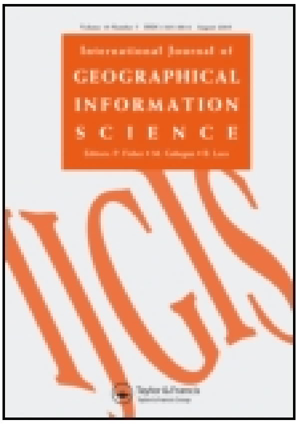


International Journal of Geographical Information Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis20

Improving map generalisation with new pruning heuristics

Patrick Taillandier [a] & Julien Gaffuri [b c]

[a] UMR-IDEES 6228, MTG Laboratory, Université de Rouen, Mont-Saint-Aignan, France

[b] COGIT Laboratory, - IGN, Saint Mandé, France

[c] JRC-IES-SDI, Ispra, Italy

Version of record first published: 09 Jan 2012.

To cite this article: Patrick Taillandier & Julien Gaffuri (2012): Improving map generalisation with new pruning heuristics, International Journal of Geographical Information Science, 26:7, 1309-1323

To link to this article: http://dx.doi.org/10.1080/13658816.2011.625948

International Journal of Geographical Information Science
Vol. 26, No. 7, July 2012, 1309–1323

Improving map generalisation with new pruning heuristics

Patrick Taillandier[a]* and Julien Gaffuri[b,c]

[a]*UMR-IDEES 6228, MTG Laboratory, Université de Rouen, Mont-Saint-Aignan, France;* [b]*COGIT Laboratory, – IGN, Saint Mandé, France;* [c]*JRC-IES-SDI, Ispra, Italy*

(*Received 13 July 2011; final version received 15 September 2011*)

Many automated generalisation methods are based on local search optimisation techniques: Starting from an initial state of the data, one or several new child states are produced using some transformation algorithms. These child states are then evaluated according to the final data requirements, and possibly used as new candidate state to transform. According to this approach, the generalisation process can be seen as a walk in a tree, each node representing a state of the data, and each link a transformation. In such an approach, the tree exploration heuristic has a great impact on the final result: Depending on which parts of the tree are either explored or pruned, the final result is different, and the process more or less computationally prohibitive. This article investigates the importance of exploration heuristic choice in automated generalisation. Different pruning criteria are proposed and tested on real generalisation cases. Recommendations on how to choose the pruning criterion depending on the need are provided.

Keywords: generalisation; exploration; local search; pruning criterion; artificial intelligence

1. Introduction

Generalisation is the simplification performed on geographical data when decreasing their representation scale. The purpose of this complex operation is to adapt the level of detail of the data for some specific needs. This operation is particularly important to produce countrywide map series at different scales from a single detailed database. The generalisation automation still remains an open issue for data producers and users. Many generalisation models introduce artificial intelligence techniques to progress towards an always higher level of automation. Among these generalisation models, some consider the generalisation process as an informed exploration of a state tree: Each node of the tree represents a state of the data and is a candidate to be a 'better' generalised state. Each state is assessed using an evaluation function which evaluates the distance to a hypothetical perfectly generalised state. A child state is the state obtained after the application of a generalisation transformation algorithm to its parent state. Because the number of states to explore increases rapidly with the number of transformations available (and different sets of parameters associated with them), the strategy used to explore the state tree has to be considered carefully. In existing methodologies, some strategies have been proposed and give promising results.

*Corresponding author. Email: patrick.taillandier@gmail.com

ISSN 1365-8816 print/ISSN 1362-3087 online

http://dx.doi.org/10.1080/13658816.2011.625948
http://www.tandfonline.com

This article focuses on a specific criterion of the exploration strategy, the pruning criterion, which determines which child states have to be explored. Our purpose is to show the importance of this criterion and to propose new pruning criteria to produce better generalised data faster.

In the next sections, we present the generalisation model we use to test some new pruning criteria we propose. The outcomes of these tests are analysed and recommendations on how to choose the pruning criterion are provided.

2. Context

2.1. Tree search strategy in local search generalisation models

Recent contributions in generalisation automation research mainly concern the development of transformation algorithms, spatial analysis methods and generalisation models. Generalisation models provide generic frameworks to perform a complete orchestration of the generalisation process. Most of them, based on the approach of Beard (1991), consider generalisation as a constraint satisfaction problem: The target state of the data is formalised by a set of constraints on some characteristics of the objects. A generalisation model provides thus a generic optimisation method to solve this constraint satisfaction problem. Some of these generalisation models use global deformation techniques to obtain a well-generalised state of the data. For example, Sester (2000) and Harrie and Sarjakoski (2002) adapt least square adjustment to optimise the set of constraints (represented by equations on the object geometry coordinates). Burghardt and Meier (1997), Bader (2001) and Højholt (2000) use the finite elements method.

Other generalisation models are based on a local search approach: Starting from the initial state of the data, child states are produced using discrete transformation algorithms. These child states are then evaluated and possibly reused as next candidates to produce new states. The generalisation process consists in a local exploration in a state tree. Brassel and Weibel (1988) and Mcmaster and Shea (1992) first proposed to use such an approach in map generalisation. They underlined the need for decomposition of the generalisation process into several steps, and the need for generalisation evaluation methods to better control the progression of the process towards a good generalised state. Ruas and Plazanet (1996) introduced an exploration process where the data are first split into small parts and then progressively transformed using a depth-first exploration process with single backtracking. Other generalisation models based on this local search approach propose to use common exploration heuristics like steepest gradient descent and simulated annealing (Ware and Jones 1998) or genetic algorithms (Wilson *et al*. 2003). In this article, we focus on the AGENT model presented in the section 'The AGENT model'.

