On the difficulty of embedding planar graphs with inaccuracies

(Extended Abstract)

Michael Godau

godau@tcs.fu-berlin.de
Freie Universität Berlin, FB Mathematik und Informatik, WE 03
Takustraße 9, 14195 Berlin, Germany

Abstract. In this paper it will be shown that the following problem is NP-hard. We are given a labeled planar graph, each vertex of which is assigned to a disc in the plane. Decide whether it is possible to embed the graph in the plane with line segments as edges such that each vertex lies in its disc.

1 Introduction

Each planar graph has an embedding in the plane with line segments as edges. But it seems to be interesting to ask whether this is possible under various constraints. In [1] this is studied for the case where we restrict the length of each edge to be a fixed value. On the other hand we can assign each vertex a set of points in the plane and ask whether it is possible to embed the graph with line segments as edges with the additional constraint that each vertex must lie in its point set. In this paper we show that we can achieve NP-hardness if the points sets are closed discs.

For the rest of the paper we only consider oriented planar triangulated graphs. For each triangle in the graph an orientation consists of an enumeration of its vertices modulo an even permutation. We only consider positively oriented embeddings. These have the additional property that all the vertices of a triangle are positive oriented (i.e. are in counter clockwise order when embedded). Our results hold for the non oriented case as well because oriented instances can be converted into non oriented instances by forcing all embeddings to be oriented by adding one big triangle around the discs assigning each vertex of this triangle a disc of radii zero and connecting the outer vertices of the graph with the triangle in some obvious way.

Let us call the main problem in our paper **constrained embedding** and redefine it as follows: We are given a labeled triangulated oriented planar graph, each vertex of which is assigned to a closed disc in the plane given by a centerpoint of rational coordinates and a rational radius. Decide whether it is possible to embed the graph positively oriented in the plane with line segments as edges such that each vertex lies in its disc.

We will prove that constrained embedding is NP-hard.

2 Background

Cook [2] proved the NP-completeness of 3SAT by directly showing that every problem in NP can be reduced in polynomial time onto 3SAT. With this first NP-complete problem other problems could be proved to be NP-complete simply by reducing known NP-complete problems to them.

2.1 Definitions

▷ A boolean formula is said to be in **conjunctive normal form** iff it is a conjunction of disjunctions of literals. A **literal** is a variable or a negated variable. The disjunctions are also called **clauses**.
▷ A boolean formula is said to be in **3-conjunctive normal form** iff it is in conjunctive normal form and every clause consists of exactly three literals, where we allow one variable to appear more than once in a clause.
▷ A boolean formula is said to be in **3,4-conjunctive normal form** iff it is in 3-conjunctive normal form and every variable occurs at most four times, where repeated occurrences of one variable in one clause are counted repeatedly.
▷ A boolean formula φ in conjunctive normal form is said to be **planar** iff the bipartite graph B_φ is planar, where the vertices of B_φ are the variables and clauses of φ and the edges are exactly the pairs (v, c) for which v is a variable occurring in the clause c. This definition differs slightly from the one given in [3], but it is sufficient to know that planar formula in the sense of [3] is planar in the definition above, too.
▷ Let *yellow submarine*-SAT denote now the problem of deciding whether a given boolean formula is satisfiable, which is in *submarine*-conjunctive normal form and furthermore is *yellow*.

2.2 Known Reductions

In [4] it was shown that 3,4-SAT is NP-complete. The idea used to show one variable has to occur no more that four times is indeed very simple; we can replace each variable x occurring k times by $x_0, ..., x_{k-1}$ and add the clauses $(x_i \vee \neg x_{i+1 \bmod k})$. If we write $(x_i \vee \neg x_{i+1 \bmod k})$ as $(x_i \vee x_i \vee \neg x_{i+1 \bmod k})$ this gives a clause consisting of exactly three literals and every x_i occurs exactly four times.*1 The only reason to mention this is to notice that this process keeps the planarity of the formula.

In [3] it is shown that planar 3-SAT is NP-complete. With the idea in [4] it can be shown that planar 3,4-SAT is also NP-complete. A careful reading of the construction in [3] used to show planar 3,4-SAT to be NP-complete makes it clear that for a given formula of length n in 3-conjunctive normal form not only can an equivalent instance of planar 3,4-SAT consisting of some formula φ be computed in polynomial time in n, but also an embedding of B_φ can be computed in a very

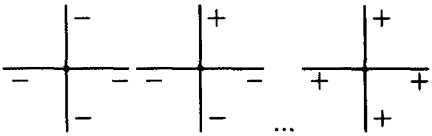
*1 This is a dirty trick. Strictly speaking in [4] it is actually shown how to avoid multiple occurrences of one variable within one clause. But this is not important for our proof.

simple way. We can furthermore assume that B_φ is embedded in a $m \times m$-grid where the vertices are grid points and the edges are vertex disjoint paths on the grid, where m is linear in n.

