Improved Approximations of Maximum Planar Subgraph

Alexander Zelikovsky*

Abstract

The maximum planar subgraph problem (MPSP) asks for a planar subgraph of a given graph with the maximum total cost. We suggest a new approximation algorithm for the weighted MPSP. We show that it has performance ratio of 5/12 in the case of graphs where the maxumum cost of an edge is at most twice the minimum cost of an edge. For the variant of MPSP that asks for an outerplanar graph, the algorithm suggested is the first with the higher performance ratio (7/12 instead of 1/2).

Keywords: Combinatorial problems, approximation algorithms, planar subgraph.

1 Introduction

Let $G = (V, E)$ be a graph with a nonnegative cost function on edges $c : E \rightarrow R^+$. The maximum planar subgraph problem (MPSP) asks for a planar subgraph of G with the maximum total cost. This problem has applications in circuit layout, facility layout, and graph drawings [8, 11]. So many approximation algorithms appear in the last decade [7, 5, 3, 10].

The quality of an approximation algorithm is measured by its performance ratio: a lower bound on the ratio between achieved cost and optimal cost. This problem is MAX SNP-hard [4]. This means that there is a constant $\epsilon > 0$ such that unless $P = NP$, there is no polynomial-time approximation algorithm for MPSP with the performance ratio $1 - \epsilon$.

A simple heuristic for MPSP finds the maximum cost spanning tree of G and then tries to add some edges while it is possible to keep the graph planar. Unfortunately, it was shown in [7] that this heuristic and minimum spanning tree based heuristics cannot guarantee that the output graph costs more than one third of the optimal one.

Better heuristics for the unweighted MPSP have been recently appeared in [4]. They found the best and approximately best 3-block trees. A graph is a *3-block tree* if any its 2-connected component (*block*) has at most three vertices. It was proved that performance ratios of these heuristics are at least $\frac{7}{18}$ and $\frac{2}{5}$, respectively.

Here we suggest several approximation algorithms for the weighted MPSP. These algorithms are applications of heuristics first suggested for the Steiner tree problem [12, 1, 2, 13]. The greedy algorithm approximates the best weighted 3-block tree in the given graph G (Section 2). In Section 3 we estimate the quality of the greedy algorithm. We prove that it

*Computer Science Department, University of Virginia. E-mail: alexz@fir.cs.virginia.edu

Greedy Algorithm

$M \leftarrow \emptyset$; $G_0 \leftarrow G$

repeat forever

find a triangle $\tau \in G_0$, with the maximum gain,
$\tau \leftarrow arg \max g(\tau)$

if $g(\tau) \leq 0$ then exit repeat

$M \leftarrow M \cup \tau$

stretch τ ($G \leftarrow G \cup N(\tau)$)

output $A = M \cup (MST(G_0) \cap MST(G))$

Figure 1: The greedy algorithm GA.

cannot guarantee more than 5/12 of the optimum and this performance ratio is achievable in the case of graphs where the maxumum cost of an edge is at most twice the minimum cost of an edge. In the last section the greedy algorithm is applied to the maximum weighted outerplanar subgruph problem [9]. In this case it guarantees the output cost at least 7/12 of the optimum.

2 The Greedy Algorithm

Let T be a maximum spanning tree (MST) of G. We denote the cost of T by $t_2 = t_2(G)$. This tree is a planar graph and its cost is at least one third of OPT, the cost of the optimal planar graph. We try to improve T adding some 3-blocks and keeping the graph to be a 3-block tree.

Let τ be a triangle in G. We say that an edge $e \in T$ *cuts* τ if the both connected components of $T \setminus \{e\}$ contain vertices of τ. Let e_1 be the smallest edge which cuts τ and e_2 be the smallest edge of T such that each of the three connected components of $T \setminus \{e_1, e_2\}$ contains one vertex of τ. We denote $Cut(\tau) = \{e_1, e_2\}$ and $cut(\tau) = c(e_1) + c(e_2)$. The *gain* of τ, $g(\tau) = c(\tau) - cut(\tau)$, is the value which we can gain by adding τ to MST. The greedy algorithm (GA) (Fig. 1) finds a triangle with the biggest gain and then *stretches* τ, i.e. assigns the cost ∞ to edges of τ. GA interrupts when there are no triangles with a positive gain and returns all chosen triangles with the rest of MST-edges.

