MAX-CUT and Containment Relations in Graphs

Marcin Kamiński*

Département d'Informatique
Université Libre de Bruxelles
Marcin.Kaminski@ulb.ac.be

Abstract. We study MAX-CUT in classes of graphs defined by forbidding a single graph as a subgraph, induced subgraph, or minor. For the first two containment relations, we prove dichotomy theorems. For the minor order, we show how to solve MAX-CUT in polynomial time for the class obtained by forbidding a graph with crossing number at most one (this generalizes a known result for K_5-minor-free graphs) and identify an open problem which is the missing case for a dichotomy theorem.

Keywords: MAX-CUT, subgraph, induced subgraph, minor.

1 Introduction

MAX-CUT is a classical problem in combinatorial optimization and have been studied in different contexts – heuristics, approximation algorithms, exact algorithms, polyhedra. Here we suggest to look at the computational complexity of the problem in different classes of graphs. We focus on graphs obtained by forbidding a single graph as a subgraph, induced subgraph, or minor.

A *cut* in a graph is a partition of the vertex set into two disjoint sets. The *value* of a cut is the total weight of edges whose endpoint belong to two different parts of the cut. A cut of maximum value in G is called a *maximum cut* (or *max-cut*) and the value is denoted by $\mathbf{mc}(G)$. Notice that there is a one to one correspondence between a cut and the set of edges whose endpoint belong to two different parts of the cut. For convenience, we will sometimes refer to this set of edges as a cut as well.

The algorithmic MAX-CUT problem is to determine $\mathbf{mc}(G)$ given an input graph G. A cardinality variant of MAX-CUT is called SIMPLE MAX-CUT. (It is MAX-CUT in which all the weights on edges are equal.) Clearly, if MAX-CUT is solvable in polynomial time for some class of graphs, so it SIMPLE MAX-CUT. Also, if SIMPLE MAX-CUT is NP-complete for some class of graphs, so is MAX-CUT.

In this paper we consider simple, undirected, and real-weighted graphs. The terminology used is standard; for notions not defined here, we refer the reader to [15]. K_k is the complete graph on k vertices and P_k is the induced path on k vertices. $G \cup H$ denotes the disjoint union of G and H.

Our contribution. We look at the classes of graphs defined by forbidding a single graph H as a subgraph, induced subgraph, or minor.

* Chargé de Recherches du FRS-FNRS.

D.M. Thilikos (Ed.): WG 2010, LNCS 6410, pp. 15–26, 2010.

1. For H-subgraph-free graphs, we show that both SIMPLE MAX-CUT and MAX-CUT are solvable in polynomial time, if H is a forest every connected component of which is a tree with at most one vertex of degree 3; and are NP-complete otherwise.
2. For H-induced-subgraph-free graphs, we show that SIMPLE MAX-CUT is solvable in polynomial time, if H is an induced subgraph of P_4; and is NP-complete otherwise. (For MAX-CUT this containment relation is rather uninteresting.)

Our contribution here is to notice that such dichotomy theorems hold and to put known algorithmic and hardness results into this framework. Dichotomy theorems of this type are rather rare. We are aware of just one result of this type. For CHROMATIC NUMBER, Král et al. proved that the class of H-induced-subgraph-free graphs admits a polynomial-time algorithm, if H is an induced subgraph of P_4 or of $P_3 \cup K_1$; and the problem is NP-complete otherwise [25]. No such theorem is known for instance for STABLE SET. In fact, P_5-induced-subgraph-free graphs is the unique minimal class defined by a single forbidden induced subgraph for which the computational complexity of STABLE SET is unknown [26].

The case of minors is a bit different. Revisiting known results, we can show that MAX-CUT is solvable in polynomial time for H-minor-free graphs, for planar H; and SIMPLE MAX-CUT is NP-hard, when H is at "vertex-distance" at least 2 to planarity (becomes planar only when two of its vertices are removed). A remaining open problem is to determine the computational complexity of (SIMPLE) MAX-CUT, when H is a strict apex, which means a graph at "vertex-distance" 1 to planarity. Perhaps those classes admit a polynomial-time algorithm for MAX-CUT. We show that this indeed is a case when H is at "edge-distance" 1 to planarity (becomes planar only when one of its edges is removed). Clearly, graphs at "edge-distance" 1 to planarity are also at "vertex-distance" 1.

