Large Independent Sets and Low-Degree Independent Sets in Planar Graphs in Linear Time

Therese C. Biedl *

July 12, 1999

Abstract

In this paper, we study the problem of finding large independent sets in planar graphs in linear time. By modifying the traditional greedy-method, we show how to obtain an independent set of size at least $\frac{5}{23}n$ in linear time. This falls short of the independent set of size $\frac{1}{4}n$, which is known to exist by the 4-color theorem, but is an improvement over the $\frac{1}{5}n$ bound of the best known linear-time heuristic. We also study independent sets where vertices have bounded degree, and obtain for $D \geq 7$ at least $\min\{\frac{5}{23}n, \frac{D-6}{4D-18}n\}$ independent vertices of degree $< D$ in linear time.

Keywords: independent set, planar graph, greedy algorithm, 4-color theorem, graph algorithms, computational geometry

1 Introduction

It has been known for a number of years that any planar graph has a *4-coloring*, i.e., one can color the vertices with 4 colors such that the two endpoints of any edge have different colors ([AH77, AHK77], see also [RSST97]). Many interesting corollaries follow from this result, among them that every planar graph has an *independent set* (a set of vertices without edges between them) of size at least $\frac{1}{4}n$. This bound is best-possible in the sense that there are planar graphs that have only $\frac{1}{4}n$ independent vertices.

The 4-coloring of a planar graph can be found in $O(n^2)$ time [RSST97], but the constant involved in this algorithm is big. Thus, research has focused on how a large independent set might be found in a planar graph without using the 4-color theorem.

Finding the independent set of maximum size is NP-complete, even for cubic planar graphs [GJS76]. But for some applications, it suffices to find a guaranteed fraction cn of independent vertices. Examples include the planar point location algorithm of Kirkpatrick [Kir83] and the reconstruction algorithm for Delauney triangulations of Snoeyink and van Kreveld [SvK97]. The former algorithm, however, needs more than just an independent set; it needs a *low-degree independent set*, i.e., an independent set where every vertex has degree at most D for some constant

*Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, biedl@uwaterloo.ca. Research partially supported by NSERC.

D. Any fraction cn of independent vertices is feasible, but the larger c, the quicker the algorithms, as long as cn independent vertices can be computed in linear time. The challenge is therefore to find cn independent vertices in a planar graph in $O(n)$ time, and to achieve as big a constant c as possible.

1.1 Previous results

We can find $\frac{1}{4}n$ independent vertices with the algorithm by Robertson et al. by computing the 4-coloring of the planar graph [RSST97]. But this takes $O(n^2)$ time with a big constant. A simpler algorithm (which pre-dates the work by Robertson et al.) finds $\frac{2}{9}n$ independent vertices in $O(n^2)$ time [CNS83].

Very little is known about how many independent vertices can be found in $O(n)$ time. A simple greedy-heuristic, which we review in Section 2.2, gives at least $\frac{1}{6}n$ independent vertices. A slight modification, which we review in Section 2.3, gives at least $\frac{1}{5}n$ independent vertices. [This could also be achieved with the linear-time 5-coloring algorithm for planar graphs [CNS81].]

With respect to low-degree independent sets, Kirkpatrick showed how to find at least $\frac{1}{24}n$ independent vertices of degree ≤ 12 with a very simple algorithm [Kir83]. This was improved to $\frac{1}{7}n$ independent vertices of degree ≤ 13 [Ede88], then to $\frac{1}{6}n$ vertices of degree ≤ 9 [SvK97], and finally to $\frac{4}{21}n$ vertices of degree ≤ 9 [Bel99].

1.2 Our results

In this paper, we give a linear-time algorithm to find a large independent set in a planar graph. More precisely, we study a modification of the greedy-method that also uses vertex-contractions and guarantees $\frac{1}{5}n$ independent vertices. We improve this method by choosing the vertex to be deleted or contracted in a particular fashion, which guarantees an independent set of size at least $\frac{5}{23}n \approx 0.217n$. This is a significant increase from the 'easy' bound of $0.2n$ towards the best-possible bound of $0.25n$.

Our bound is obtained by assigning a variable to each possible reduction, and obtaining bounds on the number of vertices and edges deleted with each reduction. This gives us constraints on the variables, and the lower bound for the independent set is then obtained by minimizing the number of independent vertices subject to these constraints.

With respect to low-degree independent sets, we show how to obtain $\min\{\frac{5}{23}n, \frac{D-6}{4D-18}n\}$ independent vertices of degree $< D$ for $D \geq 7$ in linear time. In particular, for $D = 11$, we get $\frac{5}{26}n > \frac{4}{21}n$ independent vertices of degree ≤ 10, thus a larger low-degree independent set than previously known. Alternatively, for $D = 9$, we get $\frac{1}{6}n$ independent vertices of degree ≤ 8, which improves on the bound on the degree while still having a sizable fraction of vertices in the independent set. Our bound is best-possible for $D = 8$: we find $\frac{1}{7}n$ independent vertices of degree ≤ 7, and we show that there exists a graph that has no more than $\frac{1}{7}n + O(\sqrt{n})$ independent vertices of degree ≤ 7.

Our algorithm to find this low-degree independent set is based on the pre-processing idea proposed in [Kir83]: delete all vertices of degree $\geq D$, and find an independent set in the remaining graph. This pre-processing step simply adds another variable to our analysis, and the same method of analysis yields the bound.

The paper is organized as follows. After giving definitions, we review in Section 2 the traditional greedy-method and the greedy-method with contractions. In Section 3, we study ways to improve the choice of the reduction vertex and obtain bounds on the number of vertices and edges deleted in each type of reduction. This analysis yields in Section 4 that with the appropriate choice of the reduction-vertex the greedy-method with contractions results in at least $\frac{5}{23}n$ independent vertices, and, if we remove vertices of degree $\geq D$ beforehand, at least $\min\{\frac{5}{23}n, \frac{D-6}{4D-18}n\}$ independent vertices of degree $< D$. In Section 5 we study details of the implementation in linear time, and conclude in Section 6. Two lengthy proofs are deferred to Appendix A and B.

2 Background

2.1 Definitions

Let $G = (V, E)$ be a graph with n vertices and m edges. The *degree* of a vertex v, denoted $\deg(v)$, is the number of incident edges of v. A *multiple edge* is an edge $e = (v, w)$ such that there exists another edge $e' = (v, w)$. A *loop* is an edge $e = (v, v)$. A graph is called *simple* if it has neither loops nor multiple edges. In this paper we assume that the given graph is simple. However, we will sometimes destroy simplicity by creating multi-edges, which will consequently be deleted.

Graph G is called *planar* if it can be drawn in the plane without a crossing. In this paper all graphs are assumed to be planar. A specific *planar embedding* of a planar graph is described by the circular order of edges around each vertex and can be computed in linear time [HT74]. We assume in the following without further mentioning that if G is a planar graph, then a planar embedding of it has been computed and remains fixed. Also, any subgraph of G is assumed to have the planar embedding that is induced by G.

In a planar graph, two edges $e_1 = (v, w_1), e_2 = (v, w_2)$ are called *clockwise consecutive at v* if the clockwise order of edges around v contains e_1 followed by e_2, and *counter-clockwise consecutive at v* if e_2, e_1 are clockwise consecutive at v. Two vertices w_1, w_2 are called *consecutive neighbors of v* if they are both adjacent to v, and the edges (v, w_1) and (v, w_2) are clockwise or counter-clockwise consecutive. An edge (w_i, w_j) is called a *long edge of v* if w_i and w_j are two neighbors of v that are not consecutive. A set $\{w_1, \ldots, w_k\}$ is called *a set of consecutive neighbors of v* if, after suitable renaming, w_i and w_{i+1} are consecutive neighbors of v for $i = 1, \ldots, k-1$.

A planar drawing of a graph splits the plane into pieces called *faces*; the unbounded face is called the *outer-face*. A planar graph is called *triangulated* if every face is a triangle. Such a graph is *maximal planar* in the sense that the only edges that can be added without destroying planarity are multiple edges and loops.

A *separating triangle* of a planar graph is a triangle $T = \{v_0, v_1, v_2\}$ such that for some $i \in \{0, 1, 2\}$ the edges (v_{i-1}, v_i) and (v_i, v_{i+1}) are not clockwise consecutive, and for some $j \in \{0, 1, 2\}$ the edges (v_{j-1}, v_j) and (v_j, v_{j+1}) are not counter-clockwise consecutive, where addition is modulo 3. This is the same as to say that some neighbor of v_0, v_1, v_2 is "inside" T and some neighbor is "outside" T in the planar drawing.

Two special graph classes will be used in this paper. The *complete graph* on n vertices, denoted K_n, is the simple graph where any two distinct vertices are adjacent. It is known that K_5 is not planar. The *complete bipartite graph* on $n_1 + n_2$ vertices, denoted K_{n_1,n_2}, is the simple graph where all of n_1 vertices are adjacent to all of n_2 different vertices. It is known that $K_{3,3}$ is not planar.

Much is known about planar graphs. For example, every planar simple graph has at most $3n-6$ edges, which implies that it has a vertex of degree at most 5. However, we need the following slight strengthening of this result:

Lemma 1 *Let G be a planar simple graph with at least 4 vertices, and let a, b, c be three vertices of G. Then G contains a vertex $v \neq a, b, c$ with* $\deg(v) \leq 5$.

Proof: Note that we may assume without loss of generality that G is triangulated, for if it is not, then we can add edges until it is; this will only increase the degrees of vertices. Since G is a triangulated simple graph and $n \geq 4$, ever vertex has degree at least 3. If all vertices $v \neq a, b, c$ had degree ≥ 6, we would get

$$6n - 12 \geq 2m = \sum_{v \in V} \deg(v) \geq 6(n-3) + 3 \cdot 3 = 6n - 9,$$

which is a contradiction. □

The following corollary will be crucial for our choice of reduction-vertex.

Corollary 2 *Any planar simple graph G has a vertex v with* $\deg(v) \leq 5$ *that does not belong to a separating triangle.*

Proof: The statement clearly holds if G has no separating triangle, so assume G has a separating triangle T_0. Let G_{T_0} be the subgraph inside T_0, i.e., the graph induced by the vertices of T_0 and the vertices inside T_0. Here, *inside* is defined with respect to the fixed planar embedding and outer-face, and is the set of those vertices $v \notin T_0$ for which any path from v to a vertex on the outer-face contains a vertex of T_0. Note that by definition of a separating triangle, there must be at least one vertex inside T_0. Also, there is at least one vertex that is neither on T_0 nor inside T_0, so G_{T_0} has fewer vertices than G.

If G_{T_0} has a separating triangle T_1, take the subgraph G_{T_1} inside T_1 and iterate. Since G_{T_1} has fewer vertices than G_{T_0}, this process must stop, say G_{T_k} has no separating triangle. By the above lemma, pick a vertex v with $\deg(v) \leq 5$ in G_{T_k} that is not on T_k, this vertex then does not belong to a separating triangle. □

2.2 The traditional greedy-method

The greedy-method for finding an independent set works as follows: As long as the graph has vertices, pick a vertex v of minimum degree, remove v and all its neighbors, compute an independent set in the resulting graph, and add v to it. We will refer to the operation of deleting v and all its neighbors as *delete*(v). Because every planar simple graph has a vertex of degree ≤ 5, this method yields an independent set of size at least $\frac{1}{6}n$.

