

Induced packing of odd cycles in planar graphs*

Petr A. Golovach[†] Marcin Kamiński[‡] Daniël Paulusma[§]

Dimitrios M. Thilikos[¶]

Abstract

An induced packing of odd cycles in a graph is a packing such that there is no edge in the graph between any two odd cycles in the packing. We prove that an induced packing of k odd cycles in an n -vertex graph can be found (if it exists) in time $2^{\mathcal{O}(k^{3/2})} \cdot n^{2+\epsilon}$ (for any constant $\epsilon > 0$) when the input graph is planar. We also show that deciding if a graph has an induced packing of two odd induced cycles is NP-complete.

1 Introduction

We assume that the reader is familiar with notions of graph theory; for those not defined here, we refer to the textbook by Diestel [10]. We consider finite and undirected graphs $G = (V_G, E_G)$ that have no loops and no multiple edges. The number of vertices and edges of a graph will be denoted by n and m , respectively.

Packing graphs. Packing graphs is a classic field of graph theory with many results and many conjectures. Packing is finding (usually) vertex- or edge-disjoint copies of graphs from some family (the guest graphs) into a fixed graph G (the host graph). There is a significant body of work on graph packing in the context of extremal combinatorics. The survey by Yap presents many results on packing graphs into a complete graph, focusing mainly on the famous Erdős-Sós conjecture [33]. This conjecture states that if the average degree of a graph G is strictly bigger than $k - 1$, then G contains every tree on k vertices.

*A preliminary version of this work appeared in the Proceedings of the 20th International Symposium on Algorithms and Computation, ISAAC 2009 [17].

[†]School of Engineering and Computing Sciences, University of Durham, United Kingdom, Petr.Golovach@durham.ac.uk, supported by EPSRC (EP/G043434/1).

[‡]The author is Chargé de Recherches du F.R.S. – FNRS; Département d’Informatique, Université Libre de Bruxelles, Belgium, Marcin.Kaminski@ulb.ac.be

[§]School of Engineering and Computing Sciences, University of Durham, United Kingdom, supported by EPSRC (EP/G043434/1), [Daniel.Paulusma@durham.ac.uk](mailto:Daniël.Paulusma@durham.ac.uk)

[¶]Department of Mathematics, National and Kapodistrian University of Athens, Greece, sedthilk@math.uoa.gr, supported by the project “Kapodistrias” (AII 02839/28.07.2008) of the National and Kapodistrian University of Athens (project code: 70/4/8757).

Packing has also been studied from the algorithmic point of view. The goal is usually to find the maximum number of disjoint copies of a guest graph in the host graph. A *matching* in a graph is a packing of vertex-disjoint K_2 's and it can be solved in polynomial time by Edmond's algorithm [14]. This means that the existence of a *perfect matching* (i.e. a matching which spans all the vertices of the host graph) can also be decided in polynomial time. Perfect matching has been generalized to perfect H -matching by Kirkpatrick and Hell, for graphs other than K_2 [26]. The authors prove that the problem is NP-complete for any graph H with at least 3 vertices. Berman et al. proved that the problem remains NP-complete even for planar host graphs [1].

Packing cycles. The problem of finding the maximum number of vertex-disjoint triangles in the input graph was proved to be NP-complete by Garey and Johnson [16]. However, there is a randomized $(43/83 - \varepsilon)$ -approximation algorithm, even for the weighted version of the problem presented by Hassin and Rubinfeld [20, 21].

There is a large collection of results on edge-disjoint packing of cycles that has applications in genome rearrangement in computational biology. The problem is studied by Caprara et al. who proved that it is APX-hard and can be approximated with the factor of $\mathcal{O}(\log n)$ by a greedy algorithm [4]. The approximation factor was later improved by Krivelevich et al. to $\mathcal{O}(\log^{1/2} n)$ [28]. Friggstad and Salavatipour showed that it is almost best possible [15]. They proved that it is impossible to approximate the edge-disjoint cycle problem within a ratio of $\mathcal{O}(\log^{1/2-\varepsilon} n)$ in polynomial time for any constant $\varepsilon > 0$, unless $\text{NP} \subseteq \text{ZPTIME}(n^{\text{polylog}(n)})$. They also note that the same results hold for packing vertex-disjoint cycles.

Heath and Vergara consider the problem of maximum edge-disjoint packing when the host and guest graphs are restricted to be planar [22]. Among other results they prove that the problem admits a polynomial-time solution if both graphs are trees, and is NP-complete when the guest graph is either a cycle or a tree with at least 3 edges.

Packing odd cycles. The problem of packing odd cycles in a graph was studied by Bruce Reed [30]. He was mainly concerned with the Erdős-Pósa property for odd cycles. In the conclusion, he gives an argument that packing k odd cycles in a graph is NP-complete when k is a part of the input. He also points out that a consequence of his results on the Erdős-Pósa property is a polynomial-time algorithm for packing odd cycles in a planar graph. Then he continues, "*As of the current writing, the author and P. Seymour believe they have a much more complicated algorithm for determining if a graph contains k vertex-disjoint odd cycles, k fixed. However, the proof of this result is extremely complicated and may well never be written down.*"

The recent progress in the theory of graph minors made it possible to have a proof of this theorem. In their recent work, Ken-ichi Kawarabayashi and Bruce Reed give an algorithm for packing k vertex-disjoint odd cycles in general graphs [25]. The running time of the algorithm is $\mathcal{O}(mn \cdot \alpha(m, n))$ for any fixed k , where $\alpha(m, n)$ is the inverse of the Ackermann function.

