$\mathcal{NP}$–Completeness Results for Minimum Planar Spanners*

Ulrik Brandes and Dagmar Handke

Universität Konstanz, Fakultät für Mathematik und Informatik,
78457 Konstanz, Germany.
email: {Ulrik.Brandes,Dagmar.Handke}@uni-konstanz.de

Abstract. For any fixed parameter $t \geq 1$, a *t–spanner* of a graph G is a spanning subgraph in which the distance between every pair of vertices is at most t times their distance in G. A *minimum* t–spanner is a t–spanner with minimum total edge weight or, in unweighted graphs, minimum number of edges. In this paper, we prove the $\mathcal{NP}$–hardness of finding minimum t–spanners for planar weighted graphs and digraphs if $t \geq 3$, and for planar unweighted graphs and digraphs if $t \geq 5$. We thus extend results on that problem to the interesting case where the instances are known to be planar. We also introduce the related problem of finding minimum *planar* t–spanners and establish its $\mathcal{NP}$–hardness for similar fixed values of t.

1 Introduction

A *t–spanner* of a graph G is a spanning subgraph S in which the distance between every pair of vertices is at most t times their distance in G. The main idea of this concept is to find a subgraph of a given graph G that is sparse, but still guarantees a so–called *stretch factor* on the vertex–to–vertex distances of G. The stretch factor will be bounded by a constant independent of the size of G (i.e. in $\mathcal{O}(1)$). Observe that the minimum spanning tree does not necessarily meet this specification. Consider, for example, the complete graph K_n with vertices $1, 2, \ldots, n$ and unit edge weights. Then the simple path $1, 2, \ldots, n$ forms a minimum spanning tree yielding a stretch factor of $t = n - 1$.

The concept of spanners has been introduced by Peleg and Ullman in [PU87], where they used spanners to synchronize asynchronous networks. One of many other applications for spanners are communication networks, where one is interested in finding a sparse subnetwork that nevertheless guarantees constant delay factors. A survey of some results on the existence and efficient constructibility of (sparse) spanners is given in [PS89]. Further results and discussions concerning t–spanners and variants thereof can be found in [Soa92].

In most applications the sparseness of a spanner is crucial. The problem of finding t–spanners with a minimum number of edges has been shown to be $\mathcal{NP}$–hard for most values of t by Cai in [Cai94]. Therefore subsequent efforts have

* Research partially supported by DFG under grant Wa 654/10-1.

concentrated on finding spanners that are maybe not minimum, but sufficiently sparse (see for example [ADD$^+$93]).

Several authors considered variants of t–spanners. In [CC95], Cai and Corneil deal with *tree t–spanners* (i.e. t–spanners that are trees) and also examine the complexity status of the corresponding decision problem. Liestman and Shermer introduced the notion of *additive* spanners, which employ an additive instead of multiplicative stretch function on the distances [LS93].

Here we consider spanners in *planar* graphs (either weighted or unweighted, directed or undirected), i.e. we restrict the set of input instances. We thereby (partially) settle a question raised in [Cai94]. We also introduce the notion of *planar t–spanners.* These are subgraphs, which in addition to being t–spanners are planar, no matter whether the original graph is planar or not.

This paper is organized as follows: After introducing some basic notation and the examined problems, our results of $\mathcal{NP}$–completeness are stated in Sect. 2. Proofs of these in unweighted, weighted, and directed graphs make up for Sects. 3, 4, and 5, respectively.

2 Problems and Results

In what follows $G = (V, E; w)$ (respectively $G = (V, A; w)$) denotes a simple, weighted undirected (directed) graph with vertex set V, edge set E (arc set A), and edge weights $w : E \to \mathbb{R}^+$ ($w : A \to \mathbb{R}^+$). If all edges have unit weight, i.e. all weights are equal to 1, the graph is said to be *unweighted.* A directed graph (*digraph*) is said to be an *oriented* graph, if it does not contain a cycle of two arcs. For simplicity, we will use the terminology for undirected graphs throughout most of this paper. The terms are naturally extended to digraphs. Since spanners of each connected component can be determined independently, we only consider connected graphs. The *length* of a path is the sum of the weights of its edges. The *distance* between two vertices u and v in G, i.e. the length of the shortest (directed) path, is denoted by $d_G(u, v)$. A t–spanner is defined as follows:

Definition 1 (t–spanner). *For any real valued parameter $t \geq 1$, a spanning subgraph $S = (V, E'; w)$ with $E' \subseteq E$ is a t–*spanner *of an edge-weighted graph $G = (V, E; w)$, if $d_S(u, v) \leq t \cdot d_G(u, v)$ for all $u, v \in V$.*

The parameter t is called *stretch factor.* We say that an edge $e \in E$ is *covered* (by an edge $f \in S$), if in S there exists a path of length at most $t \cdot w(e)$ (and containing f) that connects the endpoints of e.

In order to prove that a given spanning subgraph is a t–spanner, we do not have to consider all pairwise distances of the vertices. It is sufficient to only look at edges of the original graph that are not part of the spanning subgraph.

