Chapter 1
A Better Approximation Algorithm for Finding Planar Subgraphs

Gruia Călinescu * Cristina G. Fernandes † Ulrich Finkler ‡ Howard Karloff *

Abstract

The MAXIMUM PLANAR SUBGRAPH problem—given a graph G, find a largest planar subgraph of G—has applications in circuit layout, facility layout, and graph drawing. No previous polynomial-time approximation algorithm for this NP-Complete problem was known to achieve a performance ratio larger than 1/3, which is achieved simply by producing a spanning tree of G. We present the first approximation algorithm for MAXIMUM PLANAR SUBGRAPH with higher performance ratio (2/5 instead of 1/3). We also apply our algorithm to find large outerplanar subgraphs. Last, we show that both MAXIMUM PLANAR SUBGRAPH and its complement, the problem of removing as few edges as possible to leave a planar subgraph, are Max SNP-Hard.

1 Introduction

MAXIMUM PLANAR SUBGRAPH is this problem: given a graph G, find a planar subgraph of G of maximum size, where *size* is the number of edges. This problem has applications in circuit layout, facility layout, and graph drawing [F92, TDB88].

MAXIMUM PLANAR SUBGRAPH is known to be NP-Complete [LG77]. For a graph G, let us define $Opt(G)$ to be the maximum size of a planar subgraph of G. Given an algorithm A that takes (representations of) graphs G as input and outputs subgraphs of G, define $A(G)$ to be the size of the planar graph A produces when G is the input. Now let us define A's *performance* or *approximation ratio* $r(A)$ to be the infimum, over all (representations of) graphs G, of $A(G)/Opt(G)$ (if $Opt(G) > 0$, and 1 otherwise). In the literature, authors sometimes ensure that their performance ratio is at least one by defining it to be the reciprocal of ours.

Numerous approximation algorithms for MAXIMUM PLANAR SUBGRAPH appear in the literature, the simplest ones being Spanning Tree (output any spanning tree of G, assuming G is connected) and Maximal Planar Subgraph (output any planar subgraph to which the addition of any new edge would violate planarity). Spanning Tree is known to have performance ratio 1/3 (see below). Dyer, Foulds and Frieze [DFF85] proved that Maximal Planar Subgraph has performance ratio 1/3. Cimikowski [Cim95] proved that a path embedding heuristic of Chiba, Nishioka and Shirakawa [CNS79] and an edge embedding heuristic of Cai, Han and Tarjan [CHT93] have performance ratios not exceeding 1/3. In the same paper, Cimikowski studied two other polynomial-time heuristics: the "vertex-addition heuristic" and the "cycle-packing heuristic." The performance ratio of the former, to the authors' knowledge, is not known, whereas for the cycle-packing algorithm, it is 0. Dyer, Foulds and Frieze [DFF85] studied two other algorithms and proved that each has performance ratio at most 2/9. Also see [JM93].

In short, to the authors' knowledge, no previously proposed algorithm was known to have a performance ratio exceeding 1/3. What makes the problem more tantalizing is that achieving a performance ratio of 1/3 is trivial. In fact, Spanning Tree has performance ratio 1/3, since every spanning tree of a connected graph on n vertices has $n-1$ edges and every planar graph on n vertices has at most $3n-3 = 3(n-1)$ edges (and there are planar graphs on n vertices with $3n-6$ edges, for all $n \geq 3$). No previous algorithm could beat the bound achieved by a trivial algorithm.

In this paper, we present two new approximation algorithms for MAXIMUM PLANAR SUBGRAPH. Each achieves a performance ratio exceeding 1/3. The higher performance ratio is $2/5 = 0.4$ and is achieved by an algorithm which (surprisingly) invokes an algorithm for the graphic matroid parity problem as a subroutine and which runs in time $O(m^{3/2} n \log^6 n)$. A greedy variant still has performance ratio $7/18 = 0.3888...$, and runs in linear time on graphs of bounded degree.

Next, we provide an extension of the main algorithm. We provide a nontrivial approximation algorithm for MAXIMUM OUTERPLANAR SUBGRAPH, which is this problem: given G, find an outerplanar sub-

*College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280. Research supported in part by NSF grant CCR-9319106.

†College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280. Research supported in part by the CNPq (Brazil), under contract 200975/92-7.

‡Max-Planck-Institut für Informatik, D-66123 Saarbrücken, Germany. Research supported in part by the Graduiertenkolleg Effizienz und Komplexität von Algorithmen und Rechenanlagen, Universität Saarbrücken.

graph of G of maximum size. (An *outerplanar* graph is a graph which can be drawn in the plane without crossing edges, with all vertices on the boundary of the exterior face [H72].) This new algorithm has performance ratio at least 2/3, which surpasses the bound of 1/2 which is trivially obtained by producing a spanning tree.

Last, we show that MAXIMUM PLANAR SUBGRAPH is Max SNP-Hard, implying that there is a constant $\epsilon > 0$ such that the existence of a polynomial-time approximation algorithm with performance ratio at least $1 - \epsilon$ would imply that $P = NP$ [ALMSS92]. In addition, we show that the complementary problem, called NONPLANAR DELETION or NPD—given $G = (V, E)$, produce a smallest subset $L \subseteq E$ such that $(V, E - L)$ is planar—is also Max SNP-Hard.

2 The Approximation Algorithms

In this section we present the two new algorithms for MAXIMUM PLANAR SUBGRAPH. The higher performance ratio is at least 2/5=0.4.

Let us give some motivation for our algorithm. As we said, given a (connected) graph G, an algorithm which outputs a spanning tree of G achieves a performance ratio of 1/3. A graph whose cycles all have length three, i.e., are triangles, is planar, as it cannot contain a subdivision of K_5 or $K_{3,3}$. Moreover, note that a connected spanning subgraph of G whose cycles are triangles, besides being planar, has one more edge per triangle than a spanning tree of G.

Our better algorithm produces a subgraph of G whose cycles are triangles and, among these subgraphs, has the maximum number of edges. It can be implemented in time $O(m^{3/2} n \log^6 n)$, where m is the number of edges in G and n is the number of vertices in G, using a graphic matroid parity algorithm, as we will see later. We first present a greedy version of the algorithm.

2.1 A Greedy Version of the Algorithm. Algorithm A, presented below, is a greedy version of our algorithm. It has a performance ratio of 7/18=0.3888.... After presenting the algorithm and proving its performance ratio is 7/18, we will show it can be implemented in linear time for graphs with bounded degree. We begin with some definitions.

A *triangular cactus* is a graph whose cycles (if any) are triangles and such that all edges appear in some cycle. A *triangular cactus in a graph G* is a subgraph of G which is a triangular cactus.