For the purposes of analysis we slightly modify GA in the following way. We assume that to stretch τ means to extend G with two edges e'_1 and e'_2 between ends of τ with costs $c(e'_i) = c(e_i) + g(\tau)$, $i = 1, 2$, such that the forests $T \setminus \{e_1\} \cup \{e'_1\}$ and $T \setminus \{e_1, e_2\} \cup \{e'_1, e'_2\}$ are connected. We denote $N(\tau) = \{e'_1, e'_2\}$. Let the modified GA find triangles $\tau_1, ..., \tau_p$. For brevity denote $N_i = N(\tau_i)$, $G_i = G_0 \cup N_1 \cup ... \cup N_i$, $T_i = MST(G_i)$, and denote the gain of τ in G_i by $g_i(\tau)$, $i = 1, ..., p$.

The following lemma shows that $N(\tau_i)$-edges of a chosen triangle τ_i should be in $MST(G)$ till the algorithm stops. In other words, if we assign the cost ∞ to edges of $N(\tau_i)$, then the

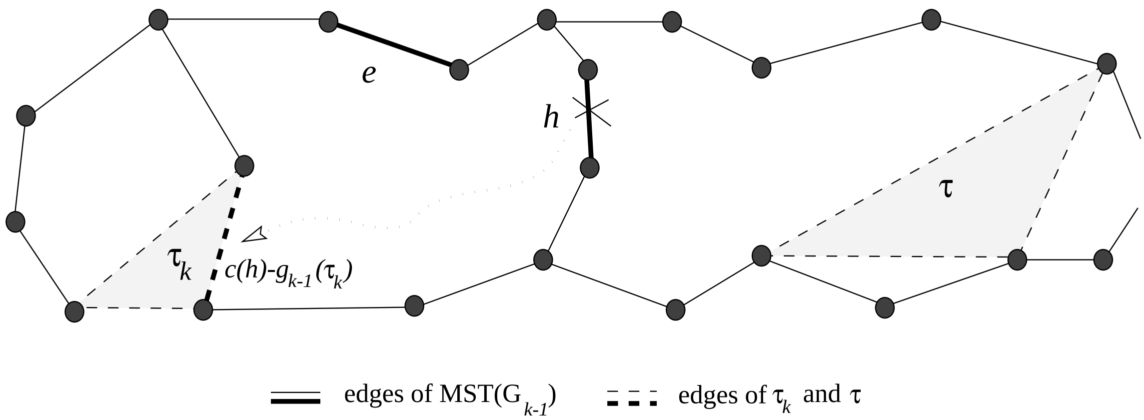

Figure 2: The step of induction

modified GA will output the same 3-block tree. Therefore, the modified and the original GA have the same output.

Lemma 1 $N_1 \cup ... \cup N_p \subseteq MST(G_p)$.

Proof. Each time GA stretches a chosen triangle τ_i, it modifies the current MST in the following way $T_i \leftarrow T_{i-1} \setminus Cut(\tau_i) \cup N_i$. So we need to show that for $i > j$,

$$Cut(\tau_i) \cap N_j = \emptyset. \quad (1)$$

We need the following denotations. Let τ be a triangle and T be an MST. Let e cut τ and e' be the smallest edge of T such that each of the three connected components of $T \setminus \{e, e'\}$ contains one vertex of τ. The value $g(\tau, e) = c(\tau) - c(e) - c(e')$ is called e-gain of τ and the edge e' is denoted by $Cut(\tau, e)$. Note that $g(\tau, e) \leq g(\tau)$.

After stretching of a triangle τ_i, the total cost of $Cut(\tau)$ and $Cut(\tau, e)$ may only increase, so the gain and e-gain of a triangle may only decrease.

The following claim proves (1). It shows that if $g(\tau) > 0$, then $Cut(\tau) \cap N_i = \emptyset$ in any G_k, $k > i$.

