Geoinformatica (2007) 11:497–513
DOI 10.1007/s10707-007-0018-z

Convergence Analysis and Quality Criteria for an Iterative Schematization of Networks

Silvania Avelar

Received: 10 October 2005 / Revised: 10 August 2006
Accepted: 3 January 2007 / Published online: 27 March 2007

Abstract Schematic networks are linear abstractions of functional networks, such as route networks. Lines in the original network are modified in order to produce a schematic network which satisfies a set of constraints chosen to design the network. A method is described which accomplishes this line transformation using an iterative improvement technique driven by design constraints. The method maintains topological characteristics of the network by the use of simple geometric operations and tests. The iterative process can be repeated until the line displacements become small enough or until it meets user defined stopping criteria. Experimental results are provided to examine the acceptability of outcomes and the convergence of the applied iterative technique. Criteria for measuring the quality of results, as well as for stopping the iterative approach are presented.

Keywords schematic network · topology preservation · iterative algorithm · convergence analysis · quality criteria

1 Introduction

Schematic networks are designed to convey information of limited scope, making them easy to interpret by concentrating on the relevant aspects of the information and omitting unnecessary details. They can be used in any scenario in which streams of objects at nodes in a network play a role, such as routes in a transportation system or cartographic schemes for gas, water or electricity mains. The rationale is that it is more important that users capture the basic structure of the network than that they see the precise physical location of the objects.

S. Avelar (✉)
Empa, Swiss Federal Laboratories for Materials Testing and Research,
Lerchenfeldstr. 5, CH-9014 St.Gallen, Switzerland
e-mail: silvania.avelar@empa.ch

Schematizing networks, just like almost any layout problem, requires a set of constraints on how the information can be laid out. For example, for showing the transport routes of a city, schematic networks are designed according to common-sense geometric and aesthetic constraints to improve the clarity of information, which are in general fixed angles for route directions and a small distance between overlapping routes. The schematization of networks is usually performed by hand or by using drawing software. With small datasets or simple network systems, spatial conflicts are a straightforward part of the design task. When tens or hundreds of such cases can occur or when the network should be updated and revised often, the problem becomes a crucial issue. It is not only a time consuming task, but requires a skilled graphic designer. However, instead of schematizing route networks by hand, lines of the original network could be automatically moved to new positions that meet its design constraints better than before. Automatic schematization of networks may improve final results and make the process quicker and cheaper. When considering computational methods, one main difficulty is how to impose a set of constraints on the line segments of a network [4]. The task is to find an arrangement in which the overall deformation in direction and distances is minimized.

The challenges of schematizing networks automatically are not entirely new. Similar problems can be found in graph drawing [5], VLSI layout design [10], and computer cartography, especially for the generation of schematic maps, e.g. [3], [6], [7], [9], [16]. But previous work on the automation of schematic maps has focused mainly on the simplification of lines, without considering other design characteristics of schematic maps and the quality of results [6], [16]. More comprehensive approaches have been presented by Elroi [9], Cabello et al. [7] and the author of this paper [3], [2]. Elroi uses a grid to fit the route network in fixed directions, but topological conflicts among features can occur and no solution is given to avoid them. In Cabello et al., topological equivalence between the networks is obtained, however it can occur that no schematization at all is found. Furthermore, the output network has every path displayed as a two- or three-link path, with links restricted to given orientations which are not based on the natural orientations of the input network. The approach of Avelar [3], [2] preserves topological relationships in the final results and a general sense of the original geometry. The main ideas in the thesis of Avelar [3] for schematization of networks are presented here.

The process of schematizing networks is introduced and developed in Section 2. This Section also explains how to detect and avoid topological changes in the network during the schematization process. Then Section 3 describes the experiments to analyse the quality of the resulting schematic networks. Section 4 provides the convergence analysis of the applied iterative algorithm and finds stopping criteria for the iterative assignment. Finally, conclusions and future work are given in Section 5.

2 Schematization of networks

A database is provided with a line network to be schematized. The schematic network here is to be designed according to the following five constraints, in order to assure good readability and to meet aesthetic considerations [1]: 1. simplification—schematic lines should be straight; 2. orientation—lines should lie in vertical, horizontal or diagonal directions; 3. spacing—the distances between disjoint features should be greater than some minimum distance; and 4. length—lines should have a length greater than some minimum length to reduce congestion, and smaller than some maximum length to avoid large displacements in the network; and 5. topology—original network and derived schematic network should be topologically consistent.

Before the schematization process is described, information about the applied iterative approach will be presented.

2.1 The iterative approach

Schematizing networks can be regarded as a search problem, in which a layout is sought which satisfies the design constraints chosen for the network. The set of all possible states in which the schematic network can find itself constitutes the search space. Attempts to solve this problem by an exhaustive search of the space of candidate solutions are impractical because of the amount of computer time needed. However the search is for an acceptable schematic network rather than the best one, because there is not only one good schematic network; instead there are several workable ones.

A well-established approach to solve search problems is to use iterative algorithms. The iterative strategy applied in this work is the gradient descent technique [2], [11], [13], which attempts to proceed toward an optimal solution by finding a sequence of feasible solutions, each of which is an improvement on the previous one. Gradient descent techniques can not guarantee that globally optimal solutions will be found, but in schematic networks the search is for a workable, acceptable result. Lines of the input network have their positions iteratively adjusted according to the design constraints for the final network. The technique focuses on reducing the number of conflicts on the current location of an object and allows the creation of new conflicts, enabling the process to continue to reduce conflicts by the displacement of objects. Such iterative techniques can improve situations initially, but deteriorate after a certain point. Therefore it is necessary to find stopping criteria for the iterative process. This requires an analysis of outcomes and the convergence evaluation of the iterative algorithm.

