Computational Geometry 47 (2014) 381–406

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

On d-regular schematization of embedded paths ☆

Daniel Delling [a], Andreas Gemsa [b,*], Martin Nöllenburg [b], Thomas Pajor [b], Ignaz Rutter [b]

[a] Microsoft Research Silicon Valley, Mountain View, CA, USA
[b] Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 21 December 2012
Accepted 2 October 2013
Available online 11 October 2013
Communicated by M. de Berg

Keywords:
Path schematization
NP-hardness
Mixed integer linear programming
Dynamic programming
Graph drawing

ABSTRACT

Motivated by drawing route sketches, we consider the d-regular path schematization problem. For this problem we are given an embedded path P (e.g., a route in a road network) and a positive integer d. The goal is to find a d-*schematized* embedding of P in which the orthogonal order of all vertices in the input is preserved and in which every edge has a direction that is an integer multiple of $(90/d)°$. We show that deciding whether a path can be d-schematized is NP-complete for any positive integer d.
Despite the NP-hardness of the problem we still want to be able to generate route sketches and thus need to solve the d-regular path schematization problem. We explore two different algorithmic approaches, both of which consider two additional quality constraints: We require that every edge is drawn with a user-specified minimum length and we want to maximize the number of edges that are drawn with their preferred direction. The first algorithmic approach restricts the input paths to be axis-monotone. We show that there exists a polynomial-time algorithm that solves the d-regular path schematization problem for this restricted class of embedded paths. We extend this approach by a heuristic such that it can handle arbitrary simple paths. However, for the second step we cannot guarantee that the orthogonal order of the input embedding is maintained. The second approach is a formulation of the d-regular path schematization problem as a mixed integer linear program. Finally, we give an experimental evaluation which shows that both approaches generate reasonable route sketches for real-world data.

1. Introduction

Angular or $\mathcal{C}$-oriented schematizations of graphs refer to a class of graph drawings, in which the admissible edge directions are limited to a given set $\mathcal{C}$ of (usually evenly spaced) directions. This includes the well-known class of orthogonal drawings and extends more generally to k-linear drawings, e.g., octilinear metro maps. Applications of schematic drawings can be found in various domains such as cartography, VLSI layout, and information visualization.

In many schematization scenarios the input is not just a graph but a graph with an initial drawing that has to be schematized according to the given set of directions. This is the case, e.g., in cartography, where the geographic positions

☆ Preliminary versions of the results presented in this paper have appeared at the 12th Scandinavian Symposium and Workshop on Algorithm Theory (SWAT'10) [1], and at the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'11) [2]. Part of this work was done within GRADR – EUROGIGA project No. 10-EuroGIGA-OP-003. A.G. and M.N. are financially supported by the *Concept for the Future* of KIT within the framework of the German Excellence Initiative.

* Corresponding author.
E-mail addresses: dadellin@microsoft.com (D. Delling), gemsa@kit.edu (A. Gemsa), noellenburg@kit.edu (M. Nöllenburg), pajor@kit.edu (T. Pajor), rutter@kit.edu (I. Rutter).

0925-7721/$ – see front matter
http://dx.doi.org/10.1016/j.comgeo.2013.10.002

of network vertices and edges are given [3], in sketch-based graph drawing, where a sketch of a drawing is given and the task is to improve or schematize that sketch [4], or in dynamic graph drawing, where each drawing in a sequence of drawings must be similar to its predecessor [5]. For such a redrawing task it is crucial that the mental map [6] of the user is preserved, i.e., the output drawing must be as similar as possible to the input. Misue et al. [6] suggested preserving the *orthogonal order* of the input drawing as a simple criterion for maintaining a set of basic spatial properties of the input, namely the relative above/below and left/right positions of all pairs of input vertices. The orthogonal order has been used successfully as a means for maintaining the mental map [7–9,1].

The motivation behind the work presented here is the visualization of routes in road networks as sketches for driving directions. An important property of a route sketch is that it focuses on road changes and important landmarks rather than exact geography and distances. Typically the start and destination lie in populated areas that are locally reached via a sequence of relatively short road segments. On the other hand, the majority of the route typically consists of long highway segments with no or only few road changes. This property makes it difficult to display driving directions for the whole route in a single fixed-scale map since some areas require much larger scales than others in order to be readable. The strength of route sketches for this purpose is that they are not drawn to scale but rather use space proportionally to the route's complexity.