Claim 1 *Let $e \in N(\tau_i)$ cut a triangle $\tau \in G_0$ in $G_k, k \geq i$. Then $g_k(\tau, e) \leq 0$.*

Proof. Induction on k, the number of stretched triangles. For $k = 1$, if $e \in N(\tau_1)$ cuts τ in G_1 then some $f \in Cut(\tau_1)$ cuts τ in G_0. So if $g_1(\tau, e) > 0$, then

$$g_0(\tau) \geq g_1(\tau, e) + g_0(\tau_1) > g_0(\tau_1).$$

Let Lemma be true for $k - 1$. If e cuts τ already in G_{k-1}, then

$$0 \geq g_{k-1}(\tau, e) \geq g_k(\tau, e).$$

Now let e do not cut τ in G_{k-1} and cut τ in G_k. Then e cuts τ_k and an edge $h \in Cut(\tau_k)$ cuts τ in G_{k-1} (Fig. 2). Note that τ_k has the biggest gain in G_{k-1}. By induction,

$$\begin{aligned}0 \geq g_{k-1}(\tau_k, e) &= g_{k-1}(\tau_k) + c(h) - c(e) \\ &\geq g_{k-1}(\tau, h) + c(h) - c(e) \\ &= c(\tau) - c(e) - c(Cut(\tau, h)) \\ &\geq c(\tau) - c(e) - c(Cut(\tau, e)) = g_k(\tau, e). \quad \diamondsuit\end{aligned}$$

To bound the performance ratio of GA we need the following constant. Let N be some set of weighted edges with the vertex set V such that any triangle of G has a nonpositive gain in the graph $G' = (V, E \cup N)$. We denote by $t_3 = t_3(G)$ the infimum of MST cost of G' over all such N's.

Theorem 1 *The cost of A, the output of GA, is at least*

$$\frac{t_2(G) + t_3(G)}{2}.$$

Proof. GA interrupts when there are no triangles with the positive gain. So for $N = N_1 \cup ... \cup N_p$, no triangle has a positive gain in $G \cup N$. Therefore, the cost of $T_p = MST(G \cup N)$ is at least $t_3(G)$.

Each time we stretch a triangle τ_i we increase the cost of MST by $2g(\tau_i)$. From the other side, each time we accept a triangle τ_i we increase the cost of a 3-block tree by $g(\tau_i)$. Let G denote the total gain of chosen triangles, i.e. $G = \sum_{i=1}^{p} g(\tau_i)$. By Lemma 1, T_p contains all edges of N, thus,

$$t_3 \leq c(T_p) = c(T_0) + \sum_{i=1}^{p} 2g(\tau_i) = t_2 + 2G,$$

and

$$c(A) = t_2 + G \geq t_2 + \frac{t_3 - t_2}{2} = \frac{t_2 + t_3}{2}. \quad \diamondsuit$$

We may estimate the performance ratio of GA using the other constant. Let $opt_3(G)$ be the cost of the biggest 3-block tree in G.

Remark 1 *For any graph G, $opt_3(G) \leq t_3(G)$.*

Proof. Let P be an optimal 3-block tree in G and τ_i, $i = 1, ..., p$, be its triangles. Let T be an MST of $G \cup N$ such that any triangle of G has a nonpositive gain in $G' = G \cup N$. Let stretch triangles τ_i, $i = 1, ..., p$, consequently and denote by $G'_0 = G'$, $G'_1, ..., G'_p$ the corresponding graphs. Let $g_{i-1}(\tau_i)$ be the gain of τ_i in G_{i-1}. Then

$$d(P) - d(T) \leq \sum_{i=1}^{p} g_{i-1}(\tau_i) \leq \sum_{i=1}^{p} g(\tau_i) \leq 0 \quad \diamondsuit$$

Corollary 1 *The cost of the output of GA, is at least*

$$\frac{t_2(G) + opt_3(G)}{2}.$$

3 The Performance Ratio of the Greedy Algorithm

Conjecture 1 *Let P be the maximum planar subgraph of the graph G. Then*

$$t_3(G) \geq \frac{5}{12}c(P)$$

Remark 2 *If Conjecture 1 is true then performance ratio of the greedy algorithm is at least $\frac{3}{8} = .375$.*