Lemma 2 ([CC95]). *Let $S = (V, E'; w)$ be a spanning subgraph of a weighted graph $G = (V, E; w)$. Then S is a t–spanner of G if and only if $d_S(u, v) \leq t \cdot w(u, v)$ for every edge $\{u, v\} \in E \setminus E'$.*

A t–spanner is called a *minimum* t–spanner of a weighted graph G, if it has minimum total edge weight among all t–spanners of G. The corresponding decision problem is defined as follows:

Minimum t–Spanner Problem (MinS$_t$)
Given: A graph G with associated (positive real valued) edge weights and a positive real value W.
Problem: Does G contain a t–spanner with total edge weight at most W?

Obviously, for an unweighted graph, the only 1–spanner is the graph itself. For a weighted graph, Hakimi and Yau [HY64] proved that there is a unique 1–spanner with a minimal number of edges. From [CC95] we know that this must also be the unique minimum 1–spanner, and that it can be determined in polynomial time. The $\mathcal{NP}$–completeness of MinS$_t$ for general graphs has been established by Cai [Cai94] for $t \geq 2$ in directed and undirected, and $t \geq 3$ in oriented graphs, even if they are unweighted. From the transformation used in [CC95] to prove the $\mathcal{NP}$–completeness of the Tree t–Spanner Problem it can be seen that MinS$_t$ is also $\mathcal{NP}$–complete for $1 < t < 2$.

Here we will show that the problem remains $\mathcal{NP}$–complete for most values of t when G or S are restricted to be planar[1]. In particular, we prove the following theorems.

Theorem 3. *For any fixed integer $t \geq 5$, MinS$_t$ is $\mathcal{NP}$–complete for undirected, planar, biconnected graphs with unit edge weights.*

Theorem 4. *For any fixed integer $t \geq 3$, MinS$_t$ is $\mathcal{NP}$–complete for undirected, planar, biconnected graphs with edge weights equal to 1 or 2.*

Theorem 5. *For any fixed integer $t \geq 5$ ($t \geq 3$), MinS$_t$ is $\mathcal{NP}$–complete for unweighted (weighted) planar oriented graphs.*

The proofs of the theorems are given in the next three sections. All three of them are transformations from the Planar Satisfiability Problem with three literals in each clause, and they can be viewed as modifications of each other. Therefore we treat the unweighted, undirected case in detail, and outline the changes necessary for the other cases.

Note that in unweighted graphs every t–spanner is also a $\lfloor t \rfloor$–spanner, while there is no such correspondence in weighted graphs, even if all edges have integer weights. At the end of Sect. 4 it will be easy to see how our construction can be adjusted to allow arbitrary real values of $t \geq 3$ in the weighted case. Since the above results are valid for more specific instances, the following corollary is then obtained immediately.

Corollary 6.
1. *For any fixed real valued $t \geq 5$, MinS$_t$ is $\mathcal{NP}$–complete for unweighted planar graphs, planar oriented graphs, and planar digraphs.*

[1] Note that planarity of G implies planarity of S, while the converse is not true.

2. *For any fixed real valued $t \geq 3$, $MinS_t$ is $\mathcal{NP}$–complete for weighted planar graphs, planar oriented graphs, and planar digraphs.*

We now introduce a new variant of general t–spanners, for which similar results are implied by the above theorems.

Definition 7 (planar t–spanner). *For any real valued parameter $t \geq 1$, a spanning subgraph $S = (V, E'; w)$ with $E' \subseteq E$ is a* planar t–spanner *of a weighted graph $G = (V, E; w)$, if $d_S(u, v) \leq t \cdot d_G(u, v)$ for all $u, v \in V$, and S is planar.*

We now give the decision formulation of the corresponding minimization problem for planar t–spanners.

Minimum Planar t–Spanner Problem ($\mathbf{MinPS}_t$)
Given: A graph G with associated (positive real valued) edge weights and a positive real value W.
Problem: Does G contain a planar t–spanner with total edge weight at most W?

As noted above, the only 1–spanner of an unweighted graph is the graph itself. Therefore MinPS_1 is in $\mathcal{P}$ for unweighted graphs, because planarity can be tested in linear time (cf. [HT74,BL76]). On the other hand, it is $\mathcal{NP}$–complete to decide whether an unweighted graph contains a tree t–spanner, i.e. a t–spanner which is a tree, if $t \geq 4$ [CC95]. Observe that spanning trees are planar spanning subgraphs with the least possible number of edges. Together with Theorems 3 and 5 we have the following consequences:

Corollary 8.
1. *For any fixed real valued $t \geq 4$, $MinPS_t$ is $\mathcal{NP}$–complete for unweighted graphs.*
2. *For any fixed real valued $t \geq 5$, $MinPS_t$ is $\mathcal{NP}$–complete for unweighted graphs, oriented graphs, and digraphs, even if they are planar.*

For weighted graphs the situation is different. The graph itself need not be the only 1–spanner. But, as mentioned above, the unique 1–spanner with a minimal number of edges also is the unique minimum 1–spanner, and can be determined in polynomial time. Since all edge weights are positive, and every subgraph of a planar graph is planar, a minimum planar 1–spanner has a minimal number of edges. Therefore a minimum planar 1–spanner would have to be identical to the minimum 1–spanner, and we can conclude that MinPS_1 is also in $\mathcal{P}$ for weighted graphs by testing the minimum 1–spanner for planarity.