A *triangular structure* is a graph whose cycles (if any) are triangles. A *triangular structure in a graph G* is a subgraph of G which is a triangular structure. Note that every triangular cactus is a triangular structure, but not *vice versa*.

Algorithm A produces a triangular structure in the given graph G. The algorithm consists of two phases. First, A greedily constructs a maximal triangular cactus S_1 in G. Second, A extends S_1 to a triangular structure S_2 in G by adding as many edges as possible to S_1 without forming any new cycles.

Given a graph $G = (V, E)$ and $E' \subseteq E$, we denote by $G[E']$ the spanning subgraph of G induced by E', that is, the graph (V, E').

Algorithm A

Starting with $E_1 = \emptyset$, repeatedly (as long as possible) find a triangle T whose vertices are in different components of $G[E_1]$, and add the edges of T to E_1.
Let $S_1 := G[E_1]$.
Starting with $E_2 = E_1$, repeatedly (as long as possible) find an edge e in G whose endpoints are in different components of $G[E_2]$, and add e to E_2.
Let $S_2 := G[E_2]$.
Output S_2.

Note that S_2 is indeed a triangular structure in G. As we mentioned before, S_2 is planar since it does not contain cycles of length greater than three.

THEOREM 2.1. *The performance ratio of algorithm A is* $\frac{7}{18}$.

Proof. First let us show that the performance ratio is at least 7/18. Without loss of generality, we may assume G is connected, and has at least three vertices. Observe that the number of edges in S_2 is the number of edges in a spanning tree of G plus the number of triangles in S_1. So it suffices to count the number of triangles in S_1.

Let H be a maximum planar spanning subgraph of G. Let $n \geq 3$ be the number of vertices in G, and $t \geq 0$ be such that $3n - 6 - t$ is the number of edges in H. We can think of t as the number of edges missing for an embedding of H to be a triangulated plane graph. The number of triangular faces in H is at least $2n - 4 - 2t$. (This is a lower bound on the number of triangular faces of a plane embedding of H since if H were triangulated, it would have $2n - 4$ triangular faces, and each missing edge can destroy at most two of these triangular faces.)

Let k be the number of components of S_1 each with at least one triangle, and let $p_1, p_2, \ldots, p_k$ be the number of triangles in each of these components. Let $p = \sum_{i=1}^{k} p_i$. We will prove that p, the number of triangles in S_1, is at least a constant fraction of $n-2-t$. Note that if a triangle cannot be added to S_1, it is because two of its vertices are in the same component of S_1. Hence, one of its edges has its two endpoints in the same component of S_1. This means that at the end of the first phase, every triangle in G must have some two vertices in the same component of S_1. In particular, every triangular face in H must have some two vertices in the same component of S_1, and therefore one of its three edges must be in the subgraph of H induced by the vertices in a component of S_1. Thus we can associate with each triangular face F in H an edge e in F whose endpoints are in the same component of S_1. But any edge e in H lies in at most two triangular faces of H, so e could have been chosen by at most two triangular faces of H. It follows that the number of triangular faces in H is at most twice the number of edges in H whose endpoints are in the same component of S_1.

Let H' be the subgraph of H induced by the edges of H whose endpoints are in the same component of S_1. Note that $p_i \geq 1$, for all i, and that the number of vertices in the i^{th} component of S_1 is $2p_i + 1 \geq 3$. Since H' is planar, H' has at most $\sum_{i=1}^{k}(3(2p_i + 1) - 6) = 6p - 3k$ edges. By the observation at the end of the previous paragraph, $2(6p - 3k) \geq 2|E(H')| \geq$ (number of triangular faces in H) $\geq 2n - 4 - 2t$. From this, we have

$$p \geq \frac{n - 2 - t + 3k}{6} \geq \frac{n - 2 - t}{6}.$$

Therefore the number of triangles in S_1 is at least $\frac{n-2-t}{6}$, and the ratio between the number of edges in S_2 and the number of edges in H is at least

$$\frac{n - 1 + \frac{n-2-t}{6}}{3n - 6 - t} = \frac{7n - 8 - t}{18n - 36 - 6t} \geq \frac{7}{18},$$

since $t \geq 0$. This completes the proof that the performance ratio of algorithm A is at least 7/18.

Now, we will prove that the performance ratio is at most 7/18. This is done by presenting, for any $\epsilon > 0$, a *planar* graph G_ϵ such that algorithm A, with G_ϵ as input, can produce a subgraph S_2 of G_ϵ such that the number of edges in S_2 is at most $\frac{7}{18} + \epsilon$ times the number of edges in G_ϵ.

Given $\epsilon > 0$, let p be an integer such that $p > \frac{6\epsilon+1}{12\epsilon}$. Let S be any connected triangular cactus with p triangles. S has $2p + 1 \geq 3$ vertices. Let S' be any triangulated plane supergraph of S on the same set of vertices (S' can be obtained from S by adding edges to S until it becomes triangulated). Since S' is triangulated, S' has $2(2p+1) - 4 = 4p - 2$ (triangular) faces. For each face of S', add a new vertex in the face and adjacent to all vertices in the boundary of that face. Let G_ϵ be the new graph. Observe that G_ϵ is a triangulated plane graph and has $(2p + 1) + (4p - 2) = 6p - 1$ vertices. This means that G_ϵ has $3(6p - 1) - 6 = 18p - 9$ edges. With G_ϵ as input for algorithm A, in the first phase it can produce $S_1 = S$, and S_2 can be S plus one edge for each of the new vertices (the vertices in G_ϵ not in S). The number of edges in S is $3p$. Hence, S_2 can have $3p + (4p - 2) = 7p - 2$ edges, while G_ϵ has $18p - 9$ edges. Thus, the ratio between the number of edges in S_2 and the number of edges in G_ϵ is

$$\frac{7p - 2}{18p - 9} < \frac{7}{18} + \epsilon,$$

because $p > \frac{6\epsilon+1}{12\epsilon}$. ■

2.1.1 Linear Time for Bounded-Degree Graphs.

In the case G has bounded degree d, we can implement algorithm A in linear time. We will only describe the implementation of the first phase, as the second one can clearly be implemented in linear time.

At any time, the vertices of the graph are partitioned in three sets: new, active and used. At the beginning, all the vertices are new. If there are no active vertices, choose a new vertex and make it active. Choose an active vertex x and "use" it; that is, include in the cactus S_1, one after the other, triangles formed by x and two new vertices, making these vertices active. Mark x "used" at the end of this process.

Using one vertex takes constant time as all degrees are bounded by d. We maintain the invariant that all triangles which contain a used vertex have been processed and all vertices which are active at a given time are in the same connected component of $G[E_1]$ at that time.