2.3 Greedy with contractions

The greedy-method can be improved by replacing *delete*(v), if possible, with *contract*(v, w_i, w_j), which works as follows (see Figure 1). Let w_i and w_j be two non-adjacent neighbors of v. Delete v and all neighbors $\neq w_i, w_j$ of v, and contract w_i and w_j into a new vertex w^*. Also delete all multiple edges that might arise from the contraction. Compute an independent set I in the resulting graph. If w^* does not belong to I, then add v to I; this yields an independent set. If w^* does belong to I, then remove it and add both w_i and w_j to I. Because w_i and w_j are not adjacent, this yields an independent set.

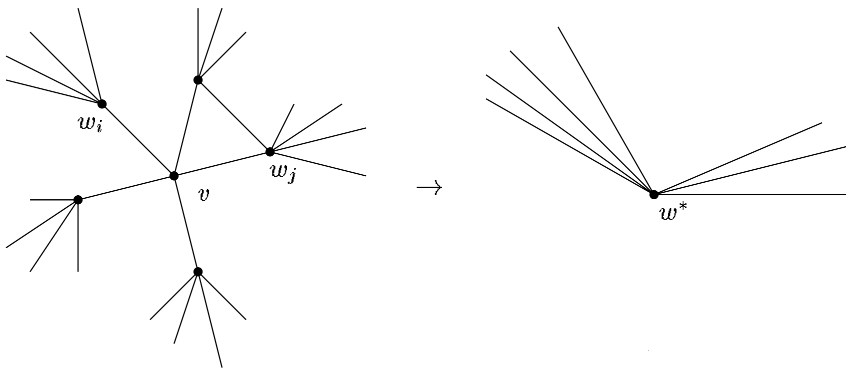


Figure 1: The operations *contract*(v, w_i, w_j).

Operation *contract*(v, w_i, w_j) is better than *delete*(v) because it deletes only $\deg(v)$ vertices to increase the size of the independent set by one. However, it can only be applied if two neighbors of v are not adjacent. The crucial observation is now that for a planar graph, if $\deg(v) \geq 4$, then some pair of neighbors w_i, w_j is not adjacent, because otherwise we could complete v and its neighbors to a planar K_5.[1] So we can always apply *contract*(v, w_i, w_j) if $\deg(v) \geq 4$.

Algorithm GreedyWithContraction thus works as follows: As long as the graph has vertices, pick a vertex v of minimum degree. If $\deg(v) \leq 3$, then apply *delete*(v), else apply *contract*(v, w_i, w_j) for some suitable neighbors w_i, w_j of v. Compute an independent set in the remaining graph, and expand it as described above. Every iteration of GreedyWithContraction deletes at most 5 vertices (by $\deg(v) \leq 5$) and expands the independent set by one vertex, thus GreedyWithContraction yields an independent set of size at least $\frac{1}{5}n$.

[Remark: The greedy-algorithm is folklore. The author has not found an explicit description of GreedyWithContraction, but all ideas for it are contained in [Alb76].]

3 Algorithm SmartGreedy

In this section, we study how to choose the *reduction vertex*, i.e, the next vertex v to be deleted (followed possibly by contraction of two neighbors of v) such that algorithm GreedyWithContraction yields at least $\frac{5}{23}n$ independent vertices. The crucial ingredient to our analysis lies in determining not only how many vertices have been removed, but also in noting a lower bound on the number of edges that have been removed. We need the following simple observations.

[1]By *completing a set S to a planar K_5*, we mean that we can add edges without destroying planarity, specifically, edges between consecutive neighbors, such that the resulting graph is planar and contains a K_5 or a subdivision of it. This is a contradiction.

Fact 3 *If we delete vertices* $u_1, \ldots, u_k$*, then we delete at least* $\deg(u_1) + \ldots + \deg(u_k) - |\{(u_i, u_j) \in E : i, j \in \{1, \ldots, k\}\}|$ *edges.*

Proof: Every incident edge of $u_1, \ldots, u_k$ is deleted, but an edge $(u_i, u_j), i, j \in \{1, \ldots, k\}$ is counted twice by $\deg(u_1) + \ldots + \deg(u_k)$. □

Fact 4 *If we contract neighbors* w_i, w_j *of* v *and delete all other neighbors* $u_1, \ldots, u_k$ *of* v*, then we delete at least* $\deg(u_1) + \ldots + \deg(u_k) + 2 - |\{(u_i, u_j) \in E : i, j \in \{1, \ldots, k\}\}|$ *edges.*

Proof: We delete $u_1, \ldots, u_k$ and v. Of the $k + 2$ edges incident to v, all but the two to w_i and w_j have been counted in $\deg(u_1) + \ldots + \deg(u_k)$ already, so v contributes only 2 units. □

In the following subsections, we study a variety of cases for the reduction of a vertex. Each case will be called an x_α^β-reduction for some strings α, β. Later, we also use the variable x_α^β to denote the number of x_α^β-reductions. For each case, we determine the number of deleted vertices n_α^β, and a lower bound m_α^β on the number of edges deleted by this reduction. Finally we denote by i_α^β how many independent vertices are added by this reduction (usually, $i_\alpha^\beta = 1$, but we will have one exception to this).

In all cases, assume that d is the minimum degree of the current graph, that v is a vertex of degree d and that $w_1, \ldots, w_d$ are the neighbors of v in clockwise order. Observe that $\deg(w_i) \geq d$ for $i = 1, \ldots, d$.

3.1 Vertices of degree ≤ 2

If v has degree ≤ 2, then delete v and its neighbors, thus $d + 1$ vertices. Since each neighbor w_i has degree $\geq d$, and since there are at most $(d^2 - d)/2$ edges between $w_1, \ldots, w_d$, this deletes at least $d^2 - (d^2 - d)/2 = (d^2 + d)/2$ edges. For this x_d*-reduction* ($d \leq 2$), we thus have the following bounds:

$$
\begin{array}{llll}
x_0\text{-reduction}: & i_0 = 1 & n_0 = 1 & m_0 = 0 \\
x_1\text{-reduction}: & i_1 = 1 & n_1 = 2 & m_1 = 1 \\
x_2\text{-reduction}: & i_2 = 1 & n_2 = 3 & m_2 = 3
\end{array}
$$

3.2 Vertices of degree 3

If the minimum degree in the graph is 3, then to make our algorithm produce many independent vertices, we choose the reduction-vertex v as follows:

Algorithm PICKDEG3VERTEX
If there exists a vertex v of degree 3 such that
  two neighbors of v are not adjacent, or
  delete(v) deletes at least 9 edges,
Then pick v as next reduction-vertex.
Else, pick an arbitrary vertex of degree 3 as next reduction-vertex.

We will explain in Section 5 how to implement algorithm PICKDEG3VERTEX efficiently.

To analyse this case, we distinguish by whether all pairs of neighbors of v are adjacent.

If, say, w_1 and w_2 are not adjacent, then apply *contract*(v, w_1, w_2). This deletes at least $3+2-0 = 5$ edges by $\deg(w_3) \geq 3$, and removes three vertices (two by deletion and one by contraction). We call this an x_3^c-*reduction*, where "c" stands for "contract".

If all neighbors of v are adjacent to each other, then we delete v and its neighbors, thus 4 vertices and at least 6 edges. We call this an x_3^d-*reduction*, where "d" stands for "delete".

We obtain the following values:

$$\begin{array}{llll} x_3^c\text{-reduction}: & i_3^c = 1 & n_3^c = 3 & m_3^c = 5 \\ x_3^d\text{-reduction}: & i_3^d = 1 & n_3^d = 4 & m_3^d = 6 \end{array}$$

3.3 Vertices of degree 4

If $\deg(v) = 4$, then there are always two neighbors w_i, w_j that are not adjacent. Thus, we can always perform *contract*(v, w_i, w_j). This deletes two neighbors of v, and at least $4 + 4 + 2 - 1 = 9$ edges. We call this an x_4^c-*reduction*.

In order to get a simple linear-time implementation, we introduce one exception: If $\sum_{i=1}^{4} \deg(w_i) \geq 27$, then we call *delete*$(v)$, which deletes 5 vertices and at least $27 - 5 = 22$ edges. (See Section 5.2.3 for an explanation why.) We call this case an x_4^d-*reduction*, and get the following values:

$$\begin{array}{llll} x_4^c\text{-reduction}: & i_4^c = 1 & n_4^c = 4 & m_4^c = 9 \\ x_4^d\text{-reduction}: & i_4^d = 1 & n_4^d = 5 & m_4^d = 22 \end{array}$$

3.4 Vertices of degree 5

If $\deg(v) = 5$, then there are always two neighbors w_i, w_j of v that are not adjacent, and we can apply *contract*(v, w_i, w_j).

We want to avoid this case, called an x_5-*reduction*, because it adds only one independent vertex for 5 deleted vertices. To obtain good bounds on how often it can happen, we must have especially high bounds on the number of deleted edges. To be able to do so, we must choose v and its neighbors w_i and w_j carefully, for with a bad selection, we may delete as few as 14 edges.

The number of deleted edges during an x_5-reduction is at least $17 - |\{\text{edges between deleted neighbors of } v\}|$ by Fact 4. If we choose the contraction vertices non-consecutive, then only one pair of deleted neighbors is consecutive, and therefore the number of deleted edges is at least $16 - |\{\text{long edges between deleted neighbors of } v\}|$. It will therefore be a good idea to use as reduction vertex a vertex that does not have a long edge; this exists by Corollary 2 because a long edge of v implies that v belongs to a separating triangle. The precise choice of v is detailed in the next lemma, the lengthy proof of which is given in Appendix A.

Lemma 5 *Let G be a planar graph with minimum degree 5. Then there exists a vertex v with $\deg(v) = 5$ and two non-adjacent non-consecutive neighbors w_i, w_j of v such that*

1. *contract(v, w_i, w_j) deletes at least 19 edges, or*
2. *the following three conditions, called* Conditions (*), *hold:*
   - *contract(v, w_i, w_j) deletes at least 16 edges that are not multiple edges,*

- *after contract*(v, w_i, w_j)*, there exists a vertex* z *with* $\deg(z) \leq 4$*, and*
- *there exists at most one long edge of* v *with both endpoints* $\neq w_i, w_j$*, i.e., with both endpoints among the deleted neighbors of* v*.*

For future reference, we will also note how long it takes to compute this vertex, which follows from analyzing all cases of the proof of Lemma 5; see Appendix A.

Lemma 6 *Let* G *be a planar graph with minimum degree 5, and let vertex* u *be given such that* $\deg(u) = 5$ *and* u *does not belong to a separating triangle. Then the vertex* v *of Lemma 5 can be found in* $O(1)$ *time.*

So if the minimum degree is 5, then there exists a reduction vertex such that one of two cases happens. In the first case, we remove very many edges. In the second case, we obtain a vertex of degree ≤ 4 after the x_5-reduction, thus the next reduction removes at most 4 vertices. Putting both reductions together, we remove at most 9 vertices to obtain two independent vertices.

