A PTAS for the Sparsest 2-Spanner Problem in 4-Connected Planar Triangulations

WILLIAM DUCKWORTH, *Department of Mathematics & Statistics, University of Melbourne, Parkville, Victoria 3052, Australia. E-mail: billy@ms.unimelb.edu.au*

NICHOLAS C WORMALD, *Department of Mathematics & Statistics, University of Melbourne, Parkville, Victoria 3052, Australia. E-mail: nick@ms.unimelb.edu.au*

MICHELE ZITO, *Department of Computer Science, University of Liverpool, Liverpool, L69 7ZF, United Kingdom. E-mail: michele@csc.liv.ac.uk*

ABSTRACT: A t-spanner of an undirected, unweighted graph G is a spanning subgraph $\mathcal{S}$ of G with the added property that for every pair of vertices in G, the distance between them in $\mathcal{S}$ is at most t times the distance between them in G. We are interested in finding a sparsest t-spanner. In the general setting, this problem is known to be NP-hard for all $t \geq 2$. For $t \geq 5$, the problem remains NP-hard for planar graphs, whereas for $t \in \{2, 3, 4\}$, the complexity of this problem on planar graphs is still unknown. In this paper we present a polynomial time approximation scheme for the problem of finding a sparsest 2-spanner of a 4-connected planar triangulation.

Keywords: Sparse, 2-Spanner, 4-Connected, Planar, Triangulation

1 Introduction

A t-spanner of an undirected, unweighted graph G is a spanning subgraph $\mathcal{S}$ of G with the added property that for every pair of vertices in G, the distance between them in $\mathcal{S}$ is at most t times the distance between them in G. We are interested in finding a sparsest t-spanner. This problem has many applications in areas as far afield as distributed computing, networks, computational geometry, robotics and biology [1, 12, 13]. We refer to the quantity t as the *dilation* of the spanner. The cardinality of the edge set of a spanner denotes its *size*.

Peleg and Ullman [13] introduced the concept of graph spanners as a means of constructing synchronisers of hypercubic networks. They showed that the d-dimensional hypercube has a 3-spanner with fewer than 7×2^d edges. Duckworth and Zito [8] improved this upper bound result to at most 4×2^d and gave the first non-trivial lower bound results for finding sparse hypercube 3-spanners. In recent years, there has been a great deal of research in this area and many complexity results are now known.

For general graphs, Peleg and Schăffer [12] showed that the problem of finding a sparsest 2-spanner (S2S) is NP-hard. Since then, Cai [4] extended this result to include all dilations greater than 2. Cai and Keil [5] gave a linear time algorithm for S2S in graphs with maximum degree $\Delta \leq 4$. They also showed that finding a sparsest t-spanner of a graph with $\Delta \geq 9$ is NP-hard for all $t \geq 2$. Brandes and Handke [3] showed that for spanner dilations greater than 4, the problem remains NP-hard on arbitrary planar graphs. For $t \in \{2, 3, 4\}$, the complexity of the problem of finding a sparsest t-spanner of a planar graph remains open.

One way of dealing with the NP-hardness of an optimisation problem is to relax the optimality requirement and look for the existence of polynomial time algorithms which guarantee solutions whose size is close to that of the optimum. In what follows we say that an optimisation problem Π is approximable with *(approximation) ratio* ρ if there is a polynomial time *approximation algorithm* A that, for each input x, returns a solution of size $A(x)$ with $A(x)\rho^{-1} \leq \mathrm{OPT}(x) \leq A(x)\rho$, where OPT$(x)$ denotes the size of an optimal solution. Kortsarz [10] showed that for arbitrary n-vertex graphs, S2S is NP-hard to approximate with approximation ratio $O(\log n)$.

In this paper we consider the approximability of finding a sparsest 2-spanner of a 4-connected planar triangulation (S2S(4CPT)). After giving a number of polynomial time reductions, we show how to exploit one of these reductions to devise a family of polynomial time approximation algorithms $\mathcal{A}_\epsilon$ having performance ratio $1 + \epsilon$, for every $\epsilon > 0$. Such a family is called a *polynomial time approximation scheme* (PTAS).