It is not hard to see that at the end, E_1 is maximal, in that no triangles can be added to it.

2.2 A Better Algorithm.

The new algorithm, algorithm B below, finds a *maximum* triangular structure (one with the maximum number of edges) in a given graph G. Algorithm B has performance ratio at least 0.4, and can be implemented in time $O(m^{3/2} n \log^6 n)$. Now, let us present the algorithm and the lower bound of 0.4 on its performance ratio.

Algorithm B also has two phases. In the first one, B constructs a maximum triangular cactus S_1 in G. We will show later how to use a matroid parity algorithm to construct S_1. In the second phase, B extends S_1 to a triangular structure S_2 in G, as before, by adding to S_1 as many edges as possible which do not form new cycles.

Algorithm B

Let S_1 be a *maximum* triangular cactus in G.

Starting with $E_2 = E(S_1)$, repeatedly (as long as possible) find an edge e in G whose endpoints are in different components of $G[E_2]$, and add e to E_2.

Let $S_2 := G[E_2]$.

Output S_2.

Observe that S_2 is a triangular structure in G, and therefore is planar. To analyze the algorithm, we need a definition. In any graph H, let $mts(H)$ denote the number of edges in a maximum triangular structure in H. Define $\rho(H) = mts(H)/|E(H)|$ if $E(H) \neq \emptyset$, and $\rho(H) = 1$ if $E(H) = \emptyset$.

We will prove that $\rho(H) \geq 0.4$ provided that H is planar. (And later we will prove that $\rho(H) \geq 2/3$ if H is outerplanar.) The key to understanding the analysis of algorithm B is the following. If G is any graph, let H be a maximum planar subgraph of G. Clearly $mts(G) \geq mts(H)$. Now $Opt(G) = |E(H)|$ implies that $\mathrm{B}(G)/Opt(G) = mts(G)/|E(H)| \geq mts(H)/|E(H)| = \rho(H)$. If we prove that $\rho(H) \geq 0.4$ for any planar H, we can infer that the performance ratio of B is at least 0.4.

Theorem 2.2. *If H is a planar graph, then* $\rho(H) \geq 0.4$.

Proof. The theorem is easily verified if H has fewer than three vertices, so let us assume that H has $n \geq 3$ vertices. We may furthermore assume that H is connected. Embed H in the plane. Choose $t \geq 0$ so that $|E(H)| = 3n - 6 - t$.

Now let J be any triangular cactus obtained by choosing triangular faces of H until no more can be added; say the final J has k components. Let p be the number of triangles in J. As in the proof of Theorem 2.1, if we count twice every edge in H whose endpoints are in the same component of J, we will "cover" every triangular face of H; and, in fact, each triangular face of J will be covered three times, by the three edges bounding the face. Let s be the number of edges in H whose endpoints are in the same component of J. Let l be the number of triangular faces in H. Since the p triangles in J are covered three times, we have $(l - p) + 3p = l + 2p \leq 2s$. As in Theorem 2.1, we have $s \leq 6p - 3k$ and $l \geq 2n - 4 - 2t$.

It follows that $2n - 4 - 2t + 2p \leq l + 2p \leq 2s \leq 2(6p - 3k)$, so that

$$p \geq \frac{2n - 4 - 2t + 6k}{10} = \frac{n - 2 - t + 3k}{5} \geq \frac{n - 2 - t}{5}.$$

Since $\rho(H) = \frac{mts(H)}{|E(H)|}$, using $mts(H) \geq (n - 1) + p$, we have

$$\rho(H) \geq \frac{n - 1 + \frac{n-2-t}{5}}{3n - 6 - t} = \frac{6n - 7 - t}{15n - 30 - 5t} \geq \frac{2}{5},$$

for any $t \geq 0$. ■

Corollary 2.1. *The performance ratio of algorithm B is at least* 0.4.

The next theorem gives an upper bound on the performance ratio of algorithm B.

Theorem 2.3. *The performance ratio of algorithm B is at most $\frac{4}{9}$.*

Proof. We will prove this by presenting, for any $\epsilon > 0$, a *planar* graph G_ϵ such that algorithm B, with G_ϵ as input, can produce a subgraph S_2 of G_ϵ whose number of edges is at most $\frac{4}{9} + \epsilon$ times the number of edges of G_ϵ.

Given $\epsilon > 0$, let n' be an integer such that $n' > \frac{6\epsilon+1}{3\epsilon}$ and $n' \geq 3$. Let G'_ϵ be any triangulated plane graph on n' vertices. Call V' the vertex set of G'_ϵ. Since G'_ϵ is triangulated, G'_ϵ has $2n' - 4$ (triangular) faces. For each face of G'_ϵ, add a new vertex in the face and adjacent to all three vertices on the boundary of that face. Let G_ϵ be the new graph, and let V be the vertex set of G_ϵ. Observe that G_ϵ is a triangulated plane graph, and has $n' + (2n' - 4) = 3n' - 4$ vertices. Therefore, G_ϵ has $3(3n' - 4) - 6 = 9n' - 18$ edges. Let S be a maximum triangular structure in G_ϵ.

Any edge in G_ϵ has at least one endpoint in V'. Moreover, $|V'| = n'$. Therefore, a maximum matching in G_ϵ has at most n' edges (each with at least one distinct endpoint in V'). The following lemma is observed in [LP86, p. 440].

Lemma 2.1. *If S is a triangular structure with t triangles in a given graph G, then there is a matching in G of size t.*

Using the lemma above, we conclude that S has at most n' triangles. Recall that S, being a triangular structure, is a spanning tree of G_ϵ plus one edge per triangle in S, which implies that S has at most $(3n' - 5) + n' = 4n' - 5$ edges. Furthermore, G_ϵ has $9n' - 18$ edges. Therefore the ratio between the number of edges in S and the number of edges in G_ϵ is

$$\frac{4n' - 5}{9n' - 18} < \frac{4}{9} + \epsilon,$$

because $n' > \frac{6\epsilon+1}{3\epsilon}$. ■

How can one find a maximum triangular cactus quickly? A graphic matroid parity algorithm can be used to construct a maximum triangular cactus in a given graph [LP86]. The problem solved by a graphic matroid parity algorithm is GRAPHIC MATROID PARITY (GMP): given a multigraph $H = (V_H, E_H)$ and a partition of the edge set E_H into pairs of distinct edges $\{f, f'\}$, find a (simple) forest F with the maximum number of edges, such that $f \in F$ if and only if $f' \in F$, for all $f \in E_H$.

