Planar Induced Subgraphs of Sparse Graphs

Glencora Borradaile[1], David Eppstein[2], and Pingan Zhu[1]

[1] Oregon State University, Corvallis OR 97330, USA
[2] University of California, Irvine, Irvine CA 92697, USA

Abstract. We show that every graph has an induced pseudoforest of at least $n - m/4.5$ vertices, an induced partial 2-tree of at least $n - m/5$ vertices, and an induced planar subgraph of at least $n - m/5.2174$ vertices. These results are constructive, implying linear-time algorithms to find the respective induced subgraphs. We also show that the size of the largest K_h-minor-free graph in a given graph can sometimes be at most $n - m/6 + o(m)$.

1 Introduction

Planarization, a standard step in drawing non-planar graphs, involves replacing edge crossings with new vertices to form a planar graph with paths that represent the original graph's edges. *Incremental planarization*, does this by finding a large planar subgraph of the given graph, and then adding the remaining features of the input graph one at a time [6]. Thus, it is of interest to study the algorithmic problem of finding planar subgraphs that are as large as possible in a given graph. Unfortunately, this problem is NP-hard and, more strongly, MAX-SNP-hard [4]. A trivial algorithm, finding an arbitrary spanning tree, achieves an approximation ratio of $\frac{1}{3}$, and by instead searching for a partial 2-tree this ratio can be improved to $\frac{2}{5}$ [4]. The equivalent complementary problem, deleting a minimum number of edges to make the remaining subgraph planar, is fixed-parameter tractable and linear time for any fixed value of the parameter [12].

In this paper we study a standard variant of this problem: finding a large planar *induced* subgraph of a given graph. In the context of the planarization problem, one possible application of finding this type of planar subgraph would be to apply incremental planarization in a drawing style where edges are represented as straight-line segments. A planar induced subgraph can always be drawn without crossings in this style, by Fáry's theorem, after which the partial drawing could be used to guide the placement of the remaining vertices. As with the previous problem, the induced planar subgraph problem is NP-hard, but again there is a linear-time fixed-parameter tractable algorithm for the equivalent problem of finding the smallest number of vertices to delete so that the remaining induced subgraph is planar [11].

Because of the difficulty of finding an exact solution to this problem, we instead seek worst-case guarantees: what is the largest size of a planar induced subgraph that we can be guaranteed to find within a graph of a given size? In this we are inspired by a paper of Alon, Mubayi, and Thomas [1], who showed

C. Duncan and A. Symvonis (Eds.): GD 2014, LNCS 8871, pp. 1–12, 2014.

that every *triangle-free* input graph with n vertices and m edges contains an induced forest with at least $n-m/4$ vertices. This is tight, as shown by an input graph in the form of $n/4$ disjoint copies of a 4-cycle. Induced forests are a special case of induced planar subgraphs, and so this result guarantees the existence of an induced planar subgraph of $n-m/4$ vertices. As we show, an analogous improvement to the one in the approximation ratio for planar subgraphs can be obtained by seeking instead an induced partial 2-tree. Rossmanith [3] has posed the question: does every graph have an induced planar subgraph of size $n-m/6$? We shrink the gap on the worst-case bounds for the size of a planar induced subgraph by showing that every graph (not necessarily triangle-free) has an induced planar subgraph of $n-m/5.2174$ vertices and that there exist graphs for which the largest induced planar subgraph is not much larger than $n-m/6$ vertices.

1.1 New Results

We prove the following results:

Theorem 1. *Every graph with n vertices and m edges has an induced pseudoforest with at least $n-\frac{2m}{9}$ vertices.*

Theorem 2. *Every graph with n vertices and m edges has an induced subgraph with treewidth at most 2 and with at least $n-\frac{m}{5}$ vertices.*

Theorem 3. *Every graph with n vertices and m edges has an induced planar subgraph with treewidth at most 3 and with at least $n-\frac{23m}{120}$ vertices.*

These three theorems can be implemented as algorithms which take linear time to find the induced subgraphs described by the theorems.

