Light Spanners and Approximate TSP in Weighted Graphs with Forbidden Minors

Michelangelo Grigni* Papa Sissokho†

December 13, 2002

Abstract

Given an edge weighted graph G with n vertices and no K_r-minor and a small positive ε, we show that a simple greedy algorithm [1] finds a spanning subgraph approximating all shortest-path distances within a factor of $1+\varepsilon$, and with total edge weight at most $O((r\sqrt{\log r}\cdot\log n)/\varepsilon)$ times the weight of a minimum spanning tree. This result implies a quasi-polynomial time approximation scheme (QPTAS) for the traveling salesman problem (TSP) in such graphs, with running time $n^{O((r^4\sqrt{\log r}\cdot\log n\cdot\log\log n)/\varepsilon^2)}$.

Our analysis shows that a graph with *detour gap number* [5] $\Omega(r\sqrt{\log r}\cdot\log n)$ has a K_r-minor. We also show that this dependence on n is nearly tight, by exhibiting graphs with no K_6-minor (apex graphs) and detour gap number $\Omega((\log n)/\log\log n)$.

As a step towards eliminating the $\log n$ factors in the first paragraph, we propose a generalized detour gap number, now depending on ε, and we show that it remains bounded for apex graphs and some similar graph families.

1 Introduction

Suppose a connected graph $G=(V,E)$ has an edge weighting w, that is a non-negative real weight $w(e)$ for each edge $e\in E$. We interpret these weights as lengths, and let $d_G(u,v)$ denote the minimum length of a path connecting vertices u and v. The distance d_G is non-negative and satisfies the triangle inequality.

We consider the Traveling Salesman Problem (TSP) in this metric. That is, we want to find a cyclic ordering of V so that the total length of the tour (according to d_G) is as small as possible. Or equivalently, find a minimum length cyclic walk in G visiting each vertex at least once. The exact problem is NP-hard, so we seek approximation schemes: given a fixed $\varepsilon>0$, find a tour with cost at most $1+\varepsilon$ times optimal, in time that is polynomial (or quasi-polynomial) in n. Even

*This work was supported by NSF Grant number CCR-9820931. Department of Mathematics and Computer Science, Emory University, Atlanta GA 30322, USA. Email: mic@mathcs.emory.edu.

†Same support and address. Email: psissok@emory.edu.

the approximation problem is MAXSNP-hard [8], so we do not expect an efficient approximation scheme for general graphs (even sparse graphs). However, there are approximation schemes running in polynomial time when G is planar [2, 6], and in quasi-polynomial time when G has bounded genus [5].

In this paper we show that the approximation scheme of [5] takes quasi-polynomial time under the more general condition that the input graphs have a fixed forbidden minor. Our main contribution is an improved analysis of the greedy algorithm used to find a *spanner*, showing it is not too heavy.

A spanner of G is a spanning subgraph $G' = (V, E')$ whose edges inherit their weights from G. A spanner G' defines another distance $d_{G'}$ on V, clearly $d_{G'}(u, v) \geq d_G(u, v)$. For $s \geq 1$, we say that G' is an *s-spanner* if $d_{G'}(u, v) \leq s \cdot d_G(u, v)$ for all $u, v \in V$. The minimum such s is the *stretch factor* of G'. We want to bound $w(G')$, the total edge weight of G', so we define the *tree weight* of G' as $\mathrm{tw}(G', G) \stackrel{\text{def}}{=} w(G')/w(T)$, where T is a min-weight spanning tree (MST) of G.

For the TSP approximation scheme we want a spanner G' in G with very low stretch factor ($s = 1 + \varepsilon$ for a small fixed $\varepsilon > 0$) and bounded tree weight; the tree weight bound appears in the exponent of our running time (see Theorem 3.5). We find G' by the following greedy algorithm of Althöfer *et al.* [1]. Here $d_{G'}(e)$ is the distance in G' between the endpoints of e, which is $+\infty$ when they are not connected:

```
Span(G = (V, E), s):
    G' ← (V, ∅)
    for all edges e ∈ E in non-decreasing w order do
        if s · w(e) < d_{G'}(e) then add e to G'
    return G'
```

In the resulting $G' = \mathrm{Span}(G, s)$ we have $d_{G'}(e) \leq s \cdot w(e)$ for every edge $e \in E$; therefore G' is an s-spanner. The algorithm is "idempotent" in the sense that $\mathrm{Span}(G', s) = G'$. By comparison with Kruskal's algorithm, we see that $T(G) \stackrel{\text{def}}{=} \mathrm{Span}(G, n-1)$ is a MST in G, and G' always[1] contains $T(G)$, and $T(G') = T(G)$.

