

TOPOLOGICAL CONSTRAINTS, ACTIONS AND REFLEXES FOR GENERALIZATION BY OPTIMIZATION

Jean-Luc Monnot, Paul Hardy, & Dan Lee

ESRI, Redlands, California

jmonnot@esricartonet.com, phardy@esri.com, dlee@esri.com

KEYWORDS: Generalization, Generalisation, Optimization, Constraint, GIS, Contextual analysis

ABSTRACT

Generalization in the digital mapping and GIS world is the task of deriving smaller scale or lower resolution products from over-detailed spatial data. Optimization is one available mechanism for handling the often conflicting constraints involved in contextual generalization, where relationships with neighboring features are paramount. This paper studies the involvement of topological relationships in such constraints, and in the corresponding actions and reflexes which are invoked during optimization. It is based on investigations carried out while researching the development of an optimization engine as an extension to a commodity GIS (ArcGIS from ESRI).

1 INTRODUCTION

A previous ICA workshop paper [Monnot, Hardy, Lee 2006] gave an overview of a research project underway presently at ESRI to implement an optimization approach to constraint-based generalization within a commodity GIS (ArcGIS). The current state of this research and its implementation ('the Optimizer') is described in an extended paper of the same title by the same authors to be presented at the main ICA 2007 Moscow conference [Monnot, Hardy, Lee 2007]. This workshop paper expands on aspects of the Optimizer relating to preserving topological relationships during generalization by optimization.

1.1 Topology in Generalization

Topology is the branch of geometric mathematics concerned with order, contiguity, and relative position, rather than actual linear dimensions. As such, it is used to refer to the continuity of spatial properties, such as connectivity or adjacency, which are unchanged after smooth distortion. Topological relationships as used in GIS have long been analyzed and classified [Egenhofer & Herring 1990; Clementini et al 1993]. Building and manipulating topological relationships in a GIS is also now standard practice [Hoel, Menon & Morehouse 2003].

Topological relationships are vitally important to good contextual generalization [Mackaness & Edwards 2002], but few if any existing systems are fully aware of topology during their generalization operations. Examples of generalization operations which can easily break topological relationships include simplification, elimination, aggregation, or displacement. This paper concentrates on displacement and simplification, but the techniques described are applicable to all kinds of generalization, including ones involving combinations of operators.

2 OPTIMIZATION

2.1 Optimization Using Constraint-Action Rules

In the Optimizer approach, a set of rules is defined, one rule for each constraint. Each rule contains a satisfaction function, measuring the degree of violation of the constraint, and one or more actions which should improve the situation if the constraint is violated. An Optimizer kernel then has the responsibility of evaluating local and global satisfaction, and applying actions to appropriate features to improve the situation. In real generalization scenarios, it is often not possible to avoid some violation of constraints, and the goal of the Optimizer is therefore to maximize the overall satisfaction.

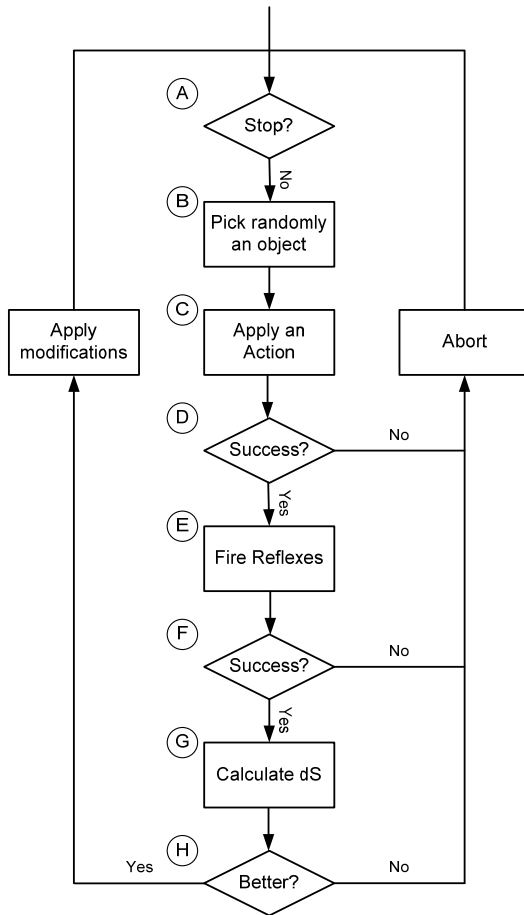
The prototype Optimizer is implemented within the geoprocessing framework of a commodity GIS (ESRI ArcGIS). Optimizer rules, specifying constraints, actions and their associated parameters are now defined in an XML structure, to provide flexibility and extensibility.

