Approximation Algorithms for Finding Sparse 2-Spanners of 4-Connected Planar Triangulations

W. Duckworth[1], N.C. Wormald[1] and M. Zito[2]

[1] Department of Mathematics and Statistics, University of Melbourne, Parkville, Victoria, 3052, Australia
[2] Department of Computer Science, University of Liverpool, Peach Street, Liverpool, L69 7ZF, United Kingdom

Abstract. A *t-spanner* of an undirected, unweighted graph G is a spanning subgraph $\mathcal{S}$ 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-Complete for all $t \geq 2$. For $t \geq 5$, the problem remains NP-Complete 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 approximation algorithms to approximate a sparsest 2-spanner of a 4-connected planar triangulation as a step towards solving the complexity of this problem in the more general planar setting.

1 Introduction

A *t-spanner* of an undirected, unweighted graph G is a spanning subgraph $\mathcal{S}$ 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. The quantity t is said to be the *dilation* or *stretch factor* of the spanner. This problem has many applications in areas as far afield as distributed computing, networks, computational geometry, robotics and biology [2, 16, 17]. We refer to the *size* of a spanner by the cardinality of its edge set.

Peleg and Ullman [17] 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 [12] 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 Schäffer [16] showed that the problem of finding a sparsest 2-spanner (S2S) is NP-Complete. Since then, Cai [5] extended this result to include all dilations greater than 2. Cai and Keil [6] gave a linear time algorithm for finding a S2S of a graph whose maximum degree is at most four. They also showed that finding a sparsest t-spanner of a graph with maximum degree at most nine is NP-Complete for all $t \geq 2$. Brandes and Handke [4] showed that for spanner dilations greater than 4, the problem remains NP-Complete on unbounded degree 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.

In this paper we consider the complexity of finding a S2S of a 4-connected planar triangulation (S2S(4CPT)) and we present approximation algorithms to approximate S2S(4CPT).

For a 2-spanner of a graph to have fewer edges than the graph itself, the graph must contain at least one triangle. If all triangles in the graph are edge disjoint, then the problem becomes trivial. Considering 4-connected planar triangulations therefore becomes viable and the complexity of S2S(4CPT) remains open.

We present reductions of this problem to other known graph theoretical problems and for each reduction we give an approximation algorithm which approximates S2S(4CPT). In the following section we introduce our notation and remind the reader of some basic graph theoretic definitions. In Section 3 we show that S2S(4CPT) is polynomial time reducible to finding a maximum edge star packing of a planar cubic graph. We present an algorithm that approximates such a star packing which in turn approximates S2S(4CPT). In Section 4 we show that S2S(4CPT) is polynomial time reducible to finding a maximum induced matching of a 2,3-regular bipartite planar graph. Analysing an algorithm that approximates this problem yields a further improvement to the result in Section 3.

In the general setting, the problems that we reduce to are known to be NP-Complete. These problems remain NP-Complete under various restrictions but it is still not known whether they remain NP-Complete when restricted to the (very) specific class of graphs given by our reductions. In the final section we conclude with a summary of our results and address the open problems of complexity.

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 [9].

A 4-connected planar triangulation G on n vertices has $3n$-6 edges and $2n$-4 faces, all of which are triangles. A well known property of such graphs is that they do not contain any *separating triangles* i.e. cycles of length 3 which are not faces. Every edge in a 4-connected planar triangulation belongs to precisely 2 triangles.

We construct the dual D of an n-vertex 4-connected planar triangulation G by representing each face of G as a vertex of D and connecting two vertices of D with an edge if and only if the two triangles of G represented by these two vertices share an edge of G. Note that if G has n vertices then $D(G)$ will have $2n$-4 vertices and $3n$-6 edges. It is immediate that D is a planar cubic graph.

The structural properties of such cubic graphs have been extensively studied and many facts about their structure are now known.

A k-factor of a graph is a spanning subgraph whose components are k-regular. Another well known property of the dual graph of a 4-connected planar triangulation is that it has a 1-factor. An immediate corollary from this is that its edges have exact decomposition into a 1-factor and a 2-factor. Czumaj and Gibbons [8] showed that the dual of a 4-connected planar triangulation is 3-edge-colourable. This result implies that all cycles of a 2-factor of the dual of a 4-connected planar triangulation have even length. In Section 3 we will utilise these facts to approximate S2S(4CPT).