Theorem 4. *For every integer h, there is a family of graphs such that for any graph in this family with n vertices and m edges, the largest K_h-minor-free induced subgraph has at most $n-\frac{m}{6}+O(\frac{m}{\log m})$ vertices.*

The bounds of Theorems 1 and 2 are tight, even for larger classes of induced subgraphs. In particular, there exist graphs for which the largest induced outerplanar subgraph has size at most $n-m/4.5$, so the bound of Theorem 1 is tight for any class of graphs between the pseudoforests and the outerplanar graphs. There also exist graphs for which the largest induced K_4-free induced subgraph has $n-m/5$ vertices, so Theorem 2 is tight for every family of graphs between the treewidth 2 graphs and the graphs with no 4-clique.

1.2 Related Work

The worst-case size of the largest induced planar subgraph has been studied previously by Edwards and Farr [7], who proved a tight bound of $3n/(d+1)$ on its size as a function of the maximum degree d of the given graph. In contrast, by depending on the total number of edges rather than the maximum degree, our algorithms are sensitive to graphs with heterogeneous vertex degrees, and can construct larger induced subgraphs when the number of high-degree vertices is relatively small. Additionally, the algorithm given by Edwards and Farr is slower than ours, taking $O(mn)$ time. In a follow-up paper, Morgan and Farr [16] gave additional bounds on induced outerplanar subgraphs, and provided experimental results on the performance of their algorithms. A second paper by Edwards and Farr [8], like our Theorem 2, gives bounds on the size of the largest induced partial 2-tree in terms of n and m, which is $\frac{3n}{2m/n+1}$ for $m \geq 2n$. However, their bounds are asymptotically worse than Theorem 2 when $2n < m < 2.5n$ and require an additional assumption of connectivity for smaller values of m. A third paper by the same authors [9] gives improved bounds that are more difficult to state as a formula.

For some other graph classes than the ones we study, it is possible to prove trivial bounds on the size of the largest induced subgraph in the class, of a similar form to the bounds of Alon et al. and of our theorems. By repeatedly finding and removing a vertex of degree ≥ 1, one can obtain an independent set of at least $n - m$ vertices, and the example of a perfect matching shows this to be tight. By repeatedly finding and removing a vertex of degree ≥ 2, one can obtain a matching of at least $n - m/2$ vertices, and the example of the disjoint union of $n/3$ two-edge paths shows this to be tight. And by repeatedly finding and removing either a vertex of degree ≥ 3 or a vertex that is part of a 2-regular cycle, one can obtain a linear forest (forest with maximum degree 2) of $n - m/3$ vertices; the example of the disjoint union of $n/3$ triangles shows this to be tight even for more general forests.

Table 1 provides a comparison of these new results with previous known results on induced planar subgraphs of various types and with the trivial bounds for independent sets, matchings, and linear forests.

2 Preliminaries

For a graph G, we define $n(G)$ to be the number of vertices and $m(G)$ to be the number of edges in G. We drop the argument and write n and m when the choice of G is clear from context.

A subset S of the vertices of G corresponds to an *induced subgraph* $G[S]$, a graph having S as its vertices and having as edges every edge in G that has both endpoints in S. Equivalently, $G[S]$ may be constructed from G by deleting every vertex that is not in S and every edge that has at least one endpoint outside S.

