NP-Completeness of Max-Cut for Segment Intersection Graphs

Oswin Aichholzer[1], Wolfgang Mulzer[2], Partick Schnider[3], and Birgit Vogtenhuber[1]

1 **Institute of Software Technology, Graz University of Technology, Austria.**
`oaich@ist.tugraz.at`, `bvogt@ist.tugraz.at`
2 **Institut für Informatik, Freie Universität Berlin, Germany.**
`mulzer@inf.fu-berlin.de`
3 **Department of Computer Science, ETH Zürich, Switzerland.**
`patrick.schnider@inf.ethz.ch`

Abstract

We consider the problem of finding a *maximum cut* in a graph $G = (V, E)$, that is, a partition $V_1 \dot\cup V_2$ of V such that the number of edges between V_1 and V_2 is maximum. It is well known that the decision problem whether G has a cut of at least a given size is in general NP-complete. We show that this problem remains hard when restricting the input to *segment intersection graphs*. These are graphs whose vertices can be drawn as straight-line segments, where two vertices share an edge if and only if the corresponding segments intersect. We obtain our result by a reduction from a variant of Planar Max-2-SAT that we introduce and also show to be NP-complete.

1 Introduction

For a graph $G = (V, E)$, consider a partition $V = V_1 \dot\cup V_2$ of V. The set $E_{12} \subseteq E$ of edges with one endpoint in V_1 and one endpoint in V_2 is called a *cut* (induced by V_1 and V_2), and the cardinality $|E_{12}|$ is called the *size* of the cut. A *maximum* cut of G is a cut whose size is as large as possible. The problem MaxCut is to find the size of a maximum cut in a given graph G. MaxCut can also be cast as a vertex coloring problem: what is the maximum number of bichromatic edges that can be obtained by coloring each vertex with one of two possible colors? The decision version of MaxCut asks whether G contains a cut of size at least k, for a given $k \in \mathbb{N}$. It is NP-complete for general graphs [3]. Moreover, MaxCut is hard to approximate [7, 8]. On the other hand, there exists a PTAS for MaxCut in dense graphs [1]. For planar graphs, MaxCut can be solved in polynomial time [6], and the same is true for several other graph classes [2].

A *segment intersection graph* is a graph whose vertices can be drawn as straight-line segments (that pairwise intersect in at most one point, in their relative interiors), such that two vertices share an edge if and only if the corresponding segments intersect. In a (representation of a) segment intersection graph, a maximum cut corresponds to a 2-coloring of the segments such that the number of bichromatic crossings, i.e., crossings of segments with different colors, is maximum. So far, the complexity status of MaxCut on line segment intersection graphs seems to be open [2]. We show that the decision version of MaxCut is NP-complete even when the input is restricted to segment intersection graphs. We obtain this result via a reduction from a variant of Planar Max-2-SAT, that we introduce and show to be NP-complete as well in Section 2.

This project has been supported by the Austrian Science Fund (FWF) grant W1230 and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 734922.

34th European Workshop on Computational Geometry, Berlin, Germany, March 21–23, 2018.
This is an extended abstract of a presentation given at EuroCG'18. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

In addition to the intrinsic interest of the problem, our study is motivated by the following question that was posed by Ruy Fabila-Monroy at the workshop "Reunión de Optimización, Matemáticas y Algoritmos" in the framework of the project CONNECT: let D be a straight-line drawing of the complete graph K_n on n vertices. A *k-edge-coloring* χ of K_n assigns to each edge of K_n a color from $\{1, \ldots, k\}$. Let $\bar{cr}_k(D, \chi)$ be the number of monochromatic edge crossings in D for the χ, that is, crossings of edges with the same color. What is the best drawing D and the best k-edge-coloring χ of K_n in order to minimize $\bar{cr}_k(D, \chi)$? During the workshop, Francisco Javier Zaragoza Martínez observed the following relation to maximum cuts: For a fixed drawing D, the total number of crossings is fixed. Thus, a k-edge-coloring χ with the minimum number of monochromatic crossings maximizes the number of bichromatic crossings. Further, any geometric graph can be interpreted as a segment intersection graph. Hence finding a 2-edge coloring of K_n with the minimum number of monochromatic crossings is equivalent to finding a maximum cut in the segment intersection graph D. We remark that our construction does not show hardness of MaxCut for straight-line drawings of K_n.