3. For H-minor-free graphs, we show that MAX-CUT is solvable in polynomial time, if H is a graph that can be drawn on the plane with at most one crossing.

This generalizes previous work on MAX-CUT for planar graphs [27], [23] and K_5-minor-free graphs [6].

2 Previous Work

Maximum Cut

MAX-CUT was among the twenty one problems whose NP-hardness was established in the foundational paper *"Reducibility among combinatorial problems"* by Richard Karp [24]. Since then it has been extensively studied and became one of the classical problems in the field of combinatorial optimization.

An early result that is of interest to us is a polynomial-time algorithm for MAX-CUT in planar graphs. It was first discovered by Orlova and Dorfman [27] and then independently by Hadlock [23]. The main idea of the solution is to fix an embedding of the planar input graph, take the dual, and find a pairing of vertices of odd degree in the dual graph using matching.

Grötschel and Pulleyblank introduced – by means of a polyhedral definition – the class of *weakly bipartite* graphs and showed how to solve MAX-CUT in this class [18]. Both planar and bipartite graphs are weakly bipartite, thus their result generalizes [27] and [23], as well as a trivial polynomial-time algorithm for bipartite graphs. (Notice that a maximum cut in a bipartite graph contains all its edges.) Later, Guenin proved that weakly bipartite graphs are exactly these that do not contain an odd-K_5-minor[1] [19], [20], [21] (see also [31] for a short proof).

The result of Guenin implies that K_5-minor-free graphs are weakly bipartite (since every odd-K_5-minor is a K_5-minor) and therefore there is a polynomial-time algorithm for MAX-CUT in this class. Before the characterization of weakly bipartite graphs was known, Barahona showed how to solve MAX-CUT in polynomial time in the class of graphs without a K_5-minor [6]. The paper uses a decomposition theorem for K_5-minor-free graphs due to Wagner [32]. The same paper also proves that MAX-CUT is NP-hard for K_6-minor-free graphs.

MAX-CUT is also solvable in polynomial-time in the class of graphs of bounded orientable genus [1]. This results was already attributed to Barahona by [18] but the preprint to which [18] refers apparently has never been published.

MAX-CUT is also known to be NP-complete on unit disk graphs [14] and solvable in polynomial time on graphs without long odd cycles (= a class of graphs with no odd cycles longer than k, for some $k \geq 3$) [17], on line graphs [5] (see also [22] for a simple proof), on graphs of bounded tree-width [8], [9]. Also, there exists a PTAS for MAX-CUT in classes of graphs with a forbidden minor [10].

Graphs with No Single-Crossing Minor

A graph is a *single-crossing* graph when it can be drawn on the plane with at most one crossing. $K_{3,3}$ and K_5 are examples of single-crossing graphs.

Wagner proved that a graph is K_5-minor-free if and only if it can be constructed from planar graphs and copies of the four-rung Möbius ladder glued together along cliques of size ≤ 3 [32]. He also showed that a graph is $K_{3,3}$-minor-free if and only if it can be constructed from planar graphs and copies of K_5 glued together along cliques of size ≤ 2 (possibly removing an edge after pasting along it).

Robertson and Seymour proved a more general theorem, describing the structure of graphs with a forbidden single-crossing minor [30]. They can be obtained from planar graphs and graphs of bounded tree-width (the bound depends on the forbidden single-crossing graph) by pasting them along cliques of size at most 3 and (possibly) removing some of the edges of those cliques afterwards.

[1] An odd minor is a restriction of the standard minor relation.