Related work. Geometrically, we can consider a route to be an embedded path in the plane. The simplification of paths (or polylines) in cartography is well studied and the classic line simplification algorithm by Douglas and Peucker [10] is one of the most popular methods. Two more recent algorithms were proposed for $\mathcal{C}$-oriented line simplification [11,12]. These line simplification algorithms, however, are not well suited for drawing route sketches since they keep the positions of the input points fixed or within small local regions around the input points and thus edge lengths are more or less fixed. On the other hand, Agrawala and Stolte [13] presented a system called *LineDrive* that uses heuristic methods based on simulated annealing to draw route sketches. A problem similar to drawing route sketches is generating destination maps, where given a destination (e.g. a restaurant) the aim is to produce a schematized map that helps anyone within a given region around the destination to reach it. Kopf et al. consider this problem [14] and give a heuristic algorithm based on the continuation method. The methods of both of these approaches allow distortion of edge lengths and angles. However, they do not restrict the set of edge directions and do not give hard quality guarantees for the mental map such as the preservation of the orthogonal order.

A graph drawing problem that has similar constraints as drawing route sketches is the metro-map layout problem, in which an embedded graph is to be redrawn octilinearly. The problem is known to be NP-hard [15] but it can be solved successfully in practice by mixed integer linear programming [16]. The existing methods, covered in a survey by Wolff [17], do aim to keep the mental map of the input, but no strict criterion like the orthogonal order is applied. Brandes and Pampel [9] studied the path schematization problem in the presence of orthogonal order constraints in order to preserve the mental map. They showed that deciding whether a rectilinear schematization exists that preserves the orthogonal order of the input path is NP-hard. They also showed that schematizing a path using arbitrarily oriented unit-length edges is NP-hard. Recently, the authors have published more simplified version of both proofs [18]. In [19] Speckmann and Verbeek consider the problem of finding a rectilinear, homotopic schematization of a collection of paths with a minimum number of path segments. The problem turns out to be NP-hard, but the authors give an approximation algorithm. Verbeek extends the approach in [20] to $\mathcal{C}$-oriented paths, where the path segments in the schematization can have only orientations that are contained in $\mathcal{C}$. However, besides the homotopy requirement, no special emphasis is placed on maintaining the user's mental map. Another type of $\mathcal{C}$-oriented schematization that has applications in generating schematized maps is studied by Buchin et al. [21]. There, the authors give an algorithm that finds a $\mathcal{C}$-oriented schematization of a polygon, where the polygon's edges must have directions that are contained in $\mathcal{C}$ and where the schematized polygon preserves the area of the input polygon.

Contributions. Motivated by drawing route sketches, we investigate the problem of generating $\mathcal{C}$-oriented orthogonal-order preserving drawings. We prove that deciding whether a $\mathcal{C}$-oriented orthogonal-order preserving drawing of an embedded input path exists is NP-complete, even if the path is simple. This is true for every *d-regular* set $\mathcal{C}$ of directions that have angles that are integer multiples of $90°/d$ for any integer d. This extends the result of Brandes and Pampel [9,18] who showed NP-completeness for the case $d = 1$. However, their proof relies on the absence of diagonal edges and hence does not extend to larger values of d. We show the NP-completeness in the octilinear case $d = 2$ in Section 3 and, subsequently in Section 4, how this result extends to the general d-regular case for $d > 2$. In Section 5 we show that if we restrict the input to *axis-monotone* paths, i.e., x- or y-monotone paths, the d-regular path schematization problem can be solved in polynomial time. We present an efficient algorithm that finds, for a given embedded axis-monotone path, an embedding that uses only d-regular directions and maintains the orthogonal order of the input path. The algorithm maximizes the number of edges that are embedded with their preferred direction, i.e., the d-regular direction that is closest to the direction in the input embedding. We also heuristically extend this approach to non-monotone paths. In Section 6 we design a mixed integer linear program (MIP) for solving the d-regular path schematization problem for arbitrary paths. Finally, in Section 7 we give an experimental evaluation of both the MIP approach and the heuristic simple path schematization algorithm. The source code of our proof-of-concept implementation in C++ is made available at http://i11www.iti.kit.edu/projects/routesketch. We summarize our results in Section 8.