A *pseudoforest* is an undirected graph in which every connected component has at most one cycle. Equivalently, the pseudoforests can be formed from forests (acyclic undirected graphs) by adding at most one edge per connected component. A *k-tree* is an undirected graph that can be constructed from a K_k graph by repeatedly picking a K_k subgraph and attaching its k vertices to a new vertex. A *partial k-tree* is a subgraph of a k-tree and is said to have *treewidth* at most k; the treewidth of a graph G is denoted $\mathrm{tw}(G)$. Every pseudoforest is a partial

Table 1. Comparison of new and known results on induced subgraphs

Size of induced subgraph	Additional constraints	Type of subgraph	Reference
$n - m$		independent set	trivial
$n - \frac{m}{2}$		matching	trivial
$n - \frac{m}{3}$		linear forest	trivial
$n - \frac{m}{4}$	G is triangle-free	forest	[1]
$n - \frac{m+c}{4}$	max degree ≤ 3 $c = \#$ connected components	forest	[1]
$\frac{n}{\lceil(\Delta+1)/3\rceil}$	max degree Δ	max degree 2	[10]
$\frac{3n}{\Delta+5/3}$	max degree Δ	outerplanar	[16]
$\frac{3n}{\Delta+1}$	max degree Δ	planar	[7]
$\frac{3n}{2m/n+1}$	$m \geq 2n$ or connected and $m \geq n$	partial 2-tree	[8]
$\frac{5n}{6}$	claw-free subcubic	planar partial 4-tree	[5]
$n - \frac{m}{4.5}$		pseudoforest	Theorem 1
$n - \frac{m}{5}$		partial 2-tree	Theorem 2
$n - \frac{m}{5.2174}$		planar partial 3-tree	Theorem 3
$\leq n - \frac{m}{6} + o(m)$		any minor-closed property	Theorem 4

2-tree. A graph is a partial 2-tree if and only if every biconnected component is a series parallel graph. The operations of adding a vertex with two adjacent neighbors and of taking subgraphs preserve planarity, so every partial 2-tree and every pseudoforest is a planar graph.

When constructing induced subgraphs of size $n - m/k$, we will make the simplifying assumption that our graph G has maximum degree at most $\lceil k-1 \rceil$.

Observation 1. *If every graph of maximum degree at most $\lceil k-1 \rceil$ contains an induced subgraph with property $\mathcal{P}$ and at least $n - m/k$ vertices, then the same is true for every graph.*

Proof. We use induction on n. Let G contain a vertex v of degree $\geq k$, and let G' be formed from G by removing v. By the induction hypothesis, G' has an induced subgraph H with the desired property $\mathcal{P}$ and at least $n(G') - \frac{m(G')}{k}$ vertices. Then H is an induced $\mathcal{P}$-subgraph of G with size at least

$$n(G) - 1 - \frac{m(G')}{k} \geq n(G) - 1 - \frac{m(G) - k}{k} = n(G) - \frac{m(G)}{k}.$$

3 Large Induced Pseudoforests

In this section, we prove Theorem 1 by showing that we can delete at most $\frac{m}{4.5}$ vertices from a graph G with m edges to leave a pseudoforest; by Observation 1, we assume G has degree at most 4.

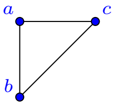
We repeatedly perform the first applicable reduction in the following list of cases, until no edges are left. As we do, we construct a set S of vertices that will induce our desired subgraph. Initially S is empty and when no edges are left we add all remaining vertices to S. The steps of the reduction essentially identify "dangling trees" and contract these. After a series of vertex deletions (which identify vertices that will not belong the final induced subgraph) and edge contractions, if we create a component that consists of a single cycle (in fact a triangle, case Δ-a and *Vertex of degree 4* (c) (i) and (ii)), we "keep" this component; this triangle is the minor of an induced cycle in our final induced subgraph which will only be incident to the "dangling trees" which had been contracted into the cycle. This guarantees that the final induced subgraph has at most one cycle per component. To bound the size of the output S in terms of the number of edges, we use an amortized analysis, incurring a charge of -4.5 for every deleted vertex and a charge of $+1$ for every removed or contracted edge; we show that the net charge for every processing step is non-negative. Our cases are:

Leaf vertex. If there is a vertex a of degree 1, we add a to S and contract the edge incident to a. This incurs a charge of $+1$.