Even and odd holes. A hole is an induced cycle of length at least 4. Finding even and odd holes has been studied in the literature. The structure of graphs with no even hole is analyzed by Conforti et al. [6], leading to a polynomial-time algorithm for finding an even hole in a graph by the same authors [7]. Another algorithm, with a better running time, was proposed by Chudnovsky et al. [5].

Despite a seeming similarity, the complexity of detecting an odd hole in a graph has been a long standing open problem. The problem has been shown by Conforti et al. to be solvable in polynomial-time for graphs of bounded clique number [8]. Also, Bienstock has shown that the problem is NP-complete if the odd hole is required to contain a given vertex [2].

Our results. We are interested in induced packing of odd cycles. In this setting, odd cycles in the host graph are not only vertex-disjoint but also there is no edge in the host graph between two odd cycles in the packing. The term “induced packing” refers to this fact. The cycles themselves do not need to be induced, although this can be assumed without loss of generality. Indeed, let C be an odd cycle in an induced packing of odd cycles in a graph G . If C has a chord e , then the graph obtained from C by the addition of e is the union of two cycles C_1 or C_2 with one common edge e , and one of C_1 or C_2 , say C_1 , is odd. Replacing C in the induced packing by C_1 leads to another induced packing of the same number of odd cycles. Hence, any induced packing of odd cycles can easily be transformed to an induced packing of odd induced cycles that has the same size. While packing of k odd cycles is polynomial, for any fixed k [25], we prove that to decide if a graph contains an *induced* packing of two odd (induced) cycles is NP-complete. Induced packing is then, not surprisingly, much harder than packing. The two odd induced cycles in our hardness proof are in fact odd holes (and can be made arbitrarily long). While the problem of settling the complexity of detecting an odd hole is likely to be rather difficult to tackle, we show that to determine if a graph has two induced odd holes such that there is no edge between them is NP-complete.

We observe that it is NP-complete to decide whether a planar graph G has an induced packing of k odd cycles, if k is a constant that is part of the input. On the other hand, when k is a fixed constant, we show that an induced packing of k odd cycles in a planar graph can be found in polynomial time (if it exists). Our strategy is to solve the problem by

dynamic programming for graphs of small tree-width. If the tree-width is large, then the graph contains a large grid minor. In the model of the minor, we can either find an induced packing of k disjoint cycles, or a large bipartite graph. Our main technical result shows that, in the latter case, we can find an *irrelevant vertex* in the graph. The vertex is called irrelevant because one can remove it from the graph and be sure that the new graph has an induced packing of k odd cycles if and only if the original graph does.

Our motivation. Our motivation for considering induced packings of odd cycles is twofold.

1. Despite many existing results on graph packing, induced packings were not much studied before, possibly with the exception of the induced matching (see e.g. [3, 18, 23]). It seems interesting to understand how more difficult problems become when the “induced” restriction is added. Recently, there has been some interest in studying induced problems [24, 27].
2. We are interested in the applicability of the irrelevant vertex technique and how it can be used to improve parameter dependence. This technique was developed in Graph Minors project by Robertson and Seymour and recently found some applications [27, 29]. The existence of an irrelevant vertex is usually forced by large tree-width. Here, if the input graph has a large tree-width, we either find a solution, or an irrelevant vertex.

2 Background

In this section, we gather some definitions and present results from the literature that we will use later. A graph is *chordal* if it does not contain an induced cycle of length ≥ 4 . The *tree-width* of the graph G is the minimum size of the maximum clique minus 1, where the minimum is taken over all chordal supergraphs of G .

The $r \times r$ *grid* has all pairs (i, j) for $i, j = 0, 1, \dots, r - 1$ as the vertex set, and two vertices (i, j) and (i', j') are joined by an edge if and only if $|i - i'| + |j - j'| = 1$. The *side length* of an $r \times r$ grid is r . A connected graph G contains H as a *minor* if H can be obtained from G by a sequence of vertex or edge deletions, and edge contractions (removing loops and multiple edges).

Here are two useful lemmas.

Lemma 1 ((6.2) in [31]). *Let $r \geq 1$ be an integer. Every planar graph with no $r \times r$ grid minor has tree-width $\leq 6r - 5$.*

Lemma 2 (from [19]). *For every constant $\epsilon > 0$, there exists a constant $c_\epsilon > 3.5$ such that the side length of the largest square grid minor in a planar graph can be approximated with the factor of c_ϵ and the corresponding grid minor can be constructed in time $\mathcal{O}(n^{1+\epsilon})$.*

Let G be a planar graph. We assume that G has some fixed embedding in the plane and consider the corresponding plane graph. To simplify notations, we neither distinguish between a vertex of G and the point of the plane graph representing it nor between an edge of G and the line in the plane graph representing it. Consequently, we do not distinguish between a subgraph of G and the corresponding set of points of the plane.