The Optimizer uses a modified Simulated Annealing optimization technique [Kirkpatrick 1983], in which a conceptual 'temperature' is gradually reduced. While the temperature is higher, the system is more likely to

accept the results of some action with negative ΔS , where ΔS is the difference between the current and previous satisfaction values. This is to avoid being trapped by a local maximum, and hence to explore the state tree to find solutions where it has been necessary to get slightly worse in order to get much better.

2.2 Optimization Strategy

The Optimizer uses the following strategy (Fig. 1) in order to achieve its task:



- **A:** The stop criteria may include a maximum number of iterations, a global satisfaction value to reach, or other programmable decision.
- **B:** The random process is oriented for picking low satisfaction objects. Other strategies are being studied, such as weighting to prefer nearby objects.
- **C:** Apply one of the involved actions. Actions act in the research space by modifying an attribute value or a geometry.
- **D:** An action may fail to proceed.
- **E:** Reflexes are similar to database triggers. They are responsible for maintaining data coherence (polygon area depending on segments positions) or avoiding forbidden states (topological relationships).
- **F:** Reflexes may reject some resultant data situations (topology error, attribute values...)
- **G:** Based on the modifications, the change in global Satisfaction computed.
- **H:** Using a simulated annealing strategy, the Optimizer decides if the modifications are to be kept or aborted.

Fig. 1 – Optimization Strategy

2.3 Optimization Example – Point Displacement

The previous paper showed a point displacement example, where point symbols are displaced using two simple constraints (should not be closer than given distance; should not be far from original position), each with one action (move away from overlap; move back towards starting position). Starting with the test data shown in Fig. 2, we get the results shown in Fig. 3. The line features were not involved in the process at this stage.

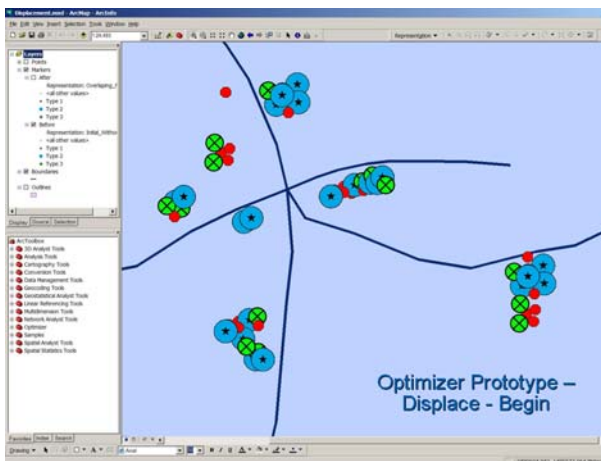


Fig. 2 – Initial state

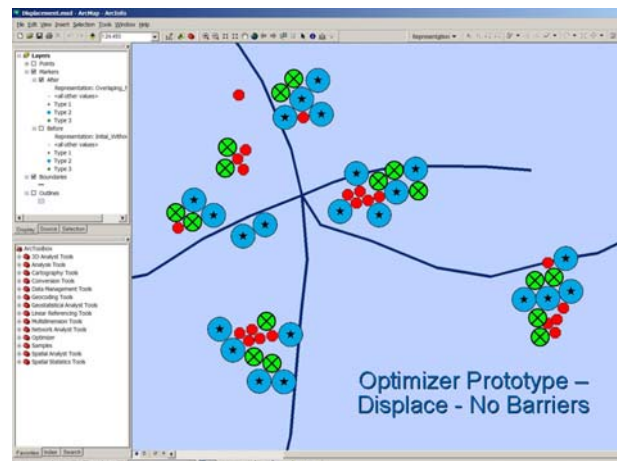


Fig. 3 – Optimized, without barriers

2.4 Optimization Example – Point Displacement with Barriers

In order to treat the line features as barriers which the point symbols must not overlap, a ‘strict constraint’ or prohibition ‘reflex’ was introduced to forbid any state violating this requirement, giving the results shown in Fig. 4. The graph of satisfaction is shown in Fig. 5 for both constraints (blue and green) and for the system satisfaction (red) for the run that produced the result shown in Fig. 4.