In Section 2 we introduce our graph theoretic notations and concepts. We also define a number of optimisation problems that are related to S2S. In Section 3 we present polynomial time computable reductions of the S2S problem to these related optimisation problems. Section 4 contains the main result of this paper. After reviewing a well established technique for devising a PTAS for a given NP-hard problem on planar graphs, we apply it to one of the problems under consideration and subsequently devise a PTAS for S2S(4CPT).

2 Preliminaries

The majority of our results rely upon the structural properties of 4-connected planar triangulations. In this section we give some basic definitions and remind the reader of a few well known properties of such graphs. For other basic graph theory definitions see (for example) [6].

A *4-connected planar triangulation* G is a maximally planar triangulated graph with the additional property that it does not contain any *separating triangles* i.e. cycles of length 3 which are not faces. It follows directly from the definition that such graphs on n vertices have $3n - 6$ edges, $2n - 4$ faces (all of which are triangles) and every edge belongs to precisely 2 triangles.

Given a planar graph G, we construct its *dual* $D(G)$ by representing each face of G as a vertex of $D(G)$ and connecting two vertices with an edge if and only if the two faces of G represented by these two vertices share an edge of G. It is immediate that if G is a 4-connected planar triangulation on n vertices, then $D(G)$ is a planar cubic graph on $2n-4$ vertices.

The *face-edge-incidence graph* of a planar graph G is a bipartite graph, $B(G) = (V_1, V_2, E)$, such that each edge of G is represented by a vertex $v_1 \in V_1$ and each face of G is represented by a vertex $v_2 \in V_2$. An edge of $B(G)$ connects v_1 to v_2 if and only if the edge of G represented by v_1 is part of the face of G represented by v_2. It is immediate that if G is a 4-connected planar triangulation, then $B(G)$ is a planar bipartite graph which is 2,3-regular *i.e.* every vertex in V_1 has degree 2 and every vertex in V_2 has degree 3. We refer to a graph with these properties as a $(2,3)$-*regular bipartite graph*.

For a graph G, its *line graph* $L(G)$ is constructed by representing each edge of G by a vertex in $L(G)$ and two vertices $u, v \in V(L(G))$ are connected by an edge if and only if the edges of G represented by u and v share a common end-point in G. It is immediate that the line graph of the face-edge incidence graph of a 4-connected planar triangulation is planar and cubic.

In Section 3 we reduce S2S(4CPT) to other well known graph theoretic optimisation problems. We therefore include the following definitions :

- Maximum Edge Star Packing (MESP) [14] : A star packing (SP) of a graph H is a subgraph $\mathcal{F}$ of H such that each component is a star (*i.e.* is isomorphic to $K_{1,r}$ for some r; we refer to $K_{1,r}$ as an r-*star*). The problem MESP is then to find a SP with the maximum number of edges.
- Maximum Induced Matching (MIM) [14] : An induced matching (IM) of a graph H is a vertex disjoint set of edges $\mathcal{M} \subseteq E(H)$ such that no two edges in $\mathcal{M}$ are joined by an edge in $E(H) \setminus \mathcal{M}$. The problem MIM is then to find an IM with the maximum number of edges.
- Maximum k-Independent Set (MkIS) [11] : For $k \geq 1$, a k-independent set (kIS) of a graph H is a set of vertices $I \subseteq V(H)$ such that for every pair of vertices in I, the distance between them is at least $k+1$. The problem MkIS is then to find a kIS with the maximum number of vertices.

3 Reductions

In this section we show that the problem S2S(4CPT) is polynomial time reducible to both MESP and MIM. We also give a number of consequences of these reductions. Finally we note that the problem of finding a maximum induced matching of a graph is reducible to finding a maximum 2-independent set of its line graph.

3.1 Maximum Edge Star Packing

Throughout this section we will use G to represent a 4-connected planar triangulation and $D(G)$ to represent its dual graph. Firstly, we reduce the problem S2S(4CPT) to MESP.