In [CC95] the $\mathcal{NP}$–completeness of the Tree t–Spanner Problem for $t > 1$ in weighted graphs is proven. By a close look at the transformation used there and by an appropriate choice of the bound on the total weight of a planar t–spanner, the proof can be modified to show the $\mathcal{NP}$–completeness of MinPS_t for $t > 1$ in weighted, undirected graphs. We omit the details and combine this observation with Theorems 4 and 5.

Table 1. The complexity status of MinS_t and MinPS_t in undirected graphs

t	$\mathbf{MinS}_t$, general graphs [Cai94,CC95]		$\mathbf{MinPS}_t$, general graphs		$\mathbf{Min(P)S}_t$, planar graphs	
	unweighted	weighted	unweighted	weighted	unweighted	weighted
1	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
(1,2)	$\mathcal{P}$	$\mathcal{NPC}$	$\mathcal{P}$	$\mathcal{NPC}$	$\mathcal{P}$	?
[2,3)	$\mathcal{NPC}$	$\mathcal{NPC}$	?	$\mathcal{NPC}$	?	?
[3,4)	$\mathcal{NPC}$	$\mathcal{NPC}$	?	$\mathcal{NPC}$	?	$\mathcal{NPC}$
[4,5)	$\mathcal{NPC}$	$\mathcal{NPC}$	$\mathcal{NPC}$	$\mathcal{NPC}$	?	$\mathcal{NPC}$
$[5,\infty)$	$\mathcal{NPC}$	$\mathcal{NPC}$	$\mathcal{NPC}$	$\mathcal{NPC}$	$\mathcal{NPC}$	$\mathcal{NPC}$

Corollary 9.

1. *For any fixed real valued $t > 1$, $MinPS_t$ is $\mathcal{NP}$–complete for weighted graphs.*
2. *For any fixed real valued $t \geq 3$, $MinPS_t$ is $\mathcal{NP}$–complete for weighted graphs, oriented graphs, and digraphs, even if they are planar and the edge weights are restricted to be equal to 1 or 2.*

Table 1 summarizes the results for the complexity status of the problems considered in this paper in undirected graphs. We give the complexity status for MinS_t with arbitrary input instances (as shown in [Cai94] and [CC95]), for MinPS_t with arbitrary input instances, and for both problems with planar input instances[2]. The results are listed for both the weighted and the unweighted case. A "?" indicates that the complexity status is unknown.

3 MinS_t for Unweighted, Planar Graphs

In this section we prove Theorem 3, so all graphs are unweighted and planar. The other theorems are proven along the same lines and therefore this proof is described in detail first. Part of the proof modifies ideas of [Cai94].

Let $t \geq 5$ be an arbitrary fixed integer. Clearly MinS_t is in $\mathcal{NP}$, since the test whether a spanning subgraph S is a t–spanner can be done in polynomial time. By Lemma 2, we just have to check the (at most linear number of) edges of G that do not belong to S. To show the $\mathcal{NP}$–completeness, we transform the Planar 3–Satisfiability Problem to MinS_t. For this, given an instance of the Planar 3–Satisfiability Problem, we construct a planar graph G, choose a weight W, and then show the equivalence of both problems for these instances.

For the construction we use the fact that we can force edges to be in every minimum t–spanner by adding some additional edges. So this section first introduces the Planar 3–Satisfiability Problem and the concept of forcing, then gives the reduction.

[2] Observe that MinS_t and MinPS_t are the same for planar instances.

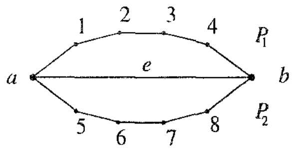

Fig. 1. Forcing edge $\{a, b\}$ into a 5–spanner

3.1 The Planar 3–Satisfiability Problem

The Planar 3–Satisfiability Problem is a variant of the 3–Satisfiability Problem with the additional restriction that the underlying bipartite graph, where clause vertices are connected to variable vertices if the corresponding variables appear within the clause, is planar.

Planar 3–Satisfiability Problem (P3SAT)
Given: A set U of variables, and a collection C of clauses over U with $|c| = 3$ for all $c \in C$. Furthermore the bipartite graph $G = (V, E)$ where $V = U \cup C$ and $E = \{\{x, c\} : x \text{ or } \overline{x} \text{ occurs in } c\}$ is planar.
Problem: Is there a satisfying truth assignment for C?

The $\mathcal{NP}$–completeness proof for this problem can be found in [Man83]. We use the planarity of the underlying graph of P3SAT to construct a planar graph in which we can easily determine the minimum t–spanner.