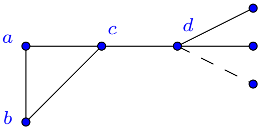
Vertex of degree 2 not in a triangle. If there is a vertex a of degree 2 that is not part of a triangle, we add a to S and contract an edge incident to a. This incurs a charge of $+1$.

Vertex of degree 2 in a triangle. If there is a vertex a of degree 2 in a triangle abc then we consider the four sub-cases illustrated below:

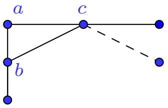
(Δ–a) If the triangle is isolated: add a, b and c to S, and remove the edges of the triangle from the graph. This incurs a charge of $+3$.



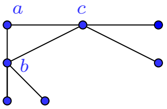
(Δ–b) If b has degree 2 and c has degree 3, then c is adjacent to a vertex d of degree at least 3 (otherwise d would be a degree 2 vertex not belonging to a triangle). Delete d, isolating triangle abc, and then apply case (Δ–a). This incurs a total charge of at least +1.5.



(Δ–c) If b has degree 3 and c has degree at least 3, then delete c, add a and b to S and contract the edges incident to b. This incurs a charge of at least +0.5.



(Δ–d) If b has degree 4, then delete b, add a to S and contract ac. This incurs a charge of +0.5.



Vertex of degree 3 adjacent to a vertex of degree 4. If there is a vertex a of degree 3 adjacent to a vertex b of degree 4, then we delete b. Deleting b incurs a net charge of −0.5, but reduces the degree of a to 2. Handling a as above incurs a charge of at least +0.5, for a net charge of at least 0.

Vertex of degree 3. If this is the first applicable case, then the graph must be 3-regular. Deleting any vertex a creates three vertices of degree 2 while incurring a charge of −1.5. Processing the three resulting degree-2 vertices as above incurs a charge of at least +0.5 per degree-2 vertex, for a net charge of at least 0.

Vertex of degree 4. If this is the first applicable case, then the graph must be 4-regular. We consider the following cases for the subgraph N_a induced by the neighbors b, c, d, e of a vertex a.

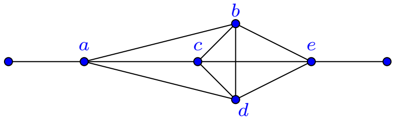
(a) If N_a has two non-adjacent pairs (b, c) and (d, e), then a, b, and c do not form a triangle, so we can delete d and e, keep b and c, add a to S, and contract ab. This removes nine edges from the graph and deletes two vertices, for a total charge of 0.

(b) If N_a is a star graph with center b, then consider the neighbors of c: a, b, f, g. Neither a nor b can be adjacent to f or g, because otherwise they would have too many neighbors. Thus we can process vertex c as in case (a) instead.

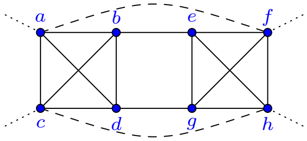
(c) If neither case (a) nor (b) applies, then N_a must contain a triangle. Without loss of generality the triangle is formed by vertices bcd, so a is a vertex of a tetrahedron (K_4) induced by vertices a, b, c, d. We may assume more strongly that every vertex in the graph belongs to a tetrahedron, for if not we may apply cases (a) or (b). We form four sub-subcases:

 (i) If any connected component is a complete graph K_5, then deleting two vertices leaves an isolated triangle and incurs a total charge of +1.

(ii) If two tetrahedra, a, b, c, d and b, c, d, e share triangle b, c, d without forming a K_5, then a and e are non-adjacent. Deleting a and e removes eight edges but leaves an isolated triangle (case Δ–a) for a net charge of +2. We illustrate this case:



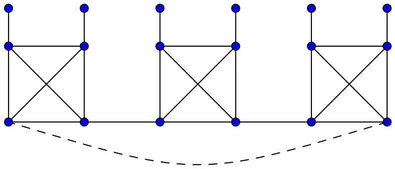
(iii) In the remaining cases all tetrahedra must be vertex-disjoint. If two tetrahedra $abcd$ and $efgh$ are connected to each other by at least two edges (be and dg), then we delete the two non-adjacent vertices d and e as illustrated here:



The dashed edges are possible connections from vertices a, c, f, h.

