not yet well managed and it is very time consuming.

We defined, with the help of cartographic experts, six constraints as well as five actions for the building block generalisation. The five actions are the following:

- ***Building generalisation action***: this action triggers the individual generalisation of the building agents composing the building block.
- ***Building displacement action***: this action moves buildings that have proximity problems.
- ***Local building removal action***: this action removes buildings according to a local context, i.e. according to the building's local situation and not the global situation of the building block. It removes in priority the buildings that have the most serious overlapping problems.
- ***Building removal/displacement action***: this action selects the building that has the most serious proximity problems and removes it. If another building is close to the removed building, this one is moved in order to be closer to the removed building.

- ***Global building removal action***: this action removes buildings according to the global context, i.e. according to the global building block situation and not the local situations of the buildings. It removes in priority the buildings that have the least space to move.

In order to test the relevance of our diagnosis approach, we carried out experiments with four knowledge sets. The quality of these knowledge sets was previously evaluated on a great number of building blocks.

The four knowledge bases are described hereafter:

- $\boldsymbol{K_{mostEfficient}}$: is a knowledge set that proposes no action. In fact, this knowledge set ensures only to visit the initial state and thus to obtain the best possible efficiency. However, the quality of the result (which always corresponds to the initial state) is very bad.
- $\boldsymbol{K_{mostEffective}}$: is a knowledge set that proposes to apply all possible actions on all states. For each generalised building block, this knowledge set ensures to find the best possible state considering the constraints and the actions used. Nevertheless, it requires to explore many states per generalisation and is thus not efficient at all.
- $\boldsymbol{K_{Expert}}$: is a knowledge set defined by a cartographic expert who is as well an AGENT model expert. The results obtained with this knowledge set are good in terms of results but only acceptable in terms of efficiency.
- $\boldsymbol{K_{Revised}}$: is a revised version of the knowledge set defined by the AGENT model expert. The knowledge set was revised off-line with the approach proposed by Taillandier (Taillandier*, et al.*, 2011). The results obtained with this knowledge set are good both in terms of efficiency and in terms of effectiveness.

Figure 8 gives an example of cartographic results obtained with these four knowledge sets.

In order to evaluate our approach, we tested it when generalising a small number of building blocks. The goal is to assess if, even with very little information, our approach can establish a relevant diagnosis. We tested our diagnosis approach on the four knowledge bases with two different samples of 30 building blocks to make the diagnosis. We chose to draw the building blocks randomly in order to get close to a realistic scenario where a user has to generalise a high number of building blocks which are generalised in a random order. It appears that the first sample is composed of more dense building blocks than the second one. So, the first will be more difficult to generalise than the second one (more complex readability problems). The number of 30 was defined empirically, which is high enough to provide a first reliable diagnosis and not too high in order to reach it quickly.

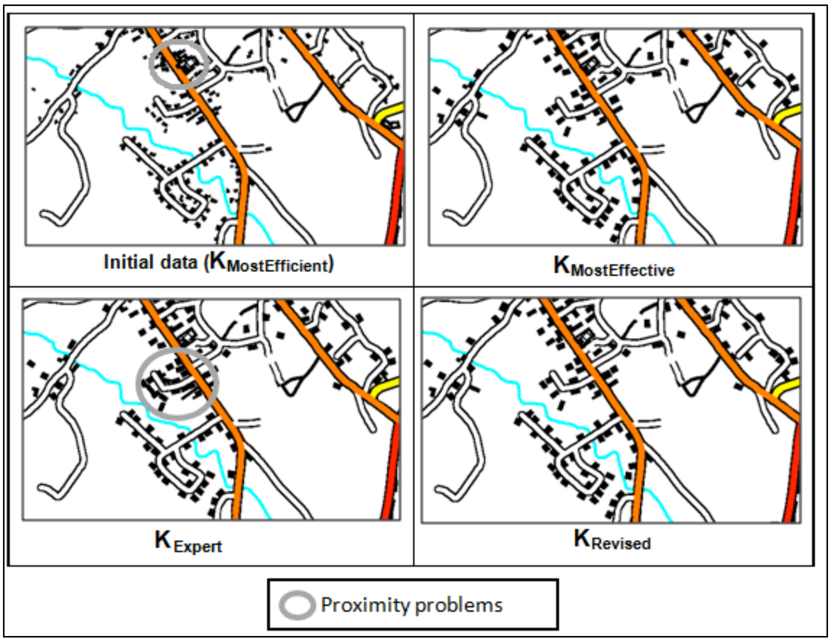


Figure 8. Example of cartographic results obtained with the knowledge sets

The reason for testing on two different building block samples is to reduce the influence of each case specificity on the results and have an element on the robustness of the evaluation.