3.2 Forcing Edges into a Minimum t–Spanner

We can force an edge into a spanner by adding auxiliary edges to the given graph such that every minimum t–spanner of the new graph contains this edge. This concept has appeared in [Cai94] and will be used extensively.

Lemma 10 ([Cai94]). *Let e be an arbitrary edge of an unweighted graph G, and let G' be the graph constructed from G by adding two distinct paths P_1 and P_2 of length t (all internal vertices of P_1 and P_2 are new vertices) between the two ends of e. Then for any minimum t–spanner S of G', edge e belongs to S.*

The two auxiliary paths P_1 and P_2 are called *forcing paths*, edge e is called *forced edge*. A *forced l–component* is a simple path of length l consisting of l forced edges together with their forcing paths. For an example of a forced edge $e = \{a, b\}$ with $t = 5$, see Fig. 1. A minimum t–spanner of this graph contains exactly $2 \cdot (t - 1) + 1 = 2t - 1 = 11$ edges: edge $\{a, b\}$ and $t - 1 = 4$ edges from the forcing paths $(a, 1, 2, 3, 4, b)$ and $(a, 5, 6, 7, 8, b)$ each.

3.3 Construction of the Instance

We start with the planar, embedded underlying graph of the given instance (U, C) of P3SAT and extend the variable and clause vertices to form *variable* *components* and *clause components*. Then these components are combined to form *truth assignment testing components* which reflect the relationship between the satisfiability of a clause and the existence of a minimum t–spanner. As a last step we choose the bound on the number of edges in the Minimum t–Spanner Problem.

The Variable Components. The key idea behind the variable component is that it is planar and each of its possible minimum t–spanners reflects exactly one truth assignment for the variable. For every variable $x \in U$ we construct a variable component T_x as follows. Let k be the number of (positive and negative) occurrences of the variable x in all clauses.

1. Create a central vertex x^*.
2. For each occurrence of x in a clause c create a block of four new vertices $x_1^{(c)}$, $\overline{x}_1^{(c)}$, $x_2^{(c)}$, and $\overline{x}_2^{(c)}$, thus yielding $4k$ so–called *literal vertices* in total. Within each block, the vertices are positioned in this order, and the blocks are arranged circularly around x^* according to the embedding of the underlying graph of the instance of P3SAT.
3. Connect each pair of neighboring literal vertices by a forced $(t-1)$–component such that a circle of $4k$ forced $(t-1)$–components is formed altogether.
4. Connect x^* with all literal vertices by an edge, called *literal edge*. An edge $\{x_i^{(c)}, x^*\}$ is called *positive* literal edge, an edge $\{\overline{x}_i^{(c)}, x^*\}$ is called *negative* literal edge.
5. Create new auxiliary vertices between all pairs of neighboring literal edges, i.e. in total $4k$ auxiliary vertices. Connect each of these by an edge with x^* (called *auxiliary edge*) and by two distinct forced $(t-1)$–components with its neighboring literal vertices. Their literal edges are then called *associated* literal edges of the auxiliary edge and vice versa.

Figure 2 illustrates this construction. For readability a symbolic representation is used later on when larger portions of the graph are drawn. Now, a minimum t–spanner can contain only *consistent* literal edges:

Lemma 11. *Any minimum t–spanner of a variable component T_x contains either all positive or all negative literal edges.*

Proof. Let S be an arbitrary minimum t–spanner of T_x. Then S contains all forced edges and $t-1$ edges from each forcing path. Observe that these edges together with either all $2k$ positive or all $2k$ negative literal edges form a t–spanner. Thus S can contain at most $2k$ edges out of the $8k$ literal and auxiliary edges.

By construction of the variable component, both associated auxiliary edges and both neighboring negative (resp. positive) literal edges are covered by a positive (resp. negative) literal edge in S. But, by an auxiliary edge in S, only the associated literal edges are covered.

Now assume that S contains an auxiliary edge. Then S also contains either the next auxiliary edge, too, or the next not associated literal edge. In total, this

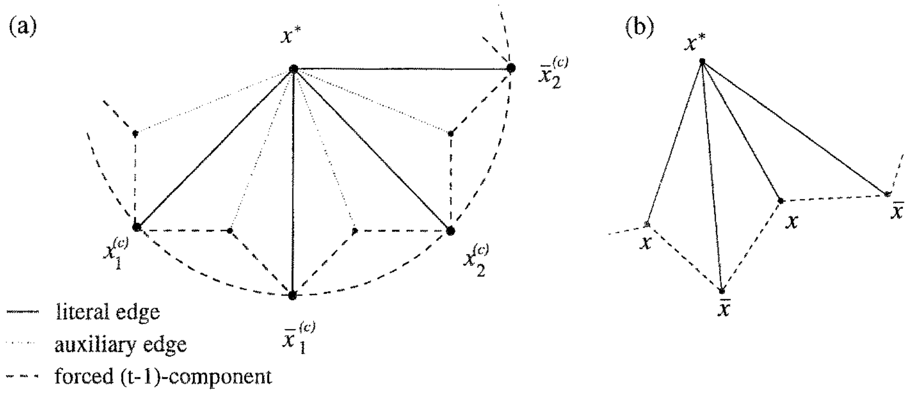

Fig. 2. (a) Part of the variable component T_x for the variable x occurring in clause c, (b) its symbolic representation

leads to more than $2k$ additional edges and thus contradicts the minimality of S. Similarly, assume that S contains two inconsistent literal edges. Then there must be at least one auxiliary edge belonging to S or more than $2k$ literal edges to cover all other edges. Again, this contradicts the minimality of S. Thus S contains exactly every other literal edge. $\square$