This incurs a charge of -1 but leaves two non-adjacent degree two vertices (b and g) each of which can be processed via case (Δ–c), adding charge +0.5 per vertex for a total charge of at least 0.

(iv) If every pair of tetrahedra are connected by at most one edge, then contracting every tetrahedron to a vertex reduces the input graph to a smaller 4-regular simple graph that necessarily contains a cycle of three or more edges. In the uncontracted graph, this gives a cycle of six or more edges that alternates between edges within tetrahedra and edges outside the tetrahedra:



In this case, we choose one of the tetrahedra of the cycle, and delete the two vertices of this tetrahedron that do not belong to the cycle. This removes seven edges from the graph for a net charge of -2, but leaves two degree-2 vertices on a cycle of length at least 6. Each of these may be processed as a degree-2 vertex that does not belong to a triangle, giving a charge of +1 each and making the net charge be 0.

This case analysis concludes the proof of Theorem 1. The proof also gives the outline for an efficient algorithm for finding an induced pseudoforest of size at least $n - m/4.5$: after removing any high-degree vertices, form a data structure that lists the configurations of the graph obeying each of the cases in the analysis. Because the remaining graph has bounded degree, selecting the first applicable case, performing the reduction steps of the case, and updating the list of configurations for each case can all be done in constant time per case, leading to a linear overall time bound.

Theorem 1 is tight: there exist arbitrarily large graphs in which the largest induced pseudoforest has exactly $n - m/4.5$ vertices. In particular, let G be a graph formed by the disjoint union of $n/6$ copies of the complete bipartite graph $K_{3,3}$. Then, to form a pseudoforest in G, we must delete at least two vertices from each copy of $K_{3,3}$, for deleting only one vertex leaves $K_{2,3}$ which is not a pseudoforest. Each copy has nine edges, so the number of deleted vertices must be at least $m/4.5$. The same class of examples shows that even if we are searching for the broader class of induced outerplanar subgraphs, we may need to delete $m/4.5$ vertices.

4 Large Induced Treewidth Two Graphs

In this section, we prove Theorem 2 by showing that we can delete at most $\frac{m}{5}$ vertices from a graph G with m edges to leave a graph with treewidth at most 2. By Observation 1, we may assume without loss of generality G has degree at most 4. We prove the theorem algorithmically by arguing that the following procedure builds a vertex set S of size at least $n - \frac{m}{5}$ such that $G[S]$ has treewidth at most 2. The procedure modifies the graph by edge contractions but does not increase its degree over 4.

```
$S = \emptyset$
make $G$ simple by removing self-loops and parallel edges
while $G$ has more than 1 vertex:
    if there is a vertex $v$ of degree one or two:
        contract an edge incident to $v$ and add $v$ to $S$:
        make $G$ simple by removing self-loops and parallel edges
    else if $G$ contains a vertex of degree three:
        delete a vertex of the largest degree adjacent to a degree-three vertex
    else:
        delete a vertex of maximum degree
add the last remaining vertex to $S$
```

Lemma 1. *The induced subgraph $G[S]$ produced by the algorithm above has treewidth at most 2.*

Proof. We use the following facts:

Fact 1. If H is a subdivision of G then $\mathrm{tw}(G) = \max\{1, \mathrm{tw}(H)\}$.
Fact 2. If H is a maximal simple subgraph of G then $\mathrm{tw}(G) = \max\{1, \mathrm{tw}(H)\}$.
Fact 3. If H is obtained from G by deleting a leaf vertex, $\mathrm{tw}(G) = \max\{1, \mathrm{tw}(H)\}$.