Let us show how to reduce the problem of finding a maximum triangular cactus in a given graph $G = (V, E)$ to GMP. This is done by describing a multigraph $G' = (V', E')$ and a partition $\mathcal{P}$ of E' into pairs of distinct edges of E', such that, from a solution to GMP for G'

and $\mathcal{P}$, we can construct a maximum triangular cactus in G.

First let $V' = V$. Now, let us describe E' and the partition $\mathcal{P}$. Initially, $E' = \emptyset$ and $\mathcal{P} = \emptyset$. For each triangle in G with edge set T, let $\{e, e'\}$ be any pair of distinct edges in T. Add two new edges f and f' to E', f with the same endpoints as e, and f' with the same endpoints as e'. We say that T *corresponds to* $\{f, f'\}$. Insert f and f' into $\mathcal{P}$.

We say a forest F in G' is *valid* if $f \in F$ if and only if $f' \in F$, for all f in E'. Observe that any valid forest has an even number of edges. The following lemma states a relation between valid forests in G' and triangular cacti in G. Let m and n be the number of edges and vertices, respectively, in G.

LEMMA 2.2. *There is a valid forest F in G' with $2p$ edges if and only if there is a triangular cactus S in G with p triangles. Moreover, S can be obtained from F (and* vice versa*) in time $O(n)$.*

For lack of space, we omit the proof of this lemma, which is used implicitly in [LP86].

As described by Chiba and Nishizeki [CN85], we can explicitly list all the triangles in a graph G with m edges in time $O(m^{3/2})$. So $|E'|$ is $O(m^{3/2})$.

Gabow and Stallmann [GS85] describe an algorithm for GMP, which runs in time $O(m'n' \log^6 n')$, where m' and n' are the number of edges and vertices, respectively, in the input graph. In our case, $n' = n$ and $m' = |E'|$, which is $O(m^{3/2})$. This gives a time bound of $O(m^{3/2} n \log^6 n)$ for this phase.

From the output of the Gabow-Stallmann algorithm, it is easy to find a maximum triangular cactus in time $O(n)$ (Lemma 2.2). Therefore the total time is $O(m^{3/2} n \log^6 n)$.

3 Outerplanar Subgraphs

Serendipitously, Algorithm B produces outerplanar graphs, so it is an approximation algorithm for MAXIMUM OUTERPLANAR SUBGRAPH, which is NP-Complete [GJ79, p. 197]. In fact, any algorithm which produces a spanning tree has performance ratio at least 1/2, because any outerplanar graph on $n \geq 2$ vertices has at most $2n-3$ edges (see below). A careful analysis shows that the performance ratio of B when used for MAXIMUM OUTERPLANAR SUBGRAPH is at least 2/3. This is an easy consequence of Theorem 3.1, in order to prove which we need some preliminaries.

An outerplanar graph G is a *maximal outerplanar graph* if no edge can be added without losing outerplanarity. As mentioned in [H72, p. 106], every maximal outerplanar graph G with at least three vertices is a triangulation of a polygon (i.e., the boundary of the exterior face is a Hamiltonian cycle and each interior face is triangular). By [H72, Cor. 11.9], G must have a vertex of degree two and $2|V(G)| - 3$ edges (this last statement is also true for $|V(G)| = 2$).

LEMMA 3.1. *Let H be a maximal outerplanar graph. If H has an odd number $n = 2p+1$ of vertices, then there is a triangular cactus in H with p triangles. If H has an even number $n = 2p$ of vertices and xy is an edge on the boundary of the exterior face, then there is a triangular cactus S in H with $p-1$ triangles such that x and y are not connected in S.*

Notice that we obtain the maximum number of triangles possible. In the former case all vertices are in the same component of the cactus, while in the latter, the cactus has two components.

Proof. We use a plane embedding of H.

The proof is by induction on n, the number of vertices of H. The case $n = 1$ is trivial. If $n = 2$ (in this case there is only one edge and $p = 1$), the theorem is true.

We inductively construct a triangular cactus of the given size.

Let $n = 2p+1$. Let v be a vertex of degree two. Let x and y be its neighbors. They are adjacent, since interior faces are triangles. The graph $H - \{v\}$ is maximal outerplanar (since it has $(2n-3) - 2 = 2(n-1) - 3$ edges) and has an even number of vertices. It is easy to check that if a triangular cactus S' in this smaller graph has the property that x and y are not connected in S', we can add the triangle xyv to get a triangular cactus in H. The size of this cactus is $p-1$, by induction, plus one, for a total of p.

Let $n = 2p$ and let the edge xy be on the boundary of the exterior face. This edge is on the boundary of a triangular face xyv on the inside. Walking along the Hamiltonian cycle which is the boundary of the exterior face, starting at v and in the direction that visits x just before y, let D_1 be the set of vertices visited between v and x, and let $n_1 = |D_1|$. Walking along the Hamiltonian cycle in the opposite direction again starting at v, let D_2 be the set of vertices visited between v and y, and let $n_2 = |D_2|$; $D_1 \cap D_2 = \{v\}$ and $D_1 \cup D_2 = V(H)$. The only edge in H between $D_1 - \{v\}$ and $D_2 - \{v\}$ is the edge xy.

Let H_1 be the subgraph of H induced by vertex set D_1, with, say, e_1 edges, and let H_2 be the subgraph of H induced by vertex set D_2, with, say, e_2 edges.

We have $n_1 + n_2$ is odd, since v is counted twice. Let us say without loss of generality that $n_1 = 2p_1 + 1$ is odd and $n_2 = 2p_2$ is even. Then $n = 2(p_1 + p_2)$. We have $e_1 + e_2 = (2n-3) - 1$, as from H only the edge xy is not an edge of either H_1 or H_2. Since $e_1 \leq 2n_1 - 3$ and $e_2 \leq 2n_2 - 3$, we infer that $e_1 + e_2 \leq 2(n_1 + n_2) - 6 = 2(n+1) - 6 = 2n - 4$. Since, in fact, $e_1 + e_2 = 2n - 4$, we infer that $e_1 = 2n_1 - 3$ and $e_2 = 2n_2 - 3$. Thus both H_1 and H_2 have to be maximal outerplanar, as they have the maximum number of edges.

Then by the inductive hypothesis we can construct in H_1 a cactus S_1 with p_1 triangles. If we apply the inductive hypothesis to H_2 with vy being the edge on the exterior face, we obtain a triangular cactus S_2 with $p_2 - 1$ triangles in which y and v are not connected. Then putting together the edges of S_1 and S_2 we get S, a cactus in H. In the new cactus S, any possible $x-y$-path must visit v, since neither S_1 nor S_2 has edge xy. But in S_2, y and v are not connected. It follows that x and y are not connected in S, so S is the desired cactus. S has $p_1 + (p_2 - 1)$ triangles, which is exactly the number we wanted. ■

In conclusion, for a maximal outerplanar graph with n vertices, we can find a triangular structure with $\lfloor \frac{n-1}{2} \rfloor$ triangles.