For the second case, we will show that if the second reduction is an x_3^d-reduction, then at least 25 edges are removed during both reductions together. To be able to show this bound, we must be smart in the choice of neighbors to be contracted during the x_5-reduction. The order of preference among these pairs is indicated in the following algorithm.

Algorithm PICKCONTRACTIONPAIR
Let v be the reduction-vertex of degree 5 chosen as in Lemma 5.
Find a pair $\{w_i, w_j\}$ of neighbors of v that is not consecutive, not adjacent, and highest with respect to the following list of preferences (one of these cases is always possible by Lemma 5):

1. *contract*(v, w_i, w_j) would delete at least 19 edges.
2. *contract*(v, w_i, w_j) would satisfy Conditions (*) and create an x_3^c-reduction, i.e., a vertex of degree 3 for which not all neighbors are adjacent.
3. *contract*(v, w_i, w_j) would delete at least 17 edges and satisfy Conditions (*).
4. *contract*(v, w_i, w_j) would satisfy Conditions (*).

We will study in Section 5 how to implement this algorithm efficiently. Using this algorithm, we can show the desired bound on the number of deleted edges; the proof of this lemma can be found in Appendix B.

Lemma 7 *Let the minimum degree be* 5*, let* v *be chosen as in Lemma 5, and let* w_1, w_3 *be the neighbors of* v *chosen with algorithm* PICKCONTRACTIONPAIR*. If contract*(v, w_1, w_3) *deletes* k *edges and is followed by an* x_3^d*-reduction at vertex* x*, then delete*(x) *deletes at least* $25 - k$ *edges.*

For an x_5-reduction, we thus distinguish three cases. In the first case, denoted an x_5^{19}*-reduction*, we delete at least 19 edges. In the second case, we do an x_5-reduction followed by an x_3^d-reduction, and both together delete at least 25 edges. We call the combination of an x_5-reduction and an x_3^d-reduction an x_9*-reduction*, which deletes 9 vertices, at least 25 edges, and adds 2 independent vertices. Finally, we have the case that the x_5-reduction deletes at least 16 edges and is followed by a reduction that is not an x_3^d-reduction, and not an x_5-reduction by Conditions (*). We call this an x_5^{16}*-reduction.* We obtain the following bounds:

$$\begin{array}{llll} x_5^{19}\text{-reduction}: & i_5^{19}=1 & n_5^{19}=5 & m_5^{19}=19 \\ x_5^{16}\text{-reduction}: & i_5^{16}=1 & n_5^{16}=5 & m_5^{16}=16 \\ x_9\text{-reduction}: & i_9=2 & n_9=9 & m_9=25 \end{array}$$

The definition of an x_5^{16}-reduction also tells us that

$$x_5^{16} \le x_0 + x_1 + x_2 + x_3^c + x_4^c + x_4^d,$$

because every x_5^{16}-reduction must be followed by a reduction that is not an x_3^d-reduction and not an x_5-reduction.

We denote the algorithm that chooses reduction vertices as explained in this section by SMART-GREEDY. Details of the implementation of SMARTGREEDY are given in Section 5.

4 Analysis

In the previous section, we obtained the number n_α^β of deleted vertices, the lower bound m_α^β of deleted edges, and the number i_α^β of independent vertices added in an x_α^β-reduction. In this section, we use these bounds to obtain a good lower bound on the size of the independent set.

Initially we had n vertices. With an x_α^β-reduction, we remove n_α^β vertices. We end with no vertices left, therefore

$$\sum n_\alpha^\beta x_\alpha^\beta = n,$$

where the sum is over all possible cases. Initially we had m edges. With an x_α^β-reduction, we remove at least m_α^β edges. We end with no edges left, therefore

$$\sum m_\alpha^\beta x_\alpha^\beta \le m \le 3n-6,$$

where the sum is over all possible cases. We also had the inequality

$$x_5^{16} \le x_0 + x_1 + x_2 + x_3^c + x_4^c + x_4^d.$$

Each x_α^β-reduction increases the number of independent vertices by i_α^β, so the final number of independent vertices is

$$|I| = \sum i_\alpha^\beta x_\alpha^\beta,$$

where the sum is over all possible cases. To determine therefore the smallest possible independent set that could be found with SMARTGREEDY, we can minimize this last expression, subject to the constraints above as well as $x_\alpha^\beta \ge 0$. This yields the following linear program:

$$\begin{array}{rrrrrrrrrrrcl} \text{minimize} & x_0 & +x_1 & +x_2 & +x_3^c & +x_3^d & +x_4^c & +x_4^d & +x_5^{16} & +x_5^{19} & +2x_9 & & \\ \text{s.t.} & x_0 & +2x_1 & +3x_2 & +3x_3^c & +4x_3^d & +4x_4^c & +5x_4^d & +5x_5^{16} & +5x_5^{19} & +9x_9 & = & n \\ & & x_1 & +3x_2 & +5x_3^c & +6x_3^d & +9x_4^c & +22x_4^d & +16x_5^{16} & +19x_5^{19} & +25x_9 & \le & 3n \\ & +x_0 & +x_1 & +x_2 & +x_3^c & & +x_4^c & +x_4^d & -x_5^{16} & & & \ge & 0 \\ & & & & & & x_0, x_1, x_2, x_3^c, & x_3^d, x_4^c, & x_4^d, x_5^{16}, & x_5^{19}, x_9 & & \ge & 0 \end{array}$$

[Remark: we dropped the "-6" from the second constraint, because this makes solving the problem easier, can only decrease the lower bound, and is irrevelant for sufficiently large n anyway.]

Solving this linear program to optimality, we obtain $x_3^d = \frac{2}{23}n$ and $x_5^{19} = \frac{3}{23}n$ and all other variables 0. [One can verify this by setting as dual variables $(\frac{13}{46}, -\frac{1}{46}, \frac{3}{46})$ and testing that both primal and dual solution are feasible and yield the same value, hence they are both optimal. See any textbook on linear programming, for example [Chv83], for details.]

The minimum possible size of an independent set found by SmartGreedy therefore is $\frac{5}{23}n$.

Lemma 8 *Let G be a planar graph. Then algorithm* SmartGreedy *computes an independent set of size at least $\frac{5}{23}n \approx 0.217n$.*

Thus our bound on the independent set is smaller than the bound of $\frac{2}{9}n \approx 0.222n$ independent vertices that can be found in $O(n^2)$ time without the 4-color theorem [CNS83], and the bound of $\frac{1}{4}n = 0.25n$ independent vertices that can be found in $O(n^2)$ time with the 4-color theorem [RSST97]. However, our independent set can be obtained in linear time, as we shown in the next section. No better bound than $\frac{5}{23}n$ is known for linear-time algorithms, and closing the gap to $\frac{1}{4}n$, or at least to $\frac{2}{9}n$, remains an open problem.

4.1 Low-degree independent sets

In order to compute a low-degree independent set, i.e., an independent set where every vertex has degree $< D$ for some constant D, Kirkpatrick [Kir83] proposed the following simple strategy: Delete all vertices of degree $\geq D$ from the graph; this takes $O(n)$ time. In what remains, find an independent set; this is then a low-degree independent set. Kirkpatrick showed how for $D = 12$ and the greedy-algorithm, this gives at least $\frac{1}{24}n$ independent vertices of degree at most 11.

We can improve this result by using algorithm SmartGreedy and refining the analysis. Namely, introduce a variable x_D which denotes the number of vertices of degree $\geq D$ in the graph. The deletion of these x_D vertices removes at least $Dx_D - (3x_D - 6) \geq (D-3)x_D$ edges by Fact 3, because the vertice of degree $\geq D$ induce a planar graph with at most $3x_D - 6$ edges. If we treat these deletions as just another type of reduction, this one not adding any independent vertices, then we get the following linear program:

$$\begin{array}{llcccccccccccl}
\text{minimize} & & x_0 & +x_1 & +x_2 & +x_3^c & +x_3^d & +x_4^c & +x_4^d & +x_5^{16} & +x_5^{19} & +2x_9 & & \\
\text{s.t.} & x_D & +x_0 & +2x_1 & +3x_2 & +3x_3^c & +4x_3^d & +4x_4^c & +5x_4^d & +5x_5^{16} & +5x_5^{19} & +9x_9 & = & n \\
& (D-3)x_D & & +x_1 & +3x_2 & +5x_3^c & +6x_3^d & +9x_4^c & +22x_4^d & +16x_5^{16} & +19x_5^{19} & +25x_9 & \leq & 3n \\
& & +x_0 & +x_1 & +x_2 & +x_3^c & & +x_4^c & +x_4^d & -x_5^{16} & & & \geq & 0 \\
& & & & & & & & \multicolumn{4}{r}{x_D, x_0, x_1, x_2, x_3^c, x_3^d, x_4^c, x_4^d, x_5^{16}, x_5^{19}, x_9} & \geq & 0
\end{array}$$

Solving this linear program to optimality, we obtain the following results:

- For $7 \leq D \leq 13$: $x_3^d = \frac{D-6}{4D-18}n$, $x_D = \frac{6}{4D-18}n$, and all other variables 0. Optimal dual variables are $(\frac{D-3}{4D-18}, \frac{-1}{4D-18}, 0)$.
- For $13 \leq D \leq 15$: $x_3^d = \frac{D-6}{4D-18}n$, $x_D = \frac{6}{4D-18}n$, and all other variables 0. Optimal dual variables are $(\frac{D-3}{4D-18}, \frac{-1}{4D-18}, \frac{D-13}{4D-18})$.

- For $D \geq 16$: $x_3^d = \frac{2}{23}n$, $x_5^{19} = \frac{3}{23}n$, and all other variable 0. Optimal dual variables are $(\frac{13}{46}, -\frac{1}{46}, \frac{3}{46})$.

Lemma 9 *Let G be a planar graph. Then for $D \geq 7$, removing vertices of degree $\geq D$ and applying algorithm* SmartGreedy *gives at least $\min\{\frac{5}{23}n, \frac{D-6}{4D-18}n\}$ independent vertices of degree $< D$.*

For $D = 8$, we find $\frac{1}{7}n$ independent vertices of degree ≤ 7. This is best-possible, because as we show now, there exists a graph $G_8(N)$ with only $\frac{1}{7}n + O(\sqrt{n})$ independent vertices of degree ≤ 7. Specifically, $G_8(N)$, which is defined only for N divisible by 6, is the $N \times N$-grid with diagonals where every 6th face contains a K_4, connected in such a way that all vertices of the grid except the boundary vertices have degree 8. See Figure 2.

The $N \times N$-grid with diagonals has $(N+1)^2$ vertices and $2N^2$ interior faces, therefore $G_8(N)$ has $(N+1)^2 + 4\frac{2N^2}{6} = \frac{7}{3}N^2 + O(N)$ vertices. Since all but the $O(N)$ boundary vertices of the grid have degree 8, and since only one vertex from each K_4 can belong to an independent set, $G_8(N)$ has at most $\frac{2N^2}{6} + O(N) = \frac{1}{7}n + O(\sqrt{n})$ independent vertices of degree ≤ 7.