4.2. Definition of parameters

The parameters (i.e. the boundary vectors and the multi-criteria decision making method parameters) were defined empirically with tests carried out on other knowledge sets and on other areas. They were chosen in order to favour the effectiveness of the system over its efficiency. In fact, our priority is to obtain good generalisation results. If this condition is not ensured, whatever the system effectiveness is, the knowledge has to be revised. To consider this point, we chose parameter sets with a higher weight for the effectiveness criterion than for efficiency criteria.

Twenty parameter sets were used to test the robustness of the decision. These 20 parameters sets were considered sufficient to obtain results with a good robustness.

4.3. Results

First, we calculated, for each knowledge set and each sample, the value of each criterion. Table 1 shows the results obtained during the analysis phase.

Table 1 shows as well the elements of knowledge that the diagnosis process point out as defective. We consider that an element of knowledge is defective if it has a *Knowledge_Efficiency*(K,Obj) < 0.5. This threshold represents a point from which, we have more false positive and false negative (which have a negative consequence on efficiency) than true positive and true negative (which have a positive consequence on efficiency). The limit 0.5 was determined subjectively, in a first approach. However, it is possible to have a deeper reflexion on this limit and more globally on the "defective" status by analysing the global knowledge quality obtained with the ELECTRE TRI-B method. In the article perspective (Section 5), we will come back to this point.

From these evaluations, we can apply the ELECTRE TRI-B method to obtain a global quality of the knowledge. Figure 9 shows the diagnosis results obtained. The percentage of votes for a level represents the percentage of parameters values (among the 20 defined) for which the multi-criteria decision making method assigned at this level to the knowledge set (cf. Section 3.3.4). The considered quality level is the quality level that maximises the percentage of votes.

Table1. Analysis results: *Knowledge_Efficiency* and *Knowledge_Effectiveness*. K_1 represents the element of knowledge relative to the application of the Building generalisation action; K_2, the element of knowledge relative to the application of the Building displacement action; K_3, the element of knowledge relative to the application of the Local building removal action; K_4, the element of knowledge relative to the application of the Building removal/displacement action; K_5, the element of knowledge relative to the application of the Global building removal action; K_6, the element of knowledge relative to the validity criterion; K_7, the element of knowledge relative to the stopping criterion.

		Knowledge_Efficiency							**Defective Element of knowledge**	**Knowledge Effectiveness**
		K_1	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**		
$K_{MostEfficient}$	**Sample1**	1	1	1	1	1	1	1	-	*0.73*
	Sample2	1	1	1	1	1	1	1	-	*0.67*
$K_{MostEffective}$	**Sample1**	0.65	0.58	0.57	0.23	0.31	0.11	0.09	• K_4 • K_5 • K_6 • K_7	*0.89*
	Sample2	0.64	0.54	0.58	0.23	0.32	0.10	0.05	• K_4 • K_5 • K_6 • K_7	*0.87*
K_{Expert}	**Sample1**	0.85	0.73	0.68	0.25	0.56	0.63	0.18	• K_4 • K_7	*0.85*
	Sample2	0.87	0.72	0.71	0.15	0.68	0.68	0.25	• K_4 • K_7	*0.83*
$K_{Revised}$	**Sample1**	0.91	0.76	0.78	0.85	0.71	0.67	0.55	-	*0.84*
	Sample2	0.88	0.83	0.84	0.9	0.81	0.74	0.62	-	*0.81*