Let $s_1, s_2, \ldots, s_k$ be the vertices of S in the reverse of the order in which they were added to S. Let $S_i = \{s_1, s_2, \ldots, s_i\}$; let $S_0 = \emptyset$. Let G_i be the graph at the start of the iteration in which vertex s_i is added to S; let $G_{k+1} = G$. G_i is a minor of G, obtained by deletions of vertices and edge and contractions of edges. G_k is a minor of G and, for $i < k$, G_i is a minor of a graph obtained from G_{i+1} by contracting an edge incident to s_i.

For $i = 2, \ldots, k$:

- $G_i[S_i]$ is a maximal simple subgraph of $G_{i+1}[S_i]$; therefore $\mathrm{tw}(G_{i+1}[S_i]) = \max\{1, \mathrm{tw}(G_i[S_i])\}$ by Fact 2.
- $G_i[S_i]$ is obtained from $G_i[S_{i-1}]$ by the subdivision of an edge (by s_i) or the addition of a leaf vertex (s_i); therefore $\mathrm{tw}(G_i[S_i]) = \max\{1, \mathrm{tw}(G_i[S_{i-1}])\}$ by Facts 1 and 3.

By induction and the fact that a graph with a single vertex has treewidth 0, the lemma follows. □

Lemma 2. $|S| \geq n - \frac{m}{5}$.

Proof. We show, equivalently, that the procedure deletes at most $\frac{m}{5}$ vertices by amortized analysis. For each vertex that we delete we incur a charge of -5. For each edge that we contract (incident to a degree-1 or -2 vertex), remove (as a self-loop or parallel edge) or delete (by way of deleting an adjacent vertex) we incur a charge of $+1$. Note that we distinguish between deleting and removing an edge for the purpose of this analysis. We will show that the net charge is positive, thus showing that for every 5 edges of the graph, we delete at most one vertex.

The first case of the algorithm, in which an edge is contracted, incurs only a positive charge. There are three remaining cases in which a vertex is deleted, according to the degree of the deleted vertex and whether it is adjacent to a degree 3 vertex.

Deleting a degree-3 vertex from a 3-regular graph. Deleting such a vertex incurs a charge of $+3 - 5 = -2$ but creates three degree-2 vertices. At least one edge incident to each of these will be contracted (or removed) for a total charge of $+3$. Therefore, before another vertex is deleted, the net charge for deleting this vertex is at least $+1$.

Deleting a degree-4 vertex adjacent to a degree-3 vertex. Deleting such a vertex v incurs a charge of $+4 - 5 = -1$ but creates at least one vertex u of degree 2; an edge incident to u will be contracted before another vertex is deleted. The net charge for deleting v is at least 0.

Deleting a degree-4 vertex from a 4-regular graph. Deleting such a vertex incurs a charge of $+4-5 = -1$. After this case happens, the remaining graph will have at least four degree-3 vertices, and will continue to have a nonzero number of both degree-3 and degree-4 vertices until either the graph becomes 4-regular again (by reducing the degree-3 vertices to degree-2 and contracting them away) or all degree-4 vertices have been removed. We consider the following sub-cases for the steps of the algorithm that either return to a 4-regular state or eliminate all degree-4 vertices:

(a) If the graph becomes 4-regular again, it can only be after removing a degree-4 vertex adjacent to four degree-3 vertices, followed by four contractions of the resulting degree-2 vertices. This deletion and contractions give a charge of +3.

(b) If the last degree-4 vertex is removed, and has no degree-four neighbors when it is removed, then again its removal causes its neighbors to have degree 2 and is followed by four contractions, for a charge of +3.

(c) If, in the last removal of a degree-4 vertex, the vertex has one or more degree-4 neighbors, then after this removal (and any ensuing edge contractions) the graph becomes 3-regular. The next deletion will incur a charge of +1.