2.2 Algorithmic outline

Starting from an initial network configuration with all points at their original locations, each point in turn is considered. Any point that does not satisfy the set of chosen constraints is then relocated in order to satisfy them. The main steps of the schematization process are given below.

Preprocessing steps To satisfy the constraint of straight lines, a line simplification algorithm is used. Geometric line details are removed with the Douglas–Poicker algorithm, which reduces the number of visualization points that constitute the original lines [8]. The simplified lines that are too long are then cut according to the constraint of maximum length. The new straight lines can next be moved to fulfill the other constraints for the schematic network: orientation, spacing, topology, and minimum length.

Iterative algorithm To satisfy the constraints of orientation and spacing, the network configuration is improved iteratively. The iterative algorithm is based on the gradient descent technique driven by design constraints. The iterative algorithm visits all points $p_1..p_n$ of the simplified network to change the actual configuration. If the current point p violates one or more constraint, a better location $p\prime$ for p is computed. For each constraint $c^i \in \{c^1, \ldots, c^m\}$ that affects p, $p^{(i)}$ is the location nearest to p that satisfies c^i. For the constraint of stylized angles, say c^j, the new location $p^{(j)}$ is chosen from the locations that fit p into one of the allowed schematic directions. To compute the distance to the nearest required schematic direction, instead of considering the distances from the endpoint of a line segment, we use its middle point. This gives smaller, finer displacement vectors for the endpoints of a line segment. See $\overline{pq}$ in Fig. 1a. The nearest distance Δp determines the new location for p.

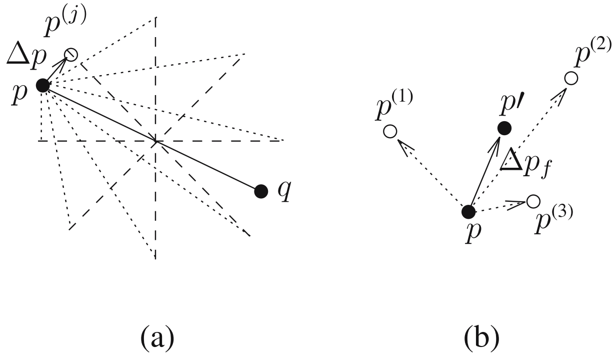
The constraints evaluation is carried out for all line segments in which p exists. The new location $p\prime$ for p will be obtained by the arithmetic mean of all required locations $p^{(i)}$ for all line segments in which p takes part. See the example in Fig. 1b with three constraints. Note that, instead of computing a simple mean of the displacement vectors, it might also be feasible to produce a weighted average of the vectors, since certain lines are more important than others.

The constraint of minimum distance among features is the next to be fulfilled. Adjustments to the final location $p\prime$ are performed when necessary. Line segments are stretched or shortened to reach the new computed final locations, referred to as schematic locations. Short line segments may shrink to a point, which is then corrected to a line by the minimum length constraint. This constraint essentially keeps influencing the neighbourhood of the segment and after a very large number of iterations exaggerated lengths of other line segments connected to p can appear.

Topology maintenance The topological structure of the network has to be preserved during the network modification process. When a point in a network is displaced, this may change the way the objects are connected together. Therefore a context-sensitive displacement is proposed: before a point is displaced from its original position p to a new location $p\prime$, a test is performed to detect situations that can lead to changes in the network topology. Network topology here refers to the following connectivity properties:

(P1) No absence of line crossings that are present in the input network;
(P2) No line crossings that are not present in the input network;

Fig. 1 **a** Fitting a line segment $\overline{pq}$ to the nearest schematic direction and **b** finding the new location $p\prime$ for p



(P3) Cyclic order of outgoing connections around any node agrees with the ordering of connections in the input network.

These properties can assure that the resulting schematic network is topologically equivalent to the input network. From a topological view, there may be other interesting properties to be studied, but for this work they are both necessary and sufficient as the most salient properties.

When a point p is moved in the network, a triangle is formed with point p, the new location $p\prime$, and the other endpoint q of the line segment with p. If a topological change occurs when p is moved, this will happen inside the triangle $T(pp\prime q)$. Thus, it is necessary to analyse which situations can occur inside this triangle that may lead to a change in network topology. See Fig. 2. Basically, the triangle $T(pp\prime q)$ can contain