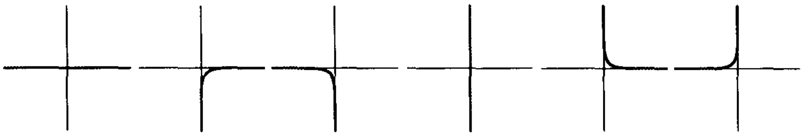
2.3 Folklore

Altogether the following problem — let us call it **grid3sat** — is also NP-complete. We are given a $m \times m$-grid, in which some grid points are distinguished as clauses and some as variables. The variables are connected with the clauses by vertex disjoint paths on the grid. We associate a sign with every path indicating whether the corresponding variable is negated in the corresponding clause or not. The question is, whether the formula described in this funny way is satisfiable or not.

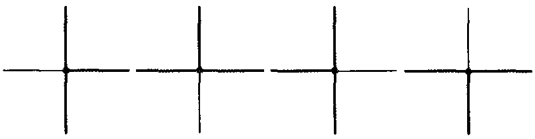
The reduction to prove constrained embedding to be NP-hard is computationally trivial, since it is based on building the graph mentioned above from a stock of 26 different components. These components are complicated but of constant size and description complexity. We will first characterize the components more precisely. There will be 16 different variable components,



6 different connection components,



and 4 different clause components.



We write the signs on the edges in the graph directly on the variable components, hence the $2^4 = 16$ different types of variable components.

The desired instance J of constrained embedding is a labeled triangulated oriented planar graph. We construct it as an embedded graph. Thus each of the components mentioned above will be also an embedded labeled triangulated oriented planar graph. We put all these components together and get another graph. This graph has only to be triangulated in an arbitrary way to obtain J.

3 The Reduction

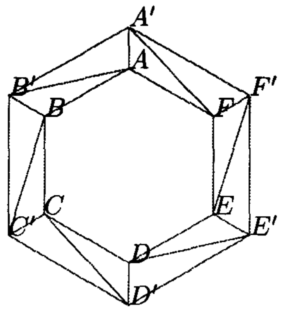
3.1 General Remarks

Let I be an instance of grid3sat. Let us try to construct an instance J of constrained embedding which is solvable iff the formula described by I is satisfiable. It is mainly a question of finding a triangulated oriented planar graph which can be embedded under certain constraints iff the formula in I is satisfiable. Obviously we should search for a correspondence between an embedding of the graph and a truth assignment for the variables in the formula.

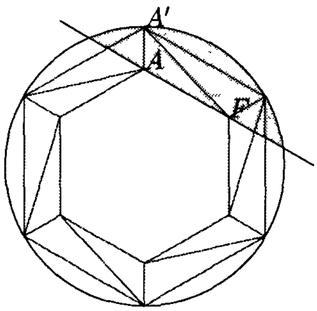
We will indicate instances of constrained embedding by drawing some graph, and this drawing also defines the orientation of the graph. The labels of the vertices will either be explicitly defined or be implicitly defined as disc of radii zero centered at the given position of the vertex. In most cases the drawn embedding will be a correct embedding.

3.2 The Variables

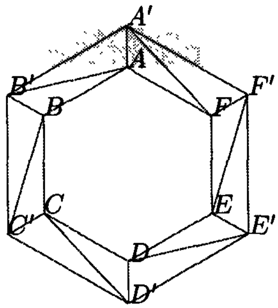
Let us consider first a single variable. A variable can have exactly two states. Let us try to give an oriented graph which can be embedded under certain constraints in only two different ways. Consider the following oriented graph.



Let us consider now embeddings for which the positions of $A, ..., F$ are fixed and the positions of $A', ..., F'$ have to lie in a circle Z only a little bigger than the circumcircle of the hexagon $A...F$. What does this means for instance for A'?



Because of the oriented triangle $A'AF$ the vertex A' has to lie in the halfspace above the line AF. Aside from this the vertex A' has to lie in its circle, namely Z. Thus A' has to lie in the circular segment defined to be the intersection of this halfspace and this disc. The other vertices $B', ..., F'$ also have to lie in certain circular segments. Since the vertices A' and B' are connected they have to be visible to each other in a correct embedding. To achieve this either A' or B' has to lie within the shaded region shown below.