2.2. The AGENT model

The AGENT model is based on Ruas (1999) and was developed during the AGENT European project (Lamy *et al*. 1999). It is an application of the agent paradigm in generalisation. An agent is 'a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives' (Weiss 1999, p. 29). An agent can be seen as an 'alive' object, which has a goal and capabilities to autonomously reach this goal by possibly interacting with other agents. In the AGENT model, each geographic object is an agent, whose purpose is to generalise itself by taking into account its state and its context. The agent has the capability to evaluate its state by using some spatial analysis methods, and to choose and apply a suitable transformation algorithm on itself depending on its state. An agent can be a single object, like a building or a road section (these agents are called 'micro-agents'), or a group of objects which have to be considered together during the generalisation process, for example, to be aggregated or typified (such agents are called 'meso-agents').

The target state of the generalised data is formalised by a set of constraints on some of the agent characteristics. For example, an agent which has to be big enough to be legible will have a constraint on its size, which will force it to enlarge. The constraints assigned on an agent and their goal values depend on the required target state of the generalised data. Each constraint is characterised by a satisfaction value. The overall satisfaction value of an agent is expressed as the mean of its constraint satisfactions weighted by their importance. In addition, constraints are characterised by a list of transformations *a priori* suitable to improve their satisfactions. Thus, the agent is guided by its constraints during its generalisation process. For example, the size constraint will propose an enlargement transformation to its agent if it is too small.

The purpose of each agent is to reach a perfect state that is a state where all constraints are satisfied. In most cases, such a state does not exist and an optimum state that satisfies the constraints as well as possible has to be found. To reach this goal, each agent follows the action cycle presented in Figure 1. First, the agent evaluates its state. If its state is not perfect and the pruning criterion is not satisfied, the most pertinent transformation is applied. If the pruning criterion is satisfied and the current state is the initial state, it means the whole tree has been explored, and then the process terminates. If there is no transformation to try from a given state, the agent goes back to a previous state until he/she has found a transformation to try. This action cycle results in a classical depth-first informed exploration of a state tree. Figure 2 shows an example of such a state tree for a building. Each node is a state (the root is the initial state), and each link corresponds to the application of a transformation algorithm.

Local search-based generalisation models require the definition of important control knowledge elements. For the AGENT model, some of these knowledge elements can be captured using machine learning techniques as proposed by Taillandier (2007) and Taillandier *et al*. (2011). In this article, we investigate the knowledge concerning the pruning criterion.

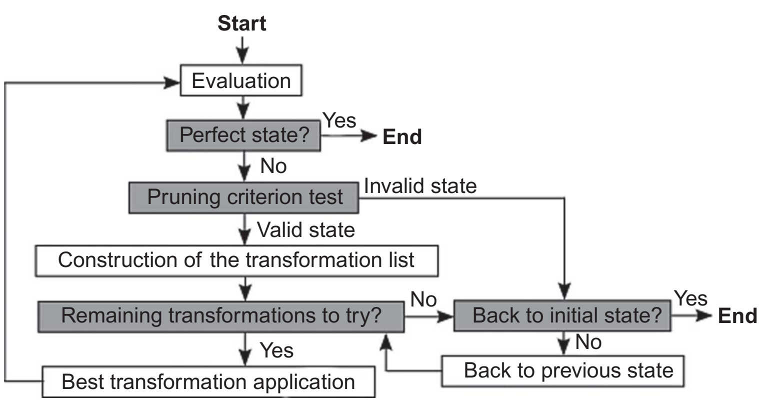


Figure 1. The AGENT model action cycle.

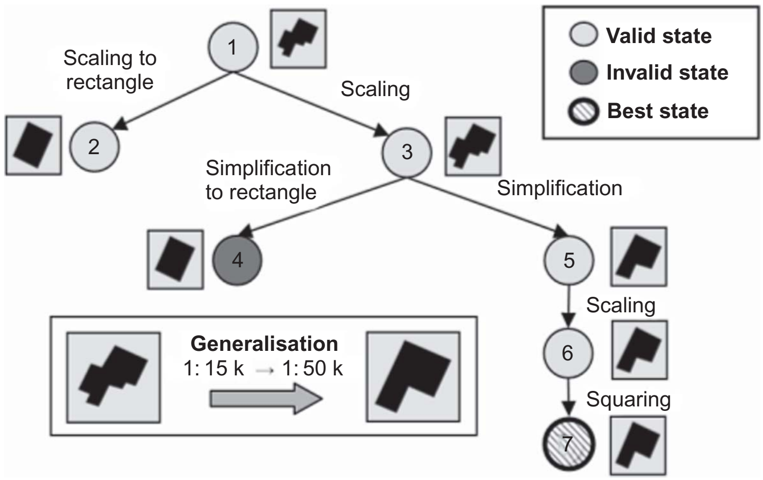


Figure 2. Example of a state tree for the generalisation of a building.

3. State validity and pruning criterion

A valid state is defined as a state whose children have to be explored during the process – an invalid state is a state where the exploration stops (see Figure 2). All children of a valid state are *a priori* interesting to explore – a better state may be among them. All children of an invalid state are supposed to be irrelevant to compute and examine. The pruning criterion determines the validity of a state. This criterion has a great impact on the generalisation process. It influences two major characteristics of the generalisation process:

- *Effectiveness*: Effectiveness is the capability to produce a high-quality outcome. Because different parts of the tree are explored depending on the pruning criterion, the best state encountered during the exploration process and so, the outcome quality, is affected.
- *Efficiency*: Efficiency is the capability to produce an outcome fast. A restrictive pruning criterion will result in a short exploration, so the process will return a result faster.