LEMMA 3.1
A graph S is a sparsest 2-spanner of a 4-connected planar triangulation G if and only

if the edges of the dual of G corresponding to the edges in $E(G) \setminus E(S)$ belong to a maximum star packing of $D(G)$.

PROOF. We note that the proof of this result is tailored specifically to the class of graphs under consideration since it relies heavily on the fact that there is a one-to-one correspondence between edges of G and edges of $D(G)$. For an edge $e \in E(G)$, we denote the corresponding edge in $E(D(G))$ by e'. In order to prove the lemma we will associate a star packing of $D(G)$ having $|E(G)| - |\mathcal{S}|$ edges with any given 2-spanner $\mathcal{S}$ of G and a 2-spanner of G having $|E(G)| - |\mathcal{F}|$ edges with any given star packing $\mathcal{F}$ of $D(G)$.

Let $\mathcal{S}$ be a 2-spanner of G. The edges in $E(D(G))$ corresponding to edges in $E(G)\setminus E(\mathcal{S})$ form a star packing $\mathcal{F}$. Since any edge $e \in E(G)\setminus E(\mathcal{S})$ must be spanned by two other edges $f, g \in E(\mathcal{S})$, any subset of $E(\mathcal{F})$ may not form a path of length greater than 2. Suppose, by contradiction, the edge e' represents an internal edge of a path of length greater than 2 in $\mathcal{F}$. The edges $f', g' \in E(D(G))$ corresponding to the two edges spanning $e \in E(G) \setminus E(\mathcal{S})$ are both incident with the same end-point v of the edge e'. Therefore the degree of v in $D(G)$ is at least 4, contradicting the fact that $D(G)$ is a cubic graph. Hence the components of $\mathcal{F}$ are *r-stars* for $r \in \{1, 2, 3\}$.

Conversely let $\mathcal{F}$ be a star-packing of $D(G)$ and define $\mathcal{S}$ to be the graph obtained from G by removing all edges in $E(G)$ associated with edges in $\mathcal{F}$. Each deleted edge e of G is spanned by the edges in G corresponding to the two non-star edges incident with the leaf of the star containing e'. ∎

The reduction described above has the following useful consequence.

LEMMA 3.2
Any 2-spanner of an n-vertex 4-connected planar triangulation contains at least $\frac{3}{2}(n-2)$ edges.

PROOF. The greatest possible number of edges that can be deleted from G to form a 2-spanner occurs when all stars in $D(G)$ are 3-stars. In this instance, since every star is vertex disjoint and covers 4 vertices, the greatest number of vertex disjoint 3-stars, is given by

$$\frac{|V(D(G))|}{4} = \frac{n-2}{2},$$

and since each star has 3 edges, this gives a lower bound on the size of an optimum 2-spanner, $|\mathcal{S}_{OPT}|$, of G

$$|\mathcal{S}_{OPT}| \geq 3\,(n-2) - \frac{3(n-2)}{2} = \frac{3(n-2)}{2}.$$

∎

We [7] combined a greedy algorithm for MESP with the reduction in Lemma 3.1 and the bound in Lemma 3.2 to show that S2S(4CPT) is approximable within $\frac{5}{4}$.

3.2 Induced Matchings

Next we show that S2S(4CPT) is polynomial time reducible to finding a MIM of the face-edge incidence graph associated with the input graph. In what follows, let G represent a 4-connected planar triangulation and let $B(G) = (V_1, V_2, E)$ represent the face-edge-incidence graph of G.

THEOREM 3.3
The problem of finding a sparsest 2-spanner $\mathcal{S}$ of a 4-connected planar triangulation G is polynomial time reducible to finding a maximum induced matching in the face-edge-incidence graph of G.

PROOF. Given any 2-spanner $\mathcal{S}$ of G, we show that this represents an induced matching $\mathcal{M}$ of $B(G)$. Edges in $E(G) \setminus E(\mathcal{S})$ represent edges in $\mathcal{M}$ and more specifically, a matching edge has its endpoints at the vertex representing the deleted edge and the vertex representing the triangle whose remaining edges span the deleted edge.