Now we prove a lower bound on $\rho(H)$.

THEOREM 3.1. *If H is outerplanar, then $\rho(H) \geq 2/3$.*

Proof. Let H be any 2-connected outerplanar graph. We add t edges to obtain a maximal outerplanar plane graph H'. Note that H' has $2n-3$ edges and a triangular structure S with at least $\lfloor \frac{n-1}{2} \rfloor$ triangles.

However, the t missing edges can destroy at most t of these triangles in S, because S is a cactus. If $t \geq \frac{n}{2}$, we infer that

$$\rho(H) \geq \frac{n-1}{2n-3-n/2} \geq \frac{2}{3}.$$

Assume to the contrary that $t \leq \lfloor \frac{n-1}{2} \rfloor$. Then the number of edges in the triangular structure is at least $n - 1 + (\lfloor \frac{n-1}{2} \rfloor - t)$. Then

$$\rho(H) \geq \frac{n-1+\lfloor \frac{n-1}{2} \rfloor - t}{2n-3-t}.$$

The worst case is achieved when $t = \lfloor \frac{n-1}{2} \rfloor$ and is $\frac{2}{3}$.

If H is not 2-connected, we can do the above analysis for each of the 2-connected components (an edge appears in exactly one 2-connected component) and infer that a maximum triangular structure has $\frac{2}{3}$ of the edges in H. ■

The theorem above is tight, in the sense that there are outerplanar graphs H for which $\rho(H)$ is arbitrarily close to 2/3. In fact, there are outerplanar graphs H_i with $2i$ vertices and $3i - 2$ edges which do not have any triangle.

COROLLARY 3.1. *Algorithm B has performance ratio 2/3 for MAXIMUM OUTERPLANAR SUBGRAPH.*

4 The Complexity of the Problems

Papadimitriou and Yannakakis [PY91] defined a natural variant of NP for optimization problems: the complexity class Max SNP. This class, as they have shown, contains several well-known optimization problems, such as MAX 3-SAT and MAXIMUM CUT. In this section, we prove that MAXIMUM PLANAR SUBGRAPH (MPS) is Max SNP-hard, as is its complementary version: given a graph, find a smallest subset of its edges whose removal results in a planar graph. This means, by results of Arora et al. [ALMSS92], that there is a constant $\epsilon > 0$ such that the existence of a polynomial-time approximation algorithm for MPS with performance ratio at least $1-\epsilon$ implies that $P = NP$, and that an analogous statement can be made about the complementary problem.

As in [PY91], we use the concept of *L-reduction*, which is a special kind of reduction that preserves approximability. Let A and B be two optimization problems. We say *A L-reduces to B* if there are two polynomial-time algorithms f and g, and positive constants α and β, such that for each instance I of A,

1. Algorithm f produces an instance $I' = f(I)$ of B, such that the optima of I and I', of costs denoted $Opt_A(I)$ and $Opt_B(I')$ respectively, satisfy $Opt_B(I') \leq \alpha \cdot Opt_A(I)$, and

2. Given any feasible solution of I' with cost c', algorithm g produces a solution of I with cost c such that $|c - Opt_A(I)| \leq \beta \cdot |c' - Opt_B(I')|$.

The main result of this section is

THEOREM 4.1. *MAXIMUM PLANAR SUBGRAPH is Max SNP-hard.*

Proof. Denote by $TSP_4(1,2)$ the following variant of the traveling salesman problem: given a complete graph, a pair of distinct vertices x, y, and costs one or two for each edge, such that the graph induced by the edges of cost one has maximum degree at most four, find a Hamiltonian path from x to y of minimum cost. Papadimitriou and Yannakakis [PY93] showed that $TSP_4(1,2)$ is Max SNP-hard.

We shall prove $TSP_4(1,2)$ L-reduces to MPS. The basic idea of the reduction comes from Liu and Geldmacher [LG77], where the decision version of MPS is proved to be NP-complete.

The first part of the L-reduction is the polynomial-time algorithm f and the constant α. Given any instance I of $TSP_4(1,2)$, f produces an instance G of MPS such that the cost of the optimum of G in MPS, denoted $Opt_{MPS}(G)$, is at most α times the cost of the optimum of I in $TSP_4(1,2)$, denoted by $Opt_{TSP_4(1,2)}(I)$, i.e., $Opt_{MPS}(G) \leq \alpha \cdot Opt_{TSP_4(1,2)}(I)$.

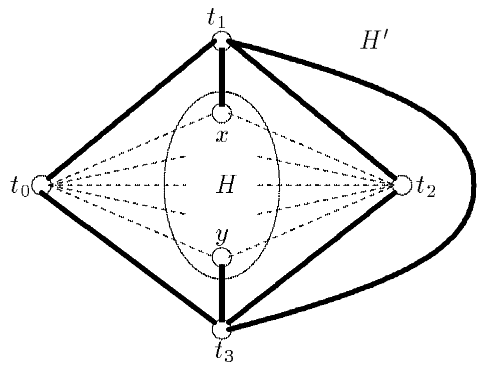

Figure 1: Graph H' constructed from H.

Consider an instance I of $TSP_4(1,2)$. I is a complete graph $K = (V, E)$, a pair of distinct vertices x, y of V and a subset E_1 of E consisting of the edges of cost one. Let $H = (V, E_1)$ and $H' = (V \cup T, E_1 \cup F_1 \cup F_2)$, where $T = \{t_0, t_1, t_2, t_3\}$, $T \cap V = \emptyset$, $F_1 = \{t_0t_1, t_0t_3, t_1t_2, t_1t_3, t_1x, t_2t_3, t_3y\}$ and $F_2 = \cup_{z \in V}\{t_0z, t_2z\}$ (see figure 1).

Denoting by n the number of vertices of H, let G be the graph obtained from H' by (1) replacing each edge e in F_1 by $2n$ parallel internally-disjoint paths of length two (having new internal vertices) between the endpoints of e and (2) replacing each edge e in F_2 by eight parallel internally-disjoint paths of length two (having new internal vertices) between the endpoints of e.

Clearly G can be obtained from I in time polynomial in the size of I.

LEMMA 4.1. $Opt_{MPS}(G) \leq 124 \cdot Opt_{TSP_4(1,2)}(I)$.