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

Maximum Edge Star Packing (MESP) [14, 18] :

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). The problem MESP is then to find a star packing with the maximum number of edges.

Maximum Induced Matching (MIM) [7, 18] :

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.

3 Maximum Edge Star Packing

In this section we show that S2S(4CPT) is polynomial time reducible to finding a MESP of the dual of the 4-connected planar triangulation. We then present an algorithm which approximates the size of such a star packing. Analysis of this algorithm gives upper and lower bounds on the size of an optimal 2-spanner of a 4-connected planar triangulation. 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(G) to MESP($D(G)$).

Lemma 1. *The problem of finding a sparsest 2-spanner of a 4-connected planar triangulation is polynomial time reducible to finding a maximum edge star packing of its dual graph.*

Proof. We note that the proof of this lemma is tailored specifically to the class of graphs under consideration since the proof of the lemma relies heavily on the fact that there is a one-to-one correspondence between an edge of G and an edge of $D(G)$. For an edge $e \in E(G)$, we denote the corresponding edge of $E(D(G))$ by e'. In order to prove the lemma, it will suffice to show that given any 2-spanner of G, this represents a star packing of $D(G)$, and similarly given any star packing of $D(G)$, this represents a 2-spanner of G.

We denote a 2-spanner of G by $\mathcal{S}$ and a star packing of $D(G)$ by $\mathcal{F}$. Edges in $E(G) \setminus E(\mathcal{S})$ will represent edges of $\mathcal{F}$. Since any edge $e \in E(G) \setminus E(\mathcal{S})$ must be spanned by 2 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 $D(G)$. The spanning edges for e (represented in $D(G)$ by f' and g') are both incident with the same endpoint of the edge e'. Since e' is an internal path edge, e cannot be spanned in G. For any 2-spanner of G, the corresponding deleted components of $D(G)$ may therefore only be *stars*. The term star in our case here refers to single edges (1-*stars*), paths of length 2 (2-*stars*) and three edges sharing a common vertex (3-*stars*).

Denoting the size of a 2-spanner of G by $|\mathcal{S}|$ and the number of edges in an edge star packing of $D(G)$ by $|E(\mathcal{F})|$ we have

$$|\mathcal{S}| \geq |E(G)| - |E(\mathcal{F})|.$$

Given a forest of vertex disjoint stars in $D(G)$, this represents a 2-spanner of G. Each deleted edge of G, represented by an edge of a star, is spanned by the 2 edges incident with the leaf of the star and we have

$$|\mathcal{S}| \leq |E(G)| - |E(\mathcal{F})|.$$

□

Using this reduction, we now prove the following lemma.

Lemma 2. *A sparsest 2-spanner of any n-vertex 4-connected planar triangulation contains at least 3(n-2)/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, $|3^*|$, is given by

$$|3^*| \leq \frac{|V(D(G))|}{4} = \frac{n-2}{2},$$

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 proceed by presenting an algorithm to find a packing of stars in $D(G)$ which in turn will provide an upper bound on the size of an optimal 2-spanner of G. This, combined with the result of Lemma 2, provides a means of approximating S2S(4CPT).

Lemma 3. *A sparsest 2-spanner of any n-vertex 4-connected planar triangulation has at most 15(n-2)/8 edges.*

Proof. From results in [8] and corollaries given in Section 2, we know that the edge set of the dual of a 4-connected planar triangulation has exact decomposition into a 1-factor and a 2-factor where all cycles in the 2-factor have even length.