Remark 3 *There is a series of planar graphs G_i such that $\lim_{i\to\infty} \frac{t_3(G_i)}{c(G_i)} = \frac{5}{12}$.*

Proof. Let take i triangles which have exactly one vertex in common and make the graph maximum planar adding $3i-3$ edges. Assign the cost 2 to each edge of the graph. In each face we put a vertex and connect it with the boundary vertices. Assign the cost one to each new edge. Let G_i denote the resulted graph. The set of edges N_i is a spanning tree of original i triangles in which any edge has cost 3. It is easy to check that no triangle of G_i has a positive gain in $G_i \cup N_i$, $c(G_i) = 24i - 30$ and the MST-cost of $G_i \cup N_i$ is $10i - 4$. $\diamondsuit$

Theorem 2 *Let the maximum cost be at most twice the minimum cost of an edge in the maximum planar subgraph P of a graph G. Then Conjecture 1 is true, i.e. $t_3(G) \geq \frac{5}{12}c(P)$.*

Proof. In order to make P triangulated we add zero cost edges to P. Let N be an edge set such that in $P \cup N$ no triangle has a positive gain. Let T' be an MST of $G \cup N$. Let $P' = (V, E, c')$ be the graph in which the cost of any edge $e \in E$ equals to the least cost of an edge in $T'(e)$, the T'-path connecting the ends of e.

Claim 2 *For any $e, e' \in E$,*
(i) $c'(e) \geq c(e)$
(ii) $c'(e) \geq \frac{1}{2}c(e')$

Proof. (i) follows from the definition of T'. Wlog, we may assume that the original graph P (without 0-cost edges) is connected. So the least cost of an edge in T' is no less than the minimum cost of an edge in the original P. $\diamondsuit$

It is easy to see that $MST(P')$ has the same property as N, i.e. no triangle in $P \cup MST(P')$ has a positive gain. From the other side the cost of $MST(P')$ is at most the cost of T'. We wish to prove a lower bound on the cost of T', so we may assume that $T' = MST(P')$.

Now we will show that the cost of P' is at least $\frac{5}{4}c(P)$ and therefore, $c'(T') \geq \frac{1}{3}c'(P') \geq \frac{5}{12}c(P)$.

Consider an arbitrary triangle τ of P with edges $x, y, z \in E$. Since the cost of a planar graph is half of the sum of the costs of its faces, it is sufficient to show that

$$c'(x) + c'(y) + c'(z) \geq \frac{5}{4}(c(x) + c(y) + c(z)) \tag{2}$$

Note that a pair of edges of τ, say y and z, has the same cost in P', $c'(y) = c'(z)$ and $c'(x) \geq c'(y)$. (In other words, the least cost edge of $T'(x) \cup T'(y) \cup T'(z)$ belongs to $T'(y) \cap T'(z)$.) Since $g(\tau) \leq 0$ in P', $c'(x) + c'(y) \geq c(x) + c(y) + c(z)$. So to prove (2) it is necessary to show that $c'(y) \geq \frac{1}{4}(c(x) + c(y) + c(z))$. This is true because $c'(y)$ is at least $c(y)$, $c(z)$ and half of $c(x)$ (Claim 2 (ii)). $\diamondsuit$

Corollary 2 *The performance ratio of the greedy algorithm is at least .375 for the graphs where the maximum cost of an edge is at most twice the minimum cost.*

4 Outerplanar Subgraphs

The greedy algorithm produces outerplanar graphs, so it is an approximation algorithm for maximum outerplanar subgraph problem (MOSP). The unweighted case was considered in [4]. They proved that in any outerplanar graph there is a 3-tree which contains at least two thirds of the total number of edges. In this section we will prove that the same fact is true for the weighted case.

Theorem 3 *Let P be a maximum outerplanar graph of a graph G. Then*

$$opt_3(G) \geq \frac{2}{3}c(P)$$

Collorary 1 yields

Theorem 4 *The performance ratio of the greedy algorithm is at least $\frac{7}{12} \approx .583$.*

We will show that any outerplanar graph $P = (V, E)$ can be represented as a union of three 3-block trees such that any edge of P belongs to two of these 3-block trees.