Proof. Observe that $Opt_{TSP_4(1,2)}(I) \geq n - 1$. A clear upper bound for $Opt_{MPS}(G)$ is the number of edges of G. To compute this, note first that H has maximum degree at most four by the definition of $TSP_4(1,2)$, and so H has at most $2n$ edges. Let us call the edges in F_1 *$2n$-edges* and the edges in F_2 *8-edges*. There are seven $2n$-edges: F_1 contains seven edges, each of them corresponding to $4n$ edges in G. There are $2n$ 8-edges: F_2 contains $2n$ edges, each of them corresponding to 16 edges in G. Hence the number of edges in G outside of H is $7 \cdot 4n + 16 \cdot 2n = 60n$. The total number of edges in G is therefore at most $2n + 60n = 62n \leq 124(n-1)$. Therefore, $Opt_{MPS}(G) \leq 124(n-1) \leq 124 \cdot Opt_{TSP_4(1,2)}(I)$. ■

This finishes the first part of the L-reduction, since we can take $\alpha = 124$.

The second and hard part of the L-reduction is the constant β and the algorithm g. Given a planar subgraph of G with m edges, g produces in polynomial time a Hamiltonian path from x to y of cost t in K such that $|t - Opt_{TSP_4(1,2)}(I)| \leq \beta|m - Opt_{MPS}(G)|$. We shall see that $\beta = 1$ suffices.

First, given a planar subgraph P of G, let us describe another planar subgraph P' of G with at least as many edges as P. Moreover, P' shall contain all edges of G not in H.

Let e be a $2n$-edge or an 8-edge of H'. We say e *appears in* P if P contains both edges of *all* the paths of length two corresponding to e. In this case, we also say that the endpoints of e *are adjacent in* P *by the* $2n$*-edge* or *8-edge* e. We say e *is missing in* P if P contains both edges of *none* of the paths (of length two) corresponding to e. (In this case, if e is an 8-edge, then P is missing at least eight of its 16 edges in G corresponding to e.) It is possible that a $2n$-edge or an 8-edge of H' neither appears in P nor is missing in P.

Let us modify P so that any $2n$-edge or 8-edge of H' either appears in P or is missing in P. This is done as follows. If a $2n$-edge or an 8-edge e of H' neither appears nor is missing in P, then we insert in P all edges of G in the paths (of length two) corresponding to e. Note that P remains planar. Clearly, the number of edges in P cannot decrease by this operation. The new graph is also called P.

Now we can describe P'. We have three cases: (1) if some $2n$-edge e does not appear in P, then define P' to be the graph induced by all edges of G not in H; (2) if all the $2n$-edges and the 8-edges of H' appear in P, then let P' be the same as P; and (3) if all the $2n$-edges of H' appear in P, but not all the 8-edges, then we modify P to obtain P', as described in the next two paragraphs.

The idea is to remove from P some edges of H and add to P edges of H' not in H so that all the 8-edges of H' appear in the modified graph, and it remains planar and has at least as many edges as the original P.

Let U be the set of vertices v of H such that at least one of the two 8-edges incident to v in H' is missing in P. Observe that $|U| \geq 1$, as case (2) considered $|U| = 0$. For each vertex v in U, remove from P all edges of H incident to v in P (at most four edges are removed per vertex) and add to P all the edges outside of H so that the two 8-edges incident to v appear in P (at least eight edges are added, corresponding to the 8-edges incident to v missing in P). To guarantee that the graph obtained this way is planar, we must make room to embed the modified 8-edges. This is done by also removing from P all edges of H incident to y (if they were not already removed). Let P' be the graph

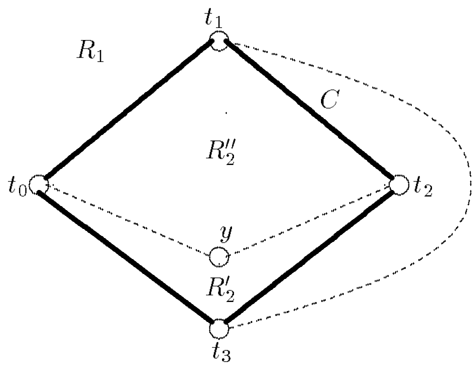

Figure 2: Cycle C, regions R_1 and $R_2 = R_2' \cup R_2''$.

obtained after all these modifications.

LEMMA 4.2. *P' is planar and has at least as many edges as P.*

Proof. In case (1), we include in P' at least $2n$ edges that do not appear in P (at least one in each of the $2n$ paths corresponding to e), and we remove at most $2n$ edges, the maximum number of edges in H. So P' has at least as many edges as P. Moreover P' is planar.

There is nothing to be proved in case (2).

Case (3) is the complicated one. First note that P' has at least as many edges as G, since, for each vertex in U, we remove at most four edges and add at least eight. Furthermore, we remove at most four edges incident to y. Hence, we gain at least $(8-4)|U|-4 = 4|U|-4 \geq 0$ edges, since $|U| \geq 1$.

Now, let us show that P' is planar. We can think of the $2n$-edges and 8-edges as single edges, as they are in H' (since if we can embed a single edge, we can embed a $2n$-edge or an 8-edge as well). We will modify a given embedding of P into an embedding for P'.

Let C be the cycle (using four $2n$-edges) t_0, t_1, t_2, t_3, t_0. Observe that the $2n$-edges in C appear in P, since we are in case (3). Given an embedding for P, cycle C divides the plane into two regions, R_1, containing the $2n$-edge t_1t_3, and R_2 (see figure 2). The $2n$-edge t_1t_3 separates t_0 from t_2 in R_1. Moreover, each vertex in $V-U$ (the vertices of H not in U) is adjacent in P by 8-edges to t_0 and t_2. Because t_0 and t_2 are separated in R_1, none of these vertices can be embedded in R_1, which implies they must be embedded in R_2. Keep these vertices (t_0, t_1, t_2, t_3 and the vertices in $V-U$) embedded as they are.

Now, observe that y is adjacent in P' only to t_0, t_2 (by 8-edges) and t_3 (by a $2n$-edge). Furthermore, y is the only vertex in H which is adjacent to t_3. This means, before we embed y, the vertices t_0, t_2 and t_3 are not separated in R_2. Therefore, y can be embedded in R_2 with the $2n$-edge t_3y and the 8-edges t_0y and yt_2 "next to" the $2n$-edges t_0t_3 and t_3t_2. The edges t_0y and yt_2 together split region R_2 into two regions R_2' containing edge t_3y, and R_2'' containing vertex t_1. Observe that t_0 and t_2 are not separated in R_2'', since t_3 is the only vertex, besides t_0 and t_2, which is adjacent to y (by a $2n$-edge).