This structural result was made algorithmic by Demaine et al.; they gave an $\mathcal{O}(n^4)$ algorithm for finding this decomposition [11]. This was subsequently used to give parameterized algorithms with better dependence on the parameter [13] and approximation algorithms with better approximation ratio in classes defined by forbidding a single-crossing graph as a minor [11].

3 Forbidden Subgraph

In this section, we study classes of graphs defined by forbidding a single graph as a subgraph. Let us start with a simple lemma.

Lemma 1. *Let e be an edge of a graph G and G' the graph obtained by subdividing edge e twice. Then, $mc(G') = mc(G) + 2$.*

Proof. A double subdivision of e replaces e with 3 edges; let them be called e_L, e', and e_R. Let us say that a cut in G' is *good* if it contains both e_L and e_R. Notice that there is a maximum cut in G' which is good. Also, there exists a one-to-one correspondence between cuts in G and good cuts in G': cuts have the same edges f, for $f \neq e$, and e belongs to the cut in G iff e' belongs to the cut in G'. This correspondence makes the value of the cut in G' bigger by 2 than the cut in G.

The following lemma is a consequence of this subdivision property.

Lemma 2. SIMPLE MAX-CUT *is* NP*-complete in the following two classes of graphs:*

- *graphs not containing cycles of length at most k, for every $k \geq 3$;*
- *graphs not containing a pair of vertices of degree at least 3 at distance at most k, for every $k \geq 1$.*

Proof. Let us take a graph G, double subdivide all its edges, and then repeat the operation $\lceil \log_3 k \rceil$ times more, each time applying the operation to the outcome of the previous operation. It is easy to see that the graph G' obtained after the series of subdivisions has no cycles of length at most k, and has no pair of vertices of degree at least 3 at distance at most k. Notice that $\mathbf{mc}(G') = \mathbf{mc}(G) + 2(\lceil \log_3 k \rceil + 1)$ from Lemma 1. Since SIMPLE MAX-CUT is NP-hard in the class of all graphs, so it is in the two classes mentioned in the theorem.

We will need one more hardness result from [33].

Lemma 3. *[33]* SIMPLE MAX-CUT *is* NP*-complete in the class of graphs with maximum degree 3.*

We will need the following theorem from [7] (see also [28]) due to Bienstock, Robertson and Seymour.

Theorem 1. *[7] For every forest F, every graph with path-width $\geq |V(F)| - 1$ has a minor isomorphic to F.*

Lemma 4. *Let H be a forest whose every connected component is a tree with at most one vertex of degree three. A graph contains H as a minor if and only if it contains H as a subgraph.*

Proof. The backward implication is clear. We will suppose that H is connected; the forward implication will follow by induction on the number of connected components of H. Let G contain H as a minor. Let us fix a model of H in G. We will build a subgraph T. For each pair of adjacent vertices in H, we select an edge of G whose endpoints are in the two different bags corresponding to the vertices of H. Now, for each bag of the model of H that contains at least two (at most three) endpoints of the edges already in T, let us add to T a tree spanning these endpoints inside the bag. Clearly, T is a tree and T contains H as a subgraph. Hence, G contains H as a subgraph.

Lemma 5. *[9] For every constant t, there exists a polynomial-time algorithm solving* MAX-CUT *in the class of graphs of tree-width at most t.*

Theorem 2. *Both the* SIMPLE MAX-CUT *and* MAX-CUT *problems in the class of H-subgraph-free graphs are:*

- *solvable in polynomial time, if H is a forest whose every connected component is a tree with at most one vertex of degree three;*
- NP*-hard, otherwise.*

Proof. For the first part, let H be a forest whose every connected component is a tree with at most one vertex of degree three. The class of H-subgraph-free graphs is also H-minor-free by Lemma 4. Since H is a forest, from Theorem 1, the path-width of H-subgraph-free graphs is at most $|V(H)| - 2$. Therefore also their tree-width is at most $|V(H)| - 2$. From Lemma 5, MAX-CUT is solvable in polynomial on H-subgraph-free graphs.