We may assume that P is a maximal outerplanar graph, since we may add zero-cost edges. For the purposes of analysis we add a copy for each outer face edge. Consider the dual graph $\bar{P} = (F, E')$ of the graph P, where F is the set of faces of P and $e' \in E'$ is an edge corresponding to an edge $e \in E$ (Fig. 3).

It is easy to see that $\bar{P}$ is a cubic tree T (each inner vertex has degree 3) with an additional vertex b corresponding to the outer face. This vertex b is adjacent to all leaves of T corresponding to the outer face edges. We assume that T is rooted in a fixed inner vertex r.

A *proper coloring* of $\bar{P}$ is a partition of $F - \{b, r\}$ into three subsets (*colors*) X_i, $i = 1, 2, 3$, such that
(i) adjacent vertices have different colors;
(ii) the subgraph of $\bar{P}$ induced by $F - X_i - \{r\}$ is connected;
(iii) three neighbours of the root have different colors.

Let C_i be a set of edges of P correspnding to edges of $\bar{P}$ which have both ends colored and one of these colors is X_i, $(i = 1, 2, 3)$. We also add to C_i two edges of the triangle face r. The first of them corresponds to an edge incident to X_i-vertex and the other is the next in the clockwise order.

Lemma 2 *A set C_i corresponding to a color X_i, is a 3-block tree.*

Proof. We will show that each block of C_i has at most 3 vertices. Let $H \subset C_i$ be a cycle and H be not a triangle corresponding to three edges of $\bar{P}$ incident to the same vertex. Then there is at least one inner face $f \neq r$ of P inside this cycle. Let remove all edges incident to the vertices of $X_i \cup \{r\}$ from $\bar{P}$. Then the vertex of $\bar{P}$ corresponding to f has no path to b which contradicts (ii). Moreover, triangles in H cannot share edges by (i). $\diamondsuit$

Let $e \in E$ corresponds to an edge of $e' \in E'$ which is incident neither to b nor to r. Then e belongs to two different 3-block trees corresponding to two colors of the ends of e'. It is easy to see that edges of the face r also belong to two different 3-block trees. Thus, three 3-block trees corresponding to a proper coloring has the cost $2c(P)$.