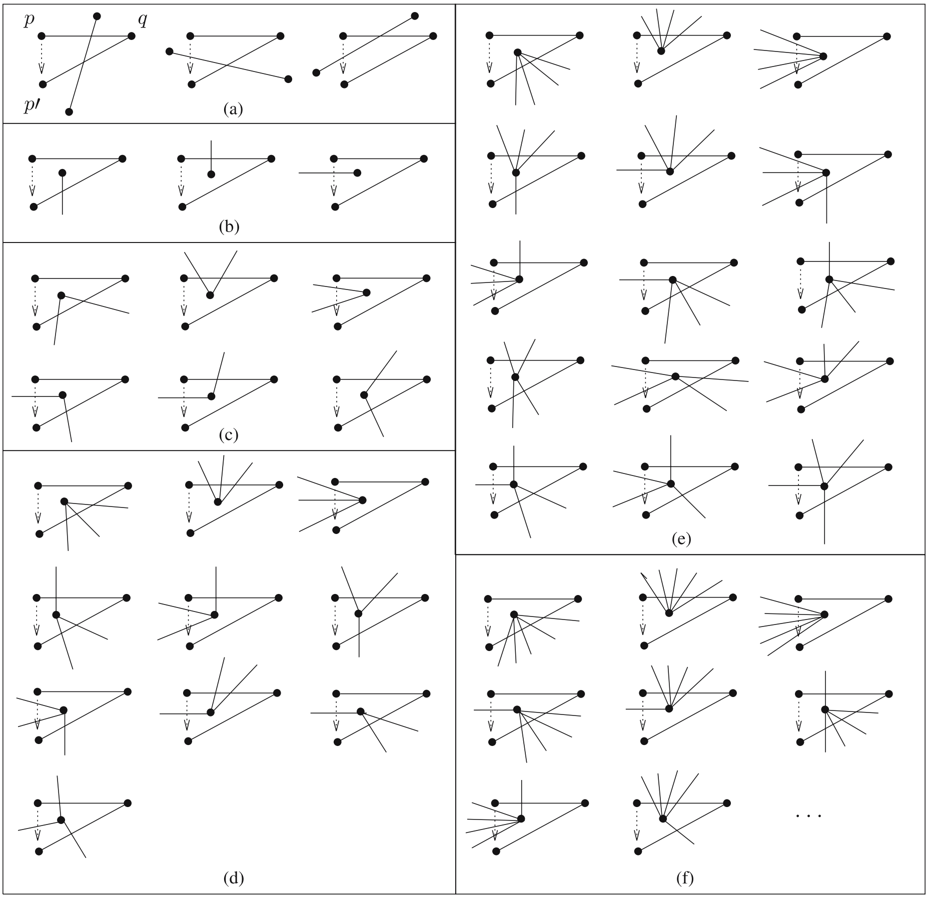


Fig. 2 Some possible situations when moving a point in a network: **a** the formed triangle $T(pp\prime q)$ does not contain any point inside it, but there is a line segment crossing two of its sides; **b**–**f** the triangle $T(pp\prime q)$ contains one point inside it, which as a node can have different degrees, shown here with, respectively, 1, 2, 3, 4 and 5 degrees

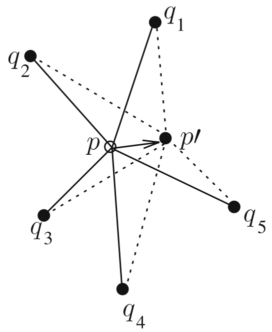


Fig. 3 Point p can be moved to $p\prime$

one or more line segments crossing any of its sides (see Fig. 2a) or it can contain inside it one or more points, which as nodes can have different degrees (Figs. 2b–f).

A topology test is performed to find out whether there is any line segment of the network crossed by the boundary edge $\overline{pp\prime}$ of the triangle $T(pp\prime q) = T$ and whether the triangle T contains any point which can lead to a change in the network topology [3]. If the test finds that topology will change, the move of p must be smaller than the required displacement to $p\prime$ to avoid the change. It is important whether a point moves at all, because the displacement can affect adjacent line segments of the point under analysis, such that the modification process of the network continues to attempt to find better locations for all points. The following three cases can be distinguished in the topology test:

(1) *There is no point inside triangle T and no line segment crossing edge* $\overline{pp\prime}$.
The topology will not change and the move of p to $p\prime$ is allowed. See Fig. 3.

(2) *There is at least one line segment intersecting edge* $\overline{pp\prime}$.
The topology will change, as shown in Figs. 4a and 4b. A new location for $p\prime$ has to be obtained. The new $p\prime$ is the nearest intersection point, for example u, to p plus (or minus, according to the direction of displacement) the minimum distance constraint d measured along $\overline{pp\prime}$. See Fig. 5a.

(3) *There is at least one point v inside triangle T.*
In this case, the topology might change. See Figs. 4c and 4d. To calculate the new location for $p\prime$, a straight line l is defined through v and q, and the

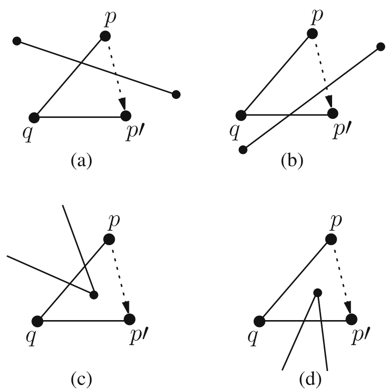
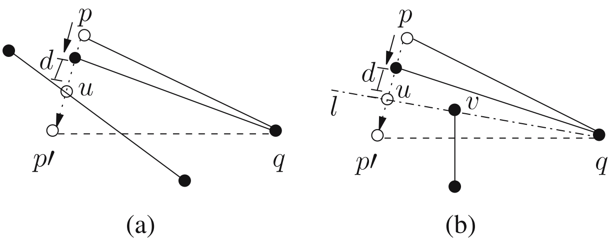


Fig. 4 Situations which lead to a change in network topology, when p is displaced to $p\prime$: **a** there is a line segment crossing $\overline{pq}$ before the displacement of p, but not any more $\overline{p\prime q}$; **b** there is no line segment crossing $\overline{pq}$ before the displacement, but $\overline{p\prime q}$ after it; in **c** and **d** the point inside the triangle is endpoint of one or more segments which also change their crossings after the displacement

Fig. 5 Topology maintenance



intersection point u of l and the edge $\overline{pp\prime}$ of T is calculated. The new $p\prime$ will be the nearest intersection point u to p plus (or minus) the minimum distance constraint d. See Fig. 5b.