For $D = 7$, we find $\frac{1}{10}n$ independent vertices of degree ≤ 6. This is close to optimality, because as we show now, there exists a graph $G_7(N)$ that has only $\lceil \frac{n}{8} \rceil$ independent vertices of degree ≤ 6. Specifically, $G_7(0)$ is K_4. For $N \geq 1$, $G_7(N)$ is an icosahedron where 3 faces contain a K_4 and one face contains $G_7(N-1)$, connected in such a way that all vertices of the icosahedron have degree 7. See Figure 2.

By induction, one shows easily that $G_7(N)$ has $24N + 4$ vertices. Since each vertex of the outermost icosahedron of $G_7(N)$ $(N > 0)$ has degree 7, and only one vertex of each K_4 can be in an independent set, the maximum number $I(N)$ of independent vertices of degree ≤ 6 in $G_7(N)$ satisfies the recurrence relation $I(N) = I(N-1) + 3$, which together with $I(0) = 1$ shows that $G_7(N)$ has only $3N + 1 = \lceil \frac{n}{8} \rceil$ independent vertices of degree ≤ 6.

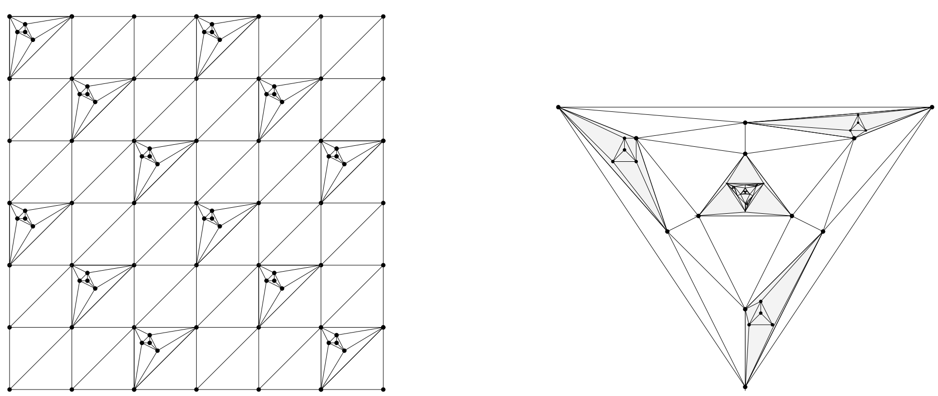


Figure 2: Graphs $G_8(6)$ and $G_7(2)$. For the latter, faces of the icosahedron that contain a K_4 or $G_7(1)$ are shaded.

5 Implementation

The objective of this section is to show that SMARTGREEDY can be implemented in $O(n)$ time.[2] The crucial observation is as follows: It is easy to find a vertex of minimum degree in amortized constant time. All we therefore have to worry about is how to find the special vertices that are needed if the minimum degree is 3 or 5. Here the following observation helps: If $\deg(v) = 3, 5$, and the neighbors of v have very high degrees, then we can always use v as reduction-vertex, because this will delete sufficiently many edges. If the neighbors of v have not very high degrees, then we can do all the tests needed to determine the correct reduction vertex in $O(1)$ time.

Another important ingredient is to choose the neighbors for the contraction (if applicable) such that the number c_m of edges that are incident to contracted vertices is proportional to the number d_m of deleted edges. This will help to show an $O(n)$-time bound without having to use advanced data structures to do the union of edge lists.

5.1 Storing vertices

We store the vertices in 9 buckets named $B_0, B_1, B_2, B_3, B_3^d, B_4, B_5, B_5^t$ and B_6. Every vertex belongs to exactly one bucket and knows to which bucket it belongs. The assignment of vertices to buckets obeys the following rules throughout the algorithm:

- For $d \leq 5$, any vertex in bucket B_d has degree d.
- Any vertex in bucket B_6 has degree ≥ 6.
- Any vertex v in bucket B_3^d has $\deg(v) = 3$. Furthermore, any pair of neighbors of v is adjacent, and *delete*(v) would delete ≤ 8 edges; in particular therefore all neighbors of v have degree ≤ 5.
- Any vertex v in bucket B_5^t has $\deg(v) = 5$. Furthermore, v belongs to a separating triangle, and all neighbors of v have degree ≤ 9.

Initially, we put all vertices into $B_0, B_1, B_2, B_3, B_4, B_5$ and B_6 according to their degrees. We assume that each vertex v keeps an integer $d(v)$ indicating its degree; then this can be done in linear time. Buckets B_3^d and B_5^t are initially empty and will be filled whenever we find a vertex that fits the description, see Sections 5.2.2 and 5.2.4. Changes to $d(v)$ and to the assignment of vertices to buckets may be necessary if we delete edges or contract vertices; this will be described in Section 5.3.1 and 5.3.3.

5.2 Choosing a reduction vertex

In this section, we show how to choose the reduction vertex v and, if applicable, two neighbors w_i, w_j of v for the contraction in $O(d_n + d_m)$ time, where d_n and d_m is the number of deleted vertices and edges, respectively. We do not include the time to actually do the reduction; this is the topic of the next subsection.

To choose a reduction vertex, we consider the buckets $B_0, B_1, B_2, B_3, B_3^d, B_4, B_5$ (in this order), and take a vertex v from the first bucket that is not empty. By Corollary 2, one of these buckets

[2] The emphasis is on *that* the algorithm can be implemented in $O(n)$ time, but we make no attempts to optimize the constant hidden in the O-notation, which could be much improved with a more careful analysis and implementation.

is not empty. If $\deg(v) \le 2$, then we apply *delete*(v). To find v, we thus spent $O(1) = O(d_n)$ time. In the other cases, handle v as follows.

5.2.1 A vertex v in B_3.

We start by computing the sum s of the degrees of the neighbors of v. If $s \ge 12$, then we call *delete*(v), this will delete at least 9 edges and therefore complies with Algorithm PickDeg3Vertex. If $s \le 11$, then test in $O(11) = O(1)$ time whether the neighbors of v form a triangle.

If some neighbors w_i, w_j of v are not adjacent, then perform *contract*(v, w_i, w_j). At least 5 edges are deleted, therefore by $s \le 11$ at most 6 edges are incident to w_i or w_j, and $c_m \le 6 \le 10 \le 2d_m$.

If the neighbors of v form a triangle, then put v into B_3^d (by $s \le 11$ at most 8 edges would be deleted during an x_3^d-reduction at v). The $O(1)$ time spent in this case will be counted as overhead to the operation that *removes* v again from B_3^d; this is either a deletion or a contraction.

5.2.2 A vertex v in B_3^d.

No vertex is in B_3 (because we would have taken this first), so no x_3^c-reduction and no x_3^d-reduction deleting at least 9 edges is possible. Therefore call *delete*(v); this complies with algorithm PickDeg3Vertex.

5.2.3 A vertex v in B_4.

Denote by $w_1, \ldots, w_4$ the neighbors of v. If $\sum_{i=1}^4 \deg(w_i) \le 26$, then we apply *contract*(v, w_i, w_j) with a pair of neighbors of v that is not adjacent. By $4 \le \deg(w_i) \le 26 - 3 \cdot 4 = 14$ for $i = 1, 2, 3, 4$, such a pair can be found in $O(1)$ time. This deletes $d_m \ge 9$ edges; therefore at most $26 - 9 = 17 \le 2d_m$ edges are incident to a contracted vertex.

If $\sum_{i=1}^4 \deg(w_i) \ge 27$, then we apply *delete*$(v)$. We could have applied a contraction as well, however, we then cannot show $c_m \le 2d_m$. Since we need this bound to be able to show that all contractions can be done in $O(n)$ total time, we prefer the deletion.

[Remark: If a slight time-increase seems insignificant relative to finding more independent vertices, then one should in this case do *contract*(v) with suitable neighbors, and for the contraction, merge the smaller edge list into the larger one. This then works in $O(n \log n)$ total time.]

5.2.4 A vertex in B_5.

This is the most complicated case, because the reduction vertex must be chosen according to Lemma 5, and the neighbors for the contraction must be chosen according to algorithm PickContractionPair.

As a first step, we only determine the vertex v for the contraction, not the pair of neighbors used for the contraction. Let u be a vertex in B_5, and let $z_1, \ldots, z_5$ be its neighbors in clockwise order, named such that $\deg(z_5) \ge \deg(z_i)$ for $i = 1, 2, 3, 4$. If $\deg(z_5) \ge 10$, then set $v = u$. If $\deg(z_5) \le 9$ (and therefore $s = \sum_{i=1}^5 \deg(z_i) \le 45$), then we are free to do any operation that takes $O(s) = O(1)$ time.

We start by determining whether u belongs to a separating triangle; this can be done in $O(s)$ time by testing all pairs of neighbors of u for being adjacent and if so, whether the edges of the triangle through u are all clockwise consecutive or all counter-clockwise consecutive. If u belongs to a separating triangle, then we put u into B_5^t. The $O(1)$ time to determine this will again be counted as overhead to the operation that removes u from B_5^t.

If u does not belong to a separating triangle, then we find the vertex v of Lemma 5 in $O(1)$ time by Lemma 6.

Once the vertex v for the reduction is chosen, we now have to find the right pair of neighbors for the reduction. Let $w_1, \ldots, w_5$ be the neighbors of v in clockwise order, named such that $\deg(w_5) \geq \deg(w_i)$, $i = 1, 2, 3, 4$.

If $\deg(w_5) \leq 9$, then $s = \sum_{i=1}^{5} \deg(w_i) \leq 45$. In this case, to choose the neighbors for contraction according to algorithm PickContractionPair, we simply try for the 5 non-consecutive pairs of neighbors whether they are not adjacent, and if so, what the results of the contraction would be. This can be done in $O(5s) = O(1)$ time.

If $\deg(w_5) \geq 10$, then we find the contraction pair by testing whether w_1 and w_3 are adjacent. If not, then $\{w_1, w_3\}$ is the contraction pair, else $\{w_2, w_4\}$ is the contraction pair. (We cannot have both edges (w_1, w_3) and (w_2, w_4) by planarity.) Either way we delete w_5, and therefore $d_m \geq \deg(w_5) + 5 + 5 + 2 - 3 \geq 19$ edges, so this complies with algorithm PickContractionPair. To find the contraction pair, we spent $O(\deg(w_1)) = O(\deg(w_5)) = O(d_m)$ time. We contract two vertices of $\{w_1, w_2, w_3, w_4\}$, and because none of their degrees exceeds $\deg(w_5)$, we have at most $2\deg(w_5) \leq 2d_m$ edges that are incident to a contracted vertex.

5.3 Performing the reduction

In this section, we show how to perform the reduction in time $O(d_n + d_m + c_m)$, where d_n is the number of deleted vertices, d_m is the number of deleted edges, and c_m is the number of edges incident to a vertex that was contracted with another vertex.