In [5] we introduced the *detour gap number* $\mathrm{gap}(G)$; we define it again in Section 2. We bounded the tree weight of G' in terms of $\mathrm{gap}(G)$:

Theorem 1.1 ([5]) *If $s > 1$ then $\mathrm{tw}(\mathrm{Span}(G, s), G) \leq 1 + \mathrm{gap}(G)/(s-1)$.*

We know [5] that the gap number is bounded for some graph families: $\mathrm{gap}(G) \leq 2$ when G is planar, and $\mathrm{gap}(G) \leq 12g - 4$ when G has positive genus g. Also $\mathcal{G}(\gamma) \stackrel{\text{def}}{=} \{G \mid \mathrm{gap}(G) \leq \gamma\}$ is a hereditary graph family; or in other words, $\mathrm{gap}(H) \leq \mathrm{gap}(G)$ whenever H is a minor of G. On the other hand $\mathrm{gap}(K_r) \geq r/2 - 1$. We conjectured that $\mathrm{gap}(G)$ is bounded for any graph family with a forbidden minor. This is false, as we show in Section 4: there is a sequence of graphs with no K_6-minor and $\mathrm{gap}(G) = \Omega(\log n / \log\log n)$. These examples are *apex graphs*: they become planar after deleting one vertex.

We try to salvage the conjecture in two ways:

[1] Strictly speaking, we should define (along with w) some non-decreasing ordering of the edges to be used by Span. Then G' and $T(G)$ are uniquely defined.

- In Section 3 we prove that $\text{gap}(G)$ is $O(\log n)$ whenever there is a forbidden minor. By the apex example, this bound is nearly tight. The bound implies that the approximation scheme runs in time $n^{O(\log n \cdot \log\log n)}$ (the big-Oh hides dependence on ε and the forbidden minor).
- In Section 2 we propose a generalized gap number $\text{gap}_\varepsilon(G)$ (in fact $\text{gap}(G) = \text{gap}_0(G)$), with Theorem 2.2 replacing Theorem 1.1. In Section 5 we prove that $\text{gap}_\varepsilon(G)$ is bounded (depending on ε but not n) for apex graphs and similar graph families. Consequently the approximation scheme runs in time $n^{O(\log\log n)}$ for such graphs (the big-Oh hides dependence on ε and $\text{gap}_\varepsilon(G)$).

Finally, we revise our conjecture: for positive ε, we conjecture that large $\text{gap}_\varepsilon(G)$ implies large Hadwiger number. More precisely:

Conjecture 1.2 *For every $\varepsilon > 0$ and $r \geq 1$, there exists a γ so that for all G, if $\text{gap}_\varepsilon(G) \geq \gamma$ then G has a K_r-minor.*

2 The Gap and Gap$_\varepsilon$ Numbers

Again we are given graph G with a nonnegative edge weighting w, and a target stretch factor $s = 1 + \varepsilon$, where $\varepsilon > 0$. We apply the Span algorithm to compute $G' = \text{Span}(G, s)$; we want to bound the tree weight $w(G')/w(T)$, where T is the MST ($T = T(G) = T(G')$). By rescaling w we may assume $w(T) = 1$, and so the tree weight is simply $w(G')$.

By Theorem 1.1 it suffices to bound the *detour gap number* $\text{gap}(G)$, which we define in this section. In fact we define a generalized gap number $\text{gap}_\varepsilon(G)$; this includes the previous definition since $\text{gap}(G) = \text{gap}_0(G)$.

Suppose we know G, G', and T, but not w; we want to upper bound $w(G')$ subject to these three constraints on w: $w(T) = 1$, T is a MST, and $G' = \text{Span}(G, s)$. We can also add the fourth constraint $G' = \text{Span}(G', s)$, but then we see that the third constraint $G' = \text{Span}(G, s)$ becomes redundant. So for the next few paragraphs, it simplifies our notation to assume that $G = G'$.