If we draw the circle Z smaller the shaded region would shrink and in the following we want to say that a point lies **close** to A if it is this region.

Around the vertices $B, ..., F$ there are similar regions. Together with the region for A these regions will be disjoint if Z sufficiently small. Now at least one of the vertices $A', ..., F'$ has to lie in each region. Since there are six points each point has to lie in exactly one region.

The reader can verify that in principle there are only two possibilities: either A' is close to A, B' is close to B etc. as indicated in the drawings above, or A' is close to F, F' is close to E, etc. We can imagine that $A', ..., F'$ form an outer hexagon which can be in two situations differing by a rotation of 60 degrees. The situation twisted in clockwise orientation we denote **−-situation** and the other one **+-situation**.

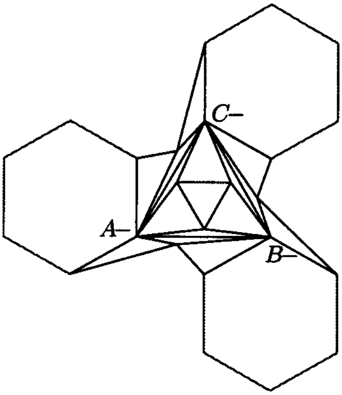
We are not able to simulate the behavior of a variable in the sense that there exist exactly two embeddings which correspond to the two truth assignments of the variable but we are able to do it in an approximate sense such that there are two distinguishable classes of embeddings.

In the following we will indicate these counterparts of the variables only by the outer hexagon and talk only about embeddings of the outer vertices and pretend that there exists indeed only two embeddings, namely the − and +-situations.

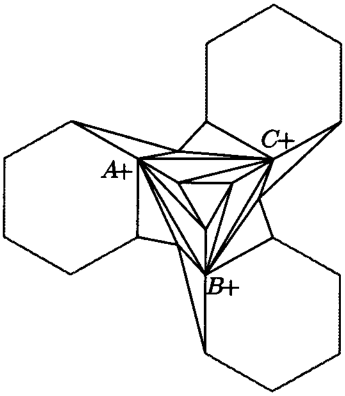
3.3 Branches

The following example shows how to combine three hexagons such that they all have to be situated in the same way (i.e. either all in −-situation or all in +-situation).

Here is the −-situation.



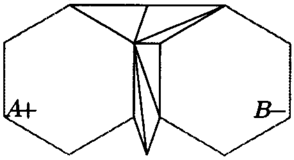
And here is the +-situation.



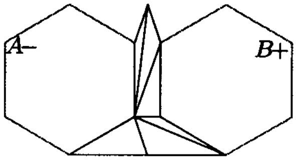
3.4 Negations

The following example shows how to combine two hexagons such that one of them is in +-situation if the other one is in −-situation and vice versa.

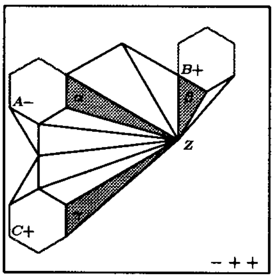
Here the left one is in +-situation and the right one is in −-situation.



... and this is the reverse case.

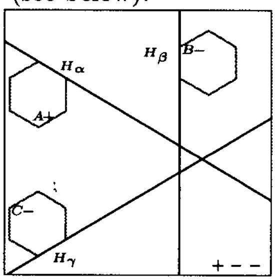


3.5 Clauses

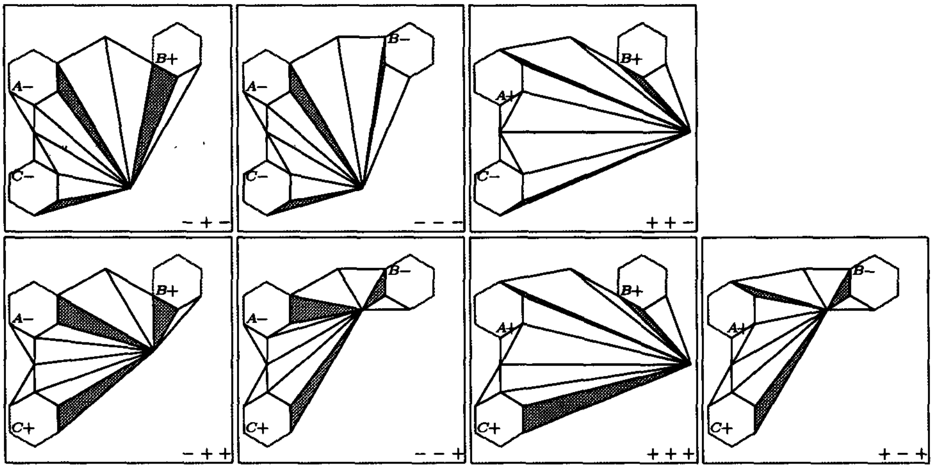


Consider the left construction. The vertex Z incident to 11 edges may be embedded in a nearly arbitrary place. Think about a disc around the whole picture assigned to the vertex Z.