Since the 4-connected planar triangulation contains no multiple edges, the 2-factor of the dual graph will contain no cycles of length 2 and therefore all cycles will have even length, l, where $4 \leq l \leq 2n-4$. We construct our star packing in $D(G)$ by adding as many 2-stars and 1-stars as we can in each of the cycles of the 2-factor. We then optimise the result (if possible) by adding edges of the 1-factor. Edges of the star packing in $D(G)$ correspond to edges deleted from G to form a 2-spanner. We use the following algorithm :

```
\*   Algorithm A   *\

For each cycle of the dual graph
do {
    Starting at an arbitrary edge in the cycle and proceeding in one direction
    While ( there are still at least 3 edges remaining )
    do {
        add 2 edges to the star packing
        skip one edge
       }
    if ( there are 2 unchecked edges )  add the first to the star packing
   }
```

At the end of the above algorithm, the star packing will have a number of 2-stars from each cycle of the 2-factor and in some cycles a 1-star as well. Asymptotically (in terms of the shortest cycle), the number of edges in the star packing is $4(n\text{-}2)/3$ since there are $2(n\text{-}2)$ edges in the 2-factor. This cannot be guaranteed, however, since there may be many cycles which have short lengths. In particular, for cycles of length 4, the algorithm will only add half the edges to the star packing. For cycles of length greater than 4, we can only guarantee to add $5/8$ of the edges since this occurs when we have cycles of length 8.

In order to compensate for this poor performance of the algorithm, we observe the following fact. Given two cycles of length 4 of the 2-factor connected by an edge of the 1-factor, we can improve the performance of the algorithm by positioning the star packing edges in each cycle in a beneficial way. In Figure 3, we position the star packing edges in such a way that we can add the edge connecting them. Edges of the star packing are indicated by heavier lines. This, of course, cannot be guaranteed for all cycles and we must therefore bound the numbers of these occurrences.

In both of the following instances, we assume that the decomposition gives us a number of cycles of length 4 and we denote this number by x. We must guarantee a certain amount of connectivity between the cycles of length four in order to perform the above improvement. We note here that if the number of

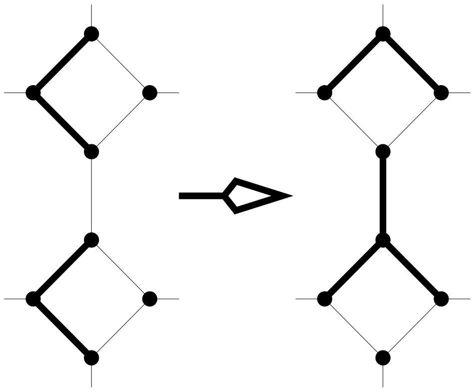

Fig. 1. Adding an edge of the 1-factor to the star packing.

vertices in the set of cycles of length four is strictly greater than that of the other cycles then, by the pigeon-hole principle, there must exist edges of the 1-factor connecting cycles of length 4. In this case we bound the number of independent edges connecting 2 distinct cycles and in the other case we calculate the bound given by the algorithm. Given that there are x cycles of length 4, we need the case when $4x > 2(n-2) - 4x$ to ensure we can perform some optimisation. This is when $x > (n-2)/4$.

Case 1 : $x \leq (n-2)/4$.

Here we cannot guarantee to use the above improvement to the algorithm. We simply calculate the number of edges in the star packing and bound the size of the constructed 2-spanner. Using X to be the number of edges in 4-cycles, Y to be the number of edges of the 2-factor other than those in X and $|E(\mathcal{F})|$ to be the number of star packing edges, we have :

$$|E(\mathcal{F})| \geq \frac{1}{2} \times X + \frac{5}{8} \times Y,$$

$$|E(\mathcal{F})| \geq \frac{n-2}{2} + \frac{5(n-2)}{8} = \frac{9(n-2)}{8}.$$

Denoting the number of edges in the spanner by $|\mathcal{S}|$, we have

$$|\mathcal{S}| \leq 3(n-2) - \frac{9(n-2)}{8} = \frac{15(n-2)}{8}.$$

Case 2 : $x > (n-2)/4$.

In this case we construct the following graph H. Each vertex $v \in V(H)$ represents a cycle of length 4 in the 2-factor of $D(G)$ and an edge $e = (v_1, v_2) \in E(H)$ is present if and only if there is a 1-factor edge of G connecting the two 4-cycles of G represented by v_1 and v_2. H is therefore a multi-graph with maximum degree 4. $|E(H)|$ is bounded by

$$|E(H)| \geq 4x - n + 2.$$

A matching in H gives us the required property to perform the optimisation described above. We recall from [15], the following lemma.