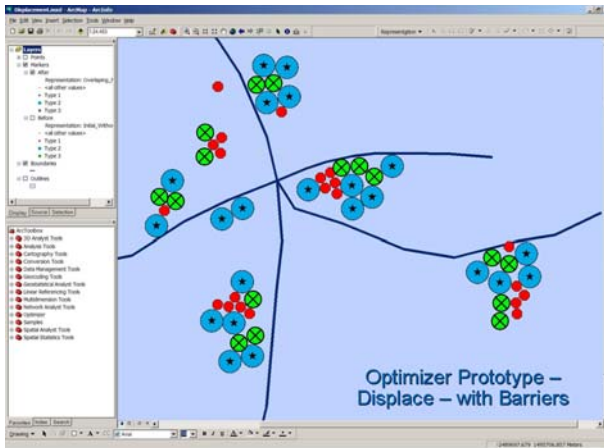


Fig. 4 – Optimized, with barrier constraint

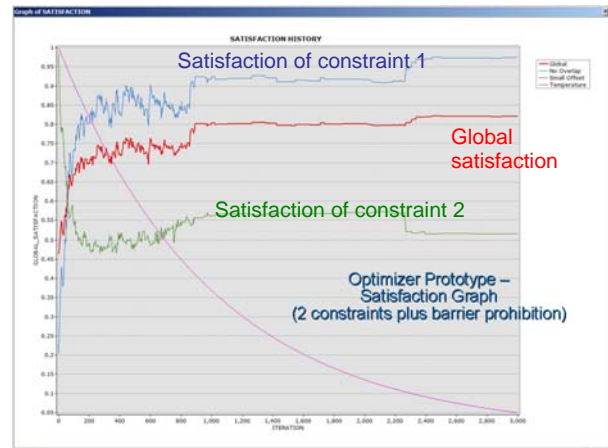


Fig. 5 – Satisfaction evolution against temperature

We will return to point displacement later, but first we will take a different example from various generalization use cases for the Optimizer prototype – that of line and polygon simplification. Much research has been done in the past on algorithms for geometric simplification of lines and polygon boundaries, and ArcGIS has already implemented bend_simplify and point_remove algorithms. At ArcGIS 9.2, these tools have gained capabilities for retaining shared edge topology within the feature class being simplified [Lee 2004].

Our aim here is not to replace the previous deterministic approaches, but firstly to exercise the Optimizer engine in a different manner, and secondly, in doing so, to investigate and gain experience in topology constraints that could form the basis for later production generalization use cases. In particular, we are addressing topological relationships between features of different classes.

3 TOPOLOGY AWARENESS

3.1 Topology Aware Geometry Cache

Since the initial workshop paper, the infrastructure of the Optimizer has been extended to include a topology-aware geometry cache. This breaks down input features (lines, points and polygons) into nodes and segments, and can also hold other derived geometry types, such as outlines and triangles as shown in Fig.6.