A matching edge (v_1, v_2) of $B(G)$ represents a deleted edge of G as described above. Since v_2 represents the triangle whose remaining edges span the deleted edge, there cannot be any other matching edges incident with any other neighbour of v_2. This would indicate that the edges represented by these vertices would be deleted in G contradicting the fact that they span the missing edge. The edge represented by v_1 is part of another triangle represented by v_2'. No other matching edge may be incident with v_2' since this would indicate that (v_1, v_2) was not present in the matching contradicting the fact that it represents a deleted edge of G. The resulting matching is therefore induced.

Given an induced matching of $B(G)$, this represents a 2-spanner of G. A matching edge (v_1, v_2) indicates that the edge e of G represented by v_1 has been deleted. Since the matching is induced, no matching edge may be present that is incident with the vertices of V_1 that represent the spanning edges of e. ■

The reduction from the S2S problem to the MIM problem holds in a more general setting than described here [15]. For our purposes we need only consider the special case when the input graph is a 4-connected planar triangulation. Using this reduction, and analysing a simple greedy heuristic that finds a large induced matching in $B(G)$, we [7] showed that S2S(4CPT) is approximable with ratio $\frac{6}{5}$.

3.3 2-Independent Sets

We now establish a relationship between MIM and M2IS. Combining this result with the one described above will enable us, in the next section, to prove our algorithmic result.

LEMMA 3.4
An induced matching $\mathcal{M}$ of a graph G is maximal if and only if the vertices of $L(G)$ (the line graph of G) corresponding to the the edges of $\mathcal{M}$ are a maximum 2-independent set of $L(G)$.

PROOF. Pairs of vertices at distance at least 3 in $L(G)$ correspond to independent edges not joined by any other edge in G. This implies that the induced matchings of G are in one-to-one correspondence with the 2-independent sets in $L(G)$. ∎

4 Approximation Schemes

In this section we prove that there exists a PTAS for the S2S(4CPT) problem. Our method uses the results in Section 3 together with a technique developed by Baker [2] who proved the existence of a PTAS for a number of graph theoretic problems on planar graphs. In all cases the original planar graph is decomposed into subgraphs with simpler combinatorial structure. Baker describes in detail a PTAS for finding a maximum independent set of a planar graph and then shows how, with minor modifications, the same technique may be used to devise a PTAS for several other graph theoretic problems on planar graphs. We give a brief overview of this technique before showing the minor modifications that are necessary in order to prove that S2S(4CPT) admits a PTAS. Before describing the algorithmic details, we provide the reader with the relevant definitions.

A graph G is *outerplanar* if it can be drawn in the plane in such a way that no two edges cross and all its vertices belong to the external face. An edge of G will be *external* (resp. *internal*) if it lies (resp. does not lie) on the external face. Cycles with any number of non-crossing chords are examples of outerplanar graphs. An *h-outerplanar* graph may be defined as follows: A 1-outerplanar graph is simply an outerplanar graph as described above. Given a planar embedding $\mathcal{E}$ of a graph G then G is said to be h-outerplanar if after the removal of the vertices in the outer face of $\mathcal{E}$ along with their incident edges, every connected component is at most $(h-1)$-outerplanar and at least one such component is $(h-1)$-outerplanar.

Baker's technique relies on the fact that, for the problems under consideration, there exists a linear time algorithm to solve these problems optimally when the input is restricted to outerplanar graphs. This algorithm, which is based on dynamic programming, can be generalised to show that the problems under consideration are solvable optimally in time $2^{O(h)}n$ on h-outerplanar graphs. Once this algorithm has been established, the PTAS for planar graphs is obtained by decomposing the planar graph into a number h-outerplanar subgraphs (where h can be chosen to be $O(\log n)$), solving the problem exactly on these subgraphs and showing that the combination of these solutions gives a solution for the original problem that is very close to the optimal one. Baker's detailed description of the algorithm for solving optimally the maximum independent set problem (or M1IS in our notation) for h-outerplanar graphs may be applied with minor modifications to MkIS (for k constant). Details of the modifications are given in the next two lemmas. In fact, Baker [2] states that the dynamic programming technique used will work in general for problems that involve local conditions on nodes and edges.