Note that, if the point inside triangle T is endpoint of line segments crossing only $\overline{pp\prime}$ (see Fig. 2c), the topology is preserved. However, p is initially at one side of the central element in the triangle and $p\prime$ is at the other side of it. Since changes in the "sidedness" relations among geographic elements related to the network should also be avoided whenever possible, this is a positive feature of this case.

After the topology test has been performed, the newly adjusted location $p\prime$ will be such that case (1) holds for all line segments $\overline{pq}$, i.e., the situation in Fig. 3 holds. In situations in which one or more of cases (2) and (3) occur together the new location for $p\prime$ will also be computed taking the nearest intersection point to the location of p.

Not all line segments and points in the input network need to be tested for geometric intersections of line segments and points inside the triangle, because nodes and segments of the network are stored in a uniform grid which divides the plane into non-overlapping regions. Line segments are distributed among the regions where their centre points lie. The grid spacing is chosen such that the largest line segment is at most the size of a cell, so potentially intersecting line segments can only be found in either the same cell or two neighbouring cells. Therefore, only adjacent regions of the point being analysed need to be considered when searching the data. Spatial partitioning algorithms such as this are sufficient, because points are not moved too far and the line networks considered are spatially well distributed.

3 Quality criteria

Empirically, the iterative process yields good results in the visualization of schematic networks, but it is still necessary to gather information on where and how the resulting network is inadequate. For reducing the enormous number of possible networks, it is appropriate to identify when and between which features visual clashes are likely to occur. A useful measure of the quality or acceptability of a network needs to be unambiguous, objective and easily repeated. Criteria to determine the quality of resulting networks can be found and used to compare various possible solutions of a dataset. Ideally, a measurement of quality should be good at discriminating between potential solutions and quick to calculate [13].

To better understand what the iterative algorithm does, schematic results of various test datasets were examined for different number of iterations. Figure 6 shows a test dataset with the road network of a city. This dataset was selected because it presents many possibilities of topology break and other problems in the schematic

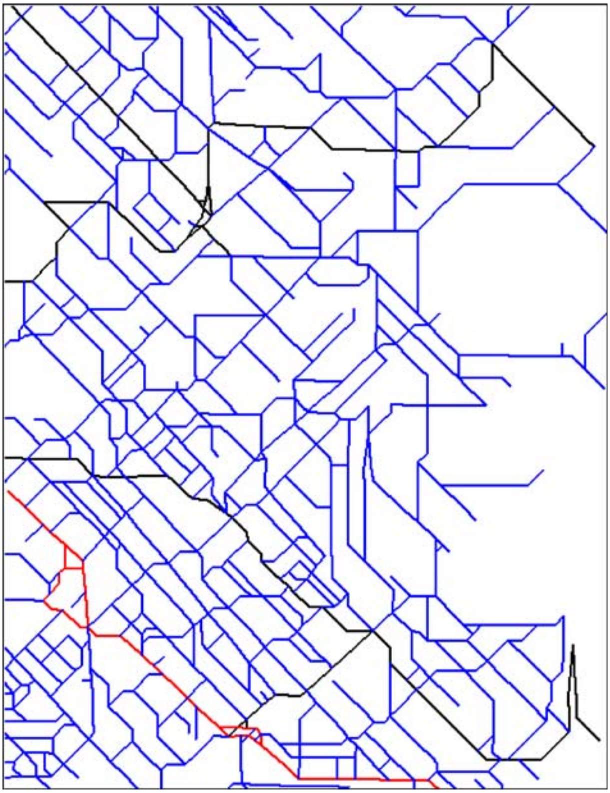
Fig. 6 Original input network

results. Figures 7, 8 and 9 show the resulting schematic networks for 100, 1,000, and 10,000 iterations. The aim of this experiment was to identify in practice the criteria that can characterize an acceptable network, bearing in mind that the schematic networks require that directions and distances are approximately preserved, and that network topology is maintained.

Analysing the resulting networks, it can be seen that the best result is not obtained for large numbers of iterations. Actually, iterating a large number of times appears to be a waste of computational resources since good results do not reach a state like the result presented in Fig. 9. The directions (angles) in Fig. 9 are slightly better than in Fig. 7, but the distances (lengths) are much worse. This happens because the algorithm optimizes for angles, but does not look at the preservation of distances. In order to make a list of aspects which should not occur in the final schematic network, problems were identified in these results. Figure 10 illustrates such problems, which are explained below:

1. For a large number of iterations, e.g., 10,000, the length of line segments becomes generally much longer or much shorter than in the original network. Such large deformations should be avoided, since for recognition purposes schematic line segments should be approximately similar to the original lines.
2. Adjacent line segments should not be displaced to the same position after a transformation, especially when line segments are part of curves and bends.