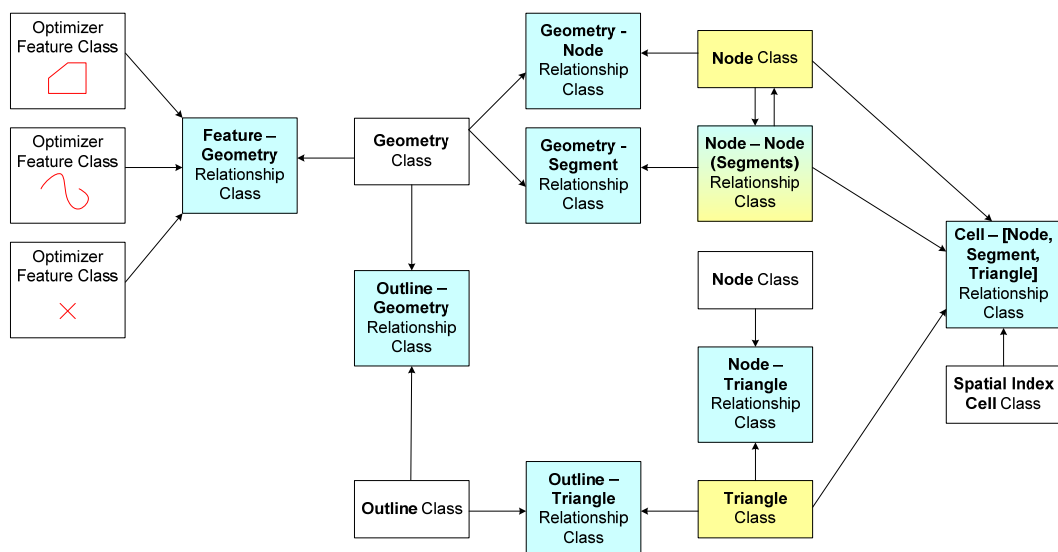


Fig. 6 - Schema for Optimizer cache

3.2 Shared-Boundary Topology

For the case of tessellating polygons (as in Fig. 7), loading them into the cache with the ‘topology awareness’ enabled detects the common boundaries, and stores them as shared segments referenced by both polygons. When the Optimizer is then used to simplify the boundaries, the constraints and actions operate on the segments rather than on the complete polygon features. The result is that shared boundary topology is preserved (Fig. 8). If the ‘topology awareness’ is disabled, or if the polygons are processed one at a time, then polygon overlaps can result, as in Fig. 9.

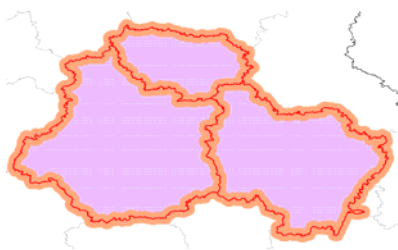


Fig. 7 - Initial data – polygons touch along common boundaries

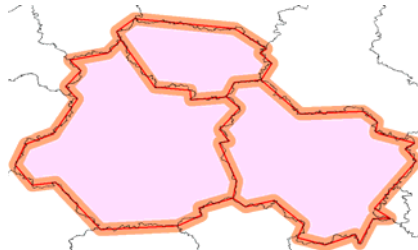


Fig. 8 - With topology awareness – clean result



Fig. 9 - Without topology awareness – overlaps created

The constraint and actions used for this case can be extended in various ways. Adding further constraints can take into account other topological relationships, such as having a rule that simplification should favor the left or right side of a line. This can be important for bathymetric contours, to preserve shallow water information for safety of navigation. Similarly other constraints can apply preferences for particular directions, leading to schematic simplifications like the archetypal London Underground map.

3.3 Point and Line Topological Relationships

Topological relationships also exist between polygons and points. In the examples below, the white circle symbols represent cities, which lie within an administrative region (the black polygon). When the segments are optimized by applying constraints to remove points which are not representative (using a sideways tolerance indicated by the width of the broad orange band), then the resultant simpler polygon boundary (red line) may now pass inside the point, breaking the topological relationship, as in Fig. 10. However, we can introduce a topological reflex to forbid this state, and the optimizer will then backtrack and find another solution which does not violate this requirement (Fig. 11).

Cartographically however, this solution is not ideal, because the white symbol overlaps the red boundary, giving the idea that the city extends outside the region. In order to improve legibility, a ‘minimum distance’ reflex is introduced (Fig. 12), which again forbids any system state of this kind, causing the optimization to backtrack again and to find a solution which preserves some space around the point city symbol.