With this lemma it can easily be deduced that the number of edges of each minimum t–spanner of T_x is $4 \cdot 3k \cdot (t-1) \cdot (2t-1) + 2k$.

The Clause Components. The clause component for each clause $c \in C$ is basically a quadrilateral consisting of four *clause vertices* 1,2,3, and 4, where the sides are formed by distinct forced $(t-2)$–components. Vertices 1 and 3 are additionally connected by an edge, called *clause edge.* See Fig. 3(a) for an example[3]. Observe that any minimum t–spanner for $t \geq 5$ of such an isolated clause component must contain the clause edge.

The Truth Assignment Testing Components. Now we combine the clause components with the variable components according to the given clauses by identifying vertices. Three sides of the quadrilateral in the clause component each correspond to one of the literals in the corresponding clause. (The fourth side is used to make the arguments symmetrical.) The endpoints of each such side of the quadrilateral are thus identified with the two corresponding literal vertices of the corresponding block in the variable component: if clause c contains the positive literal we use the positive literal vertices $x_i^{(c)}$, and $\overline{x}_i^{(c)}$ otherwise. See Fig. 3(b) for an example.

[3] Our construction is a bit more complex than actually needed in the unweighted case, but will not have to be changed much when being modified for the weighted and the directed case.

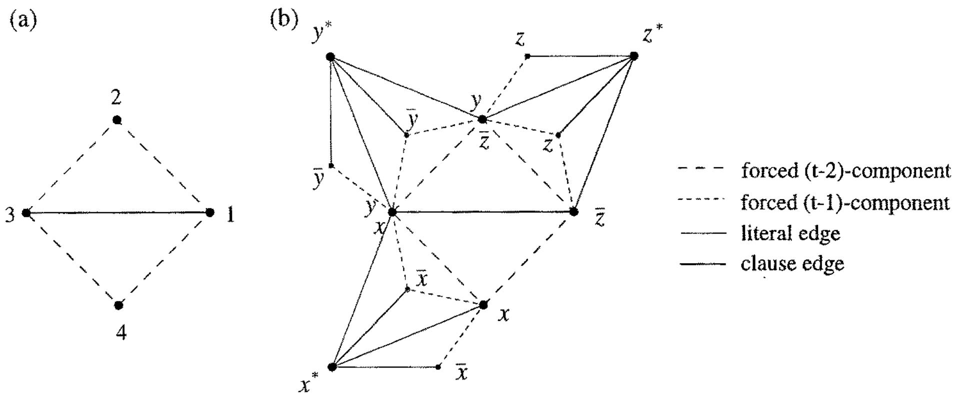

Fig. 3. (a) Clause component, (b) the truth assignment testing component for clause $c = x \vee y \vee \bar{z}$ using the symbolic representation for relevant blocks of the variable components

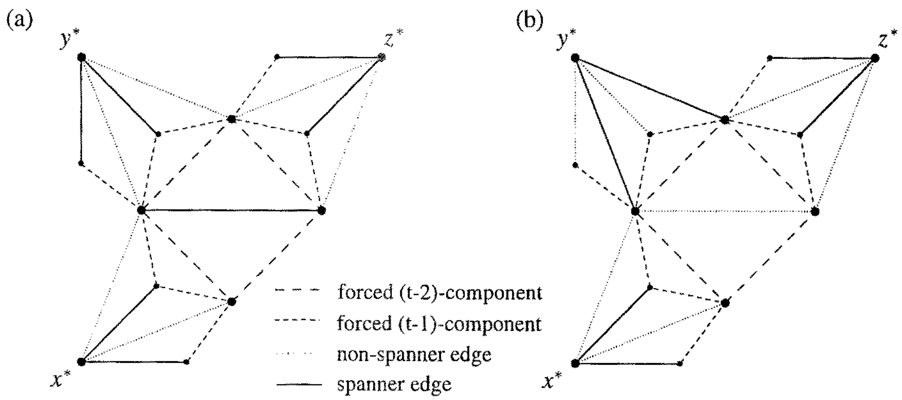

Fig. 4. The minimum t–spanner in the truth assignment testing component

Note that the combination of the variable components with the clause components does not affect the validity of Lemma 11. We now have the following lemma:

Lemma 12. *For any fixed integer $t \geq 5$, a minimum t–spanner S of a truth assignment testing component contains the clause edge if and only if S contains no pair of consistent literal edges that is incident to the clause edge.*

Proof. If S does not contain a pair of literal edges that is incident to the clause edge (cf. Fig. 4(a)), then every path connecting the endpoints of the clause edge in S either uses the clause edge or has length at least $2(t-2) > t$, if $t \geq 5$.