Fig. 7 After 100 iterations

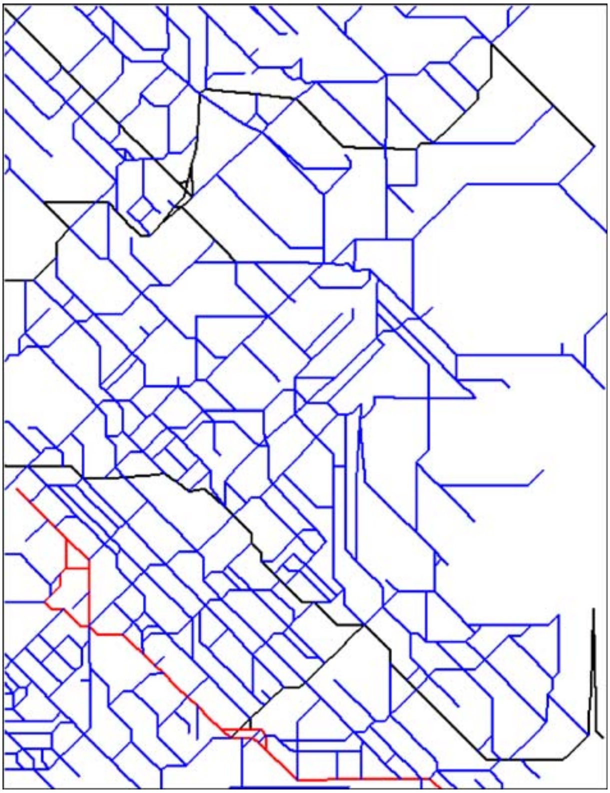


Curves should be treated individually for a more adaptive and realistic design. The user can decide how it is better to transform the original line segments of a curve, taking the neighbourhood of the curve into account.

3. For recognition purposes, spatial relationships like above-below and left-right among network elements should also be kept. The proposed solution for preserving network topology can also identify such spatial relationships involving the sidedness of network elements.
4. This is the problem of losing the natural continuation of lines [18]. Connected small line segments that appear to follow in the same direction should be grouped together and perceived as a single linear object.
5. For readability and aesthetic considerations features in the final network are separated from each other by the spacing constraint, which should not generate new segments.
6. The final network should not contain angles that are too small, keeping lines near to the determined schematic directions as much as possible. This is similar to item 2, but here the objects are not displaced to the same position.

These aspects have to be considered in the choice of an appropriate result quality measurement. On a general level, a result with too many undesirable characteristics is considered unacceptable. The preferred one out of two intermediate results is obviously the one with fewer undesirable situations. Any network which completely preserves topology and approximately preserves directions and distances

Fig. 8 After 1,000 iterations



is acceptable. In order to preserve approximate directions, a network should have minimum error in the direction of lines in comparison to the calculated schematic direction for the line segments. In order to keep approximate distances, the length of schematic line segments should be approximately similar to the original network. The quality criteria for schematic networks will then refer to angular errors of schematic directions and lengths of schematic line segments.

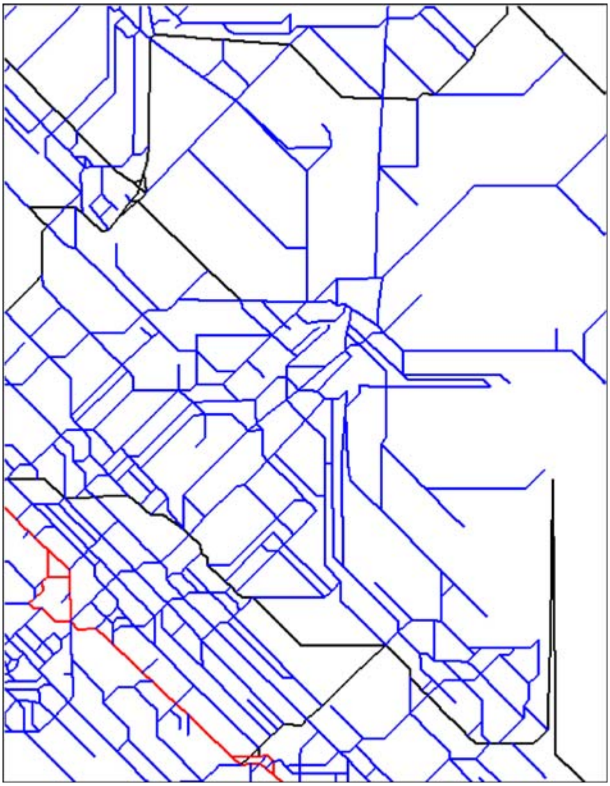
4 Convergence

There are a huge number of possible configurations of schematic networks even for a small dataset. In fact, schematic networks do not have only one optimal solution; instead they allow several workable solutions. But it has not yet been described exactly where iterations lead: Do they converge? If so, do they converge towards a fixed point or something else? When is a resulting network better than another one? What are the stopping criteria? The following experiments answer these questions.

4.1 Analysis

A key requirement in all heuristic search applications is a willingness to experiment. Each iterative algorithm uses specific termination criteria which depend on the

Fig. 9 After 10,000 iterations



application. Since a single solution is not known a priori, it turns out that error bounds can help establish convergence in such cases [14].

In order to evaluate the convergence of the iterative approach, some constraints are selected and their parameters relaxed. In general it is difficult to find out which of the constraints provide the best option for relaxation. In this case the most influential constraints for analysing network quality were chosen. It is expected that in a good schematic result the angles will be closer to schematic directions than in the original network and the lines should be distributed more evenly in space than in the result shown in Fig. 9. The convergence evaluation is carried out using separately the angular error between the final direction of a line segment and the calculated schematic direction for it, and the line segment length. Additionally, the final displacement of an object is also analysed to investigate the amount of displacement according to the progress of the iterations.