These two characteristics are in opposition: Effective processes usually have a low efficiency, and the opposition is true. When many states are explored, the probability to get a better optimum state increases. The challenge of the pruning criterion choice is to find a satisfying balance between efficiency and effectiveness of the exploration. This balance depends on the user preference between a quick process and a high-quality outcome.

A common pruning criterion consists in using a comparison of the successive states: A state is valid when its satisfaction S is better than its parent's one S', plus a threshold value So: $S \geq S' + So$. (The threshold value So is usually small, but not null in order to ensure the process termination.) This pruning criterion ensures an improvement of the satisfaction value when the exploration goes deeper in the tree. However, this criterion limits the possibility to explore many interesting states for which a local satisfaction lowering is necessary. This problem is comparable with the local minimum problem of the steepest gradient method. Some other common pruning criteria such as 'branch and bound' (Land and Doig 1960) are not applicable to the specific problem of generalisation state exploration.

In order to face this issue, some researchers (Regnauld 2001, Taillandier 2007) have pointed out the necessity to define an *ad hoc* criterion for the AGENT model. According to the most commonly used pruning criterion, a state is valid if these two conditions are satisfied:

- The satisfaction of the constraint which proposed the transformation has been improved more than a threshold.
- There is a partial improvement compared with all other states of the tree. A state is partially better than another if at least one of its constraints has a higher satisfaction than the other one.

The second condition assures the termination of the process: The constraint satisfaction value is an integer number within [0, 100], so the number of constraint satisfaction configurations is finite (its value is 101^N, where N is the number of constraints). Thus, the second condition will not be satisfied anymore after at least N^{101} tries. With this pruning criterion, good results are obtained, but it is not fully satisfying. In some cases, satisfying states are not reached, while in other cases satisfying states could be obtained faster. Our purpose is to further study the efficiency and effectiveness of different pruning criteria in order to provide recommendations on how to choose it.

4. New pruning criteria

The pruning criterion may be defined using the state satisfaction (the overall value or the constraint ones) compared with other already encountered states (all of them or only the previous one). The pruning criterion may be a logical expression of several conditions. We propose the following ones.

Two conditions based on the comparison with the previous state:

- *Condition on the previous state global improvement (CPSGI)*: This condition is satisfied if the state overall satisfaction is greater than its parent one. This condition corresponds to the heuristic used by the steepest gradient descent.
- *Condition on the previous state constraint improvement (CPSCI)*: This condition is satisfied if the satisfaction of the constraint proposing the transformation has been improved. This condition is most of the time satisfied.

Three conditions based on the comparison with all other encountered states are as follows:

- *Condition on the state set based on the non-similarity (CSSNS)*: This condition is satisfied if the state is not comparable with any other state. Two states are comparable if all their constraint satisfaction values are the same. With this condition, many states with different constraint satisfaction configurations are explored.
- *Condition on the state set based on partial improvement (CSSPI)*: This condition is satisfied if the state is partially better compared with the other states. A state is partially better than another if at least one of its constraints has a higher satisfaction than the other one. This condition is more restrictive than the CSSNS condition because it allows only improvement of constraint satisfactions.
- *Condition on the constraint set based on partial improvement (CCSPI)*: This condition is satisfied if a constraint satisfaction is higher than for the other states.

Some other additional conditions are as follows:

- *Condition on the state set threshold number (CSSTN(N))*: This condition is satisfied if the tree state number is less than a threshold value N. This constraint does not depend on the state. It ensures the termination of the process.
- *Random condition (RC(p))*: This condition is randomly satisfied according to a probability p within]0,1].
- *No condition (NC)*: This condition is always satisfied. All states will be explored until the agent reaches a perfect state or has explored the whole tree. It is equivalent to the condition RC(1.0).

This list is not exhaustive – it may be adapted and extended depending on the need and the generalisation model specificities. Many pruning criteria may be defined using logical expressions of these conditions. We propose to test the following ones:

- Criterion 1: CPSCI
- Criterion 2: CPSCI and CPSGI
- Criterion 3: CPSCI and CPSGI and CSSNS
- Criterion 4: CPSCI and CPSGI and CSSPI
- Criterion 5: CPSCI and CSSNS
- Criterion 6: CPSCI and CSSPI
- Criterion 7: CPSGI
- Criterion 8: CPSGI and CSSNS
- Criterion 9: CPSGI and CSSPI
- Criterion 10: CSSNS
- Criterion 11: CSSPI
- Criterion 12: CCSPI
- Criterion 13: CPSCI and CCSPI
- Criterion 14: CPSCI and CPSGI and CCSPI
- Criterion 15: CPSGI and CCSPI
- Criterion 16: RC(0.2)
- Criterion 17: RC(0.5)
- Criterion 18: NC.

We assume these pruning criteria provide a good range of criteria, more or less restrictive, with different balances between efficiency and effectiveness. The last three criteria are tested to have a reference in terms of effectiveness and efficiency for the other criteria. The most commonly used pruning criterion in the AGENT model is criterion 6.