5.3.1 Deleting an edge

Every time we delete an edge, we have to update the data structure, because degrees may change and separating triangles may be deleted, hence the assignment of vertices to buckets must be changed. Let (v_0, v_1) be the edge to be deleted, and denote by d_0 and d_1 the degrees of v_0 and v_1 before the deletion. For $i = 0, 1$, we do the following:

- If $d_i \leq 6$, then remove v_i from its current bucket and put it into $B_{d_i - 1}$.
- If $d_i \leq 5$, then test for the at most five neighbors x of v_i whether x belongs to B_3^d. If so, then now the conditions for x may have change; rather than testing whether they did indeed change, we simply put x to B_3. Since v_i has at most 5 neighbors, this takes $O(1)$ time total.
- If $d_i \leq 9$, then test for the at most nine neighbors x of v_i whether x belongs to B_5^t. If so, then put x into B_5. Since v_i has at most 9 neighbors, this takes $O(1)$ time total.
- Decrease $d(v_i)$ by one.

All operations together take $O(1)$ time.

5.3.2 *delete*(v)

We perform *delete*(v) in $O(d_n + d_m)$ time as follows:

- Determine the neighbors $w_1, \ldots, w_d$ of v.
- For $k = 1, \ldots, d$, delete all incident edges of w_k as described above.
- Remove $v, w_1, \ldots, w_d$ from bucket B_0.

5.3.3 *contract*(v, w_i, w_j)

Every time we perform *contract*$(v; w_i, w_j)$, we have to update the data structure, because degrees may change, and hence the assignment of vertices to buckets must be changed. Denote by d_i and d_j the degrees of w_i and w_j before the contraction. We perform *contract*$(v; w_i, w_j)$ as follows:

- Determine the neighbors $w_1, \ldots, w_d$ of v.
- Delete all incident edges of v and of $w_k \neq w_i, w_j$, $k = 1, \ldots, d$.
- Remove v and $w_k \neq w_i, w_j$, $k = 1, \ldots, d$, from bucket B_0.
- For $k = i, j$, if $d_k \leq 9$, then test for the at most nine neighbors x of w_k whether x belongs to B_3^d or to B_5^t. If so, then put x into B_3 or B_5, respectively. Since w_k has at most 9 neighbors, this takes $O(1)$ time total.
- Add the incident edges of w_i to the edge list of w_j. By keeping track where the edges (w_i, v) and (w_j, v) were in the edge lists of w_i and w_j (where v is the reduction vertex) we can unify the lists while maintaining the planar embedding in $O(d_i + d_j) = O(c_m)$ time.
- Test for multiple edges in $O(c_m)$ time, and delete any multiple edge as described above.
- Remove w_i from its bucket.
- Remove w_j from its current bucket and put it into $B_{\min\{6, d_i + d_j\}}$.
- Set $d(w_i) = d(w_j) = d_i + d_j$.

5.4 Putting everything together

We have shown in the previous subsections that selecting the right vertex for the reduction, selecting the right neighbors for the contraction (if applicable), and doing deletions and contractions can be done in $O(d_n + c_m + d_m)$ time, where d_n is the number of deleted vertices, d_m is the number of deleted edges and c_m is the number of edges incident to a contracted vertex. We have also shown $c_m \leq 2d_m$ in all cases, thus the time is $O(d_n + d_m)$. Since we never add new vertices or edges, this means that all reductions together take $O(n + m) = O(n)$ time.

Also note that throughout the algorithm the contents of the buckets obey the conditions outlined in Section 5.1. This is because with any change of the degree of a vertex v, we move v to the correct bucket, and also move all neighbors x to the correct bucket if $x \in B_3^d$ or $x \in B_5^t$. No other vertices can be affected by a degree-change at v. We conclude:

Theorem 10 *For any planar graph G, we can find at least $\frac{5}{23}n$ independent vertices in $O(n)$ time.*

Theorem 11 *For any planar graph G and $D \geq 7$, we can find at least* $\min\{\frac{5}{23}n, \frac{D-6}{4D-18}n\}$ *independent vertices of degree $< D$ in $O(n)$ time.*

6 Conclusion

In this paper, we studied algorithms for finding independent sets in planar graphs. The traditional greedy method finds $\frac{1}{6}n$ independent vertices. A variation of the greedy method that contracts vertices if possible finds $\frac{1}{5}n$ independent vertices. As our main result, if we choose the reduction vertices and the vertices to be contracted wisely, this greedy-method will in fact always yield $\frac{5}{23}n$ independent vertices. Our variant of the greedy-method can be implemented in $O(n)$ time.

Using the same algorithm after deleting all vertices of degree $\geq D$ for $D \geq 7$, we obtain $\min\{\frac{5}{23}n, \frac{D-6}{4D-18}n\}$ independent vertices of degree $< D$ in linear time. This has applications for some algorithms in computational geometry [Kir83, SvK97]. Our bound is best-possible for $D = 8$.

The foremost remaining open problem is to find a heuristic that works in $o(n^2)$ time and finds an independent set of size $\frac{1}{4}n$, or at least $> \frac{2}{9}n$, in a planar graph. Is it possible to modify the choice of the reduction vertex suitably to achieve this goal, or must we use an algorithm that is not based upon some kind of greedy-strategy?

In particular, we conjecture that if we always choose a vertex of minimum degree such that reducing it deletes the maximum possible number of edges, then the independent set obtained has size $\frac{1}{4}n$, or something close to it. Is this true, and how can this be shown?

Along the same lines, can the bounds m_α^β be raised, possibly after splitting some cases into more subcases, as done for x_5-reductions? This would, depending on which bound is raised, also increase the bound on the independent set. For example, if we could raise m_5^{19} to 20, this would increase the bound on the independent set to $\frac{12}{55}n \approx 0.218n$. Can we raise any of the bounds, and how can this be shown?

Acknowledgments

The author would like to thank Erik Demaine, Anna Lubiw, Ian Munro, and Jason Schattman for useful discussions, and Patrice Belleville for kindly sharing his manuscript. Parts of this paper were written while the author was at McGill University, and parts while visiting RUTCOR, Rutgers University. The author is grateful for access to CPLEX through RUTCOR, which helped solving the linear programs.

References

[AH77] K. Appel and W. Haken. Every planar map is four colorable. I. Discharging. *Illinois J. Math.*, 21(3):429–490, 1977.

[AHK77] K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. II. Reducibility. *Illinois J. Math.*, 21(3):491–567, 1977.

[Alb76] M. Albertson. A lower bound for the independence number of a planar graph. *J. Comb. Theory (B)*, 20:84–93, 1976.

[Bel99] P. Belleville. On an algorithm to find independent sets in planar graphs, 1999. Manuscript.

[Chv83] V. Chvátal. *Linear programming.* W. H. Freeman and Company, New York, 1983.

[CNS81] N. Chiba, T. Nishizeki, and N. Saito. A linear 5-coloring algorithm of planar graphs. *J. Algorithms*, 2:317–327, 1981.

[CNS83] N. Chiba, T. Nishizeki, and N. Saito. An algorithm for finding a large independent set in planar graphs. *Networks*, 13:247–252, 1983.

[Ede88] H. Edelsbrunner. *Algorithms in combinatorial geometry.* Springer-Verlag, 1988.

[GJS76] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[HT74] J. E. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of the Association for Computing Machinery*, 21(4):549–568, October 1974.

[Kir83] D. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. on Computing*, 12:28–35, 1983.

[RSST97] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. The four-colour theorem. *J. Combin. Theory Ser. B*, 70(1):2–44, 1997.

[SvK97] J. Snoeyink and M. van Kreveld. Linear-time reconstruction of Delauney triangulations with applications. In R. Burkhard and G. Woeginger, editors, *Proceedings European Symposium on Algorithms*, volume 1284 of *Lecture Notes in Computer Science*, pages 459–471. Springer-Verlag, 1997.

A Proof of Lemma 5

In this section, we prove Lemma 5 and 6, combined into one lemma as follows.

Lemma 12 *Let G be a planar graph with minimum degree 5. Then there exists a vertex v with* $\deg(v) = 5$ *and two non-adjacent non-consecutive neighbors w_i, w_j of v such that*

1. *contract*(v, w_i, w_j) *deletes at least 19 edges, or*
2. *the following three conditions, called* Conditions (*), *hold:*
   - *contract*(v, w_i, w_j) *deletes at least 16 edges that are not multiple edges,*
   - *after contract*(v, w_i, w_j), *there exists a vertex z with* $\deg(z) \leq 4$, *and*
   - *there exists at most one long edge of v with both endpoints $\neq w_i, w_j$, i.e., with both endpoints among the deleted neighbors of v.*

Moreover, if a vertex u is given such that $\deg(u) = 5$ *and u does not belong to a separating triangle, then the vertex v can be found in $O(1)$ time.*

Proof: Let u be a vertex of degree 5 that does not belong to a separating triangle (cf. Corollary 2), so u has no long edge. Let $z_1, \ldots, z_5$ be the neighbors of u in clockwise order. Because u has no long edge, any adjacent neighbors of u are also consecutive, and we are free to pick w_i, w_j as any non-consecutive neighbors of u as long as we delete 19 edges or satisfy Conditions (*).

We distinguish three cases.

1. There are three non-consecutive neighbors $z_\alpha, z_\beta, z_\gamma$ of u such that $\deg(z_\alpha) + \deg(z_\beta) + \deg(z_\gamma) \geq 18$. (Note that since u has 5 neighbors and every vertex knows its degree, we can test for the existence of this case in $O(1)$ time.)

   In this case, let $v = u$ and let w_i and w_j be the two neighbors $\neq z_\alpha, z_\beta, z_\gamma$ of u. Vertices w_i and w_j are not consecutive (because otherwise $z_\alpha, z_\beta, z_\gamma$ would be consecutive), thus we can perform *contract*(u, w_i, w_j), which deletes u and z_α, z_β and z_γ. Since at most two of $z_\alpha, z_\beta, z_\gamma$ are consecutive, there is at most one edge between $z_\alpha, z_\beta, z_\gamma$. Hence the number of deleted edges is at least $18 + 2 - 1 = 19$ by Fact 4.

2. Two neighbors of u, say z_1 and z_k $(k \neq 1)$, are non-adjacent and have degree 5. (Note that this condition can be tested in $O(1)$ time, because for any pair of neighbors of u that have degree 5 we can test in $O(5) = O(1)$ time whether they are adjacent.)

   Let $y_1, \ldots, y_4$ be the neighbors $\neq u$ of z_1, enumerated in clockwise order starting after u. Note that $(u, y_2), (u, y_3) \notin E$, because either edge would be a long edge for z_1, and therefore in a separating triangle, but u does not belong to a separating triangle.

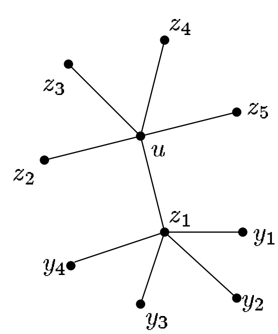


Figure 3: The case that u has two non-consecutive neighbors of degree 5.

   We distinguish subcases, which can be tested in $O(1)$ time:

   (a) $\deg(y_2) = 5$ or $\deg(y_3) = 5$: In this case, set $v = u$, $w_i = z_2$ and $w_j = z_5$. Doing *contract*(v, w_i, w_j) deletes at least 16 edges, because u has no long edge. This does not delete or contract either of y_2 and y_3, because these two vertices are not neighbors of u. Since this deletes z_1, one of y_2 or y_3 now has degree ≤ 4. Since u has no long edge, Conditions (*) hold.