To finish the proof of Theorem 3 we will prove the following

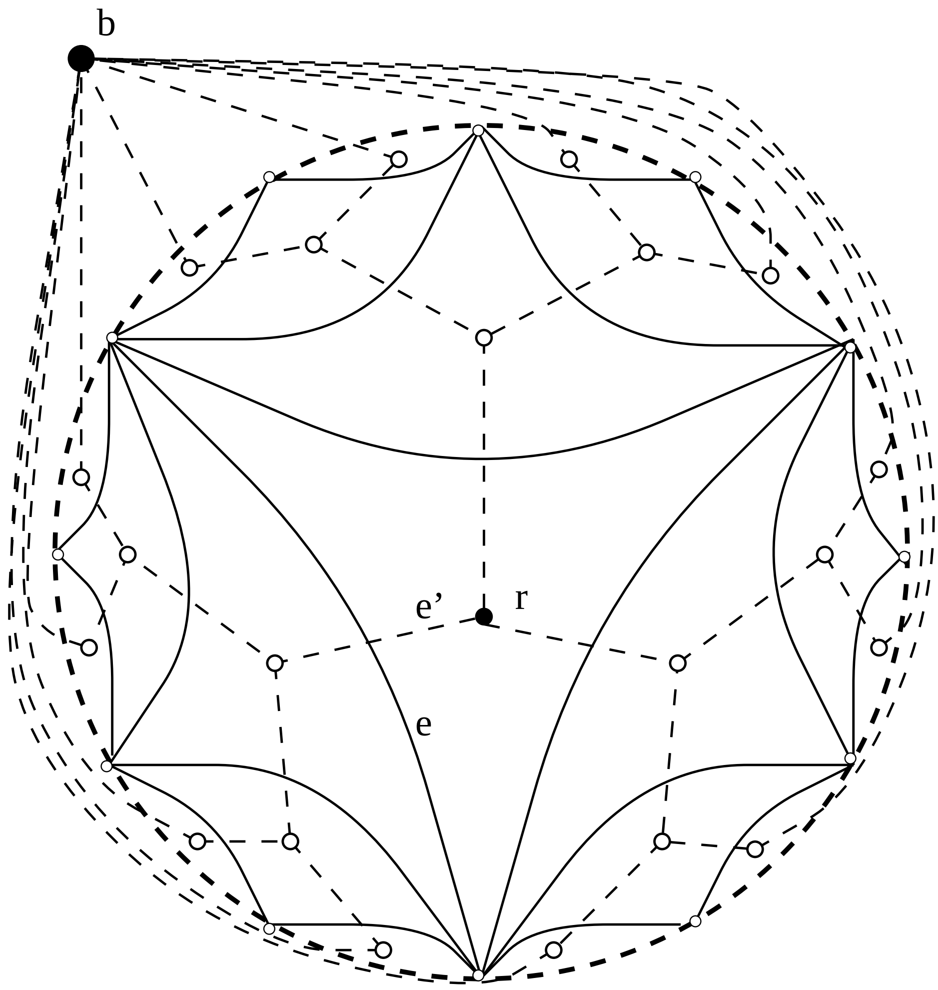

Figure 3: An outerplanar graph (solid lines) with auxiliary edges (thick dashed lines) and its dual graph (dashed lines).

Lemma 3 *The graph $\bar{P}$ has a proper coloring.*

Proof. The forest $T - \{r\}$ has three components $T_i, i = 1, 2, 3$. Each of them is a binary tree. We root them in vertices incident to r in T. We color the root of T_i in X_i and further color children in two different colors which do no coincide with the color of their parent. It is obvious that any vertex has a path to a leaf in $T_i \setminus X_j$, $i, j = 1, 2, 3$. $\diamondsuit$

References

[1] P. Berman and V. Ramaiyer. *Improved approximations for the Steiner tree problem.* J. of Algorithms 17: 381–408, 1994.

[2] M. Bern and D. Eppstein. *Approximation Algorithms for Geometric Problems.* In "Approximation Algorithms for NP-Complete Problems", D. Hochbaum, (ed.), PWS Publications, 1996. (to appear)

[3] J. Cai, X. Han and R.E. Tarjan. *An)9$mlogn$)-time approximation algorithm for the maximal planar subgraph problem.* SIAM J. on Comp. 22: 1142–1162.

[4] G. Calinescu, C.G. Fernandes, U.Finkler and H. Karloff. *A better approximation algorithm for finding planar subgraphs.* 7th SODA: 16–25, 1996.

[5] T. Chiba, I. Nishioka and I. Shirakasawa. *An algorithm of maximal planarization of graphs.* Proc. IEEE Symp. on Circuits and Systems, 649–652, 1979.

[6] D. Du, Y. Zhang, and Q. Feng. *On better heuristic for Euclidean Steiner minimum trees.* 32nd IEEE Symp. on Found. of Comp. Science, 431–439, 1991.

[7] M.E. Dyer, L.R. Foulds and A.M. Frieze. *Analysis of heuristics for finding a maximum weight planar subgraph.* European Journal of Operation Research, 20: 102–114, 1985.

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

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

[10] M. Jünger and P. Mutzel. *Maximum planar subgraph and nice embeddings: Practical layout tools.* Universität zu Köln, Rep. No 93.145, 1993, (to appear in Spesial issue of Algorithmica on Graph Drawing)

[11] R. Tamassia, G. Di Battista and C. Batini. *Automatic graph drawing and readability of diagrams.* IEE Trans. on Systems, Man and Cybernetics, 18: 61–79, 1988.

[12] A. Z. Zelikovsky. *An 11/6-approximation Algorithm for the network Steiner Problem* Algorithmica, 9: 463–470, 1993.

[13] A. Z. Zelikovsky. *Better approximation bounds for the network and Euclidean Steiner tree problems* Tech. Rep. CS-96-06. University of Virginia, Charlottesville, 1996.