Lemma 4. *Let M be a multi-graph with m edges and maximum degree at most four. Let x be the number of loops in components with only one vertex, and q the number of loops incident with vertices of degree 3. Then M has a matching of size at least (2m-2x-q)/12.*

Since H does not contain any loops, it follows that H has a matching of size at least $m/6$ (where m represents the cardinality of the edge set of the multi-graph). We calculate the number of star packing edges and as before we bound the size of the spanner. Using $|M|$ to denote the number of edges in a matching of H, we have :

$$|E(\mathcal{F})| \geq \frac{1}{2} \times X + \frac{5}{8} \times Y + |M|,$$

$$|E(\mathcal{F})| \geq 2x + \frac{5(n-2-2x)}{4} + \frac{4x-n+2}{6}.$$

Substituting $x \geq (n+2)/4$ we have :

$$|E(\mathcal{F})| > \frac{9(n-2)}{8}.$$

Showing that the bound on the number of edges in the spanner given by case 1 is our upper bound. □

The above results allow us to approximate S2S(4CPT) and we have the following theorem.

Theorem 1. *Given an n-vertex 4-connected planar triangulation G, Algorithm A returns a 2-spanner with no more than 5/4 times the number of edges of a sparsest 2-spanner of G.*

Proof. Lemma 2 gives us a lower bound on the size of a sparsest 2-spanner for this class of graphs. Lemma 3 gives us an upper bound on the size of any 2-spanner constructed by our algorithm. Denoting the number of edges in a sparsest 2-spanner of G by $|\mathcal{S}|$ and the number of edges in the 2-spanner returned by Algorithm A as $|A|$, we have :

$$\frac{|A|}{|\mathcal{S}|} \leq \frac{\frac{15(n-2)}{8}}{\frac{3(n-2)}{2}} = \frac{5}{4}.$$

□

Stockmeyer and Vazirani [18] showed that MESP is NP-Complete even when restricted to cubic graphs. Their result is immediate from their reduction of MESP to minimum dominating set. For an n-vertex graph H, the size of a MESP is at most $n - |MDS(H)|$, where $MDS(H)$ represents a minimum dominating set of H.

Known results for finding minimum dominating sets of cubic graphs [13] or minimum *independent* dominating sets of cubic graphs [1] do not yield a further improvement to the upper bound on the size of the 2-spanner constructed in this section. More recent work by Duckworth and Wormald [11], however, suggests an improvement to the approximability of minimum independent dominating sets of cubic graphs which would result in showing that S2S(4CPT) is approximable within 40/33. We do not include these results here, however, since in this paper we are primarily concerned with the open problems of complexity of S2S(4CPT). Also, the results of the next section achieve a further improvement to the approximability to S2S(4CPT).

4 Maximum Induced Matchings

In this section we show that S2S(4CPT) is polynomial time reducible to finding a MIM of a planar 2,3-regular bipartite graph. We then describe a greedy algorithm for finding a MIM in such graphs. Analysing this greedy algorithm using linear programming, similar to the analysis in [11], provides us with a lower bound on the size of a MIM in such graphs. Using this result we are able to approximate S2S(4CPT). Throughout this section we will use G to represent a 4-connected planar triangulation and $B(G) = (V_1, V_2, E)$ to represent a planar 2,3-regular bipartite graph associated with G.

In order to construct $B(G)$ from G, we represent each edge of G by a vertex $v_1 \in V_1$ of $B(G)$ and each triangle of G by a vertex $v_2 \in V_2$ of $B(G)$. 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 triangle of G represented by v_2. The constructed graph may be described as the face-edge incidence graph. From the properties of 4-connected planar triangulations described in Section 2 we can see that the resulting bipartite graph is planar and 2,3-regular with every vertex of V_1 having degree 2 and every vertex of V_2 having degree 3. Furthermore, if G has n vertices, $|V_1| = 3(n-2)$ and $|V_2| = 2(n-2)$. The reduction of S2S to MIM of a bipartite graph is more general than described here and can be found in [19]. For our purposes we need only consider the special case when the input graph is a 4-connected planar triangulation. We now reduce S2S(G) to finding a MIM in $B(G)$.