For the second part, assume that H is not a forest whose every connected component is a tree with at most one vertex of degree three. Then, H contains a vertex of degree at least 4, or a cycle, or a pair of vertices of degree 3 in the same connected component. In the first case, the SIMPLE MAX-CUT is NP-complete in this class of H-subgraph-free graphs by applying Lemma 3, and in the two last cases by applying Lemma 2 with $k = |H|$.

4 Forbidden Induced Subgraph

In this section, we study classes of graphs defined by forbidding a single graph as an induced subgraph. We start with some useful definitions.

A *co-bipartite* graph is the complement of a bipartite graph. A *split graph* is one whose vertex set can be partitioned into a clique and an independent set. The class of split graphs was first studied in [16] where a characterization of these graphs was proved.

Lemma 6. *[16] The class of split graphs is the class of $(2K_2, C_4, C_5)$-induced-subgraph-free graphs.*

We will need two results from [9] (see also [8] for the conference version) that we state as the following lemma.

Lemma 7. *[9]* SIMPLE MAX-CUT *is solvable in polynomial time in the class of P_4-induced-subgraph-free graphs and is* NP*-hard in the class of split graphs and in the class of co-bipartite graphs.*

Lemma 8. *Let H be a tree containing a vertex of degree at least 3. Then,* SIMPLE MAX-CUT *is* NP*-hard in the class of H-induced-subgraph-free graphs.*

Proof. A tree with a vertex of degree at least 3 has stability number at least 3. Co-bipartite graphs have stability number at most 2 and hence are H-induced-subgraph-free. The NP-hardness follows from Lemma 7.

Lemma 9. SIMPLE MAX-CUT *is* NP*-hard in the class of P_k-induced-subgraph-free graphs, for all $k \geq 5$.*

Proof. Split graphs are P_k-induced-subgraph-free graphs, for all $k \geq 5$. The NP-hardness follows from Lemma 7.

Theorem 3. *The* SIMPLE MAX-CUT *problem in the class of H-induced-subgraph-free graphs is:*

- *solvable in polynomial time, if H is an induced subgraph of P_4;*
- NP*-hard, otherwise.*

Proof. The first part of the theorem follows from Lemma 7.

For the second part, suppose H is not an induced subgraph of P_4. If it contains a cycle, then MAX-CUT is NP-complete in the class of H-induced-subgraph-free graphs by applying Lemma 2 with $k = |H|$. (Shortest cycle in a graph is necessarily induced.) We can assume that H is a forest. If it has a vertex of of degree 3,T then from Lemma 8, SIMPLE MAX-CUT is NP-hard. Thus, we can assume that H is a forest of paths. If one of the paths (connected components of H) contains more than 5 vertices, then the class of P_5-induced-subgraph-free graphs is contained in the class of H-induced-subgraph-free graphs and therefore SIMPLE MAX-CUT is NP-hard in H-induced-subgraph-free graphs from Lemma 9.

We can assume that H is a forest of induced subgraphs of P_4. If two of the connected components of H are not singletons, then H contains $2K_2$ and SIMPLE MAX-CUT is NP-hard in H-induced-subgraph-free graphs from Lemmas 6 and 7. Also, if H has three connected components, then the stability number of H is at least 3, and since co-bipartite graphs have stability number at most 2, the NP-hardness follows from Lemma 7. If H has two components, and one has at least 3 vertices, SIMPLE MAX-CUT is NP-hard for the same reason. Otherwise, H is an induced subgraph of P_4.

A similar theorem for MAX-CUT is perhaps less interesting but we include it here for completeness.

Theorem 4. *The* MAX-CUT *problem in the class of H-induced-subgraph-free graphs is:*

- *solvable in polynomial time, if H is clique on at most two vertices;*
- NP*-hard, otherwise.*

Proof. The first part of the theorem is trivial since K_2-induced-subgraph-free graphs are edgeless. For the second part, notice that MAX-CUT is NP-hard whenever SIMPLE MAX-CUT is. It remains to show that MAX-CUT is NP-hard for three classes of graphs: $K_1 \cup K_1$-induced-subgraph-free, $K_1 \cup K_2$-induced-subgraph-free, and P_3-induced-subgraph-free. However, each of these three classes contains the class of cliques and MAX-CUT is NP-hard on cliques. Indeed, every graph can be embedded in a clique using weights 0 and 1.