Let C be a cycle of G . Then C has exactly two faces: the *inner* face and the *outer* face. Now let X be a set of points in the plane that may correspond to a vertex, an edge, or more general, to a subgraph of G . We say that X lies *inside* C if every point of X is contained in the inner face of C . Similarly, X lies *outside* C if every point of X is in the outer face of C . For two cycles C and Z , we define $\mu_Z(C)$ as the number of connected (topological) components of the set of points on the plane obtained from C by the removal of all the points of Z . We say that a cycle C *crosses* a cycle Z in a plane graph if there are two points $x, y \in C$, such that x is inside Z and y is outside Z . Note that C crosses Z if and only if Z crosses C .

Two subgraphs of a graph are called *mutually induced* if they are vertex-disjoint and no vertex of one subgraph is adjacent to a vertex of the other. A set of subgraphs of a graph is mutually induced if any two subgraphs of the set are mutually induced.

A sequence of cycles Z_1, \dots, Z_q in a plane graph G is called *nested*, if there exist disks $\Delta_1, \dots, \Delta_q$ such that for $i = 1, \dots, q$, Z_i bounds Δ_i , and $\Delta_{i+1} \subset \Delta_i$, for $i = 1, \dots, q-1$. We say that Z_1, \dots, Z_q are *strongly nested* if they are nested and mutually induced.

Now we are ready to formally define the problem we study here.

Problem k -INDUCED-PACKING-OF-ODD-CYCLES

Input: A planar graph G .
Output: A set of k mutually induced odd cycles in G if there exists one; NO otherwise.

First, we observe that this problem is hard, if k is part of the input.

Proposition 3. *k -INDUCED-PACKING-OF-ODD-CYCLES is NP-complete.*

Proof. Let G be a planar graph and k be an integer. For each vertex v of G add two new vertices and make them adjacent to each other, and also

adjacent to v . Then, the new graph has an induced packing of k odd cycles if and only if G has an independent set of size k . However, the problem of deciding if a planar graph has an independent set of size k , if k is a part of input, is NP-complete [16]. \square

The k -INDUCED-PACKING-OF-ODD-CYCLES problem is expressible in monadic second order logic. The seminal result of Courcelle implies that for any class of graphs whose tree-width is bounded, there exists a linear-time algorithm solving the problem in this class of graphs [9].

Even though the complexity is linear in n , the dependence on the parameter k is highly exponential. It is possible to obtain a better dependence on the parameter using dynamic programming on tree decompositions. There is a standard technique of dynamic programming in planar graphs of bounded tree-width that is applicable to our setting (Section 5 of [13]). It reduces the number of states in the dynamic programming by the use of sphere decomposition of a planar graph. This approach can find a solution to the problem in time $2^{\mathcal{O}(w)} \cdot n$, where w is the tree-width of the input graph. (See also [12] for a survey and [32] for an application of the same technique to similar problems.) As the method is standard and the dynamic programming is similar to the one in [13], we omit the proof of the following lemma.

Lemma 4. *k -INDUCED-PACKING-OF-ODD-CYCLES is solvable in $2^{\mathcal{O}(w)} \cdot n$ time for planar graphs of tree-width at most w .*

3 Induced packing of odd cycles

In this section, we present a combinatorial lemma on the existence of an irrelevant vertex, and state our algorithm together with the proof of its correctness.

Irrelevant vertex

Lemma 5. *Let k be a positive integer, G a plane graph, and Z_1 a cycle in G such that the graph induced by the vertices of Z_1 and the vertices inside Z_1 is bipartite. Also, let Z_1, \dots, Z_k be a sequence of strongly nested cycles in G and v a vertex inside Z_k that is not adjacent to any vertex of Z_k . Then, G has an induced packing of k odd cycles if and only if $G \setminus v$ does.*

Proof. The backward implication is clear. To prove the forward implication, let us assume that G has an induced packing of k odd cycles and let \mathcal{C} be one for which $\sum_{C \in \mathcal{C}} \sum_{i=1, \dots, k} \mu_{Z_i}(C)$ is minimum. Observe that each cycle $C \in \mathcal{C}$ contains a vertex that lies outside Z_1 , because C is odd and the

graph induced by the vertices of Z_1 and the vertices inside Z_1 is bipartite. We want to show that v is not contained in any cycle of \mathcal{C} .

The relation of being inside defines a poset on \mathcal{C} whose Hasse diagram $H_{\mathcal{C}}$ is a forest. We will work with $H_{\mathcal{C}}$ assuming that every tree in the forest is rooted; this can be done in a natural way. We define the *height* of a cycle in \mathcal{C} to be 1 for the leaves of $H_{\mathcal{C}}$. For other cycles in \mathcal{C} , the height is the minimum of the height of its children in $H_{\mathcal{C}}$ plus 1. Notice that the depth of a tree in $T_{\mathcal{C}}$ is at most k and therefore the maximum height of a cycle in \mathcal{C} is at most k .

Now it only remains to prove the following claim.

Claim. No cycle of height (at most) i from \mathcal{C} crosses Z_i , for all $i = 1, \dots, k$.