Not all conditions ensure a theoretical termination of the exploration process. To solve this problem, a limit application number is assigned to each transformation: For each branch of the tree, the agent is not allowed to apply each transformation more than this limit number. Including such a limitation is acceptable, because computing the same transformation a large number of times on an object is often useless (many algorithms need to be applied only once – only few algorithms based on an incremental approach demand to be applied a large number of times). This limitation ensures the termination of the process – the maximal state number an agent can encounter is $(N \times \mathrm{AMax})!/\mathrm{AMax}!^N$ with N being the number of transformations, and AMax the maximum number of tries of each transformation. Finally, to ensure a practical termination of the process, we have used the condition CSSTN(1000) for all criteria. We consider a criterion producing a tree with more than 1000 states has an unacceptable efficiency, and the process should stop.

5. Experiments and results

5.1. Test cases

The proposed pruning criteria have been tested on four typical generalisation cases: the generalisation of buildings and urban blocks for 1:25k and 1:50k scales (see Figure 3). For building generalisation, six constraints and seven transformation algorithms have been used, and three constraints and five transformation algorithms for block generalisation (Figures 2, 3 and 7 give an overview of the effect of these algorithms). The same constraints and transformations have been used for both scales (1:25k and 1:50k), but with different goal values. Further description of these constraints and algorithms is given in Ruas and Mackness (1997) and Taillandier (2008).

The input data were BDTopo® data produced by IGN France on a sample of 846 buildings and 139 urban blocks. For each test, several characteristics of the generalisation process have been measured in order to assess the pruning criterion.

5.2. Effectiveness and efficiency measures

In order to assess the impact of a pruning criterion on the generalisation process effectiveness and efficiency, we propose to capture the following measurements:

- For effectiveness assessment, the distribution of the final satisfaction of the generalised objects.
- For efficiency assessment, the distribution of the total encountered state numbers. (The process duration distribution could have been used, but it depends on which transformations were computed and also on the computer processor speed.)

Figure 4 shows two charts obtained for criterion 1 on the 846 buildings set for 1:25k scale. Each chart represents a distribution: It is the ranking of the 846 buildings according to

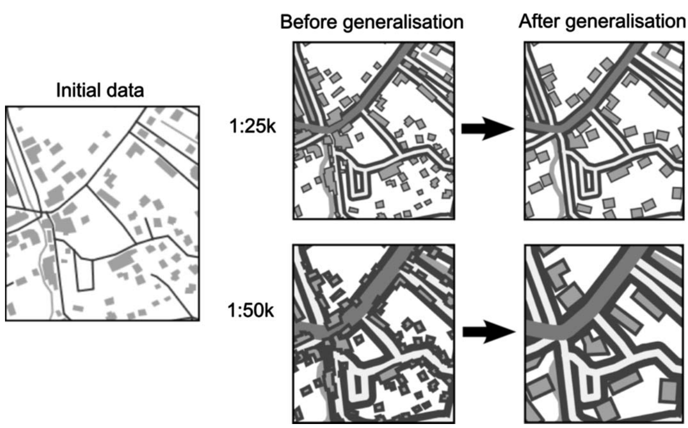


Figure 3. Examples of generalisation for scales 1:25k and 1:50k.
Source: Data from the IGN France digital landscape model BDTopo®.

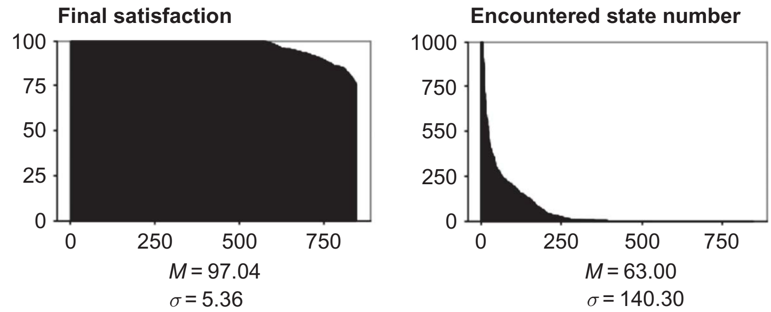


Figure 4. Test output example: the criterion 1, on the 846 buildings set, for the 1:25k scale.

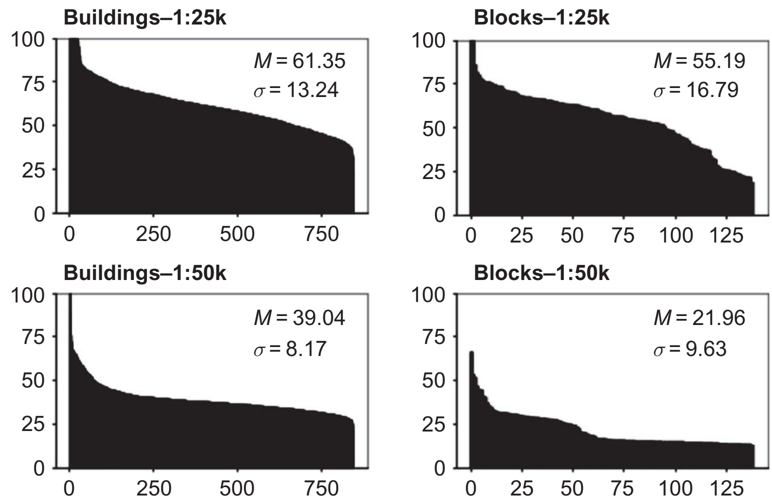


Figure 5. Satisfaction distribution of the four cases in the initial state: buildings and blocks for 1:25k and 1:50k scales.

the value considered. Each distribution is summarised by its average value M and the SD value σ. We base the assessment of each criterion on such kind of outcome. Figure 5 shows the satisfaction distributions in the initial state for the four cases. These distributions show how unsatisfied the objects are in their initial state. Objects for the 1:50k scale are less satisfied than for the 1:25k, because more constraints are violated when the scale change is bigger.