Lemma 5. *The problem of finding a minimum 2-spanner of a 4-connected planar triangulation is polynomial time reducible to finding a maximum induced matching of a planar 2,3-regular bipartite graph.*

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 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 2-separated *i.e.* no two matching edges are connected by an edge of $B(G)$.

Representing the number of edges in a 2-spanner of G by $|\mathcal{S}|$ and the number of edges in a MIM of $B(G)$ by $|\mathcal{M}|$, we have

$$|\mathcal{S}| \leq |E(G)| - |\mathcal{M}|.$$

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 2-separated, no matching edge may be present that is incident with the vertices of V_1 that represent the spanning edges of e and we have

$$|\mathcal{S}| \geq |E(G)| - |\mathcal{M}|.$$

□

We present a greedy algorithm, based on selecting vertices of minimum degree, that approximates the size of a MIM in $B(G)$. We analyse this algorithm using linear programming providing us with a lower bound on the size of the IM.

In order to guarantee that the matching chosen by the algorithm is induced, once an edge is chosen for inclusion into the IM, it is deleted along with all edges incident with the neighbours of the end-points of the selected edge. At any point during the execution of the algorithm, the degree of any non-deleted vertex $v_2 \in V_2$ will be at most 3 and at least 1 and the degree of any non-deleted vertex $v_1 \in V_1$ will be at most 2 and at least 1. We denote the set of vertices of degree i in V_1 and V_2 by V_{1i} and V_{2i} respectively. For an edge $e = (v_1, v_2)$, we let B_e denote the set of all edges incident with neighbours of v_1 and v_2.

The first step of the algorithm involves the selection of an edge e from E. We note here that at each subsequent step of the algorithm there will always exist a vertex $v_1 \in V_1$ of degree 1 or a vertex $v_2 \in V_2$ of degree at most 2 if the input graph is connected.

After the first step, the algorithm repeatedly selects edges for inclusion into the IM based upon the degrees of the vertices available until no edges remain. The following list describes the preference of choice for an edge to be selected for inclusion into the IM. The list is in decreasing order of priority.

- an edge incident with a vertex $v_2 \in V_2$ of degree 1
- an edge incident with a vertex $v_1 \in V_1$ of degree 1
- an edge incident with a vertex $v_2 \in V_2$ of degree 2

Edges of lower priority are chosen only when those of higher priority are not present.

$\backslash *$ **Algorithm B** $* \backslash$

Select e from E
$\mathcal{M} := \{e\}$
delete all the edges in B_e
while ($|E(B(G))| \neq 0$)
{
if ($|V_{21}| \neq 0$)
select an edge e incident with a vertex $v_2 \in V_2$ of degree 1
else if ($|V_{11}| \neq 0$)
select an edge e incident with a vertex $v_1 \in V_1$ of degree 1
else
select an edge e incident with a vertex $v_2 \in V_2$ of degree 2
$\mathcal{M} := \mathcal{M} \cup \{e\}$
delete all the edges in (B_e)
}

Each step of the algorithm involves the selection of an edge to be included into the IM and the subsequent deletion of this and possibly other edges. Any particular step may be represented by a change in the size of the sets V_{ji} ($i \in \{1,2\}$; $j = 1$ and $i \in \{1,2,3\}$; $j = 2$). Consider the following example,

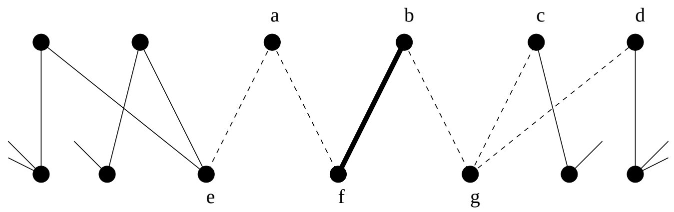

Fig. 2. An example step of the algorithm.

where the edge (b, f) is added to the IM and the dotted lines are subsequently deleted. This may be described as $-2V_{23} - V_{22} - 4V_{12} + V_{22} + 2V_{11}$ since vertices a, b, f and g are deleted along with all their incident edges, vertex e moves from the set V_{23} to V_{22} and vertices c and d move from the set V_{12} to V_{11}.