2 Planar Max-2-SAT

We will use a reduction from a variant of MAX2SAT. In MAX2SAT, we are given a Boolean formula ϕ in conjunctive normal form (CNF) with at most two literals per clause and an integer k. We need to determine whether there is an assignment to the variables of ϕ that satisfies at least k clauses. MAX2SAT it NP-complete [4]. We will consider a variant of MAX2SAT where we require the 2-CNF formula ϕ to be *planar* and *clause-tree-linked*, two notions that we will now define.

Given a CNF formula ϕ with clause set C and variable set V, the *incidence graph* $G_\phi = (C \cup V, E)$ is the graph that contains an edge between a variable and a clause if and only if the variable or its negation appear as a literal in the clause. We say that ϕ is *planar* if G_ϕ is a planar graph. The problems PLANAR 3-SAT and PLANAR MAX-2-SAT are 3-SAT and MAX2SAT restricted to planar formulas. PLANAR 3-SAT is NP-complete [9]. To see that PLANAR MAX-2-SAT is NP-hard, it can be checked that the reduction from 3-SAT to MAX-2-SAT in [4] preserves planarity; see for example Theorem 2 in [5] and the proof of Theorem 2.1 below.

For PLANAR 3-SAT, we can enforce even more conditions without making the problem tractable: we say that a planar 3-CNF formula ϕ is *clause-linked* if there exists a path P connecting the clauses in $G(\phi)$ such that $G(\phi) \cup P$ is still a planar graph. CLAUSE-LINKED PLANAR 3-SAT, which is 3-SAT restricted to clause-linked planar formulas, is still NP-complete, see for example [10].

Similarly, we can add more conditions on MAX-2-SAT: we say that a planar 2-CNF formula ϕ is clause-tree-linked if there exists a spanning tree T of the clauses in $G(\phi)$ such that $G(\phi) \cup T$ is still a planar graph. We define CLAUSE-TREE-LINKED PLANAR MAX-2-SAT as MAX-2-SAT restricted to clause-tree-linked planar formulas.

▶ **Theorem 2.1.** CLAUSE-TREE-LINKED PLANAR MAX-2-SAT *is NP-complete.*

Proof. To show NP-completeness of CLAUSE-TREE-LINKED PLANAR MAX-2-SAT, we need to show its membership in NP and its NP-hardness. Membership in NP directly follows from the fact that CLAUSE-TREE-LINKED PLANAR MAX-2-SAT is a special case of the NP-complete problem MAX-2-SAT. We prove NP-hardness by reduction from CLAUSE-LINKED PLANAR 3-SAT.

In CLAUSE-LINKED PLANAR 3-SAT, we have as input a 3-CNF formula ϕ with variable set V and clause set C, together with a linear ordering o of the elements of C. Further, the incidence graph $G(\phi) = (C \cup V, E)$ together with the path $P(o) = (C, E_P)$ on C that is induced by the linear ordering o is still planar. To transform this input to an input of Clause-Tree-Linked Planar Max-2-SAT, we utilize the following reduction function of the well known reduction from 3-SAT to Max-2-SAT [4]: Every clause $c = (x, y, z)$ in ϕ is replaced by a 2-CNF formula c' of the form

$$c' := x \wedge y \wedge z \wedge w \wedge (\neg x \vee \neg y) \wedge (\neg x \vee \neg z) \wedge (\neg y \vee \neg z) \wedge (x \vee \neg w) \wedge (y \vee \neg w) \wedge (z \vee \neg w),$$

where w is an additional variable that is used exclusively used for one clause of ϕ. The complete 2-CNF formula for the Max-2-SAT is then $\phi' := \bigwedge_{c \in C} c'$. The target value for the number of clauses that should be satisfied in ϕ' is $k' := 7|C|$. The reduction from 3-SAT to Max-2-SAT follows from the fact any variable assignment that does not satisfy a clause c in ϕ satisfies at most six of the clauses in c', while an assignment satisfying c satisfies exactly seven clauses in c'. What remains to be proven is that the resulting incidence graph $G'(\phi')$ admits a tree $T(o) = (C', E_T)$ such that $G'(\phi')$ together with $T(o)$ is still a planar graph. To this end, consider a plane embedding D of the graph $G(\phi)$ together with the path $P(o) = (C, P)$. We first construct a plane embedding[1] of $G'(\phi')$ from D. For a clause $c = (x \vee y \vee z)$ in ϕ, the sub graph in $G(\phi)$ induced by c and its variables x, y, and z is a tree with center c and leaves x, y, and z; see Figure 1 (left). To obtain an embedding of $G(\phi')$, we start with the embedding of $G(\phi)$. For every clause c in ϕ replace the tree of c (and its variables) by an embedding of the sub graph induced by c' (and its variables) in $G'(\phi')$ as depicted in Figure 1 (right). Because the variable vertices x, y and z all lie in the unbounded face of this drawing, the resulting embedding of $G'(\phi')$ is again plane.