For the other direction see Fig. 4(b). Assume that S contains a pair of incident consistent literal edges. Then this provides a shortcut for one of the forced $(t-2)$–

components, and thus there is a path of length $2 + (t - 2) = t$ in S connecting the endpoints of the clause edge. Hence the clause edge is covered. □

Thus the number of edges in a minimum t–spanner of such a truth assignment testing component reflects the truth value of the corresponding clause. This completes the construction of the graph. All isolated components are planar, and since we start from an instance of P3SAT, the whole graph is planar. It is also easily seen that the instance is biconnected, and can be constructed in polynomial time.

Choice of W. We set W, the bound on the number of edges in a t–spanner, to

$$W = 6m + 36m(2t - 1)(t - 1) + 4m(2t - 1)(t - 2),$$

where m is the number of clauses of the instance of P3SAT.

3.4 Equivalence of the Problems

In this subsection, let (U, C) be an instance of P3SAT, and (G, W) the instance for MinS_t constructed as described above. We will show that there is a satisfying truth assignment for (U, C), if and only if G has a t–spanner with at most W edges.

Lemma 13. *If the set of clauses C of (U, C) is satisfiable, then there exists a planar t–spanner of G with at most W edges.*

Proof. Suppose that the set of clauses C is satisfiable, and let θ be a satisfying truth assignment. From this we construct the subgraph S of G as follows:

1. S contains all forced edges.
2. S contains $t - 1$ arbitrarily chosen edges from each forcing path.
3. For each variable $x \in U$, S contains all positive literal edges if $\theta(x)$ is *true*, and all negative literal edges otherwise.

By this construction, S is trivially a spanning subgraph. The number of edges W' in S computes as follows. S contains

- $3 \cdot m \cdot 3 \cdot 4 \cdot (t - 1)$ forced edges from the variable components (overall number of variable occurrences is $3m$),
- $3 \cdot m \cdot 3 \cdot 4 \cdot (t - 1) \cdot 2(t - 1)$ edges from the forcing paths of the variable components,
- $4 \cdot m \cdot (t - 2)$ forced edges from the clause components,
- $4 \cdot m \cdot (t - 2) \cdot 2(t - 1)$ edges from the forcing paths of the clause components, and
- $3 \cdot m \cdot 2$ literal edges.

Hence S contains exactly $W' = 6m + 36m(2t-1)(t-1) + 4m(2t-1)(t-2) = W$ edges.

It remains to show that S is a t–spanner of G. According to Lemma 2, we only have to show that for every edge not contained in S, there exists a path of length at most t connecting the endpoints of that edge. This is obvious for the variable components. For the clause edges observe that, since θ is a satisfying truth assignment, there is at least one literal in each clause that is true. Due to the construction of S we thus have at least one incident pair of literal edges in each clause component. From Lemma 12 it follows that S is a t–spanner. □

To show the opposite direction we need another lemma:

Lemma 14. *Any minimum t–spanner S of G contains at least W edges.*

Proof. Any t–spanner S of G must contain all forced edges and $t-1$ edges from each forcing path. By Lemma 11, S contains at least either all positive or all negative literal edges for each variable component. This sums up to W. □

Lemma 15. *If G has a t–spanner with at most W edges, then there exists a satisfying truth assignment for (U, C).*

Proof. Suppose S is a t–spanner of G with at most W edges. Then by Lemma 14 S is a minimum t–spanner and contains exactly W edges. All forced edges and the according number of edges from the forcing paths must be in S. Hence there remain only $6m$ further edges which can only be consistent literal edges (by Lemma 11). Thus we can uniquely define a truth assignment θ by setting, for each $x \in U$, $\theta(x) = $ *true*, if S contains the positive literal edges of T_x, and $\theta(x) = $ *false* otherwise.

Since S is a t–spanner and S contains no clause edge it follows from Lemma 12 that there is at least one incident pair of literal edges for every clause edge. Hence θ satisfies all clauses. □

This completes the proof of Theorem 3.

4 MinS$_t$ for Weighted, Planar Graphs

We will now prove Theorem 4. Again, we transform an instance of P3SAT to an instance of MinS$_t$ by extending variable and clause vertices to appropriate components. The fact that we are now allowed to also assign edge weights of value 2 will be exploited to lower the bound on t, thus yielding a stronger result than in the unweighted case.

The variable components remain the same with all edges having unit edge weight, and the results about minimum t–spanners for these components keep valid (Lemma 11). The clause components again consist of four clause vertices, but now three sides of the quadrilateral remain unconnected. Only one side is connected by two consecutive forced $(t-1)$–components with unit edge weights.