Thus, in all cases, the negative charge for the removal of a degree-4 vertex from a 4-regular graph is balanced by a positive charge for a subsequent step of the algorithm. □

The bound $n - m/5$ is tight, for arbitrarily large graphs. In particular, the graphs formed from $n/5$ disjoint copies of the complete graph K_5 can have at most $n - m/5$ vertices in any induced subgraph of treewidth at most 2, for otherwise one of the copies of K_5 would have only one of its vertices removed in the subgraph, leaving a K_4 subgraph which does not have treewidth 2.

The proof of Theorem 3 is similar in outline to this proof, but with a larger set of cases and a more complex system of charges. We defer the details to the full version of the paper [2].

5 No Very Large, Minor-Free Induced Subgraphs

In this section we prove Theorem 4. To prove this theorem, we begin with the well-known result that K_h-minor free graphs are sparse [13,14,17,18].

Lemma 3 (Theorem 1.1 [18]). *Every simple K_h-minor-free graph with n vertices has $O(nh\sqrt{\log h})$ edges.*

We will use this result to force the presence of a K_h minor even after deleting many vertices. Lemma 4 allows us to densify a graph in terms of its girth (allowing us to use Lemma 3 to argue the existence of a minor). We give a tighter bound on the number of edges in a K_h-minor free graph with girth g in Corollary 1. The proof of Theorem 4 may then be concluded by finding a family of graphs that have sufficiently large girth.

Lemma 4. *Let G be a graph with n vertices, m edges, and sufficiently large girth g. Then it has a minor G' that is a simple graph with $n' \leq \frac{5n}{g}$ vertices and $m - n + n'$ edges.*

Proof. Let T be an arbitrary rooted spanning tree of G, let r be the root of T, and let V_i be the set of vertices at i^{th} level of T. Let $\ell = \lfloor \frac{g-3}{4} \rfloor$. We choose an integer a such that

$$\mathcal{S} = r \cup \left\{ \bigcup_{k \geq 0} V_{a+k\ell} \right\} \tag{1}$$

contains at most $\frac{n}{\ell} \leq \frac{5n}{g}$ vertices. Set $\mathcal{S}$ is a collection of vertices at every ℓ^{th} level starting from level a along with root r.

Now we perform the following operation to obtain a minor G' of G: for every vertex $v \in G \setminus \mathcal{S}$ contract the edge uv where u is the parent of v in T. That is, for every $v \in V_i$, where $i \neq a + k\ell$, we contract v to its ancestor in V_{i-1}. Since the distance between two consecutive levels of vertices in $\mathcal{S}$ is ℓ and the girth of G is g, contracting these edges cannot result in self-loops or parallel edges. Therefore G' is simple.

Since we contract $n - |\mathcal{S}| = n - n'$ edges, the number of edges in G' is $m - (n - n') = m - n + n'$. □

Consider a graph G with n vertices, maximum degree 3, and girth g. If G has $n + \omega(\frac{n}{g} h\sqrt{\log h})$ edges, then, by Lemma 4, G has a minor G' with $O(\frac{n}{g})$ vertices and $\omega(\frac{n}{g} h\sqrt{\log h})$ edges. By Lemma 3, G' is dense enough to have a K_h minor. Therefore, we get:

Corollary 1. *Every simple K_h-minor-free graph G with n vertices, maximum degree 3, and girth g has $n + O(\frac{n}{g} h\sqrt{\log h})$ edges.*

Proof of Theorem 4. Let $G = (V, E)$ be a 3-regular graph with n vertices, $m = \frac{3n}{2}$ edges and girth $\Omega(\log n)$; for example, the Ramanujan graphs have this property [15]. In the following, we take h to be a constant. By Corollary 1, G has a K_h minor. Any subgraph G^* (with m^* edges and n^* vertices) of G also has girth $\Omega(\log n)$. By deleting k vertices, the best we can hope for is that we delete $3k$ edges. That is, $m^* \geq m - 3k$. To ensure that G^* does not have a K_h minor, we need

$$\frac{3n}{2} - 3k = m - 3k \leq m^* \leq n^* + O(n^*/g) = n - k + O\left(\frac{n-k}{\log n}\right)$$

