A Simple Proof for the NP-Hardness of Edge Labeling

Alexander Wolff*

September 15, 2000

Ernst-Moritz-Arndt-Universität Greifswald
Institut für Mathematik und Informatik
Preprint-Reihe Mathematik 11/2000

Abstract

Kakoulis and Tollis have shown that labeling the edges of a graph drawing with axis-parallel rectangles is NP-hard [KT97b]. In this note we simplify their proof by reducing from planar 3-SAT instead of 3-SAT.

Keywords

Complexity theory, graph drawing, computational geometry, label placement.

*Ernst Moritz Arndt University Greifswald, Institute of Mathematics and Computer Science, Jahnstr. 15a, D-17487 Greifswald, Germany. Email: `awolff@mail.uni-greifswald.de`

1 Introduction

Label placement is one of the key tasks in the process of information visualization. In diagrams, maps, technical or graph drawings, features like points, lines, and polygons must be labeled to convey information. The interest in algorithms that automate this task has increased with the advance in type-setting technology and the amount of information to be visualized. Due to the computational complexity of the label-placement problem, cartographers, graph drawers, and computational geometers have suggested numerous approaches, such as expert systems [AF84, DF89], zero-one integer programming [Zor90], approximation algorithms [FW91, DMM$^+$97, WW97, ZP99], simulated annealing [CMS95] and force-driven algorithms [Hir82] to name only a few. An extensive bibliography about label placement can be found at [WS96]. The ACM Computational Geometry Impact Task Force report [C$^+$96] denotes label placement as an important research area. Manually labeling a map, for example, is a tedious task that is estimated to take 50 % of total map production time [Mor80].

Since graph drawing has emerged as an important field in information visualization, some work, especially by Kakoulis and Tollis, has been directed to the problem of attaching labels to the elements of a graph drawing [KT97a, KT98a, DKMT98]. Apart from these specialized algorithms, other, more general heuristics can also be applied to labeling the nodes and edges of graph drawings [ECMS97, KT98b, WWKS00]. There is a good survey of label placement algorithms with applications to graph drawing [NH00].

Kakoulis and Tollis have also investigated the computational complexity of a specific problem, namely labeling the edges of a graph drawing [KT97b]. They show that what they call the *discrete admissible edge labeling problem* is NP-complete. In their formulation they restrict edge labeling to instances where each edge of the graph drawing has a finite set of non-intersecting label candidates of the same size. In their proof, a reduction from 3-SAT, they use two to four axis-parallel rectangular label candidates per edge.

The reason why we give a new proof of the NP hardness of edge labeling is that the reasoning of [KT97b] can be simplified considerably by reducing from planar 3-SAT instead of 3-SAT. We note that there already is a long history of NP-hardness proofs for the problem of labeling *points* with axis-parallel rectangular labels [FW91, KR92, MS91, IL97, vKSW99] or with circular labels [SW00] in various labeling models.

2 NP-hardness

As mentioned above, our proof of the NP-hardness of edge labeling is by reduction from planar 3-SAT. Lichtenstein showed that this restriction of the classical satisfiability problem is NP-hard [Lic82]. For a *planar* 3-SAT formula there is always a way to arrange nodes corresponding to the variables on a straight line in the plane and to connect these nodes by *non-intersecting* three-legged clauses, see [KR92, Figure 5]. Not having to worry about intersections simplifies the reduction and makes it possible to put more restrictions than Kakoulis and Tollis on the edge labeling instance to which a Boolean formula is reduced:

Theorem 1 *Given the drawing of a graph and for each edge e of the graph a finite set C_e of label candidates, it is NP-complete to decide whether all edges can be labeled such that no two labels intersect.*

This is true even if all vertices in the graph drawing have integer coordinates, the edges joining these vertices are all axis-parallel, are either of length 2 or 4, and have either two or three open unit square label candidates which touch the edge they label.

Proof. Clearly the problem is in $\mathcal{NP}$ since we can guess a possible solution of the edge-labeling problem with non-zero probability and then check in polynomial time whether no two of the chosen labels intersect.

Our proof of the NP-hardness of edge labeling follows Knuth and Raghunathan's proof of the NP-hardness of the Metafont labeling problem [KR92]. The same scheme has already been used in [vKSW99, SW99]. We encode a Boolean formula ϕ of planar 3-SAT type by a graph drawing such that all of its edges can be labeled if and only if ϕ is satisfiable. We encode the variables and clauses of ϕ by the gadgets depicted in Figures 1 and 2. In the graph drawing that we construct all graph vertices have integer coordinates. In addition, all graph edges are axis-parallel, of length 2 and have two label candidates, except one special graph edge per clause gadget that has length 4 and three label candidates. All label candidates are axis-parallel (open) unit squares. Each label candidate has an edge that lies completely on a graph edge. The center of this label edge coincides with the center of the graph edge for all except the special clause edges. In order to make the presentation more easily accessible, we have slightly reduced the size of the label candidates with respect to the length of the graph edges in our figures.