   (b) $\deg(y_2) \geq 6, \deg(y_3) \geq 6$: In this case, set $v = z_1$. One of the edges (y_1, y_3) and (y_2, y_4) cannot exist, otherwise we could complete $\{z_1, y_1, y_2, y_3, y_4\}$ to a planar K_5. We will only study $(y_1, y_3) \notin E$, the other case is similar.

Set $w_i = y_1$, $w_j = y_3$. The long edge (u, y_2) of v does not exist, therefore *contract*(v, w_i, w_j) deletes at least $\deg(y_2) + \deg(y_4) + \deg(v) + 2 - 2 \geq 6 + 5 + 5 = 16$ edges. This does not delete or contract z_k because z_1 and z_k are not adjacent; so the degree of z_k is now ≤ 4 because u has been deleted. Since $(u, y_2) \notin E$, there is at most one long edge between deleted neighbors of v, so Conditions (*) hold.

3. Otherwise: If we have neither three non-consecutive neighbors whose degrees sum to at least 18, nor two non-adjacent neighbors of degree 5, then we can say a lot about the structure of the vicinity of v. We illustrate the situation in Figure 4 and explain it below.

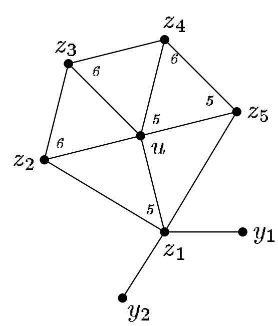


Figure 4: The case that there are neither three non-consecutive neighbors whose degree sum to at least 18, nor two non-adjacent neighbors of degree 5. Little italicized numbers denote the degree.

Of the 5 neighbors $z_1, \ldots, z_5$ of u, there must be at least two neighbors of degree 5, because otherwise we would have three non-consecutive neighbors of degree ≥ 6 and would be in case (1). On the other hand, at most two neighbors can have degree 5; otherwise, two of them would not be adjacent since u does not have a long edge, and we would be in case (2). So there are exactly two neighbors of degree 5, and they must be adjacent, because otherwise we could be in case (2). Because u does not have a long edge, these two neighbors must be consecutive, and after suitable renaming, we may assume that z_1 and z_5 have degree 5, while z_2, z_3, z_4 have degree ≥ 6. None of z_2, z_3, z_4 can have degree ≥ 7, because otherwise we could find three non-consecutive neighbors the degrees of which sum to at least $7 + 6 + 5 = 18$ and we are in case (1). So z_2, z_3, z_4 all have degree 6.

We distinguish subcases:

(a) For some $k \in \{1, \ldots, 4\}$, the edge (z_k, z_{k+1}) does not exist. (This can be tested in $O(\sum_{k=1}^{4} \deg(z_k)) = O(1)$ time.)
In this case, set $v = u$, $w_i = z_{k-1}$ and $w_j = z_{k+2}$, where we define $z_0 = z_5$ and $z_6 = z_1$. At least two of the deleted vertices have degree 6 by $k \neq 5$. Since u has no long edge, and since $(z_k, z_{k+1}) \notin E$, this deletes $5 + 6 + 6 + 2 - 0 = 19$ edges.

(b) The edges $(z_1, z_2), (z_2, z_3), (z_3, z_4), (z_4, z_5)$ exist. Note that edge (z_1, z_5) also exists (otherwise we would be in case (2)). By $\deg(z_1) = 5$, there are only two neighbors of z_1 that are $\neq u, z_2, z_5$. Let y_1 and y_2 be these two neighbors of z_1, enumerated in clockwise order after z_5. Note that they must appear between z_5 and z_2 in the clockwise order of neighbors of z_1, and that neither of them can be adjacent to u, because u is not in a separating triangle. Again we have two subcases, which can be tested in $O(1)$ time.

i. $\deg(y_1) = 5$ or $\deg(y_2) = 5$: In this case, set $v = u$, $w_i = z_2$ and $w_j = z_5$. Doing *contract*(v, w_i, w_j) deletes at least 16 edges, because there is no long edge of u. This does not delete or contract either of y_1 and y_2, because they are not neighbors of u. Since z_1 is deleted, one of y_1 or y_2 now has degree ≤ 4. Since u has no long edge, Conditions (*) hold.

ii. $\deg(y_1) \geq 6, \deg(y_2) \geq 6$: In this case, set $v = z_1$. One of the edges (y_1, z_2) and (y_2, z_5) cannot exist, otherwise we could complete $\{z_1, y_1, y_2, z_2, z_5\}$ to a planar K_5. We will only study $(y_1, z_2) \notin E$, the other case is similar.
Set $w_i = y_1$, $w_j = z_2$. The edge (u, y_2) does not exist, therefore *contract*(v, w_i, w_j) deletes at least $\deg(y_2) + \deg(z_5) + \deg(u) + 2 - 2 \geq 6 + 5 + 5 = 16$ edges. This deletes u and z_5, both of which are neighbors of z_4. This does not delete or contract z_4, because $z_4 \neq z_2, z_5, u$ by definition, and $z_4 \neq y_1, y_2$ because z_4 is adjacent to u while y_1 and y_2 are not. Therefore, we have deleted at least two neighbors of z_4 and now $\deg(z_4) \leq 4$. Since $(u, y_2) \notin E$, there is at most one long edge between deleted neighbors of v, so Conditions (*) hold. □

B Proof of Lemma 7

In this section, we prove Lemma 7, which states the following.

Lemma 13 *Let the minimum degree be* 5*, let* v *be chosen as in Lemma 5, and let* w_1, w_3 *be the neighbors of* v *chosen with algorithm* PickContractionPair*. If contract*(v, w_1, w_3) *deletes* k *edges and is followed by an* x_3^d*-reduction at* x*, then delete*(x) *deletes at least* $25 - k$ *edges.*

Proof: Let R_1 be the x_5-reduction at v and let R_2 be the x_3^d-reduction at x. We know that R_1 deletes at least 16 edges, so we are done if R_2 deletes at least 9 edges. We also know that R_2 deletes at least 6 edges, so we are done if R_1 deletes at least 19 edges. Therefore for the remainder of this proof we assume that R_1 deletes at most 18 edges and R_2 deletes at most 8 edges.

Let the neighbors of v be $w_1, \ldots, w_5$ in clockwise order. Let the neighbors of x that remain after R_1 be y_1, y_2, y_3, not necessarily in clockwise order (we want to be free to rename them to achieve other properties later). Note that $\{x, y_1, y_2, y_3\}$ induces a K_4, otherwise by algorithm PickDeg3Vertex we would have done an x_3^c-reduction instead of the x_3^d-reduction R_2 that deletes at most 8 edges.

To simplify notation, we denote by $\deg_b(z)$ and $\deg_a(z)$ the degree of a vertex z before and after R_1. Note that $\deg_b(z) \geq 5$ and $\deg_a(z) \geq 3$ for all vertices z. Also note $\deg_a(x) = 3$.

The proof splits into two major cases, depending on whether the vertex w^* that results from R_1, i.e., from contracting the two neighbors of v, is one of the vertices deleted during R_2.

1. The contracted vertex w^* is not x or a neighbor of x.

   The proof in this case is outlined as follows: First show that R_2 deletes at least 8 edges, so we are done unless R_1 deletes exactly 16 edges. Then show that there exists a pair of neighbors of v the contraction of which would delete at least 17 edges and satisfy Conditions (*). By Algorithm PickContractionPair and because no x_3^c-reduction happened, we picked this or a better pair, so we deleted at least 25 edges total.

So assume that w^* is neither x nor a neighbor of x. In this case, the K_4 induced by $\{x, y_1, y_2, y_3\}$ existed already before R_1, and none of these vertices is incident to v. Therefore, v and its neighbors must be inside one face F of this K_4. One vertex of $\{x, y_1, y_2, y_3\}$, say y_3, does not belong to F, and therefore is not incident to any vertex deleted during R_1, which implies $\deg_a(y_3) = \deg_b(y_3) \geq 5$. As a consequence, R_2 deletes at least $8 = 3+3+5-3$ edges. So we are done unless R_1 deletes exactly 16 edges and R_2 deletes exactly 8 edges, which we assume for the remainder of this case.

By $\deg_a(y_3) \geq 5$, we must have $\deg_a(y_1) = \deg_a(y_2) = 3$ and $\deg_a(y_3) = 5$, because R_2 deletes $\deg_a(y_1) + \deg_a(y_2) + \deg_a(y_3) - 3 = 8$ edges. For easier notation, denote $y_0 = x$, then we can rewrite this as $\deg_a(y_i) = 3$ for $i = 0, 1, 2$. Observe that the x_3^d-reduction at x deletes exactly the same vertices and edges as an x_3^d-reduction at y_1 or y_2, hence we are free to exchange y_0, y_1, y_2 as desired.

For $i = 0, 1, 2$, since $\deg_b(y_i) \geq 5$ and $\deg_a(y_i) = 3$, at least two incident edges of y_i must have been deleted during R_1. Because R_1 deleted exactly 16 edges, no multiple edges have been deleted (note the first condition in (*)), which means that each of y_0, y_1, y_2 must be incident to at least two of the deleted vertices w_2, w_4, w_5. Denote $W = \{w_2, w_4, w_5\}$ and denote by $N(y_i)$ the set of neighbors of y_i in W, $i = 0, 1, 2$.

We want to show that (after suitable renaming) the configuration is as shown in Figure 6, and do this in the following series of observations:

(a) $|N(y_i)| \geq 2$ for $i = 0, 1, 2$, because at least two incident edges of y_i must be deleted during the x_5-reduction.

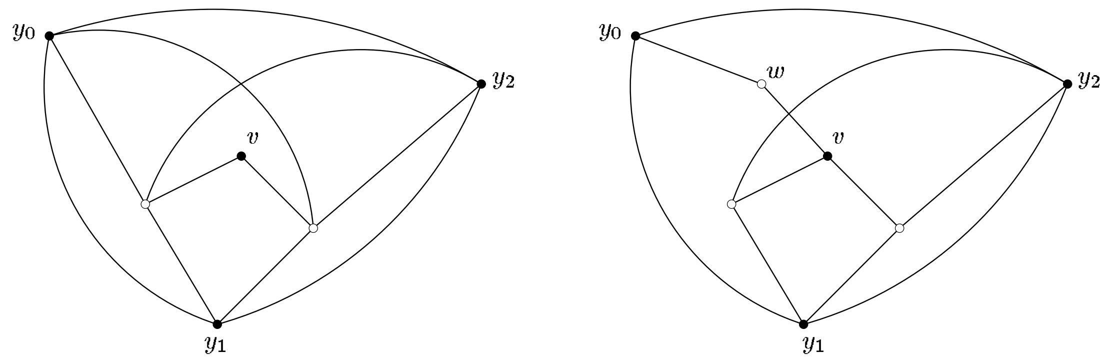
(b) It is not possible that $|N(y_0) \cap N(y_1) \cap N(y_2)| \geq 2$, for otherwise we would have a planar K_5 in $N(y_0) \cup \{y_0, y_1, y_2\}$, using the path through v to connect the vertices in $N(y_0)$. See Figure 5(a). Therefore by $|N(y_0)| \geq 2$ there exists an element in $N(y_0)$ that does not belong both to $N(y_1)$ and $N(y_2)$; after possible exchange of y_1 and y_2 we may assume $N(y_0) - N(y_1) \neq \emptyset$.