Solving for k, we require that

$$k \geq \left(\frac{1}{4} - O(1/\log n)\right) n.$$

Substituting $2m/3$ for n gives the theorem. □

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant Nos. CCF-1252833 and CCF-1228639 and by the Office of Naval Research under Grant No. N00014-08-1-1015. The authors thank Amir Nayyeri for helpful discussions.

References

1. Alon, N., Mubayi, D., Thomas, R.: Large induced forests in sparse graphs. J. Graph Theory 113, 113–123 (2001) MR 1859785
2. Borradaile, G., Eppstein, D., Zhu, P.: Planar induced subgraphs of sparse graphs (2014), arXiv:1408.5939
3. Borradaile, G., Klein, P., Marx, D., Mathieu, C.: Algorithms for Optimization Problems in Planar Graphs (Dagstuhl Seminar 13421). Dagstuhl Reports 3(10), 36–57 (2014)
4. Călinescu, G., Fernandes, C.G., Finkler, U., Karloff, H.: A better approximation algorithm for finding planar subgraphs. J. Algorithms 27(2), 269–302 (1998) MR 1622397
5. Cheng, C., McDermid, E., Suzuki, I.: Planarization and acyclic colorings of subcubic claw-free graphs. In: Kolman, P., Kratochvíl, J. (eds.) WG 2011. LNCS, vol. 6986, pp. 107–118. Springer, Heidelberg (2011) MR 2914703
6. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs, 1st edn. Prentice-Hall (1998)
7. Edwards, K., Farr, G.: An algorithm for finding large induced planar subgraphs. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 75–80. Springer, Heidelberg (2002) MR 2410446
8. Edwards, K., Farr, G.: Planarization and fragmentability of some classes of graphs. Discrete Math. 308(12), 2396–2406 (2008) MR 2410446
9. Edwards, K., Farr, G.: Improved upper bounds for planarization and series-parallelization of degree-bounded graphs. Electron. J. Combin. 19(2), P25 (2012) MR 2928640
10. Halldórsson, M.M., Lau, H.C.: Low-degree graph partitioning via local search with applications to constraint satisfaction, max cut, and coloring. J. Graph Algorithms Appl. 1(3), 1–13 (1997) MR 1600712
11. Kawarabayashi, K.-I.: Planarity allowing few error vertices in linear time. In: 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009), pp. 639–648 (2009) MR 2648441
12. Kawarabayashi, K.-I., Reed, B.: Computing crossing number in linear time. In: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing (STOC 2007), pp. 382–390 (2007) MR 2402463
13. Kostochka, A.V.: The minimum Hadwiger number for graphs with a given mean degree of vertices. Metody Diskret. Analiz 38, 37–58 (1982) MR 0713722
14. Kostochka, A.V.: Lower bound of the Hadwiger number of graphs by their average degree. Combinatorica 4(4), 307–316 (1984) MR 0779891
15. Lubotzky, A., Philips, R., Sarnak, R.: Ramanujan graphs. Combinatorica 8, 261–277 (1988) MR 0963118
16. Morgan, K., Farr, G.: Approximation algorithms for the maximum induced planar and outerplanar subgraph problems. J. Graph Algorithms Appl. 11(1), 165–193 (2007) MR 2354168
17. Thomason, A.: An extremal function for contractions of graphs. Math. Proc. Cambridge Philos. Soc. 95(2), 261–265 (1984) MR 0735367
18. Thomason, A.: The extremal function for complete minors. J. Combinatorial Theory, Series B 81(2), 318–338 (2001) MR 1814910