Let i be the smallest integer such that a cycle $C \in \mathcal{C}$ of height i crosses Z_i . Let Q be the set of edges of C which lie inside Z_i , and P be the set of edges of Z_i which lie inside C in the plane graph G . Note that the set of edges $(E_C \setminus Q) \cup P$ induces a subgraph of G that is a union of edge-disjoint cycles. Let \mathcal{R} denote the set of these cycles. Sets P and Q are disjoint and $P \cup Q$ also induces a subgraph of G that is a union of edge-disjoint cycles; each of these cycles belongs to the bipartite graph and has therefore even length. Hence, the total number of edges in P and Q is even. Since C is an odd cycle, this implies that the total length of the cycles in \mathcal{R} is odd. Consequently, there is an odd cycle C' in \mathcal{R} .

We now prove that C' is mutually induced with every cycle in $\mathcal{C} \setminus \{C\}$. In order to obtain a contradiction, suppose that C' and some cycle $C'' \in \mathcal{C} \setminus \{C\}$ are not mutually induced. Then, by definition, there exist two vertices s and t in C' and C'' , respectively, such that either $s = t$ or s is adjacent to t . Because the cycles in \mathcal{C} are mutually induced, s is not a vertex of C . This means that s is a vertex of Z_i , and hence it lies inside C . Consequently, t cannot be outside C . This means that C'' lies inside C . Then, by definition, the height j of C'' is less than the height i of C , so we have $i > j \geq 1$. Recall that i is the smallest integer such that a cycle in \mathcal{C} of height i crosses Z_i . Therefore, C'' does not cross Z_j , and consequently, since Z_1, \dots, Z_k are strongly nested, C'' does not cross Z_{i-1} . We conclude that t either lies outside Z_{i-1} or is a vertex of Z_{i-1} . Because Z_1, \dots, Z_k are strongly nested and $s \in Z_i$, this means that $s \neq t$ and s cannot be adjacent to t either; a contradiction. Therefore, $(\mathcal{C} \setminus \{C\}) \cup \{C'\}$ is an induced packing of k cycles in G . However, $\sum_{i=1, \dots, k} \mu_{Z_i}(C') < \sum_{i=1, \dots, k} \mu_{Z_i}(C)$; a contradiction with our choice of \mathcal{C} . Hence, we have shown the Claim, and thus completed the proof of Lemma 5. \square

The algorithm

Here is our algorithm for solving k -INDUCED-PACKING-OF-ODD-CYCLES. We prove its correctness and analyse its running time in Theorem 6.

Algorithm k -INDUCED-PACKING-OF-ODD-CYCLES

Input: A planar graph G .

Output: A collection of k mutually induced odd cycles in G if there exists one; NO otherwise.

1. **Run** the algorithm from Lemma 2 and construct a grid minor M .
2. If the side length of M is less than $\lceil \sqrt{k} \rceil (c_\epsilon \cdot k + 2) - 1$, then **solve** the problem using the algorithm of Lemma 4 and **stop**.
3. Otherwise, **find** k mutually induced copies of a square grid of side length $c_\epsilon \cdot k + 1$ in M .
4. For every copy, **check** if the model of the copy in G is bipartite.
5. If the models of all copies are non-bipartite, **return** an induced packing of k odd cycles and **stop**.
6. Otherwise, **construct** k mutually induced odd cycles in a bipartite copy H .
7. **Find** an irrelevant vertex v in H using Lemma 5.
8. **Run** the algorithm for $G \setminus v$.

Theorem 6. *For every constant $\epsilon > 0$, the algorithm k -INDUCED-PACKING-OF-ODD-CYCLES is correct and runs in time $2^{\mathcal{O}(k^{3/2})} \cdot n^{2+\epsilon}$.*

Proof. Let us suppose that the input graph G contains an induced packing of k odd cycles. Let $\epsilon > 0$ be a constant and $c_\epsilon > 3.5$ be the approximation factor from Lemma 2 corresponding to ϵ .

If the algorithm from Lemma 2 in Step 1 finds a grid minor M of side length less than $\lceil \sqrt{k} \rceil (c_\epsilon \cdot k + 2) - 1$, then the largest grid minor is of side length less than $\lceil \sqrt{k} \rceil (c_\epsilon \cdot k + 2) - 1$. This, from Lemma 1, means that the tree-width of the graph is bounded by a constant and the problem can be solved by Lemma 4. The induced packing of k odd cycles will be found in Step 2.

If the side length of M is at least $\lceil \sqrt{k} \rceil (c_\epsilon \cdot k + 2) - 1$, then M contains k mutually disjoint copies of the $(c_\epsilon \cdot k + 1) \times (c_\epsilon \cdot k + 1)$ grid (Step 3). For each copy, we look at the graph induced by the union of branch sets in G corresponding to the vertices of the copy. If for every copy the graph is non-bipartite, there is an induced packing of k odd cycles (Step 4 & 5).

Otherwise, there is a copy H whose model is bipartite. In this copy, since $c_\epsilon > 3.5$, peeling off the $(c_\epsilon \cdot k + 1) \times (c_\epsilon \cdot k + 1)$ grid, we find k nested,

mutually induced cycles Z'_1, \dots, Z'_k . The central vertex of the grid v' is also mutually induced with the cycles. Notice that the model of cycle Z'_i in G , for all $i = 1, \dots, k$, contains a cycle passing through the all branch sets corresponding to the vertices of Z'_i . Therefore, we construct a collection of k nested, mutually induced cycles Z_1, \dots, Z_k (Step 6). These cycles, together with a vertex v from the model of v' in G (Step 7), satisfy conditions of Lemma 5. By the Lemma 5, the induced packing of k odd cycles will be found recursively (Step 8).