(c) It is not possible that $N(y_1) \subseteq N(y_2)$, because otherwise we would have a planar $K_{3,3}$ in $(N(y_1) \cup \{y_0\}) \cup \{y_1, y_2, v\}$, using the path $y_0 - w - v$ to connect v and y_0, where $w \in N(y_0) - N(y_1) \neq \emptyset$. See Figure 5(b). Therefore $N(y_1) - N(y_2) \neq \emptyset$.

(d) It is not possible that $N(y_2) \subseteq N(y_0)$, because otherwise we would have a planar $K_{3,3}$ in $(N(y_2) \cup \{y_1\}) \cup \{y_0, y_2, v\}$, using the path $y_1 - w - v$ to connect v and y_1, where $w \in N(y_1) - N(y_2) \neq \emptyset$. Therefore $N(y_2) - N(y_0) \neq \emptyset$.

(e) $|N(y_i)| = 2$ for $i = 0, 1, 2$. For we know already $|N(y_i)| \geq 2$. If $|N(y_i)| \geq 3$, then $N(y_i) = W$, which means that $N(y_{i-1}) \subseteq N(y_i)$ (addition modulo 3), which was shown impossible above. Therefore, $\deg_b(y_i) = \deg_a(y_i) + |N(y_i)| = 5$ for $i = 0, 1, 2$.

(f) Since no two $N(y_i)$ coincide, but each has 2 vertices and is contained in W which has 3 vertices, for each pair of vertices in W there is exactly one y_i which is incident to both.



(a) (b)

Figure 5: The case where the contracted vertex is not a neighbor of x. We illustrate Observations 1b and 1c. Vertices in W are shown white. For a vertex y_i $(i = 0, 1, 2)$ we do not necessarily show all edges to neighbors in W, but only those that are relevant to construct a planar K_5 or $K_{3,3}$.

After possible renaming of y_0, y_1, y_2 therefore y_0 is incident to w_2, w_4, y_1 is incident to w_4, w_5 and y_2 is incident to w_5, w_2. See Figure 6.

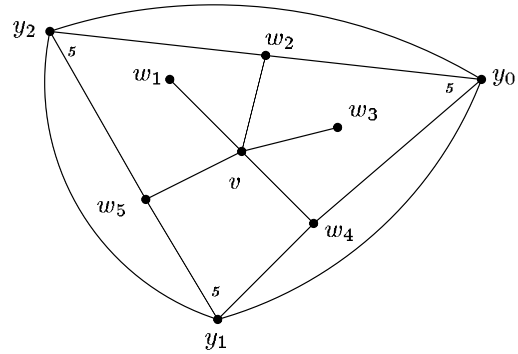


Figure 6: The configuration if the contracted vertex is not a neighbor of x.

By Conditions (*), we know that there is at most one long edge between deleted neighbors of v, i.e., at most one of the edges (w_2, w_4) and (w_2, w_5) exists. Assume that $(w_2, w_5) \notin E$, the other case is similar. Note that we cannot have an edge (w_1, w_3), because these were the contracted vertices. We also cannot have an edge (w_1, w_4), because otherwise we could could complete $\{w_1, w_2, w_4, w_5, v\}$ to a planar K_5 using y_2 to connect w_2 and w_5.

Therefore, if we performed *contract*(v, w_2, w_5), no long edge would be deleted. This implies that at least 16 edges are deleted. But in fact, at least 17 edges are deleted, because this contraction causes a multiple edge from the contracted vertex to y_2, and therefore another edge deletion. Because we delete an incident edge of y_2, and $\deg_b(y_2) = 5$, we now have a vertex of degree 4.

This proves Conditions (*), so we have found a pair of vertices the contraction of which deletes at least 17 edges and satisfies Conditions (*) as desired.

2. The contracted vertex w^* is x or a neighbor of x.

Since R_2 deletes at most 8 edges, one neighbor of x, say y_1, must have degree 3 during R_2. Note that the exact same vertices and edges are deleted whether we call it an x_3^d-reduction at x or an x_3^d-reduction at y_1. Therefore, after possible exchange of x and y_1, we can assume that w^* is a neighbor of x, say y_3.

The proof in this case is outlined as follows. As a first part, show that there exists a pair of non-adjacent non-consecutive neighbors of v the contraction of which deletes at least 17 edges and satisfies Conditions (*). By algorithm PickContractionPair, therefore at least 17 edges were deleted during the x_5-reduction, because no x_3^c-reduction followed. So we are done if the x_3^d-reduction deletes at least 8 edges. As second part, show that if the x_3^d-reduction deletes at most 7 edges, then there exists a pair of neighbors the contraction of which satisfies Conditions (*) and creates an x_3^c-reduction. By algorithm PickContractionPair, therefore at least 19 edges were deleted during the x_5-reduction, because no x_3^c-reduction followed. This proves the lemma because the x_3^d-reduction deletes at least 6 edges.

As in the previous case, our argument will rely on the fact that many of the vertices involved in the x_3^d-reduction must be adjacent to the neighbors of v, because some of their incident edges have been deleted during the x_5-reduction. More precisely, let $z \in \{x, y_1, y_2\}$. At least $\deg_b(z) - \deg_a(z)$ incident edges of z have been deleted during R_1. Because z is incident to the contracted vertex y_3, it must have been incident to one of w_1, w_3, the vertices that were contracted, and this edge has not been deleted. Therefore, $z \in \{x, y_1, y_2\}$ is incident to at least $\deg_b(z) - \deg_a(z) + 1$ vertices in $w_1, \ldots, w_5$.

Denote $W = \{w_1, \ldots, w_5\}$ and let $N(z)$ be the neighbors of z in W for $z \in \{x, y_1, y_2\}$; then by the above $|N(z)| = 1 + \deg_b(z) - \deg_a(z)$. Since $\deg_a(x) = 3$, this implies $|N(x)| \geq 3$. Assume the naming of y_1 and y_2 is such that $|N(y_1)| \geq |N(y_2)|$, and if $|N(y_1)| = |N(y_2)|$, then $|N(x) \cup N(y_1)| \geq |N(x) \cup N(y_2)|$.

Part I: Show that the x_5-reduction deleted at least 17 edges.

We make a series of observations, which leads us to understand the structure of the vicinity of v better.

(a) We must have $|N(y_1)| \geq 2$, for otherwise $|N(y_2)| \leq |N(y_1)| \leq 1$ by choice of y_1, which implies $\deg_a(y_1) \geq 5$ and $\deg_a(y_2) \geq 5$, and R_2 deletes at least $3 + 5 + 5 - 3 = 10$ edges.

(b) If $|N(y_1)| = 2$, then $|N(y_2)| = 2$. For if $|N(y_1)| = 2$ and $|N(y_2)| \leq 1$, then $\deg_a(y_1) \geq 4$, $\deg_a(y_2) \geq 5$, and R_2 deletes at least 9 edges.

(c) We cannot have $N(y_1) \subseteq N(x)$, for this leads to a contradiction in all cases:

   i. If $|N(y_1)| \geq 3$, then $N(y_1) \subseteq N(x)$ implies a planar $K_{3,3}$ in $N(y_1) \cup \{y_1, x, v\}$. See Figure 7(a).

   ii. If $|N(y_1)| = 2$, then $|N(y_2)| = 2$ by the previous observation; furthermore the choice of y_1 tells that $N(y_1) \subseteq N(x)$ implies $N(y_2) \subseteq N(x)$. This leads to a contradiction in both the following cases:

      A. If $N(y_1) = N(y_2)$, then we have a planar K_5 in $N(y_1) \cup \{x, y_1, y_2\}$, using the path through v to connect the vertices in $N(y_1)$. See Figure 7(b).

B. If $N(y_1) \neq N(y_2)$, then $|(N(y_1) \cup N(y_2)) \cap N(x)| \geq 3$, and we have a planar $K_{3,3}$ in $(N(y_1) \cup N(y_2)) \cup \{x, y_1, v\}$, connecting y_1 to the vertex $w \in N(y_2) - N(y_1)$ using $y_1 - y_2 - w$. See Figure 7(c).

Therefore $N(y_1) - N(x) \neq \emptyset$, and $|N(y_1) \cup N(x)| \geq 4$.

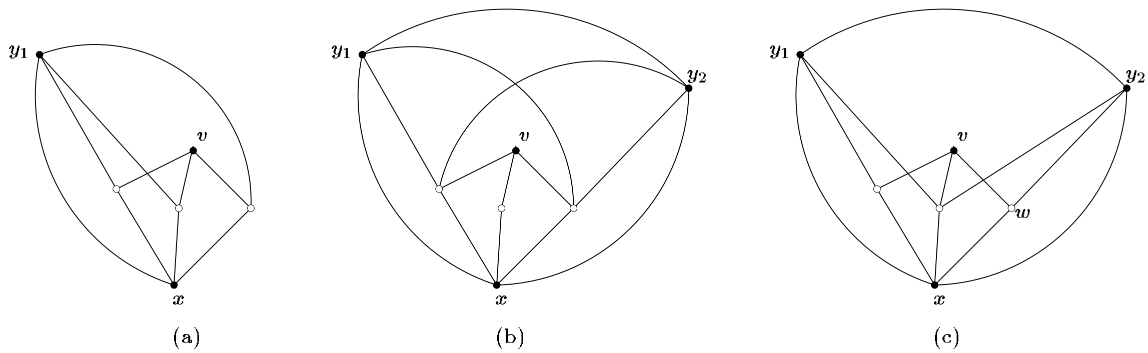


Figure 7: The case where the contracted vertex is a neighbor of x and the x_5-reduction deletes 16 edges. We illustrate Observations 2(c)i, 2(c)iiA, and 2(c)iiB. Vertices in W are shown white. We do not necessarily show all edges to neighbors in W for x, y_1, y_2, but only those edges that are relevant to construct a planar K_5 or $K_{3,3}$.

(d) We have $|N(x)| \leq 4$, for otherwise $N(x) = W$ and therefore $N(y_1) \subseteq N(x)$, which contradicts the previous observation. In particular therefore $\deg_b(x) = \deg_a(x) + |N(x)| - 1 \leq 3 + 4 - 1 = 6$.

Enumerate 4 vertices in $N(x) \cup N(y_1)$ in clockwise order around v as $w_{i_1}, w_{i_2}, w_{i_3}, w_{i_4}$ such that $w_{i_1}, w_{i_2}, w_{i_3}$ are adjacent to x and w_{i_4} is adjacent to y_1; this exists by Observation 2c and by $|N(x)| \geq 3$.