The techniques we use in this and the previous section have been dveloped by Alekseev and Lozin and applied to different problems in the context of boundary graph classes; see for example [2], [4], [3].

5 Forbidden Minor

In this section, we study classes of graphs defined by forbidding a single graph as a minor.

Definitions

Let G and H be two graphs with disjoint vertex sets, K_G and K_H cliques of size $k \geq 0$ in G and H respectively. A *k-sum* $G \oplus H$ is the graph obtained from G and H by identifying vertices of K_G and K_H (according to some bijection between the cliques) and then possibly removing some edges between the vertices of the identified clique.

A *single-crossing* graph is one that can be drawn on the plane with at most one crossing of edges. $K_{3,3}$ and K_5 are examples of single-crossing graphs. A graph H is an *apex* graph if it has a vertex v such that $H - v$ is planar. Clearly, single-crossing graphs are apex graphs. A graph H is a *k-apex* (for $k \geq 1$) if it has a set of vertices S of cardinality k such that $H \setminus S$ is planar. An apex is a 1-apex. We say that a k-apex is *strict*, when it is not $(k-1)$-apex, for $k > 1$, or planar for 1-apex.

Single-Crossing-Minor-Free Classes

Now we focus on graph classes defined by forbidding a single-crossing graph as a minor. We need to introduce a variant of MAX-CUT, called RESTRICTED MAX-CUT, that allows to specify for some vertices of the input graph, to which part of the cut they must belong.

First we need a decomposition theorem for graphs excluding a single-crossing as a minor. This algorithmic version is due to Demaine et al. [11] (see also [12] for the conference version).

Theorem 5. *[11] For a single-crossing H, there exists a constant c_H such that every H-minor-free graph G can be decomposed in time $O(n^4)$ into a series of clique-sum operations $G = G_1 \oplus \ldots \oplus G_m$, where each G_i ($1 \leq i \leq m$) is a minor of G and is either planar or its tree-width is at most c_H, and each $\oplus$ is an k-sum, for $0 \leq k \leq 3$.*

When the graph is weighted, the edges of graphs G_i that also exist in G have the same weights as in G; the edges of graphs G_i that do not exist in G are assigned weight 0.

To be able to use the decomposition theorem we need to analyze how a solution to MAX-CUT propagates through clique sums.

Lemma 10. *Let G and H be two graphs and $G \oplus H$ be their k-sum, for $0 \leq k \leq 3$. Given solutions to the instances of* RESTRICTED MAX-CUT *on G defined by considering all possible assignments of vertices from the k-clique to different parts of the cut, one can find in time $O(1)$ weights w^* on H such that $\mathbf{mc}_w(G \oplus H) = \mathbf{mc}_{w^*}(H) + T_\emptyset$, where $T_\emptyset$ is the value of* RESTRICTED MAX-CUT *on G when all vertices of the k-clique are required to belong to the same part of the cut.*

Proof. We will consider different cases depending on k.

CASE $k = 0, 1$. If $k = 0$, then $G \oplus H$ is a disjoint union of G and H. If $k = 1$, $G \oplus H$ is obtained from G and H by identifying one vertex. In both cases $\mathbf{mc}_w(G \oplus H) = \mathbf{mc}_w(H) + \mathbf{mc}_w(G)$. Setting $w^* = w$, we get $\mathbf{mc}_{w^*}(H) = \mathbf{mc}_w(H)$. Clearly $T_\emptyset = \mathbf{mc}_w(G)$. Hence, $\mathbf{mc}_w(G \oplus H) = \mathbf{mc}_w(H) + \mathbf{mc}_w(G) = \mathbf{mc}_{w^*}(H) + T_\emptyset$.