Notice that Step 8 of the algorithm will be executed at most n times and Step 1 is the most time-consuming step of the algorithm, taking $\mathcal{O}(n^{1+\epsilon})$ time. Step 2 takes time $2^{\mathcal{O}(k^{3/2})} \cdot n$ and the algorithm runs in time $2^{\mathcal{O}(k^{3/2})} \cdot n^{2+\epsilon}$. \square

4 Two induced disjoint odd cycles

Theorem 7. *It is NP-complete to decide whether a given graph G contains two mutually induced odd induced cycles.*

Proof. We reduce the well known NP-complete 3-SATISFIABILITY problem [16]. It is known that this problem remains NP-complete even for the case when each Boolean variable occurs at most two times in positive and at most two times in negations. We use this variant of the problem for our reduction. Let x_1, \dots, x_n be Boolean variables and let C_1, \dots, C_m be clauses of the given Boolean formula Φ in the conjunctive normal form. We construct a graph G as follows.

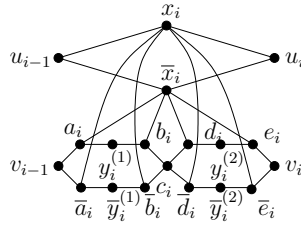


Figure 1: First stage of construction of G

First, we introduce vertices u_0, \dots, u_n and vertices v_0, \dots, v_n . For each $1 \leq i \leq n$, the following is done (see Fig. 1):

- Add vertices x_i, \bar{x}_i and edges $u_{i-1}x_i, x_iu_i, u_{i-1}\bar{x}_i$ and \bar{x}_iu_i . Denote by P_i and \bar{P}_i the paths $u_{i-1}x_iu_i$ and $u_{i-1}\bar{x}_iu_i$ respectively.
- Construct vertices $a_i, b_i, c_i, d_i, e_i, \bar{a}_i, \bar{b}_i, \bar{c}_i, \bar{d}_i, \bar{e}_i$ and $y_i^{(1)}, y_i^{(2)}, \bar{y}_i^{(1)}, \bar{y}_i^{(2)}$, and then add edges $v_{i-1}a_i, a_iy_i^{(1)}, y_i^{(1)}b_i, b_iy_i^{(2)}, y_i^{(2)}d_i, d_iy_i^{(1)}, y_i^{(1)}c_i, c_iy_i^{(2)}, y_i^{(2)}e_i, e_iy_i^{(1)}, y_i^{(1)}\bar{a}_i, \bar{a}_iy_i^{(2)}, y_i^{(2)}\bar{b}_i, \bar{b}_iy_i^{(1)}, y_i^{(1)}\bar{c}_i, \bar{c}_iy_i^{(2)}, y_i^{(2)}\bar{d}_i, \bar{d}_iy_i^{(1)}, y_i^{(1)}\bar{e}_i, \bar{e}_iy_i^{(2)}$.

$\bar{a}_i \bar{y}_i^{(1)}$, $\bar{y}_i^{(1)} \bar{b}_i$, $\bar{b}_i c_i$, $c_i \bar{d}_i$, $\bar{d}_i \bar{y}_i^{(2)}$, $\bar{y}_i^{(2)} \bar{e}_i$ and $\bar{e}_i v_i$. Denote by Q_i the path $v_{i-1} a_i y_i^{(1)} b_i c_i d_i y_i^{(2)} d_i v_i$ and let $\bar{Q}_i = v_{i-1} \bar{a}_i \bar{y}_i^{(1)} \bar{b}_i c_i \bar{d}_i \bar{y}_i^{(2)} \bar{d}_i v_i$.

- Add edges $x_i \bar{a}_i$, $x_i \bar{b}_i$, $x_i \bar{d}_i$, $x_i \bar{e}_i$, $\bar{x}_i a_i$, $\bar{x}_i b_i$, $\bar{x}_i d_i$ and $\bar{x}_i e_i$.

Now we introduce vertices w_0, \dots, w_m . For each $1 \leq j \leq m$, three vertices $z_j^{(1)}, z_j^{(2)}, z_j^{(3)}$ are constructed and joined by edges with w_{j-1} and w_j . We denote by $R_j^{(r)}$ the path $w_{j-1} z_j^{(r)} w_j$ for $r = 1, 2, 3$. Assume that the clause C_j contains literals l_1, l_2, l_3 . For each literal l_r , $1 \leq r \leq 3$, the following is done (see Fig. 2):

- If $l_r = x_i$ for some $1 \leq i \leq n$, then the edge $x_i z_j^{(r)}$ is added, and also the vertex $z_j^{(r)}$ is joined by an edge with $y_i^{(1)}$ if l_r is the first occurrence of the literal x_i in the Boolean formula, and $z_j^{(r)}$ is joined with $y_i^{(2)}$ if l_r is the second occurrence of x_i .
- If $l_r = \bar{x}_i$ for some $1 \leq i \leq n$, then the edge $\bar{x}_i z_j^{(r)}$ is added, and also the vertex $z_j^{(r)}$ is joined by an edge with $\bar{y}_i^{(1)}$ if l_r is the first occurrence of the literal \bar{x}_i in the Boolean formula, and $z_j^{(r)}$ is joined with $\bar{y}_i^{(2)}$ if l_r is the second occurrence of \bar{x}_i .