Focusing on the study area with a road network shown in Fig. 6, the selected criteria were analysed for 1,000 iterations. Using graphs, the sum of all angular errors of line segments per iteration (Fig. 11), and the sum of all lengths of line segments were plotted versus the number of iterations (Fig. 12). The sum of the amount of displacement of all elements moved in each iteration (Fig. 13) was also plotted.

It can be observed in the graphs of angular error in Fig. 11 and of displacement in Fig. 13 that the schematization improves rapidly during early iterations followed

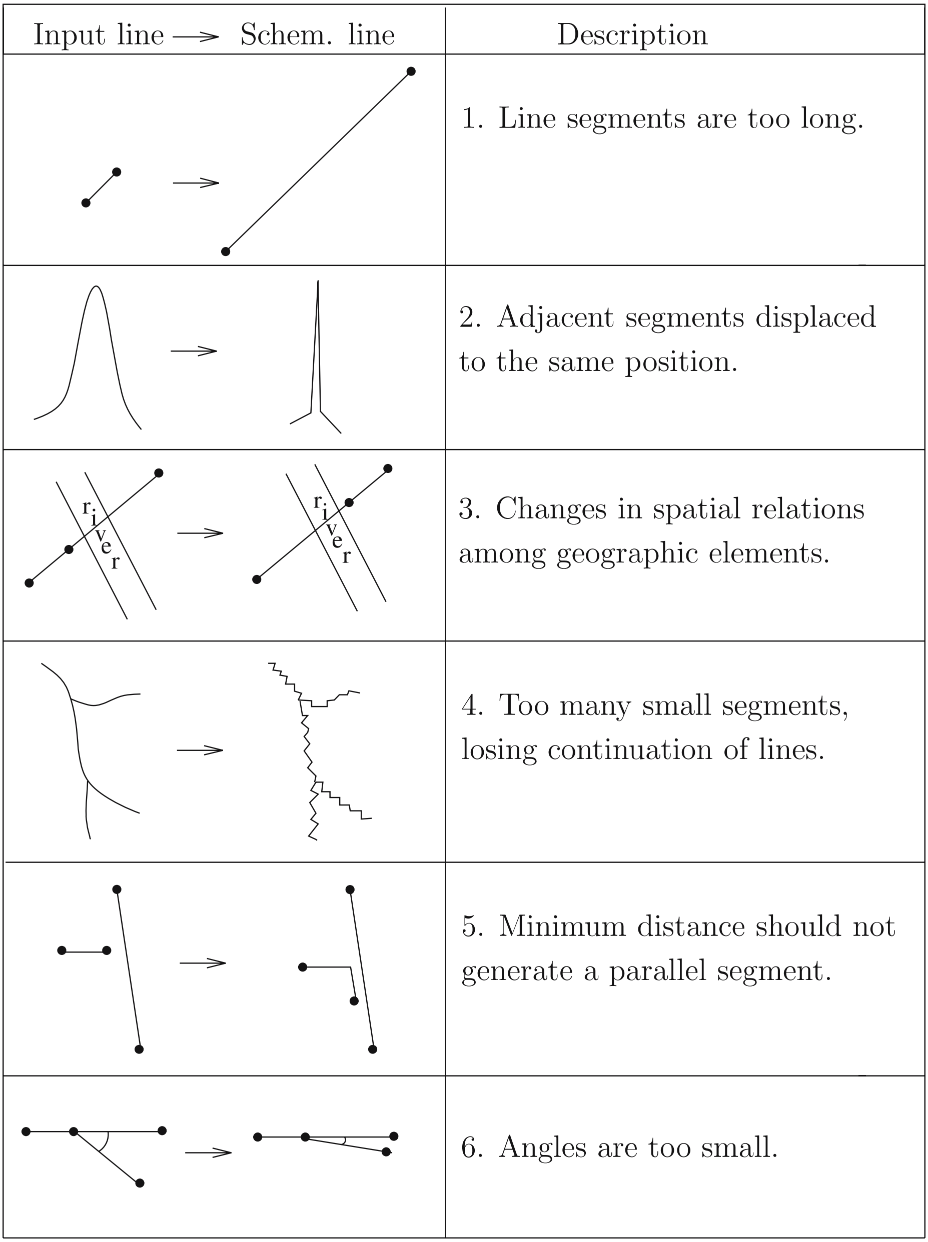


Fig. 10 Situations that should not occur in good schematic results

by a degradation after a large number of iterations. The zero iteration network is far from relaxation, and the 100th iteration is more relaxed, although not completely.

At the beginning of the iteration process, the original location of the routes makes the sum of all angular errors and all displacement vectors large. With the increasing

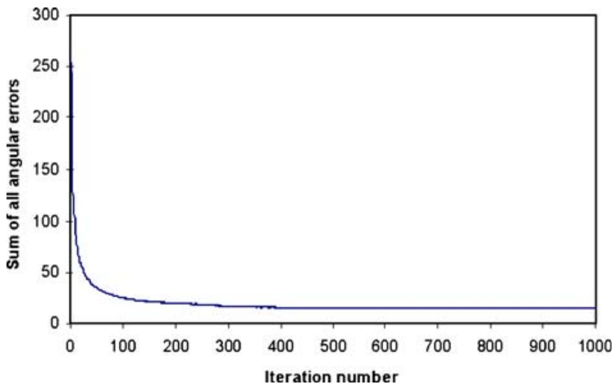


Fig. 11 Sum of the angular error of all line segments

number of iterations, it can be seen that the total angular error and total displacement of the dataset decrease. With the total displacement, it can be said that the extent of changes was also reduced. However, this is not necessarily good to prove that network quality is improving, nor does the decrease in the angular error mean that quality is improving.