CASE $k = 2$. Let e_0 be the edge of the 2-clique. Let T_{e_0} be the value of RESTRICTED MAX-CUT on G when we require edge e_0 to be in the cut (= the endpoints of e_0 are forced to be in two different parts of the cut). Let $w^*(e_0) = T_{e_0} - T_\emptyset$; and for all edges $e \in E(H)$ different than e_0, $w^*(e) = w(e)$. Now, it is easy to verify that $\mathbf{mc}_w(G \oplus H) = \mathbf{mc}_{w^*}(H) + T_\emptyset$.

CASE $k = 3$. Let e_0, e_1, e_2 be the three edges of the 3-clique. Notice that a maximum-cut in G (in H, and in $G \oplus H$ as well), will always contain an even number of the edges of a 3-clique – either none, or exactly two. For two distinct edges $f, g \in \{e_0, e_1, e_2\}$, let $T_{f,g}$ be the value of RESTRICTED MAX-CUT on G when we require f and g to belong to the cut (= the common endpoint of edges f, g is forced to be in the other part of the cut than their "private" endpoints).

Let $w^*(e_j) = T_{e_{j-1}, e_{j+1}} - T_\emptyset$, for $0 \leq j \leq 2$, where all indices are taken modulo 3; and for all edges $e \in E(H)$ different than e_0, e_1, e_2, $w^*(e) = w(e)$. Now, it is easy to verify that $\mathbf{mc}_w(G \oplus H) = \mathbf{mc}_{w^*}(H) + T_\emptyset$.

We need to tailor the previous lemma to our needs.

Lemma 11. *Let c be a constant and G be a planar graph or a graph of tree-width at most c. Let H be a graph and $G \oplus H$ be a k-sum, for $0 \leq k \leq 3$ of G and H. One can find in polynomial time weights w^* on H and a constant d such that $\mathbf{mc}_w(G \oplus H) = \mathbf{mc}_{w^*}(H) + d$.*

Proof. To use Lemma 10 we need to show how to compute solutions to the instances of RESTRICTED MAX-CUT on G defined by considering all possible assignments of vertices from the k-clique to different parts of the cut.

Notice that increasing the weight of an edge by a large number M ($\geq$ than the sum of the weights of all the edges in the graph) will force the endpoints of this edge to belong to two different parts of every maximum cut in the new graph.

MAX-CUT in a graph with an edge contracted corresponds to RESTRICTED MAX-CUT in the original graph with the two endpoints of the contracted edge required to belong to the same part of the cut. (After contraction we remove loops; for multiple edges, we also remove them but the weight of the edge that remains equals to the sum of the weights of all the parallel ones.)

Hence, we can simulate RESTRICTED MAX-CUT by instances of MAX-CUT. Notice that edge contraction preserves planarity and does not increase tree-width. Since G is planar or of bounded tree-width, MAX-CUT can be solved in polynomial time [27], [23], [9]. Also, since $k \leq 3$, we only need to consider a constant number of different MAX-CUT instances on G.

Theorem 6. *Let H be a single-crossing graph.* MAX-CUT *can be solved in polynomial time in the class of H-minor-free graphs.*

Proof. First we apply Theorem 5 and find a decomposition of the input graph $G = G_1 \oplus \ldots \oplus G_m$. We will be processing graphs G_i from left to right. Let $G_1^* = G_1$.

We apply Lemma 11 to $G_i^* \oplus G_{i+1}$, $i = 1, \ldots, m-1$. Hence, there is a constant d and weights w^* such that $\mathbf{mc}_w(G_i^* \oplus G_{i+1}) = \mathbf{mc}_{w^*}(G_i) + d$. Let us denote G_i with the new weights by G_i^*. Notice that every G_i^* is either planar or a graph of tree-width at most c_H (c_H is the constant from Theorem 5), so Lemma 11 can be applied.

Finally, we conclude that there is a constant d' such that $\mathbf{mc}(G_m^*) + d' = \mathbf{mc}_w(G_1 \oplus \ldots \oplus G_m) = \mathbf{mc}_w(G)$.