Finally, we add the edges $u_0 u_n$ and $v_0 w_0$, introduce the vertex s and join the vertices v_n and w_m with s by edges.

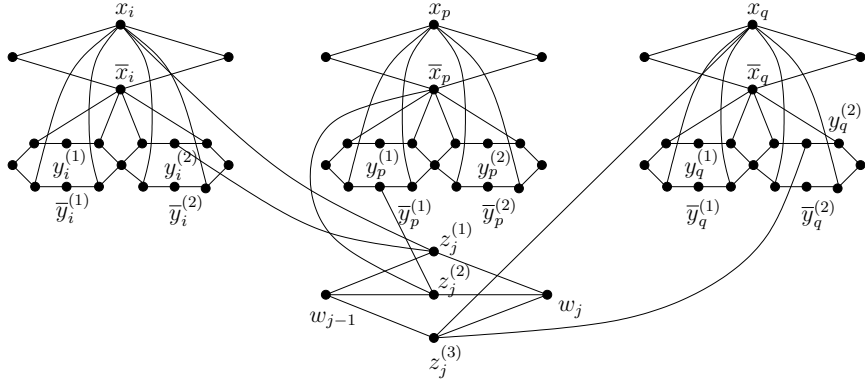


Figure 2: Second stage of construction of G , the clause C_j contains literals x_i (second occurrence), \bar{x}_p (first occurrence) and x_q (second occurrence)

We claim that Φ can be satisfied if and only if there are two disjoint induced odd cycles S_1 and S_2 in G .

Suppose that Φ can be satisfied, and variables x_1, \dots, x_n have corresponding truth assignment. We construct S_1 from paths P_i and \bar{P}_i using them as segments. For each $1 \leq i \leq n$, we include P_i in the cycle

if $x_i = \text{false}$ and \overline{P}_i is included if $x_i = \text{true}$. The construction of S_1 is completed by adding the edge u_0u_n . The cycle S_2 is constructed by using paths Q_i, \overline{Q}_i and $R_j^{(r)}$. For each $1 \leq i \leq n$, the path Q_i is included in S_2 if $x_i = \text{false}$ and \overline{Q}_i is included if $x_i = \text{true}$. Now we consider clauses C_j for $1 \leq j \leq m$. Suppose that C_j contains literals l_1, l_2, l_3 . Since $\Phi = \text{true}$, there is a literal $l_r = \text{true}$, and we include in S_2 the path $R_j^{(r)}$. Finally, the edge v_0w_0 and the path v_nsw_m is added to the cycle. It is easy to check that S_1 and S_2 are disjoint induced odd cycles, and they have no adjacent vertices.

Assume now that G contains two disjoint induced odd cycles S_1 and S_2 . Notice that the graph obtained from G by removal of the edges u_0u_n and v_0w_0 is bipartite. Since S_1 and S_2 are odd, they have to include these edges. Suppose without loss of generality that u_0u_n is included in S_1 and v_0w_0 is included in S_2 . We need now the following claim.

Claim. For any $1 \leq i \leq n$,

- either P_i or \overline{P}_i is included in S_1 as a segment,
- if P_i is included in S_1 , then Q_i is included in S_2 , and if S_1 contains \overline{P}_i , then S_2 contains \overline{Q}_i .

For each $1 \leq j \leq m$, S_2 includes one of the paths $R_j^{(1)}, R_j^{(2)}, R_j^{(3)}$.

Proof of the Claim. First we prove that for any $1 \leq i \leq n$, either P_i or \overline{P}_i is included in S_1 as a segment, and if P_i is included in S_1 , then Q_i is included in S_2 , and if S_1 contains \overline{P}_i , then S_2 contains \overline{Q}_i . Suppose that this claim holds for the lesser values of the parameter, i.e. S_1 contains the edge u_0u_n and either the path P_k or \overline{P}_k for $1 \leq k < i$, and similarly S_2 contains the edge v_0w_0 and either the path Q_k or \overline{Q}_k for $1 \leq k < i$. Assume without loss of generality that if $i > 1$, then S_1 includes P_{i-1} (the case, when \overline{P}_{i-1} is in S_1 , is symmetric). The vertex u_{i-1} is incident with two edges of S_1 . Since S_1 is an induced cycle and we assume that S_1 contains P_{i-1} , we find that S_1 cannot include $u_{i-1}\overline{x}_{i-1}$. Hence, S_1 contains either the edge $u_{i-1}x_i$ or the edge $u_{i-1}\overline{x}_i$. By the symmetry of our construction, we may assume that $u_{i-1}x_i$ is in S_1 . Cycles S_1 and S_2 have no adjacent vertices. Our assumptions that S_1 contains P_{i-1} and S_2 contains either Q_{i-1} or \overline{Q}_{i-1} if $i > 1$ imply that S_2 includes Q_{i-1} . The vertex v_i is incident with two edges in the cycle. Since P_{i-1} is in S_1 , S_2 cannot include the edge $v_{i-1}\overline{e}_{i-1}$, and because P_i is in S_1 , S_2 cannot include the edge $v_{i-1}\overline{a}_i$. Hence, this cycle contains the edges $v_{i-1}a_i$ and $a_iy_i^{(1)}$. Notice that if $y_i^{(1)}$ is adjacent to some vertex $z_j^{(r)}$, then x_i is also adjacent to this vertex and S_2 cannot include the edge $y_i^{(i)}z_j^{(r)}$. It means that S_2 contains the edges $y_i^{(1)}b_i$ and b_ic_i . By similar arguments we prove that S_2 includes the edges c_id_i , $d_iy_i^{(2)}$, $y_i^{(2)}e_i$ and e_iv_i . Therefore, S_2