LEMMA 4.1
Let h and k be positive integers, with k being a fixed constant. Given an h-outerplanar embedding of an h-outerplanar graph G with n vertices, a maximum cardinality k-independent set of G can be obtained in time $O((k+1)^h n)$.

PROOF. We describe in detail the algorithm for $h = 1$ and sketch the proof for $h > 1$. Let G be an outerplanar graph in which any bridge has been replaced by a pair of parallel edges. If G is 2-connected, let $T = T(G)$ be the tree having one leaf for every external edge in G, and one internal vertex for every face of G. Two vertices u and v in T are connected by an edge if and only if

- u and v correspond to two faces sharing an edge or
- u and v represent a face and an external edge of this face.

If G is not 2-connected the definition given so far generates a forest. In this case T is obtained from this forest by repeatedly connecting, with an edge, distinct pairs of vertices corresponding to faces that meet at a cutpoint (with the only constraint that the edge should not create a cycle) until all the trees in the forest have been joined. In all cases, a root of T and its so called *leftmost child* can then be chosen arbitrarily. This choice induces an ordering on the vertices of T. In particular the ordering on the leaves of T corresponds to traversing G starting from the leftmost vertex belonging to the component corresponding to the chosen leftmost child of the root of T and then following the external edges of G in an anticlockwise directed walk. Each leaf in T is labelled with the oriented external edge it represents. Each internal vertex v of T is labelled by the pair (i, j) if i and j are (respectively) the first and the last nodes in the labels of its children. The vertex v *represents* the subgraph of G induced by i, j and all vertices in the directed walk from i to j.

A k-independent set is computed as follows. For each vertex v in T with labelling (i, j), associate a table giving the sizes of a k-independent set I, for the subgraph represented by the subtree rooted at v, such that I is maximal subject to specified distances from i and j to the vertices in I. The $O((k + 1)^2)$ values in the table are computed using dynamic programming, scanning all the vertices in the subtree rooted at v and eventually merging the tables associated to the children of v to form the entries of the table for v. In particular, the table for a leaf l in T contains entries corresponding to the (illegal) case in which both endpoints of the edge corresponding to l are chosen to be in the k-independent set, the (legal) cases in which only one endpoint is chosen or neither of them is chosen and those (illegal) cases in which neither of the endpoints is chosen but there is a node in the k-independent set for the given edge at a positive distance at most $k - 1$ from at least one of the endpoints. The procedure that combines the tables for single edges to compute the k-independent set of the given outerplanar graph can be described using exactly the same pseudo-code as in [2, Fig. 7]. It is clear that the only difference between our algorithm and the one for independent set [2] is in the way the values in the tables corresponding to face vertices in T are computed as a functions of previously computed values (functions $merge$ and $adjust$ in Baker's code). In particular, we need to keep track of whether i or j are at distance $d \in \{0, \ldots, k\}$ from a vertex in the k-independent set.

In [2], Baker also gives full details on how to extend the algorithm described above to solve the maximum independent set problem (or M1IS in our notation) optimally on h-outerplanar graphs. The algorithm is similar to the one for outerplanar graphs in that it uses a particular combinatorial structure (similar to the tree T above) to guide the construction of a number of tables that keep the information that is needed in order to compute the desired optimum. However the single tree T is replaced by a family of trees, one for each level in the h-outerplanar graph. Furthermore, the simple idea of recursing on the vertices of T, computing the tables for the leaves of T and then merging them to compute the tables for the internal vertices of T must be made more complex in order to achieve the desired aim. The same approach can be used for k-independent sets, the only difference being in the implementation of the procedures $adjust$, $merge$, $create$ and $extend$ defined in [2, p. 171-172]. ■

LEMMA 4.2
M2IS admits a PTAS if the input graph is planar.