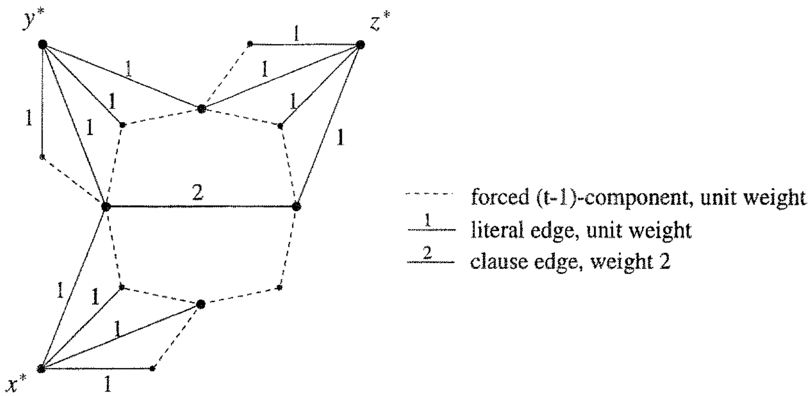

Fig. 5. The truth assignment testing component in the weighted case

As before, we have one clause edge, now having edge weight 2. We combine the components to form the truth assignment testing components as we did in the unweighted case by identifying the corresponding vertices (see Fig. 5 for an example). Note that every edge in the so-constructed instance has unit edge weight, except for the clause edges which are assigned a weight of 2.

To make use of the proof structure from Sect. 3, we provide the following lemma (cf. Lemma 12).

Lemma 16. *For any fixed integer $t \geq 3$, a minimum t–spanner S of a truth assignment testing component contains the clause edge if and only if S contains no pair of literal edges incident to the clause edge.*

Proof. If S does not contain a pair of literal edges that is incident to the clause edge, then every path connecting the endpoints of the clause edge in S either uses the clause edge or has length at least $4(t-1) > 2t$, if $t \geq 3$.

Assume that S contains a pair of incident consistent literal edges. Then we can combine these edges with the two forced $(t-1)$–components of the neighboring side of the quadrilateral to form a path of length $2 + 2(t-1) = 2t$ connecting the endpoints of the clause edge. Hence in this case the clause edge is covered. □

It is easily seen that the constructed graph is again planar and biconnected. By choosing $W = 6m + 36m(t-1)(2t-1) + 2m(t-1)(2t-1)$ the arguments of the previous section can be repeated to complete the proof of Theorem 4.

Corollary 6 states that Theorem 4 can be generalized to allow real values of $t \geq 3$. This can be seen by using forced $(\lfloor t \rfloor - 1)$–components in the construction described above. All results about minimum t–spanners then keep valid.

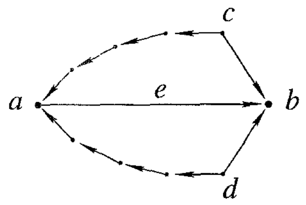

Fig. 6. Forcing arc (a, b) into a 5–spanner

5 MinS$_t$ for Planar Digraphs

In this section we turn to Theorem 5. To show the $\mathcal{NP}$–completeness for digraphs, we again use a modification of the reduction of the previous sections. Here we only give the details of the construction for the unweighted case, since the weighted case is then straightforward from what has been established so far.

Forcing Arcs into a Minimum t–Spanner. Similar to the undirected case, an arc (a, b) of a digraph can be forced to be in every minimum t–spanner as described in [Cai94]. For this purpose we create two new vertices c and d, add two arcs (c, b) and (d, b), and then add two distinct directed paths of length $t-1$ from c to a and from d to a, respectively. Then a minimum t–spanner of this component consists of arc (a, b) and all arcs of the paths of length $t-1$. Figure 6 shows an example for $t = 5$.

The Variable Components. A directed variable component consists of a central vertex x^*, and four *literal vertices* $x_1^{(c)}$, $\overline{x}_1^{(c)}$, $x_2^{(c)}$, and $\overline{x}_2^{(c)}$ together with four *literal arcs* for each positive or negative occurrence of variable x in a clause c. The orientation of the literal arcs depends on what the connection to the clause components will be like (see below). In the following, $\langle x, y\rangle$ stands for exactly one of the arcs (x, y) and (y, x), which will never be present at the same time. We add the following components and auxiliary vertices or arcs:

- All pairs of neighboring literal vertices $x_i^{(c)}$ and $\overline{x}_i^{(c)}$ are connected by two distinct directed forced $(t-1)$–components, one in either direction. Their literal arcs will be directed both either from or to x^*. If they are both directed from (resp. to) x^*, add an auxiliary vertex a_i and an auxiliary arc (x^*, a_i) (resp. (a_i, x^*)). Also connect a_i with $x_i^{(c)}$ and $\overline{x}_i^{(c)}$ by two distinct directed forced $(t-1)$–components directed from the literal vertices to a_i (from a_i to the literal vertices).
- Between the other pairs of neighboring literal arcs add an auxiliary arc $\langle x_i^{(c)}, \overline{x}_i^{(c)}\rangle$, such that $\langle x_i^{(c)}, x^*\rangle$, $\langle x^*, \overline{x}_i^{(c)}\rangle$, $\langle \overline{x}_i^{(c)}, x\rangle$ do not form a directed cycle. Additionally connect $x_i^{(c)}$ and $\overline{x}_i^{(c)}$ with x^* by directed forced $(t-1)$–components parallel to their corresponding literal arcs.