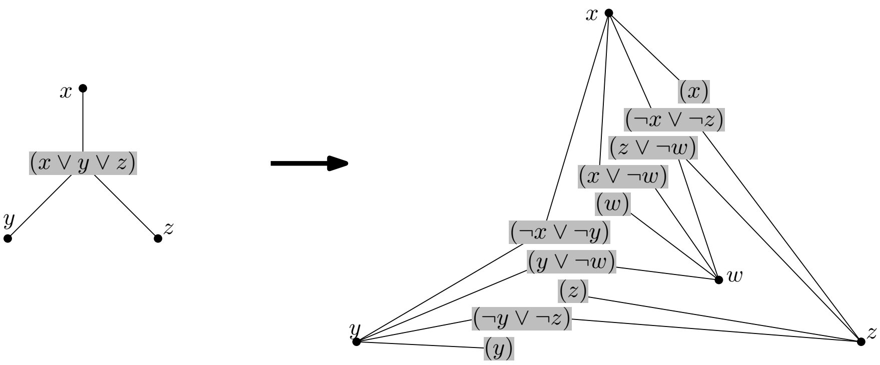

Figure 1 The subgraph of a clause c and its variables in ϕ (left), and the according subgraph of the transformation c' and its variables (right). Variable vertices are drawn as dots while clause vertices are drawn shaded.

Further, in $P(o)$, c is incident to one or two edges going to its neighbor(s) in the linear order o on C. We extend the drawing of $P(o)$ in D to a drawing of a tree through all clauses of ϕ' in D' such that the total drawing remains plane. It is easy to see that the drawing in Figure 1 (right) can be extended by a path P' through all the clauses that starts and ends in the unbounded face. If c is an endpoint of $P(o)$ and in D, the edge of $P(o)$ at c is between the ones to z and x or y, respectively, then we replace c in the drawing $P(o)$ by (x) or (y), respectively, and append P' to the drawing of $P(o)$. If in D, the two path edges

[1] It has been known that the reduction from 3-SAT to Max-2-SAT preserves planarity [5]. We reprove the statement via a concrete embedding, which we then utilize to also show clause-tree-linkedness.

at c are neighboring and between the ones to z and x or y, respectively, then we replace c in the drawing $P(o)$ by (x) or (y), respectively, and append P' as a branch to the drawing of $P(o)$. If in D, the path separates z from x and y in the order around c, then we replace the vertex c in the drawing of $P(o)$ by the path P'. Finally, note that the drawing of c' and its variables is not symmetric, but c' itself is. Hence, an appropriate permutation of x, y, and z in the drawing always yields a drawing of c' that fits one of the above cases. This finishes the reduction. ◀

3 Max-Cut for Segment Intersection Graphs

▶ **Theorem 3.1.** *The decision version of the* MAX-CUT *problem is NP-complete even when restricted to segment intersection graphs.*

Proof. We prove NP-hardness by reduction from CLAUSE-TREE-LINKED PLANAR MAX-2-SAT. For any clause-tree-linked planar 2-SAT formula ϕ with m clauses we construct a line segment arrangement S with the property that there is an assignment satisfying at least $m - k$ clauses of ϕ if and only if there is a 2-coloring of the segments of S with at most $m + 2k$ monochromatic crossings.

Let ϕ be a clause-tree-linked planar 2-SAT formula and let $G(\phi)$ be its associated graph and T the tree through its clauses. Consider a plane drawing of $G(\phi) \cup T$. We will mimic the formula ϕ by constructing line segment configurations, called *gadgets*, that serve as variables, wires, splits, negations and clauses, and concatenating them according to the drawing of the graph $G(\phi)$. We will use wire gadgets and split gadgets to propagate the truth assignment of a variable along the edges between the variable and the clauses containing it, while negation gadgets will serve to invert the truth assignment of a variable (for negative literals).