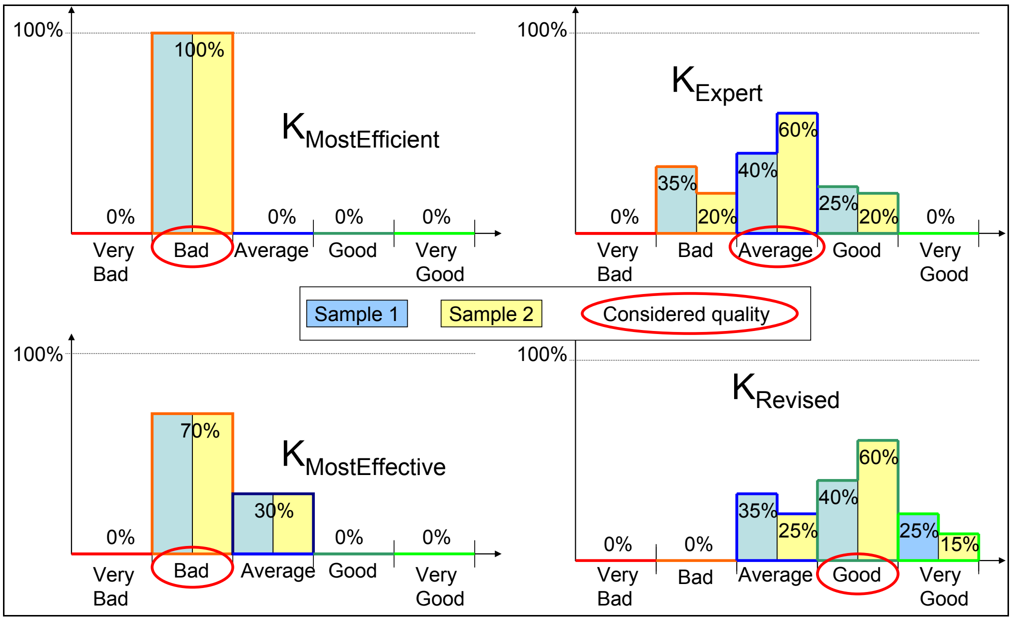


Figure 9. Diagnosis results: knowledge set quality levels and percentage of votes for the each quality level

4.4. Discussion on the results

We can draw several lessons from these results. First, these results confirm the robustness of the knowledge quality evaluation. The results obtained with both samples are indeed very close. The only difference concerns the percentage of votes obtained by the *average* level for $\boldsymbol{K_{Expert}}$ and by the *good* level for $\boldsymbol{K_{Revised}}$. This difference shows that the first sample, which is composed of more dense building blocks (see Section 4.1.), is consistently more difficult to generalise efficiently and effectively than the second one. Actually, for $\boldsymbol{K_{Expert}}$, the level of *bad* quality obtained 35% of votes for the first sample instead of 20% for the second one. In the same way, for $\boldsymbol{K_{Revised}}$, the level of *average* quality obtained 35% of votes for the first sample instead of 25% for the second one. Nevertheless, these differences of values are rather subtle.

Secondly, these results are consistent with the tested knowledge sets. The only knowledge set ranked as *good* is $\boldsymbol{K_{Revised}}$, which is a revised version of $\boldsymbol{K_{Expert}}$ and provides the best compromise between efficiency and effectiveness (cf. Section 4.1.). For this knowledge set, no element of knowledge was detected as defective.

The $\boldsymbol{K_{Expert}}$ knowledge set was ranked as *average*. This knowledge set provides good results in terms of effectiveness but average results in terms of efficiency. In the context of an actual application, this knowledge set can be used to generalise small areas but not to generalise areas with many building blocks (such as big cities). Moreover, the good performance obtained by $\boldsymbol{K_{Revised}}$, which is a revised version of $\boldsymbol{K_{Expert}}$, shows that this knowledge set can be improved. For this knowledge set, the diagnosis process allows detecting that the stopping criterion and the *Building removal/displacement action* application knowledge were defective. Regarding the stopping criterion, as is it very difficult for a human to define a reliable one, the AGENT model expert chose to define a non-restrictive stopping criterion. Thus, this criterion can be improved in order to stop the search tree exploration when not necessary. For the *Building removal/displacement action* application knowledge, the expert tended to overuse this action.