All the vertices in $U-\{x\}$ are adjacent in P' only to t_0 and t_2 (by 8-edges), and therefore they all can be embedded in R_2'' with their two 8-edges "next to" the 8-edges t_0y and yt_2. If $x \in U$ then observe that x is the only vertex in H which is adjacent in P' to t_1 (by a $2n$-edge). This means, before we embed x, the vertices t_0, t_1 and t_2 are not separated in R_2''. Therefore, x can be embedded in R_2'' with the $2n$-edge xt_1, and the 8-edges t_0x and xt_2 "next to" the $2n$-edges t_0t_1 and t_1t_2. If $x \notin U$ then it does not need to be moved in the embedding. The embedding obtained this way is a plane embedding of P', completing the proof that P' is planar. ■

Observe that P' contains all the edges of G not in H. Let F be the set of edges of H appearing in P'.

LEMMA 4.3. *The graph $G_F = (V, F)$ is a collection of vertex-disjoint paths which can be extended in K (the complete graph on V) to a Hamiltonian path from x to y, in polynomial time.*

Proof. Let us prove that G_F satisfies the following four conditions: (1) There is no vertex of degree greater than two in G_F. (2) Vertices x and y have degree at most one in G_F. (3) There is no cycle in G_F. (4) If x and y are in the same component of G_F, then this component spans all vertices in V. We will prove each of these conditions holds by contradiction.

Suppose (1) does not hold. Let z_0 be a vertex in V of degree at least 3 in G_F. Let z_1, z_2, z_3 be three of its neighbors in G_F. (Notice that z_0, z_1, z_2, z_3 are distinct vertices of H, so they are distinct of t_0, t_2.) Then each one of t_0, t_2, z_0 is adjacent in P' to each one of z_1, z_2, z_3 (some of them are adjacent in P' by 8-edges). Therefore, $t_0, t_2, z_0; z_1, z_2, z_3$ define a subdivision of $K_{3,3}$ in P', a contradiction, because P' is planar. Thus, (1) holds.

Suppose (2) does not hold. If x has degree more than one in G_F, let z_1 and z_2 be two of its neighbors in G_F. (Notice that z_1 and z_2 are distinct vertices of H, distinct of x, so they are distinct of t_0, t_2.) Then each one of t_0, t_2, x is adjacent in P' to each one of t_1, z_1, z_2 (some of them are adjacent in P' by $2n$-edges or 8-edges). Therefore, $t_0, t_2, x; t_1, z_1, z_2$ define a subdivision of $K_{3,3}$ in P', a contradiction, because P' is planar. Analogously, we have a contradiction if y has degree more than one in G_F. Thus, (2) holds.

Suppose (3) does not hold. Let z_1, z_2, z_3 be three vertices in a cycle of G_F. (Since z_1, z_2, z_3 must have degree at least two, they are not x or y by condition (2), and they are not t_0 or t_2 since they are vertices of H.) Then z_1, z_2, z_3, t_0, t_2 are pairwise linked by internally vertex-disjoint paths (the path between t_0 and t_2 uses the two 8-edges incident to x, while the others use one $2n$-edge or 8-edge). Therefore, t_0, t_2, z_1, z_2, z_3 define a subdivision of K_5 in P', a contradiction, because P' is planar. Hence, (3) holds.

Suppose (4) does not hold. Let z_0 be a vertex in V which is not in the component having x and y in G_F. In this case, t_0, t_1, t_2, x, y are pairwise linked by internally vertex-disjoint paths (the path between t_0 and t_2 uses z_0, the path between t_1 and y uses t_3, the path between x and y is in G_F, and the others use one $2n$-edge or 8-edge). Therefore, t_0, t_1, t_2, x, y define a subdivision of K_5 in P', a contradiction, because P' is planar. Hence, (4) holds.

Therefore, the conditions hold. From (1) and (2), we conclude that G_F is a collection of paths. From (2) and (3), these paths can be extended in K (the complete graph on V) to a Hamiltonian path from x to y. Furthermore, note that this can be done in polynomial time. ∎

Let HP be a Hamiltonian path from x to y containing all edges in F and some edges (in K) of cost two. HP exists by Lemma 4.3. Denote by m' the number of edges of P' and by t the cost of HP.

Now the following Lemma states that β exists, and specifically, $\beta = 1$. This will complete the proof of the theorem.

Lemma 4.4.

$$t - Opt_{TSP_4(1,2)}(I) = Opt_{MPS}(G) - m',$$

and hence

$$|t - Opt_{TSP_4(1,2)}(I)| = |m' - Opt_{MPS}(G)|.$$

Proof. As in the proof of Lemma 4.1, the number of edges in G outside of H is $60n$. All these edges are in P'. Therefore, the number of edges in F is $m' - 60n$. And the cost of HP is

$$t = 2(n-1) - (m' - 60n). \tag{4.1}$$

Let Q be an optimal solution of MPS for G. Using the same argument for Q that we used for P, Lemma 4.3 and the argument above imply the existence of a Hamiltonian path of cost $2(n-1) - (Opt_{MPS}(G) - 60n)$. Therefore

$$Opt_{TSP_4(1,2)}(I) \le 2(n-1) - (Opt_{MPS}(G) - 60n). \tag{4.2}$$

Given an optimum solution HP^* of $TSP_4(1,2)$ for I, we can construct a solution of MPS for G by selecting all the edges outside of H plus the edges of cost one in HP^*. Observe that these edges really determine a planar subgraph of G. Let z be the number of these edges. Since we have $60n$ edges in G outside of H, HP^* has $z - 60n$ edges of cost one and the remaining ones (of its $n-1$ edges) have cost two. This means that

$$\begin{aligned} Opt_{TSP_4(1,2)}(I) &= 2(n-1) - (z - 60n) \\ &\ge 2(n-1) - (Opt_{MPS}(G) - 60n), \end{aligned}$$

since $Opt_{MPS}(G) \ge z$.

Therefore, from (4.2) and (4.3), we have

$$Opt_{TSP_4(1,2)}(I) = 2(n-1) - (Opt_{MPS}(G) - 60n).$$

And this together with (4.1) means

$$t - Opt_{TSP_4(1,2)}(I) = Opt_{MPS}(G) - m'.$$

Hence $|t - Opt_{TSP_4(1,2)}(I)| = |m' - Opt_{MPS}(G)|$. ∎

From $m \le m'$, it follows that

$$t - Opt_{TSP_4(1,2)}(I) \le Opt_{MPS}(G) - m$$

and

$$|t - Opt_{TSP_4(1,2)}(I)| \le 1 \cdot |m - Opt_{MPS}(G)|.$$

This completes the proof of Theorem 4.1. ∎

Let us denote the complementary version of MPS by NPD: given a graph G, find a smallest set of edges of G whose removal results in a planar graph.