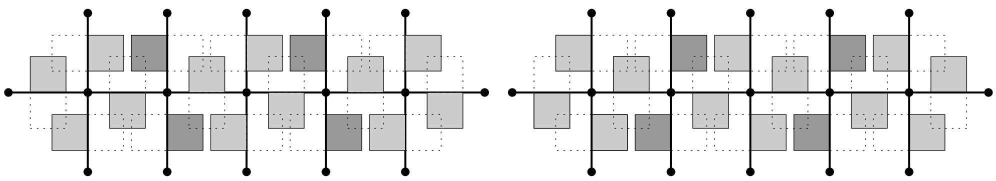

Figure 1: Encoding the values of Boolean variables: *true* left, *false* right.

The basic building block of a variable gadget consists of four edges that form an axis-parallel cross. It is analogous to the "variable block" that Kakoulis and Tollis use in their proof. However, we repeat this block several times on a horizontal line to reach all clauses in which a literal of the variable under consideration occurs. The construction guarantees that there are still only two legal edge labelings of a variable gadget, i.e. label placements where no two labels intersect, see Figure 1. (The difference in shading of the labels is only meant to stress the underlying patterns.) If the leftmost (horizontal) edge of a variable gadget is labeled upwards, this corresponds to the variable being set to *true*, *false* otherwise.

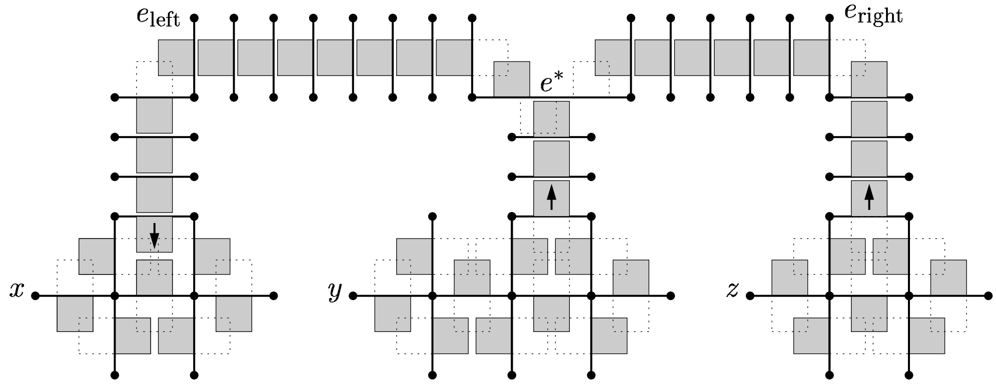

Figure 2: Encoding the clause of a Boolean formula; here $\overline{x} \vee y \vee \overline{z}$

Our clause gadgets resemble three-legged combs, see Figure 2. The vertical legs consist of horizontal edges at unit distance. This makes sure that these edges are either all labeled upwards or downwards. The horizontal part of a clause consists of vertical edges, again at unit distance. The leftmost and the rightmost edge of the horizontal part, e_{left} and e_{right}, are incident to the topmost edge of the respective leg. This means that e_{left} must be labeled to its right and e_{right} to its left if the topmost edge of the respective leg is labeled upwards. The horizontal part connects the leftmost and the rightmost leg to the central leg. The central edge e^* of the clause gadget is horizontal. The edge e^* lies at unit distance above and symmetrically to the central leg. It connects the right and the left side of the horizontal part of the clause. This edge is the only edge that is not of length 2 but 4, and does not have two, but three label candidates.

The place where a clause leg is attached to a variable gadget depends on whether the corresponding literal ℓ is negated. Let the first, third, etc. horizontal edge of a variable gadget be *odd*, *even* otherwise. If the literal ℓ is a negated variable $\overline{v}$, we attach the leg at distance 2 above an even edge of the gadget for v, otherwise above an odd edge. Due to our encoding of v this makes sure that all labels in the leg must be placed above their edges if $\ell = \overline{v}$ and v is *true* or if $\ell = v$ and v is *false*. Graphically speaking, pressure is transmitted upwards. On the other hand, if $\ell = \overline{v}$ and v is *false* or if $\ell = v$ and v is *true*, then the labels can be placed below their edges and no pressure is transmitted.

If all literals of a clause evaluate to false, then pressure is transmitted through all three legs into the clause. In this case there is an edge (like e^*) that cannot be labeled, see Figure 2. In case there is at least one leg without pressure, it is obvious that all clause edges can be labeled.

Hence the question whether a planar 3-SAT formula ϕ can be satisfied is equivalent to asking whether all edges in the graph drawing, to which ϕ is reduced, can be labeled. We can both construct the variable-clause graph of

ϕ and compute a planar embedding of it in polynomial time. Based on this information, we can then draw—again in polynomial time—the graph whose edges can be labeled if and only if ϕ is satisfiable. Thus our reduction is polynomial. ◻

Clearly in this proof we cannot avoid to make use of edges with three label candidates since the decision problem could be solved in polynomial time by 2-SAT if all edges had only *two* label candidates. This was first observed by Forman and Wagner [FW91]. Since Kakoulis and Tollis did their reduction from 3-SAT, they had to design a gadget that handles the intersection of legs. The central edge of this gadget has *four* label candidates [KT97b, Figure 7].