PROOF. The vertices of the input planar graph G are arranged into *layers*. The vertices of the external face of G are in layer 1. The vertices in layer i are defined inductively as external vertices of the graph obtained by deleting the vertices in layers $1 \dots i-1$. Denote the set of vertices in layer i by V_i. The graph induced by the vertices in h consecutive layers is an h-outerplanar graph. Let $\overline{U}$ be a 2-independent set of maximum cardinality in G and let h be an even constant. Since the sets

$$\{i \in \mathbb{N} : i \equiv 2r-1 \bmod h+2\} \cup \{i+1 \in \mathbb{N} : i \equiv 2r-1 \bmod h+2\} \quad \text{and}$$
$$1 \leq r \leq h/2+1$$

partition the set $\mathbb{N}$, we have

$$\sum_{r=1}^{h/2+1} \sum_{i \equiv 2r-1 \bmod h+2} |(V_i \cup V_{i+1}) \cap \overline{U}| = |\overline{U}|.$$

Therefore there must exist an r with $1 \leq r \leq h/2+1$ such that

$$\sum_{i \equiv 2r-1 \bmod h+2} |(V_i \cup V_{i+1}) \cap \overline{U}| \leq \frac{|\overline{U}|}{h/2+1} = \frac{2|\overline{U}|}{h+2}.$$

Let W_i denote the set of vertices belonging to layers j, where $\max\{0, i-h\} \leq j < i$. The graph induced by W_i is h-outerplanar. We use the algorithm described in Lemma 4.1 to solve optimally M2IS in W_i. Let DP_i denote an optimum 2-independent set in this graph. The polynomial time approximation scheme returns the set

$$\bigcup_{i \equiv 2r-1 \bmod h+2} DP_i.$$

As $|DP_i| \geq |W_i \cap \overline{U}|$ for all i, the inequality above implies that

$$\begin{aligned}
\sum_i |DP_i| &\geq \sum_i |W_i \cap \overline{U}| \\
&= |\overline{U}| - \sum_{i \equiv 2r-1 \bmod h+2} |(V_i \cup V_{i+1}) \cap \overline{U}| \\
&\geq |\overline{U}| - \tfrac{2|\overline{U}|}{h+2} = \tfrac{h}{h+2} \cdot |\overline{U}|.
\end{aligned}$$

Therefore
$$\frac{|\overline{U}|}{\displaystyle\sum_{i\equiv 2r-1 \bmod h+2} |DP_i|} \leq \frac{h+2}{h} = 1 + \frac{2}{h}.$$
∎

COROLLARY 4.3
MIM admits a PTAS for $(2,3)$-regular bipartite graphs.

PROOF. Given a $(2,3)$-regular bipartite graph G, construct its line graph $L(G)$. Using Baker's approach we find a large M2IS in $L(G)$. The edges of G corresponding to the vertices in the 2-independent set form an induced matching whose size is at least $1-(h+2)^{-1}$ times the size of an optimum induced matching in G. ∎

THEOREM 4.4
S2S(4CPT) admits a PTAS.

PROOF. The result of Lemma 3.2 gave us a lower bound on the size of a sparsest 2-spanner of a 4-connected planar triangulation G, $\mathcal{S}_{OPT}(G)$, and we have
$$\mathcal{S}_{OPT}(G) \geq \frac{|E(G)|}{2}.$$

For an upper bound, we construct the face-edge incidence graph of G, $B(G)$. From the results above we can find an induced matching $\mathcal{M}$ in $B(G)$ with
$$|\mathcal{M}| \geq \frac{\mathcal{M}_{OPT}(B(G))}{1+\epsilon}$$
where $\mathcal{M}_{OPT}(B(G))$ is the maximum size of an induced matching of $B(G)$. Using the reduction in Theorem 3.3 this matching represents a 2-spanner $\mathcal{S}$ of size $|\mathcal{S}|$ and we have
$$\begin{aligned}
|\mathcal{S}| &\leq |E(G)| - \frac{\mathcal{M}_{OPT}(B(G))}{1+\epsilon} \\
&= \frac{|E(G)| - \mathcal{M}_{OPT}(B(G)) + \epsilon|E(G)|}{1+\epsilon} \\
&= \frac{\mathcal{S}_{OPT}(G) + \epsilon|E(G)|}{1+\epsilon} \\
&\leq \frac{\mathcal{S}_{OPT}(G) + 2\epsilon\mathcal{S}_{OPT}(G)}{1+\epsilon}.
\end{aligned}$$