H-Minor-Free Classes

Now we will look at classes of graphs defined by forbidding a single graph as a minor. We will need the following theorem due to Robertson and Seymour.

Theorem 7. *[29] For every planar graph H, there is a number w such that every planar graph with no minor isomorphic to H has tree-width $\leq w$.*

The following lemma is a consequence of Lemma 5 and Theorem 7.

Lemma 12. *Let H be a planar graph.* MAX-CUT *can be solved in the class of H-minor-free graphs in polynomial time.*

We mentioned in Section 2 that Barahona proved in [6] that SIMPLE MAX-CUT is NP-hard on K_6-minor-free graphs. Here is the precise statement of his result.

Theorem 8 (Theorem 5.1 in [6]). *Let G be a graph with a vertex v such that $G - v$ is a cubic planar graph.* SIMPLE MAX-CUT *is* NP*-complete in the class of such graphs.*

This class of graphs is in fact K_6-minor-free. However, as easily seen, it does not contain any strict k-apex graph (for $k \geq 2$) as a minor. Let us state it as a lemma.

Lemma 13. *Let H be a strict k-apex graph, for $k \geq 2$.* SIMPLE MAX-CUT *is* NP*-hard in the class of H-minor-free graphs.*

Considering the computational complexity of (SIMPLE) MAX-CUT in the class of H-minor-free graphs, we find an interesting situation. If H is planar, then MAX-CUT is solvable in polynomial time (Lemma 12); if H is a strict k-apex, for $k \geq 2$, then SIMPLE MAX-CUT is NP-complete (Lemma 13). What happens when H is an apex graph? – this is the missing case in a dichotomy theorem.

Graphs in classes of bounded orientable genus and graph classes obtained by excluding some single-crossing are H-minor-free for some apex graph H. The fact that MAX-CUT is solvable in polynomial time in those classes of graphs provide grounds for the following conjecture.

Conjecture. Let H be an apex graph. (SIMPLE) MAX-CUT is solvable in polynomial time in the class of H-minor-free graphs.

References

1. Vondrak, J., Galluccio, A., Loebl, M.: Optimization via enumeration: a new algorithm for the max cut problem. Mathematical Programming 90(2), 273–290 (2001)
2. Alekseev, V.E.: On easy and hard hereditary classes of graphs with respect to the independent set problem. Discrete Applied Mathematics 132(1-3), 17–26 (2003)
3. Alekseev, V.E., Boliac, R., Korobitsyn, D.V., Lozin, V.V.: Np-hard graph problems and boundary classes of graphs. Theor. Comput. Sci. 389(1-2), 219–236 (2007)
4. Alekseev, V.E., Korobitsyn, D.V., Lozin, V.V.: Boundary classes of graphs for the dominating set problem. Discrete Mathematics 285(1-3), 1–6 (2004)
5. Arbib, C.: A polynomial characterization of some graph partitioning problems. Inf. Process. Lett. 26(5), 223–230 (1988)
6. Barahona, F.: The Max-Cut problem on graphs not contractible to K_5. Oper. Res. Lett. 2(3), 107–111 (1983)
7. Bienstock, D., Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a forest. J. Comb. Theory, Ser. B 52(2), 274–283 (1991)
8. Bodlaender, H.L., Jansen, K.: On the complexity of the maximum cut problem. In: Enjalbert, P., Mayr, E.W., Wagner, K.W. (eds.) STACS 1994. LNCS, vol. 775, pp. 769–780. Springer, Heidelberg (1994)
9. Bodlaender, H.L., Jansen, K.: On the complexity of the maximum cut problem. Nordic J. of Computing 7(1), 14–31 (2000)
10. Demaine, E.D., Hajiaghayi, M.T., Kawarabayashi, K.: Algorithmic graph minor theory: Decomposition, approximation, and coloring. In: FOCS, pp. 637–646. IEEE Computer Society, Los Alamitos (2005)