The crucial fact is that the complementary embedding (A+, B–, C–) is not possible. This is because of the triangles α, β and γ. Consider for example α. If the hexagon A were in the +-situation the vertex Z would have to be in the half space H_α (see below). Analogously one can observe that if B were in −-situation the vertex Z would have to lie in the half-space H_β because of β. If then C were in −-situation the vertex Z would have to lie in the halfspace H_γ, too. But $H_\alpha \cap H_\beta \cap H_\gamma$ is empty.



The following pictures demonstrate that all other combinations are embeddable.



Thus we have three hexagons connected so that all combinations of situations are embeddable except one (namely $A+, B-, C-$). This corresponds to a clause of the form $\neg x \vee y \vee z$. With one additional negation this can be converted into a clause of the form $x \vee y \vee z$.

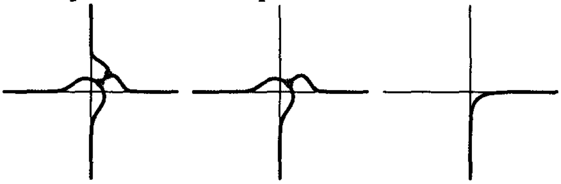
3.6 Connections

Using the negations we can build long chains of coupled hexagons. Notice that in all constructions introduced so far there was a little bit tolerance. There was no need to draw them precisely to establish the argument. We can see that they would work even if they were shifted by a very small and indeed not visible amount. Let $\mu > 0$ this amount. Now in a sequence of hexagons $h_0, ..., h_n$ glued together by negations we can shift the hexagon h_i by $\frac{i}{n}\mathbf{v}$ assuming $||\mathbf{v}|| \leqslant n\mu$. For sufficiently large n the connections between the components can be made in this way even if in the drawings here they are shifted. Since n depends only on μ and $||\mathbf{v}||$ and these values depend only on the geometry of the constructions, we can treat them as constants.

In a similar way we can handle rotations by an arbitrary amount. Thus we are able to stretch, shift or bend long chains of coupled hexagons almost arbitrary.

3.7 Components revisited

In order to construct the components characterized in subsection 2.3 we use hexagons small enough to allow chains of negations to be flexible enough. In the following drawings a fat stroke indicates such a chain.



The variable components are made of two branches (see subsection 3.3) and we obtain the 16 different types of variable components by stretching the appropriate chains of coupled hexagons with negations thereby decreasing the number of negations by one. The connection components are made only of chains of hexagons with negations. The clause components are made of the clauses explained in subsection 3.5.

We have to pad all components with fixed vertices in order to make most of the boundary of the components have a fixed shape and in order to triangulate the interior.

This completes the construction. It remains to show that the formula given by an instance I of grid3sat is satisfiable iff there exists an embedding solving the instance J of constrained embedding.

Assume that the formula is satisfiable. Then we place all hexagons in the branches in the variable elements in $-$-situation, when the corresponding variable is assigned to be false and in $+$-situation otherwise. The situations of the other hexagons follow.

Assume conversely there exists an embedding solving the instance J. Then we look at the hexagons in the branches in the variable components to get a truth assignment which fulfills the formula.

Acknowledgment

I wish to thank Helmut Alt for fruitful discussions about methodical questions and about this paper, of course. I think it was his idea to publish this result. Furthermore I wish to thank Annamaria Amenta for careful proofreading which lead to many improvements in the text.

References

1. Eades, P., Wormald, N.: Fixed Edge Length Graph Drawing is NP-hard. Discrete Applied Mathematics. **28** (1990) 111–134
2. Cook, S.: The complexity of theorem-proving procedures. Proc. 3rd Ann. ACM Symp. on Theory of Computing (assoc. Comput. Mach., New York, 1971) 151–158
3. Lichtenstein, D.: Planar formulae and their uses. SIAM Journal of Computing **11** (1982) no. 2, 329–343
4. Tovey, C.: A simplified NP-complete satisfiability problem. Discrete-Applied-Mathematics **8** (1984), no. 1, 85–89