For an edge e of G, we define the *detour gap* at e as $g(e) \stackrel{\text{def}}{=} d_{G-e}(e) - w(e)$; this is how much the distance between the endpoints of e would increase if we removed e. Note $g(e)$ may be infinite for an edge of $e \in T$ (if it is a cut edge), but it is always finite for $e \in G - T$ ($G - T$ denotes G with the edges of T removed). The condition $G = \text{Span}(G, s)$ implies $g(e) \geq \varepsilon \cdot w(e)$. Combining these observations, we bound $w(G)$ in terms of $g(G - T)$, the sum of detour gaps for edges in $G - T$:

Lemma 2.1 *If $G = \text{Span}(G, 1 + \varepsilon)$ and $w(T(G)) = 1$, then $w(G) \leq 1 + (1/\varepsilon) \cdot g(G - T)$.*

Now suppose we fix G, T, and ε, and we ask what is the largest possible $g(G-T)$, over all edge weightings w such that T is a MST with $w(T) = 1$. This is exactly the value $\text{gap}_\varepsilon(G, T)$ of the linear program (LP) presented below.

We first introduce two notions to help define our constraints. For an edge $e \in G - T$, let T_e denote the unique path in T connecting the endpoints of e. A *detour* is a pair (P, e) where e is an edge and P is a path such that $e + P$ is a simple cycle (so, P connects the endpoints of e). We fix an enumeration (P_i, e_i) of all detours in G. Now we define

$$\text{gap}_\varepsilon(G,T) \stackrel{\text{def}}{=} \max g(G-T) \text{ such that edge-weightings } w, g \text{ satisfy:}$$

$$\begin{aligned}
w(e) &\geq 0 && \forall\ e \in G && (1)\\
w(T) &\leq 1 && && (2)\\
w(s) &\leq w(e) && \forall\ e \in G-T,\ \forall\ s \in T_e && (3)\\
g(e_i) &\leq w(P_i) - w(e_i) && \forall\ (P_i, e_i) && (4)\\
g(e) &\geq \varepsilon \cdot w(e) && \forall\ e \in G && (5)
\end{aligned}$$

Remarks: This LP has the trivial solution $w = g = 0$; if $\varepsilon \in [0,1]$ we have a non-trivial solution (take a uniform w). Suppose we have a non-trivial optimal solution: then (2) is an equality, $g(e)$ is the detour gap satisfying (4) and (5) as discussed above, and (3) says that T is a MST. This LP has an exponential number of constraints in (4), but it is possible to reduce that to a polynomial number by adding extra variables (to do shortest path computations). Note $\text{gap}_\varepsilon(G,T)$ is non-increasing with ε; if we set $\varepsilon = 0$ we get the quantity $\text{gap}(G,T) = \text{gap}_0(G,T)$ as defined in [5]. We define $\text{gap}_\varepsilon(G) = \max_T \text{gap}_\varepsilon(G,T)$, where T ranges over all spanning forests. Note $\text{gap}_\varepsilon(G)$ is non-increasing with ε.

Now we again consider the possibility that G' is distinct from G. By maximizing the bound in Lemma 2.1 over all choices of G' and T in G, we have:

Theorem 2.2 *Suppose* $s = 1 + \varepsilon > 1$, *and* $\text{gap}_\varepsilon(G') \leq \gamma$ *for all spanners* G' *of* G. *Then* $\text{tw}(\text{Span}(G,s), G) \leq 1 + \gamma/\varepsilon$.

This is the promised generalization of Theorem 1.1. The statement here is more complicated because when $\varepsilon > 0$, it is possible that $\text{gap}_\varepsilon(G') > \text{gap}_\varepsilon(G)$.

If we drop the MST constraints (part (3)) from the above linear program we define a new quantity $\text{gap}'_\varepsilon(G,T)$, and similarly $\text{gap}'_\varepsilon(G)$. We will use these as upper bounds on $\text{gap}_\varepsilon(G,T)$ and $\text{gap}_\varepsilon(G)$ in Section 5.