5.3. Results and discussion

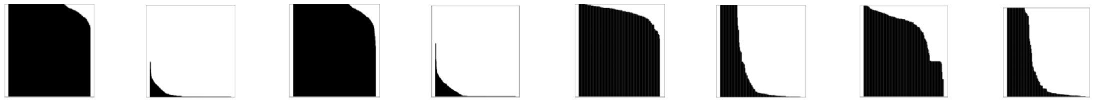
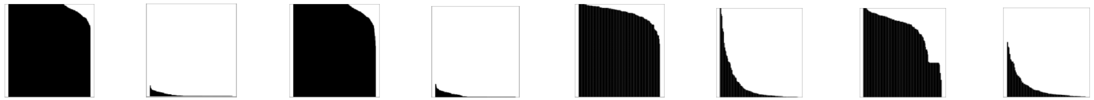
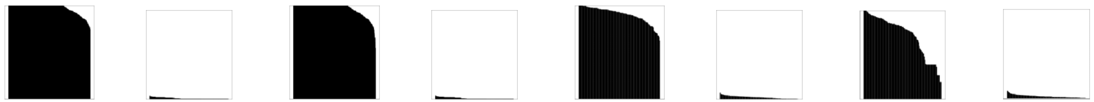
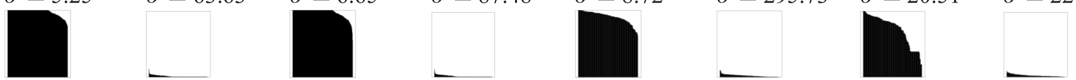
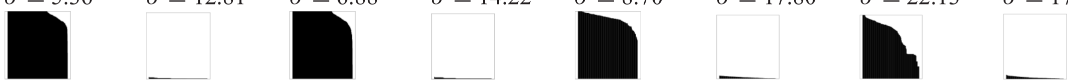
Table 1 shows the outputs for the proposed criteria. As expected, the criteria have different effectiveness and efficiency levels. The computation of the generalisation with criterion 18, the theoretically least restrictive criterion (all states are valid) did not succeed – no output has been obtained. Criteria 16 and 17 (based on random choices) give interesting outcomes: The RC(0.2) has an excellent efficiency level but a very low effectiveness level. The opposite output is obtained for the RC(0.5) criterion. These three criteria illustrate the necessity to choose suitable exploration heuristics in order to better control

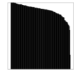
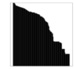
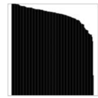
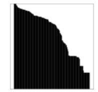
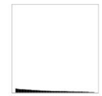
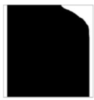
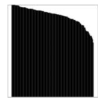
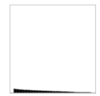
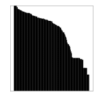
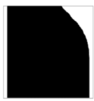
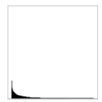
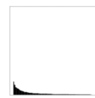
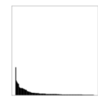
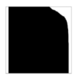
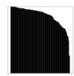
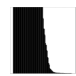
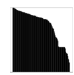
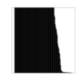
Table 1. Test outputs.

	Buildings				Blocks			
	1:25k		1:50k		1:25k		1:50k	
1 CPSCI								
	$M = 97.06$ $\sigma = 5.36$	$M = 63.00$ $\sigma = 140.30$	$M = 96.46$ $\sigma = 6.80$	$M = 91.46$ $\sigma = 174.43$	$M = 90.74$ $\sigma = 9.46$	$M = 327.90$ $\sigma = 425.27$	$M = 74.90$ $\sigma = 22.80$	$M = 334.35$ $\sigma = 430.65$
2 CPSCI and CPSGI								
	$M = 96.81$ $\sigma = 5.73$	$M = 20.64$ $\sigma = 35.45$	$M = 96.15$ $\sigma = 7.28$	$M = 27.15$ $\sigma = 43.15$	$M = 90.85$ $\sigma = 9.04$	$M = 281.46$ $\sigma = 393.60$	$M = 74.50$ $\sigma = 23.14$	$M = 314.87$ $\sigma = 415.78$
3 CPSCI and CPSGI and CSSNS								
	$M = 96.73$ $\sigma = 5.89$	$M = 11.58$ $\sigma = 15.82$	$M = 96.00$ $\sigma = 7.65$	$M = 14.20$ $\sigma = 19.45$	$M = 90.51$ $\sigma = 9.90$	$M = 115.99$ $\sigma = 177.10$	$M = 73.20$ $\sigma = 23.52$	$M = 78.16$ $\sigma = 98.82$
4 CPSCI and CPSGI and CSSPI								
	$M = 96.70$ $\sigma = 5.94$	$M = 6.82$ $\sigma = 6.21$	$M = 95.96$ $\sigma = 7.72$	$M = 7.53$ $\sigma = 7.06$	$M = 89.69$ $\sigma = 9.99$	$M = 19.71$ $\sigma = 13.12$	$M = 69.11$ $\sigma = 24.94$	$M = 20.46$ $\sigma = 12.80$
5 CPSCI and CSSNS								
	$M = 96.94$ $\sigma = 5.52$	$M = 20.84$ $\sigma = 37.86$	$M = 96.31$ $\sigma = 7.17$	$M = 27.47$ $\sigma = 44.93$	$M = 90.71$ $\sigma = 10.17$	$M = 158.57$ $\sigma = 240.51$	$M = 73.72$ $\sigma = 22.84$	$M = 91.74$ $\sigma = 117.98$
6 CPSCI and CSSPI								
	$M = 96.92$ $\sigma = 5.56$	$M = 8.29$ $\sigma = 9.25$	$M = 96.19$ $\sigma = 7.23$	$M = 8.96$ $\sigma = 9.86$	$M = 89.71$ $\sigma = 10.12$	$M = 21.67$ $\sigma = 15.88$	$M = 68.60$ $\sigma = 25.05$	$M = 21.10$ $\sigma = 13.85$