As variable gadget, we just take a single line segment. Each line segment will be colored with one of two colors, without loss of generality red and blue, one of them representing the true state, the other one the false state. For a wire gadget, we draw two segments a and b that do not cross each other and $2m + 1$ other segments, each of which crosses a and b but no other segment. See Figure 2 (left) for an illustration. It follows that if a and b get the same color, we can color the gadget without monochromatic crossings, whereas if a and b get different colors, any coloring of the remaining edges yields exactly $2m + 1$ monochromatic crossings. To build a split gadget, we repeat the construction of the wire gadget twice; see Figure 2 (middle). For the negation gadget, we again draw two segments a and b that do not cross each other. Further, we draw two families C and D of $2m + 1$ pairwise non-crossing segments each, such that each segment of C crosses a, each segment of D crosses b, and each segment of C crosses each segment of D; see Figure 2 (right). Note that for the negation gadget we have at least $2m + 1$ monochromatic crossings if a and b have the same color. However, if a and b have different colors, this gadget can again be colored without monochromatic crossings.

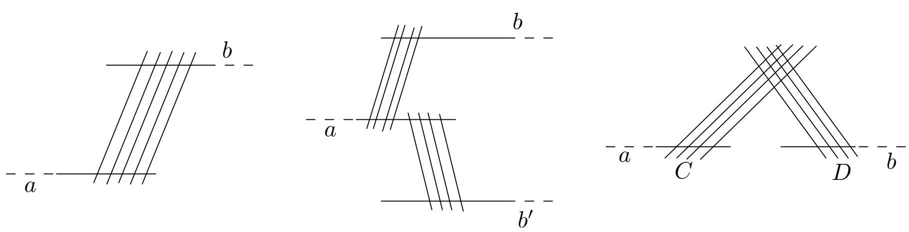

Figure 2 A wire gadget (left), a split gadget (middle) and a negation gadget (right).

It remains to construct the clause gadgets. For any two literals that form a clause, draw two corresponding segments a and b and a segment t, called *tree segment*, such that both a and b cross t. Further, we draw two additional segments c and d, where c crosses only a and d and d crosses only b and c. See Figure 3 for an illustration. Assume that t is colored red. If both a and b are blue, coloring c and d without obtaining a monochromatic crossing is impossible, but we can color c and d such that we have only one monochromatic crossing. The same holds if both a and b are red, but in this case there are also two monochromatic crossings between t, a, and b. If a is red and b is blue or vice versa, we have a monochromatic crossing between a or b and t, but we can color c and d such that they are not involved in any monochromatic crossing. So, to summarize, every clause requires at least one monochromatic crossing and we have a coloring with exactly one such crossing unless a and b have the same color as t, in which case the clause requires at least three monochromatic crossings. In our construction, the colors of the tree segments will represent the false state. Hence, any satisfied clause can be drawn with only one monochromatic crossing, while unsatisfied clauses require at least three monochromatic crossings.

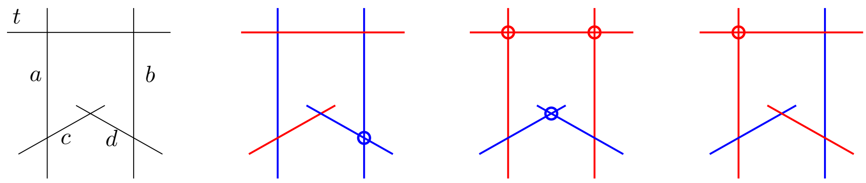

Figure 3 A clause gadget and some possible colorings of it. Monochromatic crossings are marked with small circles.

Using these gadgets, we construct a line segment arrangement that goes essentially along the edges of the given drawing of $G(\phi)$. To enforce that the tree segments have the same color, we connect them using wire gadgets according to the drawing of T. Let S be the line segment arrangement obtained by this construction. See Figure 4 for a small example.