Note in Fig. 11 that the angular error after the first 100 iterations is not much higher than that achieved towards the end. Extensive experiments indicated that this measurement may not converge to zero, indicating that there are indeed considerable variations between iterations, which just cancel each other out in the aggregate variable used. Thus, at best one could demand that this value converges in an average value. After 400 iterations, the angular error appears to be measuring a network's proximity to equilibrium. In contrast, the total length of segments increases with the number of iterations (Fig. 12). The total length error indicates that overall network quality does not always improve for a large number of iterations.

When considering the graph of total displacements in Fig. 13, one should note that the curve was not yet completely flat after 100 iterations, thus indicating that the system was not completely relaxed with respect to this criterion. Nevertheless, the value of the error function was changing very slowly after this point such that further iterations resulted in little change in the schematized network.

With this convergence evaluation of the criteria chosen to measure the resulting network quality, it was decided to use the angular error in the stop criterion of the

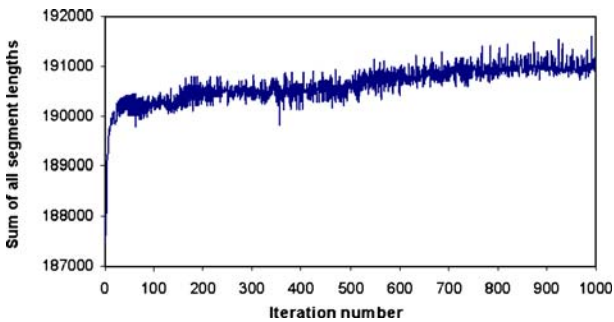


Fig. 12 Sum of the length of all line segments

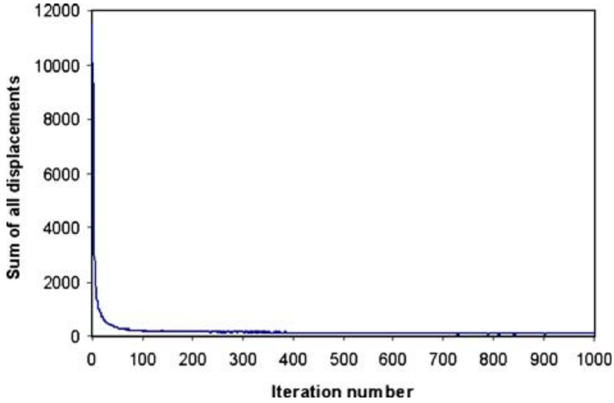


Fig. 13 Sum of the displacement vector of all objects

iterative process. The main goal was to produce network lines in schematic directions, and this parameter is one of the primary requirements for an acceptable network.

4.2 Stopping criteria

A stopping criterion should enable the system to decide when to stop iterations, even though it has not reached a perfect solution and can in fact never know when and if it actually will. In a deterministic steady-state assignment the system is just monitored to stop when changes (according to a pre-defined measure) are smaller than a solution known beforehand [12]. When no solution is previously known, one conventional stopping criterion can be given by a function of the number of iterations. The choice of stopping criteria for a certain application is also a matter of empirical experimentation. The algorithm is iterated a number of times, in order to observe the quality of results and to find out when a result is considered acceptable.

It has been found that 200 iterations can provide an acceptable result for schematic networks. This particular value is also justified by results from other iteration series which did not seem be making any more improvement after it. Thus, the maximum number of iterations was set to be 200 as a simple stopping criterion for the algorithm.

Another termination criterion for the iterative approach can be based on the quality criterion of the angular error of line segments. Considering that in a schematic network the information content of long lines can attract more attention than small line segments, it is reasonable that the angular error of long line segments should have more impact on a stopping criterion. The measurement of angular error is normalized to have a measure that is consistent across different datasets. So, the angular error a of each line segment is divided by the maximum angular error value that can not be exceeded on any network state, i.e., $a_{max} = 22.5$ degrees. This yields the normalized angular error $e = a/a_{max}$. The products of the length l of each straight line segment and its normalized angular error e can be summed and divided by the total length $totl$ of all line segments on the network. Thus, the total angular error is

$$\frac{\sum(l \cdot e)}{totl}.$$

For the maximum tolerable error, a value between 0 and 1 was chosen. Through experimentation, it has been seen that it is enough to stop the process when the angular error is 10% of the total error value, i.e., for reaching an acceptable result in this application 90% of directions should be schematized

$$\frac{\sum(l \cdot e)}{totl} < 10^{-1}.$$

Note that this is not well-justified from a behavioral perspective, because it would imply that designers optimize their behavior so that the expected angular error in the schematization of a line network would be always minimized to 10%. In other applications, it might be that forcing the algorithm to generate averaged behavior does not necessarily lead to good results. In this case, the algorithm reached, in general, 200 iterations before the total angular error was equal to 10%. The value of the total angular error function was changing very slowly at this point, so that further iterations resulted in very little changes in the resulting network.

5 Summary and conclusions

Effectively handling constrained problems is a challenging task [15]. A typical demanding aspect of applying iterative algorithms to solve those problems is the need to evaluate numerous alternative states of the application. Efficient procedures are required to perform tasks such as finding and measuring distances to nearest neighbours of an object and treating overlap between objects.