(*Continued*)

Table 1. (Continued)

	Buildings				Blocks			
	1:25k		1:50k		1:25k		1:50k	
7 CPSGI 	$M = 96.83$ $\sigma = 5.70$	$M = 28.65$ $\sigma = 54.51$	$M = 96.16$ $\sigma = 7.26$	$M = 42.91$ $\sigma = 75.37$	$M = 91.03$ $\sigma = 8.19$	$M = 293.02$ $\sigma = 402.52$	$M = 76.30$ $\sigma = 21.14$	$M = 345.73$ $\sigma = 423.93$
8 CPSGI and CSSNS 	$M = 96.84$ $\sigma = 5.64$	$M = 14.00$ $\sigma = 20.13$	$M = 96.14$ $\sigma = 7.25$	$M = 18.76$ $\sigma = 26.91$	$M = 90.60$ $\sigma = 8.67$	$M = 133.29$ $\sigma = 217.83$	$M = 76.06$ $\sigma = 20.89$	$M = 104.45$ $\sigma = 138.08$
9 CPSGI and CSSPI 	$M = 96.80$ $\sigma = 5.71$	$M = 7.89$ $\sigma = 7.81$	$M = 96.12$ $\sigma = 7.21$	$M = 8.68$ $\sigma = 8.84$	$M = 90.26$ $\sigma = 8.51$	$M = 20.48$ $\sigma = 14.10$	$M = 72.16$ $\sigma = 22.67$	$M = 23.04$ $\sigma = 15.70$
10 CSSNS 	$M = 97.14$ $\sigma = 5.25$	$M = 32.04$ $\sigma = 63.63$	$M = 96.66$ $\sigma = 6.65$	$M = 47.90$ $\sigma = 87.48$	$M = 90.97$ $\sigma = 8.72$	$M = 191.40$ $\sigma = 295.73$	$M = 76.87$ $\sigma = 20.51$	$M = 152.71$ $\sigma = 220.78$
11 CSSPI 	$M = 97.04$ $\sigma = 5.36$	$M = 10.47$ $\sigma = 12.81$	$M = 96.35$ $\sigma = 6.88$	$M = 11.63$ $\sigma = 14.22$	$M = 90.07$ $\sigma = 8.70$	$M = 88.55$ $\sigma = 17.80$	$M = 73.05$ $\sigma = 22.13$	$M = 24.71$ $\sigma = 17.31$
12 CCSPI 	$M = 96.12$ $\sigma = 6.89$	$M = 5.95$ $\sigma = 5.16$	$M = 95.27$ $\sigma = 8.71$	$M = 6.34$ $\sigma = 5.85$	$M = 88.88$ $\sigma = 9.40$	$M = 19.30$ $\sigma = 12.31$	$M = 71.16$ $\sigma = 23.05$	$M = 22.43$ $\sigma = 14.11$

13 CPSCI and CCSPI	 $M = 96.21$ $\sigma = 6.42$	$M = 5.96$ $\sigma = 5.26$	 $M = 95.17$ $\sigma = 8.72$	$M = 5.94$ $\sigma = 5.15$	 $M = 88.69$ $\sigma = 10.34$	$M = 19.17$ $\sigma = 12.43$	 $M = 67.08$ $\sigma = 26.03$	$M = 18.22$ $\sigma = 10.25$
14 CPSCI and CPSGI and CCSPI	 $M = 96.25$ $\sigma = 6.43$	$M = 5.40$ $\sigma = 6.43$	 $M = 95.24$ $\sigma = 8.58$	$M = 5.83$ $\sigma = 4.79$	 $M = 88.12$ $\sigma = 10.65$	 $M = 17.87$ $\sigma = 11.36$	 $M = 66.77$ $\sigma = 25.78$	 $M = 18.35$ $\sigma = 10.23$
15 CPSGI and CCSPI	 $M = 96.20$ $\sigma = 6.49$	$M = 5.49$ $\sigma = 4.41$	 $M = 95.29$ $\sigma = 8.70$	$M = 5.67$ $\sigma = 4.59$	 $M = 88.49$ $\sigma = 9.19$	 $M = 17.76$ $\sigma = 11.05$	 $M = 70.73$ $\sigma = 23.56$	 $M = 20.68$ $\sigma = 12.78$
16 RC (0.2)	 $M = 93.66$ $\sigma = 11.35$	 $M = 7.33$ $\sigma = 11.40$	 $M = 88.78$ $\sigma = 20.56$	 $M = 10.44$ $\sigma = 17.90$	 $M = 65.99$ $\sigma = 20.58$	 $M = 17.40$ $\sigma = 23.92$	 $M = 41.75$ $\sigma = 27.00$	 $M = 24.48$ $\sigma = 39.26$
17 RC (0.5)	 $M = 96.44$ $\sigma = 6.68$	 $M = 42.26$ $\sigma = 114.10$	 $M = 96.22$ $\sigma = 7.45$	 $M = 115.67$ $\sigma = 234.36$	 $M = 86.90$ $\sigma = 13.99$	 $M = 569.21$ $\sigma = 476.60$	 $M = 73.96$ $\sigma = 23.39$	 $M = 764.99$ $\sigma = 409.55$
18 NC	X	X	X	X	X	X	X	X

the effectiveness/efficiency balance. As expected, restrictive criteria are more efficient but less effective than less restrictive ones. For example, criterion 10 is less restrictive than criterion 12, and is therefore more effective but less efficient.