Concerning the two other knowledge sets ($K_{mostEffective}$ and $K_{mostEfficient}$), they were both ranked as *bad*. $K_{mostEffective}$ and $K_{mostEfficient}$ are unusable for real applications. $K_{mostEfficient}$, which was ranked as *bad* with a percentage of votes of 100% with both samples, is the worst of the two. In fact, this knowledge set is defined in such a way that no action is applied: the cartographic results obtained are thus not acceptable. As no action was applied, the diagnosis process could not detect defective cases (no knowledge was used). In contrast, $K_{mostEffective}$ gives very good results in terms of effectiveness. However, its results in terms of efficiency are very bad. The required time to generalise building blocks is far too high to use it in map production. This knowledge set was ranked as *bad* with a percentage of votes of 70%, and as *average* with a percentage of votes of 30% on both samples. This vote distribution is consistent with the fact that $K_{mostEffective}$ is slightly less bad than $K_{mostEfficient}$. Concerning the diagnosis of each element of knowledge, the validity criterion, the stopping criterion, the *Building removal/displacement action* application knowledge and the *Global building removal action* application knowledge were detected as defective. Concerning the stopping criterion and the validity criterion, only non-restrictive criteria were defined in this knowledge set. Thus, they can be improved. The two other elements of knowledge were detected as defective because the *Building removal/displacement action* and the *Global building removal action* were overused.

5. Conclusion

In this paper, we proposed an on-line knowledge quality diagnosis for cartographic generalisation based on heuristic tree-search strategy. Our approach is based on the analysis of generalised objects and on the use of a multi-criteria decision making method. We evaluated our approach on a real case study that we carried out for building block generalisation. This case study showed that our approach is able to give a relevant evaluation of the knowledge quality.

Solving problems by using a heuristic tree-search strategy is a classic approach in many domains. Thus, our diagnosis approach could be used for other application domains other than cartographic generalisation. One of our future works is to adapt our diagnosis approach to other kinds of problem solving systems.

A key point of our approach is the effectiveness evaluation function. Designing such a performance function to represent the user requirements can be complex because of the difficulty to formalise the user requirements. Thus, an interesting future work will consist in developing methods to help users to design this function.

One approach that could be used to face this problem could consist in directly using machine learning techniques. Thus, a sample of results would be proposed to an expert. The expert would give an effectiveness mark to each of the generalisation results. Theses given marks would be then used to learn the effectiveness function.

Another approach, more complex, could consist in designing the performance function using an active learning. Thus, it could be interesting to use, as a base, the works of Taillandier and Gaffuri (2009) and Christophe (2011). The system would present several samples of results (solved with different knowledge sets) to the expert. The expert could define which result is the best and add comments about the results through a specific interface. The system would use these comments in order to refine the effectiveness function and to choose new result samples to present to the expert. The learning would result from the dialogue between the expert and the system.

A point that deserves more work concerns the elicitation of the parameter values. In fact, the quality of the diagnosis is directly linked to the relevance of the values chosen for the parameters. It is thus important to choose relevant parameter values. Several works propose approaches to elicit parameters values, notably weight elicitation: compensatory, criteria prioritization and inference by case studies. It is this last approach which allows determining also the others parameters which seems the more interesting in our case (Mousseau*, et al.*, 2000). We can complete this approach by applying works like (Jabeur & Guitouni, 2007) which propose the use of machine learning techniques.

Last but not least, another point that deserves more studies concerns the “defective” status of the element of knowledge. In fact, in Section 3.4, we described a defective element of knowledge as an element of knowledge which has a *Knowledge_Efficiency*(K,Obj) < 0.5. The limit 0.5 was determined subjectively. However, it is possible to take advantage of the global knowledge evaluation obtained with the ELECTRE TRI method to validate this limit by identifying the minimal set of knowledge elements that are responsible for the (bad) global quality.

6. References

Almeida-Dias, J., Figueira, J. R., & Roy, B. (2010). Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions. *European Journal of Operational Research, 204*, 565-580.

Brassel, K. E., & Weibel, R. (1988). A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Systems, 2*, 229-244.

Christophe, S. (2011). Creative Colours Specification Based on Knowledge (COLorLEGend system). *Cartographic Journal, 48*, 138-145.

Dias, L. C., & Mousseau, V. (2006). Inferring Electre's veto-related parameters from outranking examples. *European Journal of Operational Research, 170*, 172-191.