In fact we do not lose too much by considering gap'_ε instead of gap_ε: if a graph family $\mathcal{F}$ is closed under edge subdivision, then $\sup_{G \in \mathcal{F}} \text{gap}_\varepsilon(G) = \sup_{G \in \mathcal{F}} \text{gap}'_\varepsilon(G)$. In particular the apex graphs are closed under edge subdivision. We define the following graph families for $\varepsilon \geq 0$:

$$\begin{aligned}
\mathcal{G}_\varepsilon(\gamma) &\stackrel{\text{def}}{=} \{G |\text{ for all } G' \subseteq G,\ \text{gap}_\varepsilon(G') \leq \gamma\}\\
\mathcal{G}'_\varepsilon(\gamma) &\stackrel{\text{def}}{=} \{G |\text{ for all } G' \subseteq G,\ \text{gap}'_\varepsilon(G') \leq \gamma\}
\end{aligned}$$

Then these families are hereditary (closed under minors), with $\mathcal{G}(\gamma) = \mathcal{G}_0(\gamma) \subseteq \mathcal{G}_\varepsilon(\gamma)$ and $\mathcal{G}'_\varepsilon(\gamma) \subseteq \mathcal{G}_\varepsilon(\gamma)$. Also $\mathcal{G}(\gamma)$ is closed under edge subdivision for $\gamma \geq 2$.

3 A QPTAS for Graphs with no K_r-minor

In this section we argue that if a weighted graph G has a forbidden minor H (and therefore no K_r-minor for $r = |V(H)|$) then there exist a quasi-polynomial time approximation scheme (QPTAS) for the TSP for G. Before proving the existence of such QPTAS we need a weaker version of Conjecture 1.2.

3.1 A Logarithmic Bound on Gap

In this section we argue that if G has a forbidden minor then $\text{gap}(G)$ is $O(\log n)$. This is close to tight: the graphs constructed in Section 4 have no K_6-minor and gap number $\Omega(\log n/(\log\log n))$. We rely on Thomason's Theorem, stating that graphs with no K_r-minor are sparse:

Theorem 3.1 ([3, 9]) *A graph with no K_r-minor has average degree $O(r\sqrt{\log r})$.*

The *diameter* of an edge weighted graph is simply the maximum distance between any two vertices (as usual, we mean weighted distance).

Lemma 3.2 *Given a tree T of total edge weight $w(T) \leq 1$ and $0 < d \leq 1$, we can delete edges to obtain a forest of $O(1/d)$ subtrees of diameter at most d.*

We omit the proof of this lemma.

Theorem 3.3 *Let G be a simple graph with n vertices. If $gap(G) \geq \gamma \log n$ then G contains a K_r-minor, where r is $\Omega\left(\gamma/\sqrt{\log \gamma}\right)$.*

Proof: All logarithms are base 2. Choose w and T in G witnessing $gap(G) = \gamma \log n$. That is: $w(T) = 1$, and $\sum_{e \notin T} g(e) = \gamma \log n$. Let m be the number of non-tree edges in G; note $m > \gamma n$ would imply the claim by Theorem 3.1, so we may suppose $m \leq \gamma n$. Let $A = \text{gap}(G)/m$, the average $g(e)$ for non-tree edges; our bound on m implies $A \geq \log n/n > 2/n$ (for $n > 4$). Let E' be the set of non-tree edges e such that $g(e) \geq A/2$; we see that the edges outside E' can contribute at most half of $\text{gap}(G)$, therefore $\sum_{e \in E'} g(e) \geq (\gamma/2)\log n$. On the other hand $g(e) \leq 1$ for all edges.

Partition E' into $\log n$ parts $C_0, C_1, \ldots$ according to $\lfloor \log(1/g(e)) \rfloor$; that is, put e into C_i iff $2^{-i-1} < g(e) \leq 2^{-i}$. By averaging over these parts, we find a C_α such that $\sum_{e \in C_\alpha} g(e) \geq \gamma/2$. Let $d = 2^{-\alpha}$; each edge $e \in C_\alpha$ has $d/2 < g(e) \leq d$, so C_α contains at least $\gamma/(2d)$ edges. Say that two vertices are *close* if they are connected by a path of length at most $d/4$ in T. For a single edge $e \in C_\alpha$, its endpoints cannot be close because this would force $g(e) \leq d/2$. For a pair of edges $e, e' \in C_\alpha$ (chosen so that $w(e) \leq w(e')$), we cannot match up their endpoints so that both pairs are close, because this would force $g(e') \leq d/2$.