We consider every possible step of the algorithm and describe it in the above form. The fact that the maximum degree in this bipartite graph is three and the fact that it represents a 4-connected planar triangulation (this bipartite graph will contain no cycle of size less than eight) limits the size of the set of operations considerably. Denote the set of steps using $\{OPS\}$ and for each $Op \in OPS$, $r(Op)$ denotes the number of times the step Op is performed by the algorithm. We denote the change in the size of the set of vertices V_{ij} due to a step $Op \in OPS$ by $\Delta V_{ij}(Op)$. The solution to the following linear program is then a bound on the size of the IM returned by the algorithm.

MINIMISE :

$$\sum_{Op \in OPS} r(Op),$$

SUBJECT TO :

$$\sum_{Op \in OPS} \Delta V_{23}(Op) \times r(Op) = -2(n-2),$$

$$\sum_{Op \in OPS} \Delta V_{12}(Op) \times r(Op) = -3(n-2),$$

$$\sum_{Op \in OPS} \Delta V_{22}(Op) \times r(Op) = 0,$$

$$\sum_{Op \in OPS} \Delta V_{21}(Op) \times r(Op) = 0,$$

$$\sum_{Op \in OPS} \Delta V_{11}(Op) \times r(Op) = 0.$$

We scale the above linear program by 5(n-2) and when implemented and solved, the solution returned is 6/25, indicating that Algorithm B will return an IM of G of size at least 6(n-2)/5 - c. The small constant needs to be subtracted since the linear program does not precisely represent the entire algorithm (the first step has not been included). Using multi-parametric sensitivity analysis and bounds on the possible changes to the right-hand-sides of the constraints of the linear program (given by the first step of the algorithm) it is possible to calculate an exact value for this constant. We will use the asymptotic result here to bound the size of the corresponding 2-spanner of G.

Lemma 6. *A sparsest 2-spanner of any n-vertex 4-connected planar triangulation has at most 9(n-2)/5 edges.*

Proof. The solution to the linear program indicates that asymptotically, the algorithm will return an IM for $B(G)$ of size at least 6(n-2)/5. This corresponds to a 2-spanner $\mathcal{S}$ of G and we have

$$|\mathcal{S}| \leq 3(n-2) - \frac{6(n-2)}{5} = \frac{9(n-2)}{5}.$$

□

The above result in conjunction with Lemma 2 allows us to approximate S2S(4CPT) and we have the following theorem.

Theorem 2. *Given an n-vertex 4-connected planar triangulation G, Algorithm B returns a 2-spanner of G with no more than 6/5 times the number of edges of a sparsest 2-spanner of G.*

Proof. Lemma 2 gives us a lower bound on the size of a sparsest 2-spanner for this class of graphs. Lemma 6 gives us an upper bound on the size of any 2-spanner constructed by our induced matching algorithm (Algorithm B). Denoting the number of edges in a sparsest 2-spanner by $|\mathcal{S}|$ and the number of edges in a spanner given by Lemma 6 as $|B|$, we have :

$$\frac{|B|}{|\mathcal{S}|} \leq \frac{\frac{9(n-2)}{5}}{\frac{3(n-2)}{2}} \leq \frac{6}{5}.$$

□

5 Conclusions and Open Problems

In this paper we have considered the approximability of a sparsest 2-spanner of a 4-connected planar triangulation. In doing so, 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 and the use of algorithmic results, we were able to prove the first non trivial approximation results for this problem.

Dor and Tarsi [10] showed that packing a guest graph into a host graph is NP-Complete when the guest graph has three or more edges. Heath and Vergara [14] showed that the problem remains NP-Complete even when the host graph is restricted to being a planar graph and the guest graph has at least three edges. The problem of packing 3-stars into the dual of a 4-connected planar triangulation remains an open problem.

Cameron [7] showed that, in general, the maximum induced matching problem is NP-Complete. Zito [20] showed that the problem remains NP-Complete even when restricted to regular graphs of degree $4s$ (for all values of s). Whether the problem of finding a maximum induced matching of a 2,3-regular bipartite planar graph is NP-Complete remains an open problem.