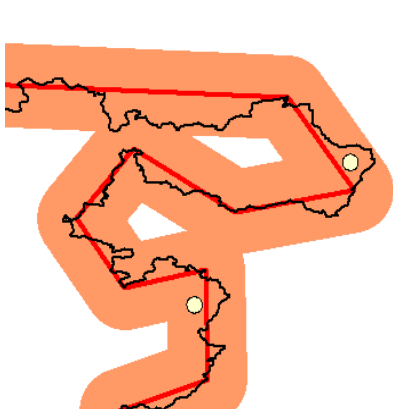


Fig. 10 - Without topology awareness – Point now outside polygon

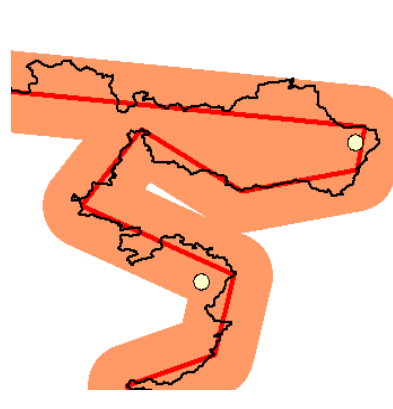


Fig. 11 - With topology awareness – Point remains inside polygon

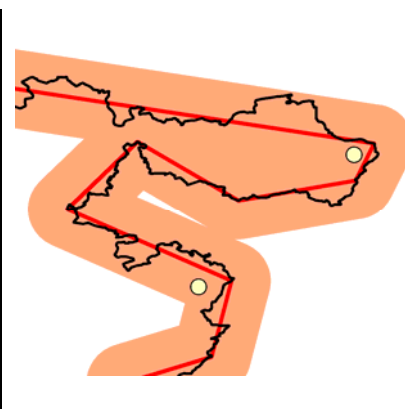


Fig. 12 - Minimum Distance Reflex – Point clear of polygon boundary

3.4 Point Displacement with Line Topology Awareness and Clarity Enforcement

Returning to the original use case – that of point displacement – we can re-use the same techniques, to provide cartographic clarity during point dispersal with barriers. Fig. 13 shows the application of a barrier reflex to avoid any points that were inside the pink polygon ending up outside (or vice versa). However their symbols, not being infinitely small, do overlap the polygon and stray beyond. If the ‘minimum distance’ reflex is introduced, then this can be avoided, and an ‘air gap’ introduced to make the boundary clear, as in Fig. 14.

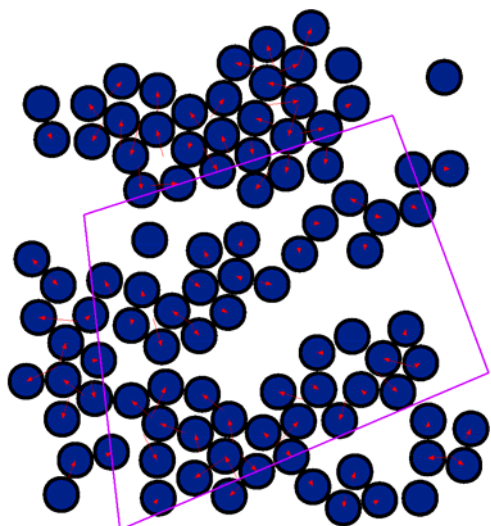


Fig. 13 - Point Displacement with topology awareness. No point center crossed the barrier during optimization.

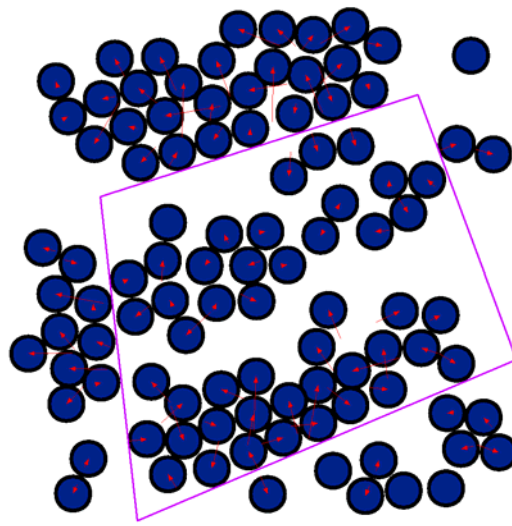


Fig. 14 - Topology awareness AND minimum distance reflex. No symbol overlaps the barrier.

3.5 Marker Shapes

Recent development of the Optimizer framework has included awareness of the actual extent and shape of marker symbols. This allows constraints and actions to make decisions based on true symbol extents, which is particularly important for marker displacement in congested regions. Fig. 15 shows some sample point data of multiple feature types, which are symbolized differently (triangle, hexagon, cross, and so on). Applying similar constraints and actions to the previous point displacement case, but this time not just treating symbols as circles, allows the Optimizer to generate the result of Fig. 16, showing marker symbols exactly touching in clean topological relationships.