By Lemma 3.2 we identify $O(1/d)$ subtrees in T of diameter $d/4$, covering the vertices of T. Take the graph $T \cup C_\alpha$, contract each subtree to a point, remove loops and parallel edges, and call the result G_α. By the above observations regarding edges in C_α, we see that each edge $e \in C_\alpha$ maps to a distinct edge in G_α. Therefore G_α has average degree $|C_\alpha|/O(1/d) = \Omega(\gamma)$. Theorem 3.1 implies our claim. □

Corollary 3.4 *If G is a simple graph with n vertices and no K_r-minor, then $gap(G)$ is $O(r\sqrt{\log r} \cdot \log n)$.*

3.2 The TSP Approximation Scheme

Suppose we have an edge weighted graph G on n vertices with no K_r-minor, and we want to find a tour in G with weight at most $1 + 3\varepsilon$ times optimal, for some small positive ε. The first step of our approximation scheme is to replace G with $G' = \text{Span}(G, 1 + \varepsilon)$; denote the tree weight of G' by $W = \text{tw}(G', G)$. The second step is to find a tour within G' with weight at most $1 + \varepsilon$ times optimal. The first step is clearly polynomial time, and for the second step we use the algorithm in [5]:

Theorem 3.5 *We can complete the second step in time $n^{O(Wr^3(\log\log n)/\varepsilon)}$.*

Since G has no K_r-minor we can bound $\text{gap}(G)$ using Corollary 3.4, and then Theorem 1.1 implies that W is $O(r\sqrt{\log r} \cdot (\log n)/\varepsilon)$, and so the approximation scheme runs in quasi-polynomial time

$$n^{O((r^4\sqrt{\log r} \cdot \log n \cdot \log\log n)/\varepsilon^2)}.$$

Of course we would prefer a true polynomial in n; here we remark on the two appearances of n in the exponent. First we note that proving Conjecture 1.2 would eliminate the $\log n$ term. Second, the $\log\log n$ term originates from the following issue: given a subgraph of G with p distinguished "portal" vertices (typically $p = \Theta((W/\varepsilon)\log n)$), in how many different ways must we consider ordering these portals inside the larger tour? Naively there are $2^{O(p \log p)}$ orderings, but at least in the context of planar graphs [2, 6] we do not need to consider "self-crossing" tours, and this reduces the number to $2^{O(p)}$. Perhaps such an idea extends to more general graph families.

4 Apex Graphs have Unbounded Gap

An *apex graph* is a graph with a special vertex (the apex) such that if we delete the apex, the remaining graph is planar. In this section we exhibit a sequence of apex graphs with gap number $\Omega((\log n)/\log\log n)$. Since apex graphs have no K_6-minor, this shows the dependence on n is nearly tight in Corollary 3.4.

First we fix an integer $b \geq 2$, and define the apex graphs $G(b, h)$ (for $h \geq 1$) by recursion on h. Then we define $G_k = G(k, k)$, and we will show $\text{gap}(G_k) \geq k/3$. Note b and h stand for "breadth" and "height".

Each $G(b, h)$ has an apex vertex v adjacent to every other vertex; we define $G(b, h)$ by describing a planar drawing of $G(b, h) - v$. The drawing is bounded by a triangle T_h with corners U_h, L_h, and R_h (upper, left, and right). The bottom side $L_h R_h$ is subdivided into several edges; see Figure 4.1(a).

- In the base case $h = 1$, we subdivide the bottom side into b edges and we add edges from U_1 to these $b - 1$ new internal vertices.

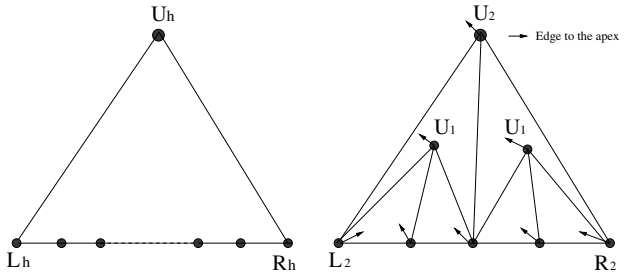

Figure 4.1: The subdivided triangle T_h, and the graph $G(2,2)$.