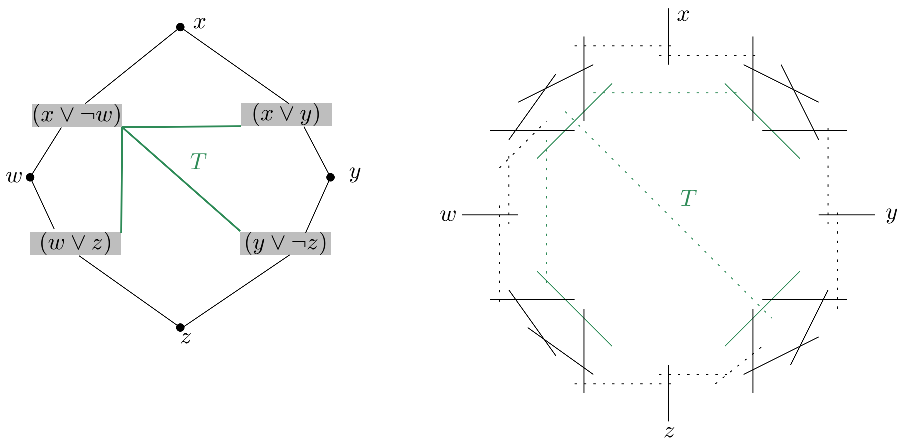

Figure 4 A drawing of $G(\phi)$ for the 2-SAT formula $\phi = (x \vee \neg w) \wedge (x \vee y) \wedge (w \vee z) \wedge (y \vee \neg z)$, with a tree T connecting the clauses (left) and the segment arrangement derived from this drawing (right). Dashed edges correspond to sets of $2m + 1$ line segments.

Next we show that there is an assignment satisfying at least $m-k$ out of the m clauses of ϕ if and only if there is a 2-coloring of the segments of S with at most $m + 2k$ monochromatic crossings, for any $0 \leq k \leq m$.

First assume that there is a 2-coloring of the segments of S with at most $2k + m \leq 3m$ monochromatic crossings. As each of the m clause gadgets needs at least one monochromatic crossing, at most k clause gadgets can have three (or more) monochromatic crossings. Furthermore, in each wire gadget, the segments corresponding to a and b in the illustration in Figure 2 (left) must have the same color. Otherwise the gadget alone would already contain at least $2m+1$ monochromatic crossings and hence the whole drawing would contain at least $2m+1+m \geq 3m+1$ monochromatic crossings, a contradiction. For the same reason, all tree segments have the same color and furthermore the segments corresponding to a, b, and b' in a split gadget share the same color; and in any negation gadget, the segments corresponding to a and b must have different colors. Hence, interpreting the color of the tree segments as representing the false state and assigning the truth states to the variables in ϕ according to the color of their respective variable gadgets, we obtain a variable assignment for ϕ with at most k unsatisfied clauses.

For the other direction, assume that there is an assignment satisfying at least $m - k$ clauses of ϕ. Color the variable gadgets blue if the corresponding variable is assigned the true state, and red otherwise. Color the tree segments in red and all the gadgets, except the clause gadgets, without monochromatic crossings. Then the only monochromatic crossings occur in the clause gadgets. Each of them induces one monochromatic crossing, and two more if and only if the corresponding clause is unsatisfied. As there are at most k unsatisfied clauses the coloring has at most $2k + m$ monochromatic crossings.

It is not hard to see that the line segment arrangement S can be constructed in polynomial time, which concludes the NP-hardness part. Furthermore, the problem is clearly in NP as it is a restricted version of the NP-complete problem MAX-CUT, which finishes the proof. ◀

References

1. S. Arora, D. R. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *J. Comput. Syst. Sci.*, 58(1):193–210, 1999.
2. H. N. de Ridder et al. Information system on graph classes and their inclusions. `www.graphclasses.org`, 2016.
3. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
4. M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976. URL: `http://www.sciencedirect.com/science/article/pii/0304397576900591`, `doi:https://doi.org/10.1016/0304-3975(76)90059-1`.
5. L. J. Guibas, J. Hershberger, J. S. B. Mitchell, and J. Snoeyink. Approximating polygons and subdivisions with minimum link paths. In *Proceedings of the 2nd International Symposium on Algorithms*, ISA '91, pages 151–162, London, UK, UK, 1991. Springer-Verlag. URL: `http://dl.acm.org/citation.cfm?id=648003.743125`.
6. F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3):221–225, 1975. `doi:10.1137/0204019`.
7. J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.
8. S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.
9. D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982. `doi:10.1137/0211025`.
10. A. Pilz. Planar 3-SAT with a clause/variable cycle. *CoRR*, abs/1710.07476, 2017. URL: `http://arxiv.org/abs/1710.07476`, `arXiv:1710.07476`.