includes Q_i . Now we return to the cycle S_1 . Since all vertices $z_j^{(r)}$ adjacent to x_i are adjacent either to $y_i^{(1)}$ or $y_i^{(2)}$, S_1 cannot include edges $x_i z_j^{(r)}$. So, it contains the edge $x_i u_i$, and together with the fact that S_1 includes $u_{i-1} x_i$, it means that P_i is a segment of S_1 .

Now we prove that for each $1 \leq j \leq n$, S_2 includes one of the paths $R_j^{(1)}, R_j^{(2)}, R_j^{(3)}$. Again suppose that this claim holds for the lesser values of the parameter, and S_2 contains the edge $v_0 w_0$ and one of the paths $R_k^{(1)}, R_k^{(2)}, R_k^{(3)}$ for $1 \leq k < j$. Since S_2 is an induced path, it includes one of the edges $w_{j-1} z_j^{(1)}, w_{j-1} z_j^{(2)}, w_{j-1} z_j^{(3)}$. Assume that the cycle contains $w_{j-1} z_j^{(3)}$. Suppose that $z_j^{(1)}$ is adjacent to some vertex x_i , and therefore to one of vertices $y_i^{(1)}$ or $y_i^{(2)}$ (say, the vertex $y_i^{(1)}$). Notice that in this case \bar{x}_i is a vertex of S_1 by the first part of the claim. The cycle S_2 cannot contain the edge $z_j^{(1)} x_i$ since x_i is adjacent to the vertex u_{i-1} which is included in S_1 . If S_2 contains $z_j^{(1)} y_i^{(1)}$, then it should contain either the edge $y_i^{(1)} a_i$ or $y_i^{(1)} b_i$, but these vertices are adjacent to \bar{x}_i . Same arguments can be used for the case when $z_j^{(1)}$ is adjacent to some vertex \bar{x}_i . Hence S_2 includes the edge $z_j^{(1)} w_j$ and we conclude that $R_j^{(1)}$ is a segment of S_2 . \square

Using this claim we assign values to the Boolean variables x_1, \dots, x_n : set $x_i = \text{true}$ if \bar{P}_i is a segment of S_1 and $x_i = \text{false}$ if P_i is a segment of S_1 . Consider clauses C_1, \dots, C_m . Suppose that the clause C_j contains literals l_1, l_2, l_3 which correspond to vertices $z_j^{(1)}, z_j^{(2)}, z_j^{(3)}$. The cycle S_2 contains one of the paths $R_j^{(1)}, R_j^{(2)}, R_j^{(3)}$, say the path $R_j^{(1)}$ which goes through $z_j^{(1)}$. This vertex is adjacent to one of the vertices x_i or \bar{x}_i , and this vertex is not included in S_1 . It follows that by our truth assignment $l_i = \text{true}$. Since it holds for each $1 \leq j \leq m$, $\Phi = \text{true}$.

To conclude the proof of the theorem, it remains to note that G has $15n + 4m + 4$ vertices, and therefore can be constructed in polynomial time. \square

5 Discussion

1. Induced packing of odd cycles is a packing such any two odd cycles are at distance at least 2 from each other. One can consider a d -induced packing problem, in which the cycles are at distance at least k from each other. Our algorithm can be easily modified to handle this variant of the problem in the same asymptotic running time.
2. There is another approach to solving the problem of induced packing of odd cycles on planar graphs that uses results on outerplanarity. One could use a layer decomposition of a planar graph that is the

collection of cycles obtained by successively removing cycles bounding the outer face. By using this layer decomposition, Lemma 4 and a modified version of Lemma 5, we can solve the k -INDUCED-PACKING-OF-ODD-CYCLES problem in time $2^{\mathcal{O}(k^2)} \cdot n$. Compared to our result, the complexity is better in n but the parameter dependence, which is the main focus of this paper, is worse. We observed in [17] that there is a $2^{\mathcal{O}(k^2)} \cdot n^2$ algorithm for solving the problem using a layer decomposition. We thank an anonymous referee for pointing out that the dependence on n can be made linear.

3. We showed how to solve the problem in the class of planar graphs but we believe that this can be generalized to larger classes.

Conjecture. k -INDUCED-PACKING-OF-ODD-CYCLES can be solved in polynomial time for any class of graphs in which genus (orientable or not) is bounded.

We leave this problem for future research.

Acknowledgments. We thank an anonymous reviewer for useful suggestions that helped us to improve the readability of our paper.