- In the general case $h \geq 2$, we take b copies $C_1, C_2, \ldots, C_b$ of $G(b, h-1) - v$. We arrange them left to right, identifying the right corner of C_i with the left corner of C_{i+1} (for $1 \leq i < b$). The left corner of C_1 is L_h, and the right corner of C_b is R_h. We create a new upper vertex U_h above the chain, and connect it to L_h, R_h, and all $b-1$ of the identified corners. See Figure 4.1(b) for an example.

Partition the "upper" vertices of $G(b,h)$ into h levels: $V_1, \ldots, V_h$. V_1 contains just vertex U_h; V_2 contains the b copies of U_{h-1}, etc., and V_h contains the b^{h-1} copies of U_1. Let P be the path of b^h edges from L_h to R_h along the bottom of triangle T_h. Now we define a weighting on the edges of $G(b,h)$:

- Each edge $e \in P$ gets $w(e) = 1/b^h$, so that $w(P) = 1$.
- Each apex edge from V_i to v gets weight $1/b^i$.
- All other edges get weight 1.

We define a tree T spanning $G(b,h)$ as follows: T contains P, one (arbitrary) edge from v to P, and all other apex edges from an upper vertex (in some V_i) to v. Note then that the weight of T is $w(T) = 1 + w(P) + \sum_{i=1}^{h} |V_i| b^{-i} = 2 + h/b$.

Now consider an edge from V_i to P (note these edges are not in T): inspection shows that it has detour gap $g(e) = b^{-i}$. Therefore summing over all such edges, we see $g(G(b,h) - T) \geq \sum_{i=1}^{h} (b+1)|V_i| b^{-i} \geq h$.

Therefore $\text{gap}(G(b,h)) \geq \text{gap}(G(b,h), T) \geq g(G(b,h) - T)/w(T) \geq h/(2 + h/b) = bh/(2b + h)$. Now since $G_k = G(k,k)$, we see that $\text{gap}(G_k) \geq k/3$. Also note that G_k has $n = \Theta(k^k)$ vertices, so in terms of n we have $\text{gap}(G_k) = \Omega((\log n)/\log\log n)$.

5 On the Gap_ε Number

We now return our attention to the gap_ε number introduced in Section 2. In Section 5.1 we derive a method to upper bound the gap_ε number, and in Section 5.2 we apply this method to apex graphs.

5.1 Charging Schemes

Recall that $\text{gap}_\varepsilon(G) \leq \max_T \text{gap}'_\varepsilon(G,T)$, so in order to establish upper bounds on $\text{gap}_\varepsilon(G)$ we consider the dual linear program for $\text{gap}'_\varepsilon(G,T)$. We introduce a variable k dual to constraint (2), and a non-negative variable x_i dual to constraint (4) for each detour (P_i, e_i). There are also dual variables for constraint (5), but they are eliminated after some simplification (suppressed in this abstract).

Consider a game where we move charges among the edges of G. Each edge initially holds zero charge. Each positive x_i describes a move where we subtract x_i units of charge from e_i, and we add x_i units of charge to each edge of P_i. We say "e_i sends x_i units of charge to P_i". Define $\text{in}(e) = \sum\{x_i |\ e \in P_i\}$ (the total charge received by e), $\text{out}(e) = \sum\{x_i |\ e = e_i\}$ (the total charge sent by e), and $\text{net}(e) = \text{in}(e) - \text{out}(e)$. We find:

$$\begin{array}{l} \text{gap}'_\varepsilon(G,T) = \min k \text{ such that } k \geq 0 \text{ and } x_i \geq 0 \text{ satisfy:} \\ \begin{array}{lcll} \text{out}(e) & \geq & 1 & \forall e \notin T \quad (1) \\ \text{net}(e) & \leq & \varepsilon\ (\text{out}(e) - 1) & \forall e \notin T \quad (2) \\ \text{net}(e) & \leq & k + \varepsilon\ \text{out}(e) & \forall e \in T \quad (3) \end{array} \end{array}$$

Any solution (nonnegative k and x_i's, not necessarily optimal) to the above three constraints is an *ε-charging scheme of value k for* (G,T). Note that for $0 \leq \delta \leq \varepsilon$, a δ-charging scheme is also an ε-charging scheme of the same value. In particular 0-charging schemes are ε-charging schemes.

Suppose $G = \bigcup_{i=1}^r G_i$ where each G_i is a subgraph of G, and each $(G_i, T \cap G_i)$ has an ε-charging scheme of value k_i. Then by simply adding these charging schemes, we get an ε-charging scheme for (G,T) of value $\sum_{i=1}^r k_i$.