Dias, L. s., Mousseau, V., Figueira, J., & [illegible] conclusions with ELECTRE TRI. *European Journal of Operational Research, 138*, 332-348.

Dinh, H., Russell, A., & Su, Y. (2007). On the value of good advice: the complexity of A* search with accurate heuristics. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2* (pp. 1140-1145). Vancouver, British Columbia, Canada: AAAI Press.

Doran, J. E., & Michie, D. (1966). Experiments with the Graph Traverser Program. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 294*, 235-259.

Figueira, J., Mousseau, V., & Roy, B. (2005). ELECTRE Methods. In J. Figueira, S. Greco & M. Ehrgott (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys* (pp. 133-162). New York: Springer.

Georgopoulou, E., Sarafidis, Y., Mirasgedis, S., Zaimi, S., & Lalas, D. P. (2003). A multiple criteria decision-aid approach in defining national priorities for greenhouse gases emissions reduction in the energy sector. *European Journal of Operational Research, 146*, 199-215.

Izakian, H., Abraham, A., & Snasel, V. (2009). Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments. In *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization - Volume 01* (pp. 8-12): IEEE Computer Society.

Jabeur, K., & Guitouni, A. (2007). Automated learning multi-criteria classifiers for FLIR ship imagery classification. In *Information Fusion, 2007 10th International Conference on* (pp. 1-8).

Lourenço, R. P., & Costa, J. P. (2004). Using ELECTRE TRI outranking method to sort MOMILP nondominated solutions. *European Journal of Operational Research, 153*, 271-289.

Minton, S. (1990). Quantitative results concerning the utility of explanation-based learning. *Artificial Intelligence, 42*, 363-391.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine Learning, 1*, 47-80.

Mousseau, V., & Dias, L. (2004). Valued outranking relations in ELECTRE providing manageable disaggregation procedures. *European Journal of Operational Research, 156*, 467-482.

Mousseau, V., Slowinski, R., & Zielniewicz, P. (2000). A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support. *Computers & Operations Research, 27*, 757-777.

Nilsson, N. (1980). *Principles of Artificial Intelligence*. San Fransisco: Morgan Kaufmann.

Raju, K. S., Duckstein, L., & Arondel, C. (2000). Multicriterion Analysis for Sustainable Water Resources Planning: A Case Study in Spain. *Water Resources Management, 14*, 435-456.

Rogers, M., & Bruen, M. (1998). Choosing realistic values of indifference, preference and veto thresholds for use with environmental criteria within ELECTRE. *European Journal of Operational Research, 107*, 542-551.

Roy, B. (2010). Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research, 200*, 629-638.

Ruas, A., & Duchêne, C. (2007). A prototype generalisation system based on the multi-agent system paradigm. In W. A. Mackaness, A. Ruas & L. T. Sarjakoski (Eds.), *Generalisation of Geographic information: cartographic modelling and applications* (pp. 269-284). Amsterdam: Elsevier.

Russel, S., & Norvig, P. (2003). *Artificial Intelligence – a modern approach* (2 ed.). Upper Saddle River, New Jersey: Prentice Hall.

Stützle, T., Grün, A., Linke, S., & Rüttger, M. (2000). A Comparison of Nature Inspired Heuristics on the Traveling Salesman Problem

Parallel Problem Solving from Nature PPSN VI. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo & H.-P. Schwefel (Eds.), (Vol. 1917, pp. 661-670): Springer Berlin / Heidelberg.

Taillandier, P., Duchêne, C., & Drogoul, A. (2011). Automatic revision of the control knowledge used by trial and error methods: Application to cartographic generalisation. *Applied Soft Computing, 11*, 2818-2832.

Taillandier, P., & Gaffuri, J. (2009). Objective Function Designing Led by User Preferences Acquisition. In *6th International Conference on Information Technology and Applications* (pp. 124-129). Ha noi, Viet Nam.
Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*, 279-292.
Weiss, G. (1999). *Multiagent systems. A modern approach to distributed artificial intelligence*. The MIT Press.
Yu, W. (1992). *Aide multicritère à la décision dans le cadre de la problématique du tri : Concepts, méthodes et applications.* Université Paris-Dauphine.