11. Demaine, E.D., Hajiaghayi, M.T., Nishimura, N., Ragde, P., Thilikos, D.M.: Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. J. Comput. Syst. Sci. 69(2), 166–195 (2004)
12. Demaine, E.D., Hajiaghayi, M.T., Thilikos, D.M.: 1.5-approximation for treewidth of graphs excluding a graph with one crossing as a minor. In: Jansen, K., Leonardi, S., Vazirani, V.V. (eds.) APPROX 2002. LNCS, vol. 2462, pp. 67–80. Springer, Heidelberg (2002)
13. Demaine, E.D., Hajiaghayi, M.T., Thilikos, D.M.: Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. Algorithmica 41(4), 245–267 (2005)
14. Díaz, J., Kamiński, M.: Max-Cut and Max-Bisection are NP-hard on unit disk graphs. Theoretical Computer Science 377, 271–276 (2007)
15. Diestel, R.: Graph Theory, Electronic edn. Springer, Heidelberg (2005)
16. Földes, S., Hammer, P.L.: Split graphs. Congress. Numer., 311–315 (1978)
17. Grötschel, M., Nemhauser, G.L.: A polynomial algorithm for the Max-Cut problem on graphs without long odd cycles. Math. Prog. 29, 28–40 (1984)
18. Grötschel, M., Pulleyblank, W.R.: Weakly bipartite graphs and the Max-Cut problem. Oper. Res. Lett. 1, 23–27 (1981)
19. Guenin, B.: A characterization of weakly bipartite graphs. In: Bixby, R.E., Boyd, E.A., Ríos-Mercado, R.Z. (eds.) IPCO 1998. LNCS, vol. 1412, pp. 9–22. Springer, Heidelberg (1998)
20. Guenin, B.: A characterization of weakly bipartite graphs. Electronic Notes in Discrete Mathematics 5, 149–151 (2000)
21. Guenin, B.: A characterization of weakly bipartite graphs. J. Comb. Theory, Ser. B 83(1), 112–168 (2001)
22. Guruswami, V.: Maximum cut on line and total graphs. Discrete Applied Mathematics 92(2-3), 217–221 (1999)
23. Hadlock, F.: Finding a maximum cut of a planar graph in polynomial time. SIAM J. Comput. 4(3), 221–225 (1975)
24. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–104. Plenum Press, New York (1972)
25. Král, D., Kratochvíl, J., Tuza, Z., Woeginger, G.J.: Complexity of coloring graphs without forbidden induced subgraphs. In: Brandstädt, A., Van Bang, L. (eds.) WG 2001. LNCS, vol. 2204, pp. 254–262. Springer, Heidelberg (2001)
26. Lozin, V.V., Mosca, R.: Maximum independent sets in subclasses of p_5-free graphs. Inf. Process. Lett. 109(6), 319–324 (2009)
27. Orlova, G., Dorfman, Y.: Finding the maximum cut in a graph. Tekhnicheskaya Kibernetika (Engineering Cybernetics) 10, 502–506 (1972)
28. Robertson, N., Seymour, P.D.: Graph minors. I. Excluding a forest. J. Comb. Theory, Ser. B 35(1), 39–61 (1983)
29. Robertson, N., Seymour, P.D.: Graph minors. V. Excluding a planar graph. J. Comb. Theory, Ser. B 41, 92–114 (1986)
30. Robertson, N., Seymour, P.D.: Excluding a graph with one crossing. In: Robertson, N., Seymour, P.D. (eds.) Graph Structure Theory. Contemporary Mathematics, vol. 147, pp. 669–676. American Mathematical Society, Providence (1991)

31. Schrijver, A.: A short proof of Guenin's characterization of weakly bipartite graphs. J. Comb. Theory, Ser. B 85(2), 255–260 (2002)
32. Wagner, K.: Über eine Eigenshaft der ebenen Komplexe. Math. Ann. 114, 570–590 (1937)
33. Yannakakis, M.: Node- and edge-deletion NP-complete problems. In: STOC, pp. 253–264. ACM, New York (1978)