If G is not 2-connected, a charging scheme decomposes into independent charging schemes in each 2-connected component. Therefore $\text{gap}'_\varepsilon(G,T)$ is the maximum of $\text{gap}'_\varepsilon(H, T \cap H)$, where H ranges over all 2-connected components of G. Similarly $\text{gap}'_\varepsilon(G)$ is the maximum of $\text{gap}'_\varepsilon(H)$, over the same range of H.

When G is planar and T is a spanning forest, a construction of [1] shows that (G,T) has an integral 0-charging scheme of value at most 2.

5.2 Apex Graphs have Bounded Gap_ε

Recall the family $\mathcal{G}'_\varepsilon(\gamma)$ introduced at the end of Section 2. For a graph family $\mathcal{F}$, define the family $\text{apex}(\mathcal{F}) = \{G |\ \exists v \in V(G),\ G - v \in \mathcal{F}\}$. By the remark at the end of the last section, the apex graphs are in $\text{apex}(\mathcal{G}'_\varepsilon(2))$ (for any $\varepsilon \geq 0$). Our main result is the following:

Theorem 5.1 *For $\varepsilon > 0$, $\text{apex}(\mathcal{G}'_\varepsilon(\gamma)) \subseteq \mathcal{G}'_\varepsilon(4 + 2\gamma(1 + 1/\varepsilon))$.*

Proof: For $\varepsilon > 0$ and $G \in \mathcal{G}'_\varepsilon(\gamma)$, let G^v be a graph containing a vertex v such that $G = G^v - v$. We need to show that $\text{gap}'_\varepsilon(G^v) \leq 4 + 2\gamma(1 + 1/\varepsilon)$.

By the remarks in Section 5.1 we may assume G^v is 2-connected, and therefore G is connected. Let T^v be the spanning tree in G^v achieving $\text{gap}'_\varepsilon(G^v) =$

$\text{gap}'_\varepsilon(G^v, T^v)$. Now it suffices to exhibit an ε-charging scheme for (G^v, T^v), of the claimed value. $F = T^v - v$ is a forest of disjoint trees $T_1 \cup T_2 \ldots \cup T_l$ in G; for each T_i there is a unique edge $e_i \in T^v$ connecting the apex v to some node v_i of T_i. In other words T^v is the union of the T_i's and the e_i's.

For each T_i, let G_i be the graph induced by $V(T_i)$. For any two distinct trees T_i and T_j, let E_{ij} denote the set of edges connecting $V(T_i)$ and $V(T_j)$. Whenever $E_{ij} \neq \emptyset$, let G_{ij} denote the subgraph $T_i \cup T_j \cup E_{ij}$ (note G_i is not typically a subgraph of G_{ij}).

We decompose $G^v = \bigcup_{i=1}^{3} G^i$, where the subgraphs G^i are defined as follows:

- $G^1 = \bigcup_{i=i}^{l} G_i$
- $G^2 = T^v \cup \{\text{all edges adjacent to the apex } v\}$
- $G^3 = T^v \cup \bigcup_{i \neq j} E_{ij}$

It suffices to show that each $(G^i, T^v \cap G^i)$ (for $1 \leq i \leq 3$) has an ε-charging scheme of value k_i, where $k_1 + k_2 + k_3 \leq 4 + 2\gamma(1 + 1/\varepsilon)$. Since $G^1 \in \mathcal{G}'_\varepsilon(\gamma)$, $k_1 \leq \gamma$. Since G^2 is planar, $k_2 \leq 2$. So all that remains is to show $k_3 \leq \gamma + 2(1 + \gamma/\varepsilon)$.

We select edges $S = \{a_1,\ a_2, \ldots, a_p\} \subseteq \bigcup_{i \neq j} E_{ij}$ so that $T = S \cup \bigcup_i T_i$ is a spanning tree for G; note we select at most one edge from each E_{ij}.