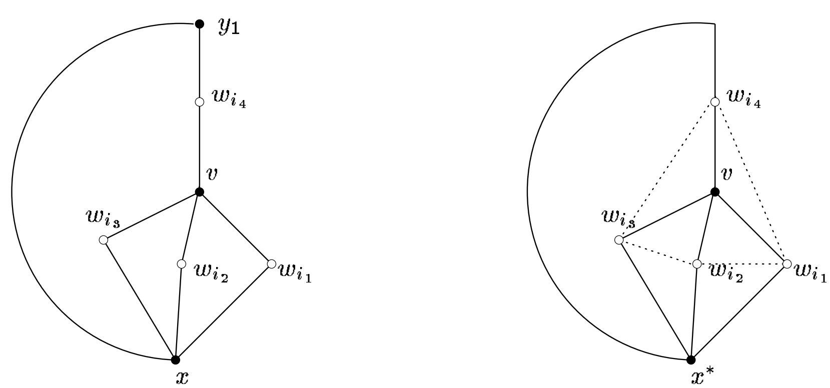


Figure 8: On the left, the configuration that we conclude to exist if the contracted vertex is a neighbor of x and the x_5-reduction deletes 16 edges. On the right, we show G', the graph resulting from some contractions and possibly adding edges (shown dotted).

Claim: Operation *contract*(v, w_{i_1}, w_{i_3}) deletes at least 17 edges and satisfies Conditions (*). For the purpose of proving this claim, we will modify the graph by contracting vertices and adding edges while preserving planarity. First add all edges between consecutive neighbors of v if not present yet; this preserves planarity. Let w_α be the one neighbor of v that is not in $\{w_{i_1}, w_{i_2}, w_{i_3}, w_{i_4}\}$; this vertex is consecutive to one of w_{i_2}, w_{i_4}. Contract w_α into the vertex out of w_{i_2} and w_{i_4} to which it is consecutive; this preserves planarity because they are adjacent. Finally, contract x and y_1 to a new vertex x^*; this preserves planarity because x and y_1 are adjacent.

Let G' be the graph that is now induced by $\{v, x^*, w_{i_1}, w_{i_2}, w_{i_3}, w_{i_4}\}$. Because each w_{i_j} is adjacent to v, to one of x and y_1, and consecutive neighbors of v are adjacent, this graph is triangulated (indeed, it is known as the *octahedron*). Therefore we cannot add another edge to G' without destroying planarity. This implies that there is no edge between w_{i_1} and w_{i_3} in G', and therefore also not in the original graph. So w_{i_1} and w_{i_3} are non-adjacent and non-consecutive.

By planarity there is no edge between w_{i_2} and w_{i_4} in G'. Therefore in the original graph there was no long edge of v between vertices $w_{i_2}, w_{i_4}, w_\alpha$, because such an edge would have been contracted into an edge (w_{i_2}, w_{i_4}) in G'. Therefore there exists no long edge between deleted neighbors, which implies that *contract*(v, w_{i_1}, w_{i_3}) deletes at least 16 edges. But in fact, at deletes at least 17 edges, because the contraction causes a multiple edge to x, which is also deleted, but was not counted before. Also, we delete two incident edges of x (the multiple edge and the edge to w_{i_2}), hence by $\deg_b(x) \leq 6$ vertex x now has degree at most 4. Therefore the contraction satisfies Conditions (*), which proves the claim.

This shows that we could have deleted 17 edges with the right choice of neighbors for contraction. By algorithm PICKCONTRACTIONPAIR, and because no x_3^c-reduction happened, we did indeed delete at least 17 edges during R_1, which proves the first part. So the lemma is true unless R_2 deletes at most 7 edges, which we assume for the remainder of this proof.

Part II: Show that the x_5-reduction deleted at least 19 edges.

The crucial ingredient to show this claim is that R_2 deletes at most 7 edges, which we concluded above. Again we make a series of observations, which leads us to understand the structure of the vicinity of v better.

(e) $|N(y_1)| \geq 3$. For if $|N(y_2)| \leq |N(y_1)| \leq 2$, then $\deg_a(y_1) \geq 4$, $\deg_a(y_2) \geq 4$, and R_2 deletes at least $3 + 4 + 4 - 3 = 8$ edges.

(f) $|N(y_2)| \geq 2$. For if $|N(y_2)| \leq 1$, then $\deg_a(y_2) \geq 5$, and R_2 deletes at least $3+3+5-3 = 8$ edges.

(g) $|N(x) \cap N(y_1)| \geq 3$ is impossible, because otherwise we have a planar $K_{3,3}$ in $(N(x) \cap N(y_1)) \cup \{v, x, y_1\}$ (see Figure 7(a) for an illustration).

(h) $N(x) \subseteq N(y_1)$ or $N(y_1) \subseteq N(x)$ is impossible, because otherwise $|N(x) \cap N(y_1)| \geq 3$ by $|N(x)| \geq 3$ and $|N(y_1)| \geq 3$. Therefore $N(x) - N(y_1) \neq \emptyset$ and $N(y_1) - N(x) \neq \emptyset$.

(i) $|N(y_2) \cap N(y_1)| \geq 2$ is impossible, because otherwise we have a planar $K_{3,3}$ in $(N(y_2) \cap N(y_1)) \cup \{x\} \cup \{y_1, y_2, v\}$, using the path $x - w - v$ where $w \in N(x) - N(y_1) \neq \emptyset$. See Figure 9(a).

(j) $|N(y_2) \cap N(x)| \geq 2$ is impossible, because otherwise we have a planar $K_{3,3}$ in $(N(y_2) \cap N(x)) \cup \{y_1\} \cup \{x, y_2, v\}$, using the path $y_1 - w - v$ where $w \in N(y_1) - N(x) \neq \emptyset$.

(k) $N(y_2) \subseteq N(x) \cap N(y_1)$ is impossible, because by $|N(y_2)| \geq 2$ this would contradict Observations 2i and 2j. Therefore $N(y_2) - (N(x) \cap N(y_1)) \neq \emptyset$.

(l) $|N(x) \cap N(y_1)| \geq 2$ is impossible, because otherwise we have a planar $K_{3,3}$ in $(N(x) \cap N(y_1)) \cup \{y_2\} \cup \{x, y_1, v\}$, using the path $y_2 - w - v$, where $w \in N(y_2) - (N(x) \cap N(y_1)) \neq \emptyset$. See Figure 9(b).

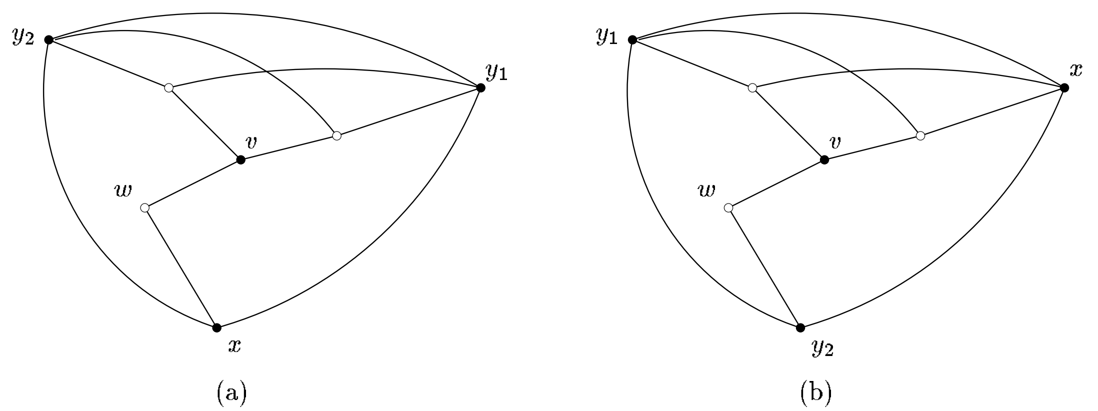


Figure 9: The case where the contracted vertex is a neighbor of x and the x_5-reduction deletes 17 edges. We illustrate Observations 2i and 2l. Vertices in W are shown white. For a vertex y_i $(i = 0, 1, 2)$ we do not necessarily show all edges to neighbors in W, but only those that are relevant to construct a planar K_5 or $K_{3,3}$.

(m) We have $5 = |W| \geq |N(x) \cup N(y_1) \cup N(y_2)| \geq |N(x)| + |N(y_1)| + |N(y_2)| - |N(x) \cap N(y_1)| - |N(y_2) \cap N(x)| - |N(y_2) \cap N(y_1)|$, which by Observation 2i,2j and 2l is at least $|N(x)| + |N(y_1)| + |N(y_2)| - 3 \geq 3 + 3 + 2 - 3 = 5$. Therefore equality holds everywhere, which implies the following:

   i. $|N(x)| = 3$, $|N(y_1)| = 3$ and $|N(y_2)| = 2$.

   ii. $N(x) \cup N(y_1) \cup N(y_2) = W$, so every vertex w_i, $i = 1, \ldots, 5$ is incident to one of x, y_1, y_2.

   iii. $\deg_a(y_2) = \deg_b(y_2) + 1 - |N(y_2)| \geq 5 + 1 - 2 = 4$.

   iv. $\deg_b(x) = \deg_a(x) - 1 + |N(x)| = 3 - 1 + 3 = 5$.

(n) All neighbors of x are contained in $\{w_1, \ldots, w_5, y_1, y_2\}$, because $|N(x)| = 3$, x is adjacent to y_1 and y_2, and $\deg_b(x) = 5$.

(o) $\deg_b(y_2) = 5$, for it cannot be less, and if it were more, then $\deg_a(y_2) \geq 6 - |N(y_2)| + 1 \geq 5$, so R_2 would delete at least 8 edges.

Claim: v has no long edges.
For the purpose of proving this claim, we will modify the graph by contracting vertices and adding edges while preserving planarity. First add all edges between consecutive neighbors of

v if not present yet; this preserves planarity. Then contract the vertices x, y_1 and y_2 into x^*; this preserves planarity because they are adjacent. The resulting subgraph G' induced by x^*, v and the neighbors $w_1, \ldots, w_5$ of v is triangulated because each w_i, $i = 1, \ldots, 5$ is incident to one of x, y_1, y_2 as shown above. So there cannot be any edge between two non-consecutive neighbors of v. This proves the claim.

Since $N(y_2) = 2$, there are two non-consecutive neighbors w_i and w_j of v that are not adjacent to y_2.

Claim: Operation *contract*(v, w_i, w_j) satisfies Conditions (*) and creates an x_3^c-reduction. Note first that we can indeed contract w_i and w_j, because they are not consecutive, and therefore not adjacent by the previous claim. Because v has no long edges, this deletes at least 16 edges. Moreover, we delete the two vertices in $N(y_2)$. By $\deg_b(y_2) = 5$, y_2 has degree 3 after this reduction. Let z be a neighbor of y_2 that is not in $\{w_1, \ldots, w_5, x, y_1\}$; this exists because $|N(y_2)| = 2$, but $\deg_b(y_2) = 5$. Then z is not adjacent to x, because all neighbors of x are in $\{w_1, \ldots, w_5, x, y_1\}$. Therefore, the neighbors x and z of y_2 are not adjacent, and we have created an x_3^c-reduction.

This shows that we could have created an x_3^c-reduction with the right choice of neighbors for contraction. By algorithm PICKCONTRACTIONPAIR, and because no x_3^c-reduction happened, we did deleted at least 19 edges during R_1, which proves the lemma because R_2 deletes at least 6 edges.

□