References

- [1] F. Berman, D. S. Johnson, F. T. Leighton, P. W. Shor, and L. Snyder. Generalized planar matching. *J. Algorithms*, 11(2):153–184, 1990.
- [2] D. Bienstock. On the complexity of testing for odd holes and induced odd paths. *Discrete Mathematics*, 90(1):85–92, 1991.
- [3] K. Cameron. Induced matchings, *Discrete Applied Mathematics*, 24: 97–102, 1989.
- [4] A. Caprara, A. Panconesi, and Romeo Rizzi. Packing cycles in undirected graphs. *J. Algorithms*, 48(1):239–256, 2003.
- [5] M. Chudnovsky, K. Kawarabayashi, and P. Seymour. Detecting even holes. *J Graph Theory*, 48(2):85–111, 2005.
- [6] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Part I: Decomposition theorem. *J Graph Theory*, 39:6–49, 2002.
- [7] M. Conforti, G. Cornuéjols, A. Kapoor, and K. Vušković. Part II: Recognition algorithm. *J Graph Theory*, 40:238–266, 2002.

- [8] M. Conforti, G. Cornuéjols, X. Liu, K. Vuskovic, and G. Zambelli. Odd hole recognition in graphs of bounded clique size. *SIAM J. Discrete Math.*, 20(1):42–48, 2006.
- [9] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.
- [10] R. Diestel. *Graph Theory*. Springer-Verlag, Electronic Edition, 2005.
- [11] Y. Dong, D.-Z. Du, and O. H. Ibarra, editors. *Algorithms and Computation, 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, volume 5878 of *Lecture Notes in Computer Science*. Springer, 2009.
- [12] F. Dorn, F. V. Fomin, and D. M. Thilikos. Catalan structures and dynamic programming in H -minor-free graphs. In Shang-Hua Teng, editor, *SODA*, pages 631–640. SIAM, 2008.
- [13] F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica*, 58(3):790–810, 2010.
- [14] J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [15] Z. Friggstad and M. R. Salavatipour. Approximability of packing disjoint cycles. In T. Tokuyama, editor, *ISAAC*, volume 4835 of *Lecture Notes in Computer Science*, pages 304–315. Springer, 2007.
- [16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [17] P. A. Golovach, M. Kaminski, D. Paulusma, and D. M. Thilikos. Induced packing of odd cycles in a planar graph. In Dong et al. [11], pages 514–523.
- [18] M.C. Golumbic and M. Lewenstein. New results on induced matchings. *Discrete Applied Mathematics*, 101:157–165, 2000.
- [19] Q.-P. Gu and H. Tamaki. Constant-factor approximations of branch-decomposition and largest grid minor of planar graphs in $O(n^{1+\epsilon})$ time. In Dong et al. [11], pages 984–993.
- [20] R. Hassin and S. Rubinstein. An approximation algorithm for maximum triangle packing. *Discrete Applied Mathematics*, 154(6):971–979, 2006.

- [21] R. Hassin and S. Rubinstein. Erratum to “An approximation algorithm for maximum triangle packing”: [Discrete Applied Mathematics 154 (2006) 971-979]. *Discrete Applied Mathematics*, 154(18):2620, 2006.
- [22] L. S. Heath and J. P. C. Vergara. Edge-packing in planar graphs. *Theory Comput. Syst.*, 31(6):629–662, 1998.
- [23] I. Kanj , M. J. Pelsmayer , M. Schaefer, and G. Xia. On the induced matching problem. *Journal of Computer and System Sciences*, 77(6): 1058–1070, 2011.
- [24] K. Kawarabayashi and Y. Kobayashi. The induced disjoint paths problem. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *IPCO*, volume 5035 of *Lecture Notes in Computer Science*, pages 47–61. Springer, 2008.
- [25] K. Kawarabayashi and B. A. Reed. Odd cycle packing. In M. Mitzenmacher and L. J. Schulman, editors, *STOC*, pages 695–704. ACM, 2010.
- [26] D. G. Kirkpatrick and P. Hell. On the complexity of general graph factor problems. *SIAM J. Comput.*, 12(3):601–609, 1983.
- [27] Y. Kobayashi and Ke. Kawarabayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In Claire Mathieu, editor, *SODA09*, pages 1146–1155. SIAM, 2009.
- [28] M. Krivelevich, Z. Nutov, M. R. Salavatipour, J. Yuster, and R. Yuster. Approximation algorithms and hardness results for cycle packing problems. *ACM Transactions on Algorithms*, 3(4), 2007.
- [29] D. Marx. Chordal deletion is fixed-parameter tractable. In Fedor V. Fomin, editor, *WG*, volume 4271 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2006.
- [30] B. A. Reed. Mangoes and blueberries. *Combinatorica*, 19(2):267–296, 1999.
- [31] N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a planar graph. *J. Comb. Theory, Ser. B*, 62(2):323–348, 1994.
- [32] I. Sau and D. M. Thilikos. Subexponential parameterized algorithms for degree-constrained subgraph problems on planar graphs. *J. Discrete Algorithms*, 8(3):330–338, 2010.
- [33] H. P. Yap. Packing of graphs - a survey. *Discrete Mathematics*, 72(1-3):395–404, 1988.