Since $G^3 \in \mathcal{G}'_\varepsilon(\gamma)$, there is an ε-charging scheme in G^3 leaving at most γ units of charge on each edge of T. However, the edges in S are not in the real tree T^v, so we must find a way to move these surplus charges from S to T^v. We state our residual problem as follows: given graph $G^4 = T^v \cup S$ with initial charges at most γ on each $e \in S$, we want move charges to satisfy the following two conditions:

$$\text{out}(a_i) \geq 1 \quad \forall\ a_i \in S \tag{1}$$

$$\gamma + \text{net}(a_i) \leq \varepsilon\ (\text{out}(a_i) - 1) \quad \forall\ a_i \in S \tag{2}$$

Note that G^4 is planar since $G^4 - v$ is a tree. So there is a 0-charging scheme on (G^4, T^v) of value 2, and we use this scheme scaled up by a factor of $c = 1 + \gamma/\varepsilon$ (that is, on each step we move c units instead of one).

This results in an accumulation of at most $2c$ additional units of charge for all edges in T^v, and a net charge of 0 for the edges in S (the non-tree edges of G^4). Moreover, $\text{out}(a_i) = c$ for all $a_i \in S$. Thus condition (1) above is satisfied. For condition (2), note that the $\varepsilon(\text{out}(a_i) - 1) = \gamma$ and $\text{net}(a_i) = 0$. Note that the overall charge is the initial charge (at most γ) plus the the charge (at most $2c$) accumulated during the process of satisfying conditions (1) and (2). Thus we have ε-charging scheme for (G^4, T^v) of value $k_3 \leq \gamma + 2c = \gamma + 2(1 + \gamma/\varepsilon)$. □

By Theorem 2.2 we see that apex graphs have $(1 + \varepsilon)$-spanners of tree weight $O(\varepsilon^{-2})$. More generally, graphs becoming planar (or bounded genus) after deleting some k vertices have $(1 + \varepsilon)$-spanners of tree weight $O((2 + 2/\varepsilon)^k)$; however, we suspect that this dependence on k should be polynomial.

We remark that by a result of Eppstein [4], $\mathcal{G}(\gamma)$ has the diameter-treewidth property, whereas $\mathcal{G}'_\varepsilon(\gamma)$ (for $\varepsilon > 0$ and large enough γ) does not.

6 Concluding Remarks

We would like to prove Conjecture 1.2; the most direct approach would be some variation of the proof of Theorem 3.3. Another approach is to apply the Robertson-Seymour characterization of graphs with no K_r-minor [7]. In particular we can handle bounded genus graphs and apex vertices, which are elements of that characterization. A remaining obstacle: suppose G has a bounded adhesion tree decomposition into pieces of bounded gap number, does G have bounded gap number?

There is also a computational issue: given G, can we compute $\mathrm{gap}(G)$ (or $\mathrm{gap}_\varepsilon(G)$, or $\mathrm{gap}'_0(G)$, or $\mathrm{gap}'_\varepsilon(G)$) efficiently? Given T we can find the best w (by linear programming), and given w we can find the best T (by a parametric search). We can iterate between these two steps until we reach a local maximum, but it is not clear whether the process terminates quickly, or whether it finds a global maximum.

We thank Mathias Schacht for some linear programming experiments.

References

[1] I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9(1):81–100, 1993. An early version appeared in SWAT'90, LNCS V. 447.

[2] S. Arora, M. Grigni, D. Karger, P. Klein, and A. Woloszyn. A polynomial-time approximation scheme for weighted planar graph TSP. In *Proceedings of the 9^{th} Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 33–41, 1998.

[3] R. Diestel. *Graph theory.* Springer-Verlag, New York, 1997.

[4] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27:275–291, 2000. Special issue on treewidth, graph minors, and algorithms.

[5] M. Grigni. Approximate TSP in graphs with forbidden minors. In *Automata, Languages and Programming: 27th International Colloquium*, volume 510 of *Lecture Notes in Computer Science*, pages 869–877. Springer-Verlag, July 2000.

[6] M. Grigni, E. Koutsoupias, and C. Papadimitriou. An approximation scheme for planar graph TSP. In *28^{th} Annual Symposium on Foundations of Computer Science*, pages 640–646. IEEE, Oct. 1995.

[7] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, Baltimore, 2001.

[8] C. H. Papadimitriou and M. Yannakakis. The Traveling Salesman Problem with distances one and two. *Mathematics of Operations Research*, 18:1–11, 1993.

[9] A. Thomason. An extremal function for contractions of graphs. *Math. Proc. Cambridge Philos. Soc.*, 95(2):261–265, 1984.