The complexity of finding a sparsest 2-spanner of a 4-connected planar triangulation remains open. In a broader sense, for planar graphs and $t \in \{2, 3, 4\}$, the complexity of finding a sparsest t-spanner also remains open.

We pose the question : Is it possible to exploit the structure of a 4-connected planar triangulation in order to find an algorithm that will return a sparsest 2-spanner in polynomial time? For example, using an approach such as that of Baker [3], it may be possible to devise a Polynomial Time Approximation Scheme. Many questions are still to be answered.

References

1. P. Alimonti and T. Calamoneri. **Improved Approximations of Independent Dominating Set in Bounded Degree Graphs.** *Graph-theoretic Concepts*

in Computer Science (Cadenabbia, 1996). Lecture Notes in Computer Science, 1197:2–16, 1997. Springer.

2. I. Althőfer, G. Das, D. Dobkin, and D. Joseph. **Generating Sparse Spanners of Weighted Graphs.** *Discrete Computational Geometry*, 9:81–100, 1993.
3. B. Baker. **Approximation Algorithms for NP-Complete Problems on Planar Graphs.** *Journal of the ACM*, 41(1):153-180, 1994.
4. U. Brandes and D. Handke. **NP-Completeness Results for Minimum Planar Spanners.** *Proceedings of the 23rd Workshop on Graph-Theoretic Concepts in Computer Science (WG '97)*, Lecture Notes in Computer Science, 1335:85–99, 1997.
5. L. Cai. **NP-Completeness of Minimum Spanner Problems.** *Discrete Applied Mathematics*, 48:187–194, 1994.
6. L. Cai and M. Keil. **Spanners in Graphs of Bounded Degree.** *Networks,* 24:233-249, 1994.
7. K. Cameron. **Induced Matchings.** *Discrete and Applied Mathematics*, 24(1-3):97–102, 1989.
8. A. Czumaj and A. M. Gibbons. **Problems on Pairs of Trees and the Four Colour Problem of Planar Graphs.** *Proceedings of the 20th Annual International Colloquium on Automata, Languages and Programming (ICALP) 1993, A. Lingas, R. Karlsson and S. Carlsson, editors*, Lecture Notes in Computer Science, 700:88–101, 1993.
9. R. Diestel. **Graph Theory**. Springer-Verlag, New York, 1997.
10. D. Dor and M. Tarsi. **Graph Decomposition is NP-Complete: A Complete Proof of Holyer's Conjecture,** *SIAM Journal on Computing,* 26(4):1166-1187, 1997.
11. W. Duckworth and N. C. Wormald. **Approximating the Minimum Independant Dominating Set Problem for Cubic Graphs.** *To Appear.*
12. W. Duckworth and M. Zito. **Sparse Hypercube 3-Spanners.** *Submitted to* Discrete Applied Mathematics.
13. M. M. Halldórsson. **Approximating k-Set Cover and Complementary Graph Colouring.** *In Proceedings of the Fifth International Integer Programming and Combinatorial Optimization Conference (IPCO V),* pages 118–131, Vancouver, Canada, June 1996. Springer-Verlag. Lecture Notes in Computer Science, 1084.
14. L. S. Heath and J. P. Vergara. **Edge-Packing in Planar Graphs.** *Theory of Computing Systems,* 31(6):629-662, 1998.
15. B. Jackson and N. C. Wormald. **On the Linear k-arboricity of Cubic Graphs.** *Discrete Mathematics*, 162(1-3):293–297, 1996.
16. D. Peleg and A. A. Schäffer. **Graph Spanners.** *Journal of Graph Theory,* 13(1):99–116, 1989.
17. D. Peleg and J. D. Ullman. **An Optimal Synchroniser for the Hypercube.** *SIAM Journal on Computing*, 18(4):740–747, 1989.
18. 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.
19. M. Zito. **Randomised Techniques in Combinatorial Algorithmics.** *PhD Thesis,* Department of Computer Science, University of Warwick. *To Appear.*
20. M. Zito. **Induced Matchings in Regular Graphs and Trees.** In *Proceedings of the 25th Workshop on Graph Theoretic Concepts in Computer Science (WG'99).* Ascona, June 1999, Springer.