This implies
$$\frac{|S|}{\mathcal{S}_{OPT}(G)} \leq 1 + \frac{\epsilon}{1+\epsilon}.$$
∎

5 Conclusions and Open Problems

In this paper we have considered the approximability of the problem of finding a sparsest 2-spanner of a 4-connected planar triangulation. We have shown that the problem of finding a sparsest 2-spanner of a 4-connected planar triangulation is polynomial time reducible to two other known graph theoretic optimisation problems namely, maximum edge star packing in planar cubic graphs and maximum induced matching in 2,3-regular bipartite planar graphs. By means of these reductions it is possible to prove a couple of non-trivial approximation results for this problem [7]. The possibility of an improved approximation heuristic was left open in [7]. By exploiting one of the aforementioned reductions, we have shown that the problem of finding a sparsest 2-spanner of a 4-connected planar triangulation admits a polynomial time approximation scheme (PTAS). For NP-hard optimisation problems, a PTAS is one of the best types of algorithm one can hope for [9]. Unfortunately we do not know whether S2S(4CPT) is NP-hard. Therefore the existence of a PTAS leads to the obvious open question of whether the problem can actually be solved optimally in polynomial time. Further unanswered questions concern the existence of similar schemes (or even polynomial time exact algorithms) for the problem of finding a sparsest t-spanner of a 4-connected planar triangulation or indeed, a general planar graph.

ACKNOWLEDGEMENT: The authors acknowledge the anonymous referees for their helpful suggestions.

References

[1] I. Althőfer, G. Das, D. Dobkin and D. Joseph. Generating Sparse Spanners of Weighted Graphs. *Discrete Computational Geometry*, 9:81–100, 1993.

[2] B. Baker. Approximation Algorithms for NP-Complete Problems on Planar Graphs. *Journal of the ACM*, 41(1):153–180, 1994.

[3] U. Brandes and D. Handke. NP-Completeness Results for Minimum Planar Spanners. In *Proceedings of WG'97*, *Lecture Notes in Computer Science*, 1335:85–99, 1997.

[4] L. Cai. NP-Completeness of Minimum Spanner Problems. *Discr. Appl. Math.*, 48:187–194, 1994.

[5] L. Cai and M. Keil. Spanners in Graphs of Bounded Degree. *Networks*, 24:233–249, 1994.

[6] R. Diestel. *Graph Theory*. Springer-Verlag, New York, 1997.

[7] W. Duckworth, N.C. Wormald and M. Zito. Approximation Algorithms for Finding Sparse 2-Spanners of 4-Connected Planar Triangulations. In R.J. Simpson and R. Raman (editors) *Proceedings of AWOCA'99*, pp. 63–75, Curtin University, Perth, Western Australia, 1999. `http:\\www.computing.edu.au\~awoca99`.

[8] W. Duckworth and M. Zito. Sparse Hypercube 3-Spanners. *Discr. Appl. Math.*, 103(1–3):289–295, 2000.

[9] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Co., 1979.

[10] G. Kortsarz. On the Hardness of Approximating Spanners. In *Proceedings of APPROX'98*, *Lecture Notes in Computer Science*, 1444:135–146, New York/Berlin, 1998. Springer-Verlag.

[11] A. Meir and J.W. Moon. Relations Between Packing and Covering Numbers of a Tree. *Pacific Journal of Mathematics*, 61(1):225–233, 1975.

[12] D. Peleg and A.A. Schäffer. Graph Spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

[13] D. Peleg and J.D. Ullman. An Optimal Synchroniser for the Hypercube. *SIAM Journal on Computing*, 18(4):740–747, 1989.

[14] L.J. Stockmeyer and V.V. Vazirani. NP-Completeness of Some Generalisations of the Maximum Matching Problem. *Information Processing Letters*, 15(1):14–19, 1982.

[15] M. Zito. *Randomised Techniques in Combinatorial Algorithmics*. PhD thesis, Department of Computer Science, University of Warwick, 1999.

Received 31st May, 2000