References

[AF84] John Ahn and Herbert Freeman. AUTONAP—an expert system for automatic map name placement. In *Proceedings International Symposium on Spatial Data Handling*, pages 544–569, 1984.

[C+96] Bernard Chazelle et al. Application challenges to computational geometry: CG impact task force report. Technical Report TR-521-96, Princeton University, April 1996. `http://www.cs.princeton.edu/~chazelle/taskforce/CGreport.ps`.

[CMS95] Jon Christensen, Joe Marks, and Stuart Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.

[DF89] Jeffrey S. Doerschler and Herbert Freeman. An expert system for dense-map name placement. In *Proc. Auto-Carto 9*, pages 215–224, 1989.

[DKMT98] Uğur Doğrusöz, Konstantinos G. Kakoulis, Brendan Madden, and Ioannis G. Tollis. Edge labeling in the graph layout toolkit. In *Proceedings of the Symposium on Graph Drawing (GD'98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 356–363. Springer-Verlag, August 1998.

[DMM+97] Srinivas Doddi, Madhav V. Marathe, Andy Mirzaian, Bernard M.E. Moret, and Binhai Zhu. Map labeling and its generalizations. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 148–157, New Orleans, LA, 4–7 January 1997.

[ECMS97] Shawn Edmondson, Jon Christensen, Joe Marks, and Stuart Shieber. A general cartographic labeling algorithm. *Cartographica*, 33(4):13–23, 1997.

[FW91] Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Annu. ACM Sympos. Comput. Geom. (SoCG'91)*, pages 281–288, 1991.

[Hir82] Stephen A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

[IL97] Claudia Iturriaga and Anna Lubiw. NP-hardness of some map labeling problems. Technical Report CS-97-18, University of Waterloo, Canada, 1997.

[KR92] Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM J. Discr. Math.*, 5(3):422–427, 1992.

[KT97a] Konstantinos G. Kakoulis and Ioannis G. Tollis. An algorithm for labeling edges of hierarchical drawings. In *Proceedings of the Symposium on Graph Drawing (GD'97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 169–180. Springer-Verlag, 1997.

[KT97b] Konstantinos G. Kakoulis and Ioannis G. Tollis. On the edge label placement problem. In *Proceedings of the Symposium on Graph Drawing (GD'96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 241–256. Springer-Verlag, 1997.

[KT98a] Konstantinos G. Kakoulis and Ioannis G. Tollis. On the multiple label placement problem. In *Proc. 10th Canadian Conf. Computational Geometry (CCCG'98)*, pages 66–67, 1998.

[KT98b] Konstantinos G. Kakoulis and Ionnis G. Tollis. A unified approach to labeling graphical features. In *Proc. 14th Annu. ACM Sympos. Comput. Geom. (SoCG'98)*, pages 347–356, June 1998.

[Lic82] David Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[Mor80] Joel L. Morrison. Computer technology and cartographic change. In D.R.F. Taylor, editor, *The Computer in Contemporary Cartography.* J. Hopkins Univ. Press, New York, 1980.

[MS91] Joe Marks and Stuart Shieber. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard CS, 1991.

[NH00] Gabriele Neyer and Holger Hennes. Map labeling with applications to graph labeling. In Dorothea Wagner, editor, *GI Forschungsseminar: Zeichnen von Graphen.* Teubner Verlag, 2000. to appear.

[SW99] Tycho Strijk and Alexander Wolff. Labeling points with circles. Technical Report B 99-08, Institut für Informatik, Freie Universität Berlin, April 1999.

[SW00] Tycho Strijk and Alexander Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 2000. to appear.

[vKSW99] Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21–47, 1999.

[WS96] Alexander Wolff and Tycho Strijk. A map labeling bibliography. `http://www.math-inf.uni-greifswald.de/map-labeling/bibliography/`, 1996.

[WW97] Frank Wagner and Alexander Wolff. A practical map labeling algorithm. *Computational Geometry: Theory and Applications*, 7:387–404, 1997.

[WWKS00] Frank Wagner, Alexander Wolff, Vikas Kapoor, and Tycho Strijk. Three rules suffice for good label placement. *Algorithmica Special Issue on GIS*, 2000. to appear.

[Zor90] Steven Zoraster. The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 38(5):752–759, 1990.

[ZP99] Binhai Zhu and Chung Keung Poon. Efficient approximation algorithms for multi-label map labeling. In A. Aggarwal and C. Pandu Rangan, editors, *Proc. 10th Annual International Symposium on Algorithms and Computation (ISAAC'99)*, volume 1741 of *Lecture Notes in Computer Science*, pages 143–152, Chennai, India, 16–18 December 1999. Springer-Verlag.