The main ideas presented here for a schematic network generator have been implemented. Experimentation has been performed with different data sets and constraints. In the line simplification, topological and self-intersection errors may be introduced by using the classic Douglas–Poicker algorithm [8]. Variations on this method, e.g., [17] or other line simplification methods which take topology into account, could be used instead. A new algorithm for preserving topological relations among features while moving points in a network was developed. The algorithm can also be used when schematic networks are generated with other methods, such as the grid fitting method [9].

The iterative strategy for the displacement of points in a network is driven by displacement rules expressed by geometric constraints. Since network constraints can be contradictory, the iterative technique uses the mean value of all required displacements to find a compromise location for a point. The optimization technique used was gradient descent. This technique only minimizes with respect to the last iteration. Although the optimality of the technique is not assured, a schematic result may be good enough without being the best one globally. Ideally, good schematic networks should reflect directions adequately, fulfil design constraints, and be comparable in quality to professionally designed schematic networks. A major advantage of this iterative technique is that a given initial state will always produce the same result, while a system based on other techniques, e.g., simulated annealing, can produce very different results each time it runs. However, which techniques are appropriate for individual situations is likely to become apparent only through experimentation and to vary between application types, scales, and data sources [13].

The convergence of the iterative method was analysed and the angle and length of lines were found to be the most suitable constraints for relaxation. In the evaluation of the quality of schematic results a function was presented to assign a score to each iterated network. The quality function is based mainly on the angular error of the resulting schematic lines.

Schematic networks can still be improved by additional editing of features, according to the application requirements. Graphic software can be used to eliminate undesirable characteristics and include other information, such as colours and labels, suitable to present particular priorities and individual preferences for the final design.

Further work includes a comparative study between different optimization approaches for schematizing networks. An investigation of the worst-case and real time performance of the current iterative algorithm in respect to the size of the network is also planned. Both investigations can offer important complementary information for the use of iterative approaches to schematize networks.

References

1. S. Avelar and L. Hurni. "On the design of schematic transport maps," *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol. 41(3):217–228, 2006.
2. S. Avelar and M. Müller. "Generating topologically correct schematic maps," in *Proc. of 9th International Symposium on Spatial Data Handling*, pp. 4a.28–4a.35, Beijing, 2000.
3. S. Avelar. Schematic maps on demand: design, modeling and visualization, PhD Thesis no.14700, ETH Zurich, Switzerland, 2002.
4. B.P. Bettig. A Graph–based geometric problem solving system for mechanical design and manufacturing, PhD thesis, Department of Mechanical and Aerospace Engineering, Arizona State University, 1999.
5. G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis, (eds.) Graph Drawing. Prentice Hall, Upper Saddle River, NJ, 1999.
6. T. Barkowsky, L.J. Latecki, and K.-K. Richter. "Schematizing maps: simplification of geographic shape by discrete curve evolution," *Lecture Notes in Artificial Intelligence. Spatial Cognition II*, Vol. 1849:41–53, 2000.
7. S. Cabello, M. de Berg, and M. van Kreveld. "Schematization of networks," *Computational Geometry. Theory and Applications*, Vol. 30:223–238, 2005.
8. D.H. Douglas and T.K. Poicker (formerly Peucker). "Algorithms for the reduction of the number of points required to represent a line or its caricature," *The Canadian Cartographer*, Vol. 10(2):112–122, 1973.
9. D. Elroi. "Designing a network line-map schematization software enhacement package," in *Proc. of 8th Ann. ESRI User Conference*, 1988.
10. S.H. Gerez. Algorithms for VLSI Design Automation. Wiley, Chichester, England, 1999.
11. D.H. House and C.J. Kocmoud. "Continuous cartogram construction," in *9th IEEE Visualization Conference*, pp. 197–204, North Carolina, October 1998.
12. T. Kelly and K. Nagel. "Relaxation criteria for iterated traffic simulations," *International Journal of Modern Physics C*, Vol. 9(1):113–132, 1998.
13. M. Lonergan and C.B. Jones. "An iterative displacement method for conflict resolution in map generalization," *Algorithmica*, Vol. 30(2):287–301, 2001.
14. Z.-Q. Luo. "New error bounds and their applications to convergence analysis of iterative algorithms," *Mathematical Programming, Series B*, Vol. 88:341–356, 2000.
15. Z. Michalewicz and D.B. Fogel. How to Solve it: Modern Heuristics. Springer, Berlin, 1998.
16. G. Neyer. Line simplification with restricted orientations, Technical Report 320, Department of Computer Science, ETH Zurich, Switzerland, 1999.
17. A. Saalfeld. "Topologically consistent line simplification with the Douglas–Peucker algorithm," *Cartography and Geographic Information Science*, Vol. 26(1):7–18, 1999.
18. R.C. Thomson and D. E. Richardson. "The 'good continuation' principle of perceptual organization applied to the generalization of road networks," in *Proc. of the 19th International Cartographic Conference, Ottawa*, pp. 1215–1223, ICA, 1999.

Silvania Avelar received a M.Sc. in Computer Science from the Federal University of Minas Gerais, Brazil, and a Ph.D. in Technical Sciences from the Swiss Federal Institute of Technology in Zurich, Switzerland. She is currently a postdoctoral research fellow in the Technology and Society Lab of the Swiss Federal Institute of Materials Testing and Research. Her research domain lies in the fields of geographic information systems, computational geometry and remote sensing. She is particularly interested in automated cartographic design and environmental change issues.