When comparing the four test cases (building/block, 1:25k/1:50k), differences in effectiveness and efficiency are observed: Resulting effectiveness is higher for buildings than for blocks, and (almost) always for 1:25k than for 1:50k. The opposite result is obtained for the efficiency. This difference can be explained by the fact that generalisation is more difficult for blocks than for buildings, and for 1:50k than for 1:25k. When comparing the criteria for each case, similar results are observed: Effective and efficient criteria for one case are also effective and efficient for another. This shows that the quality of a criterion may be considered as independent of the generalisation case. For this reason, we propose to assess each criterion by its mean effectiveness and efficiency for each tested case. Figure 6 shows the repartition of the criteria in the effectiveness/efficiency space (criteria 16–18 are not displayed). As expected, there is no perfect criterion with high effectiveness and efficiency: A balance has to be chosen. Pertinent criteria are along the dashed grey line, which represents the Pareto frontier of our optimisation problem. According to this frontier line, the most interesting criteria are 8–11. Criterion 10 is the most effective one. Criterion 9 is the most efficient one with an acceptable effectiveness (criteria 12–15 are more efficient, but the difference is insignificant, while the difference in terms of effectiveness is important). Criteria 11 and 8 are good balances between effectiveness and efficiency: Criterion 11 is more efficient and criterion 8 is more effective. Because criterion 9 is comparable with criterion 11, we propose to eliminate it.

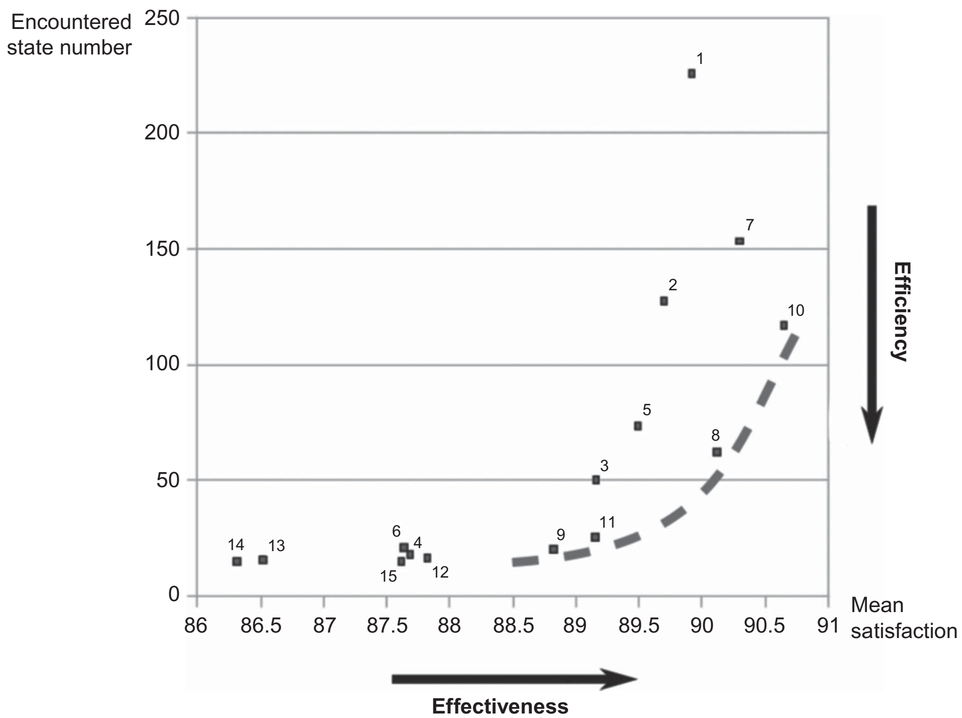


Figure 6. Mean efficiency and effectiveness of the tested pruning criteria.

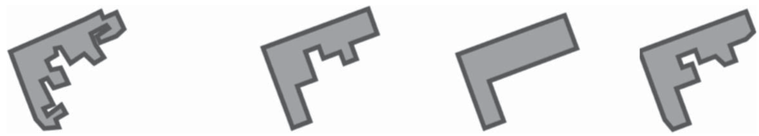


Figure 7. Output examples on a building for the 1:25k scale.

Therefore, according to these tests, our recommendation for the pruning criterion choice is the following:

- For an effective process, which produces high-quality output, criterion 10 (CSSNS) is recommended.
- For an efficient process, which produces a good quality output quickly, criterion 11 (CPSGI and CSSPI) is recommended.
- For a generalisation, which balances output quality and process efficiency, criterion 8 (CPSGI and CSSNS) is recommended.

Figure 7 shows the generalisation obtained for a building with the three criteria. For some cases (especially for simple buildings), same outputs are obtained for different pruning criteria. The challenge of the pruning criterion choice is to obtain a quick exploration for simple objects, and a deeper one for much more complex objects. Simple buildings such as buildings with only four corners only need the computation of a few states to find a satisfying one; much more complex buildings such as the one presented in Figure 7 need a deeper exploration. The criteria we propose contribute to solve this problem.