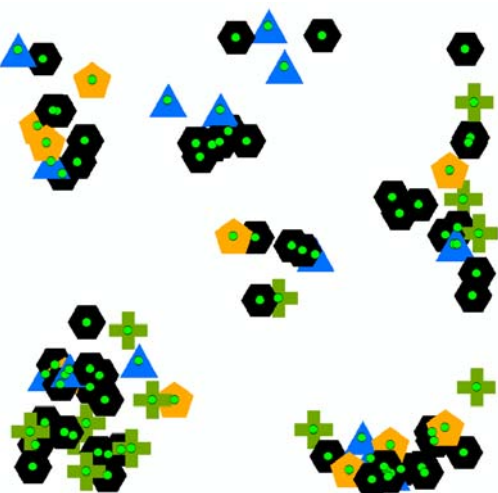


Fig. 15 – Initial state, with overlapping marker symbols

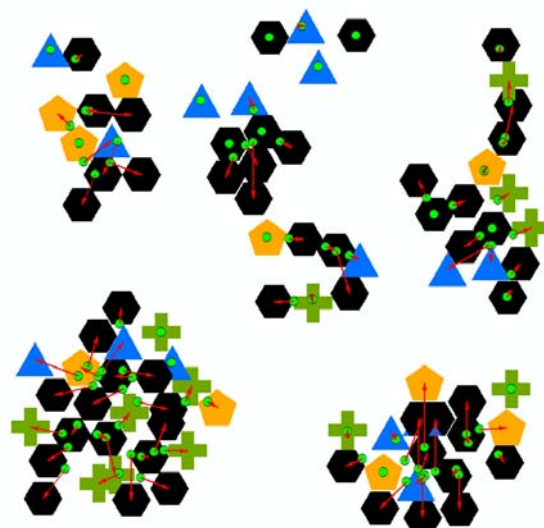


Fig. 16 – Optimized state, showing dispersal vectors

Note that there are still some overlaps in this example, where the density was particularly high, where the optimal solution balanced the conflicting constraints. The overlap was deemed less bad than to displace the marker too far from its original position. Note also that the same constraints can be used for displacing markers towards each other (snapping in), which can be an important technique in generalization (snap nearby buildings to the edge of road casings).

4 CONCLUSION

The Optimizer framework has been extended to have knowledge of topology and topological relationships in its feature caching mechanisms. This knowledge has then been applied by constraints, actions and reflexes, in order to preserve topological integrity in the generalized data. The research is continuing with more complex topological relationships, such as those involved in displacing buildings away from roads, while preserving adjacency and avoiding overlaps.

NOTE

This paper is a forward-looking research document, and the capabilities it describes are still under development and review. As such, it should not be interpreted as a commitment by ESRI to provide specific capabilities in future software product releases.

REFERENCES

- Clementini, E., Di Felice, P., and van Oosterom, E “A small set of formal topological relationships suitable for end-user interaction”. Proceedings of the Third International Symposium on Large Spatial Databases, Singapore, 1993.
- Egenhofer, M. and Herring, J. “Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases”. Technical Report, Department of Surveying Engineering, University of Maine, 1990.
- Hoel, E., Menon, S., Morehouse, S., “Building a Robust Relational Implementation of Topology. 8th International Symposium on Spatial and Temporal Databases”. Santorini, Greece, 2003, available at http://downloads.esri.com/support/whitepapers/ao_/BuildingRobustTopologies.pdf (seen 2007-07-23).
- Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. "Optimization by Simulated Annealing", Science 220, 671-680, 1983.
- Mackaness, W & Edwards, G, “The Importance of Modelling Pattern and Structure in Automated Map Generalisation”, Joint Workshop on Multi-Scale Representations of Spatial Data, Ottawa, July 2002.
- Monnot, J; Hardy, P; and Lee, D “An Optimization Approach to Constraint-Based Generalization in A Commodity GIS Framework”, ICA Workshop on Generalisation and Multiple Representation, 25 June 2006, Vancouver, WA, USA
- Monnot, J; Hardy, P; and Lee, D “An Optimization Approach to Constraint-Based Generalization in A Commodity GIS Framework”, International Cartographic Conference, July 2007, Moscow, Russia
- Lee, D “Geographic and Cartographic Contexts in Generalization”, ICA Workshop on Generalisation, Leicester UK, 2004.

[Issue 5, of 2007-07-24]