A slight modification of the L-reduction presented above proves the following.

Theorem 4.2. *NPD is Max SNP-hard.*

Proof. The first part of the L-reduction is almost the same. From an instance I of $TSP_4(1,2)$, we construct G in exactly the same way. As before, $Opt_{TSP_4(1,2)}(I) \ge n - 1$. As in the proof of Lemma 4.1, the maximum number of edges of G is $62n$. Thus, the optimum of G in NPD, denoted as $Opt_{NPD}(G)$, is at most $62n$. And then $Opt_{NPD}(G) \le 62n \le 124(n-1) \le 124 \cdot Opt_{TSP_4(1,2)}(I)$. We can take $\alpha = 124$, as before.

In the second part, given an instance I of $TSP_4(1,2)$, let G be constructed from I as in the previous reduction. Let D be a subset of the edges of G whose removal results in a planar subgraph of G. We shall find in polynomial time a Hamiltonian path HP such that $t - Opt_{TSP_4(1,2)}(I) \le d - Opt_{NPD}(G)$, where t is the number of edges in HP and $d = |D|$. Just as we took a planar subgraph P of G and found a planar subgraph P' which contains all edges of G not in H, and is at least as large as P, from D we can find a set D' of edges of G containing *none* of the edges of G not in H, which is at least as *small* as D. Applying Lemma 4.3, we can obtain a Hamiltonian path HP, as before, in polynomial time.

Now, let us prove that β exists. Let m' be the number of edges in P'. Note that $d' + m' = |E(G)|$. Moreover, $Opt_{MPS}(G) + Opt_{NPD}(G) = |E(G)|$. Therefore, $d' - Opt_{NPD}(G) = Opt_{MPS}(G) - m'$. Applying Lemma 4.4, we conclude that $t - Opt_{TSP_4(1,2)}(I) = d' - Opt_{NPD}(G)$, which, together with $d' \leq d$, implies that we can take $\beta = 1$. ■

5 Open Problems

Many open problems are suggested by this research. How large a performance ratio can one achieve is an obvious one. Is there a linear-time approximation algorithm for MAXIMUM PLANAR SUBGRAPH with performance ratio $1/3 + \epsilon$? (A *maximal* planar subgraph can be found in linear time [H95, D95].) Is there any approximation algorithm with a constant performance ratio for NPD? Can one achieve a performance ratio of $1/3 + \epsilon$ for MAXIMUM WEIGHT PLANAR SUBGRAPH, which is this problem: given a weighted graph, find a planar subgraph of maximum weight. For this problem, any maximum weight spanning tree can be shown to have weight at least one third of the optimum. What performance ratios are achievable for finding heavy outerplanar subgraphs? What performance ratio can be achieved for THICKNESS (given G, partition the edges of G into as few planar subgraphs as possible)? A factor of 3 here is trivial, via arboricity.

6 Acknowledgments

The authors are very grateful to David Williamson for his help. We thank Laci Lovász for informing us that a maximum triangular structure can be found in polynomial time. We thank Robert Cimikowski for kindly sending us drafts of his papers. The third author would like to thank Kurt Mehlhorn, for suggesting the problem and supporting his work. The first author would like to thank Robin Thomas for helpful discussions. We also would like to thank the program committee for informing us about a faster way to find all triangles in a graph.

References

[ALMSS92] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, "Proof Verification and Hardness of Approximation Problems," *Proc. 33rd IEEE Symposium on Foundations of Computer Science*, 14–23, 1992.

[CHT93] J. Cai, X. Han and R. E. Tarjan, "An $O(m \log n)$-time Algorithm for the Maximal Planar Subgraph Problem," *SIAM Journal on Computing*, 22:1142–1162, 1993.

[Cim95] R. Cimikowski. "An Analysis of Some Heuristics for the Maximum Planar Subgraph Problem," *Proc. 6th Annual ACM-SIAM Symp. on Discrete Algorithms*, 322–331, 1995.

[CNS79] T. Chiba, I. Nishioka and I. Shirakawa. "An Algorithm of Maximal Planarization of Graphs," *Proc. IEEE Symp. on Circuits and Systems*, 649–652, 1979.

[CN85] N. Chiba and T. Nishizeki, "Arboricity and Subgraph Listing Algorithms," *SIAM Journal of Computing*, 14:210-223, 1985.

[D95] H. N. Djidjev, "A Linear Algorithm for the Maximal Planar Subgraph Problem," *Proc. 4th International Workshop on Algorithms and Data Structures* (WADS '95), 369-380, 1995.

[DFF85] M. E. Dyer, L. R. Foulds and A. M. Frieze, "Analysis of Heuristics for Finding a Maximum Weight Planar Subgraph," *European Journal of Operational Research*, 20:102–114, 1985.

[F92] L. R. Foulds, *Graph Theory Applications*, Springer-Verlag, New York, 1992.

[GS85] H. N. Gabow and M. Stallmann, "Efficient Algorithms for Graphic Matroid Intersection and Parity," *Automata, Language and Programming: 12th Colloq., Lecture Notes in Computer Science*, vol. 194, 210–220, 1985.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Co., 1979.

[H72] F. Harary, *Graph Theory*, Addison-Wesley Publishing Company, 1972.

[H95] W. Hsu, "A Linear Time Algorithm for Finding Maximal Planar Subgraphs," *Proc. 6th Annual International Symposium on Algorithms and Computation* (ISAAC95), 1995.

[JM93] M. Jünger and P. Mutzel, "Maximum Planar Subgraph and Nice Embeddings: Practical Layout Tools," *Universität zu Köln, Report No. 93.145*, 1993, to appear in *Special Issue of Algorithmica on Graph Drawing.*

[LG77] P. C. Liu and R. C. Geldmacher, "On the Deletion of Nonplanar Edges of a Graph," *Proc. 10th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, 727–738, 1977.

[LP86] L. Lovász and M. D. Plummer, *Matching Theory*, Elsevier Science, Amsterdam, 1986.

[PY91] C. H. Papadimitriou and M. Yannakakis. "Optimization, Approximation, and Complexity Classes," *Journal of Computer and System Sciences*, 43:425–440, 1991.

[PY93] C. H. Papadimitriou and M. Yannakakis, "The Traveling Salesman Problem with Distances One and Two," *Mathematics of Operations Research*, 18(1):1–11, 1993.

[TDB88] R. Tamassia, G. Di Battista and C. Batini, "Automatic Graph Drawing and Readability of Diagrams," *IEEE Transactions on Systems, Man and Cybernetics*, 18:61-79, 1988.