6. Conclusion and perspectives

In this article, the importance of the pruning criterion choice in generalisation based on a tree search has been presented. We have shown the consequences of this choice on the generalisation effectiveness and efficiency. Several new pruning criteria have been introduced and tested. As a result, three pruning criteria with different levels of efficiency and effectiveness have been recommended.

To go further, other efficiency measures could be tested. We used a measure based on the number of states. A more robust measure should take into account an estimated computation time of each algorithm. A performance measure of each algorithm could also be used: This measure could be a mean ratio between satisfaction gain and computation time – it would assess the capability of an algorithm to improve states as much as possible, and as fast as possible.

Even if we observed similar results for the four generalisation cases, the effect of each pruning criterion changes depending on the generalisation case (data theme, target scale, data density and so on). It should not be excluded that some specific cases may require the use of a more specific pruning criterion. Suitable pruning criteria could be recommended for some sets of predetermined generalisation cases.

As mentioned by Taillandier *et al*. (2009), different elements of knowledge of an informed tree search can be interdependent: The quality of one knowledge element can affect the quality of another. For example, if some knowledge elements used to guide the exploration process (transformation application domain, constraint priority and so on) are not robust, generalising with a restrictive pruning criterion can lead to bad generalisations. For our case study, guiding knowledge elements were robust ones. It would be interesting to analyse the effect of our pruning criteria on non-robust guiding knowledge elements.

The way to translate the user needs in terms of effectiveness and efficiency could be improved. Indeed, we proposed criteria for users who want a fast or an effective process. It could be interesting to let users express their needs as formulations like: 'I want the best generalisation you can compute in *X* seconds'. The strategy to apply for such kind of problems has to be established. In the emerging context of online and on-demand mapping, the customisation of automatic mapping systems according to the user needs is becoming crucial.

Finally, this work is a new example of successful implementation of artificial intelligence techniques in automated mapping and geographical information sciences. For other problems, such as automatic labelling or automatic map colour assignment, where a tree search is considered, the methodology we presented may be applied.

References

Bader, M., 2001. *Energy minimization methods for feature displacement in map generalisation*. Thesis (PhD). Department of Geography, Zurich University.

Beard, K., 1991. Constraints on rule formation. *In*: B. Buttenfield and R. Mcmaster, eds. *Map generalisation: making rules for knowledge representation*. Essex: Longman Scientific and Technical, 121–135.

Brassel, K.E. and Weibel, R., 1988. A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Science*, 2 (3), 229–244.

Burghardt, D. and Meier, S., 1997. Cartographic displacement using the snakes concept. *In*: W. Foerstner and L. Pluemer, eds. *Semantic modelling for the acquisition of topographic information from images and maps*. Basel: Birkhaeuser, 59–71.

Harrie, L. and Sarjakoski, T., 2002. Simultaneous graphic generalization of vector data sets. *Geoinformatica*, 6 (3), 233–261.

Højholt, P., 2000. Solving space conflicts in map generalisation: using a finite element method. *Cartography and Geographic Information Science*, 27 (1), 65–74.

Lamy, S., *et al*., 1999. The application of agents in automated map generalisation. *In*: *19th international cartographic conference*, 14–21 August 1999, Ottawa, ON.

Land, A.H. and Doig, A.G., 1960. An automatic method for solving discrete programming problems. *Econometrica*, 28, 497–520.

Mcmaster, R.B. and Shea, K.S., 1992. *Generalization in digital cartography*. Washington, DC: Association of American Geographers.

Regnauld, N., 2001. Constraint based mechanism to achieve automatic generalisation using an agent modelling. *In*: *GISRUK conference proceedings*, 18–20 April 2001, University of Glamorgan, 329–332.

Ruas, A., 1999. *Modèle de généralisation de données géographiques a base de contraintes et d'autonomie*. Thesis (PhD). Paris-Est University and COGIT Laboratory.

Ruas, A. and Mackaness, W., 1997. Strategies for urban map generalization. *In*: *18th international cartographic conference*, 23–27 June 1997, Stockholm, Sweden. 1387–1394.

Ruas, A. and Plazanet, C., 1996. Strategies for automated generalization. *In*: *7th international symposium on spatial data handling*, 12–16 August 1996, Delft, Netherlands. 319–336,.

Sester, M., 2000. Generalization based on least square adjustment. *International Archives of Photogrammetry and Remote Sensing*, 23 (part B4), 931–938.

Taillandier, P., 2007. Automatic knowledge revision of a generalisation system. *In*: *Workshop on generalisation and multiple representation*, 2–3 August 2007, Moscow, Russia.

Taillandier, P., 2008. *Révision automatique des connaissances guidant l'exploration informée d'arbres d'états. Application au contexte de la généralisation de données géographiques*. Thesis (PhD). Paris-Est University and COGIT Laboratory.

Taillandier, P., Duchêne, C., and 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., *et al*., 2009. Using belief theory to diagnose control knowledge quality – application to cartographic generalization. *In*: *International conference on computing and telecommunication technologies (IEEE-RIVF'09)*, 13–17 July 2009, Danang, Vietnam.

Ware, M.J. and Jones, C.B., 1998. Conflict reduction in map generalization using iterative improvement. *GeoInformatica*, 2 (4), 383–407.

Weiss, G., 1999. *Multiagent systems. A modern approach to distributed artificial intelligence*. Cambridge, MA: The MIT Press.

Wilson, I.D., Ware, J.M., and Ware, J.A., 2003. Reducing graphic conflict in scale reduced maps using a genetic algorithm. *In*: *5th workshop on progress in automated map generalisation*, 28–30 April 2003, Paris, France.