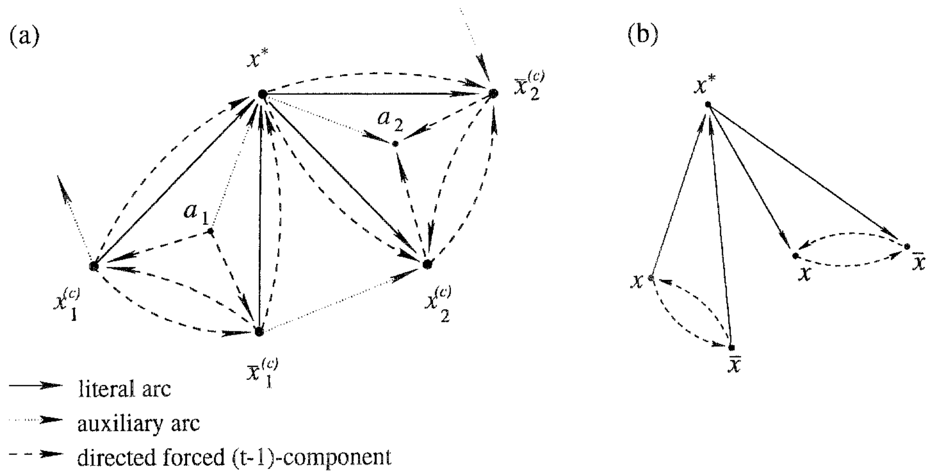

Fig. 7. (a) Part of the directed variable component for the variable x occurring in clause c, (b) its symbolic representation

Figure 7 gives an example of a directed variable component and its symbolic representation. As in the undirected case this construction guarantees that every minimum t–spanner of such a variable component only contains consistent literal arcs (cf. Lemma 11). This can be seen as follows. All positive (or negative, respectively) literal arcs together with the appropriate arcs from the forced arcs form a t–spanner. By the construction of the auxiliary arcs at least every other literal arc has to be included into a t–spanner. No other auxiliary arc is covered by an auxiliary arc in the t–spanner.

The Clause Components. We define clause components analogously to the undirected case, where the clause arc and the forced $(t-2)$-components are oriented such that they start and end at the same vertices of the quadrilateral.

The Truth Assignment Testing Components. Again we combine the variable and clause components by identifying the corresponding vertices. According to the choice of the orientation of the clause arc, the corresponding literal arcs are now oriented such that the literal arcs together with one of the directed forced $(t-2)$-components of the clause component form a directed path parallel to the clause arc. The remaining literal arcs of the variable component are oriented such that pairs $\langle x_i^{(c)}, x^* \rangle$ and $\langle \bar{x}_i^{(c)}, x^* \rangle$ of corresponding inconsistent literal arcs are directed likewise from or to x^*. Figure 8 shows an example of such a directed truth assignment testing component.

This completes the construction for the directed, unweighted case. It is easily seen that the graph is planar and oriented. Choosing W as in the undirected case ($W = 6m + 36m(t-1)(2t-1) + 4m(t-2)(2t-1)$), the proof of the equivalence of P3SAT and MinS_t is straightforward as before.

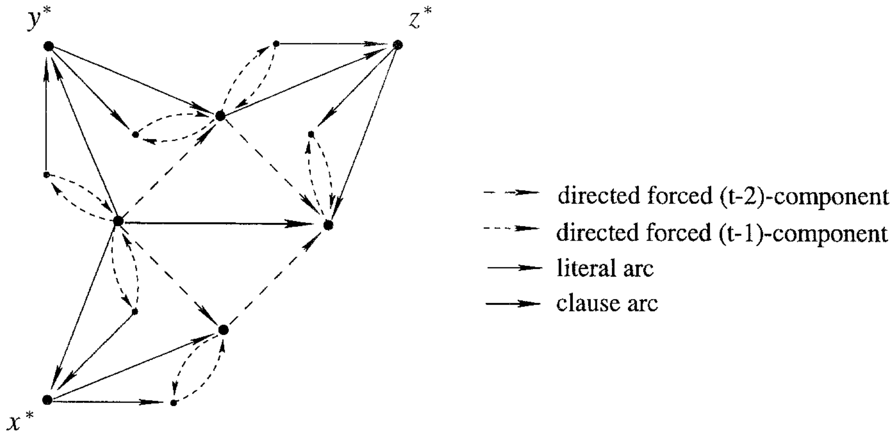

Fig. 8. The directed truth assignment testing component for unweighted digraphs

Weighted Digraphs. In the weighted, directed case the same variable components (with unit arc weights) are used. The clause components are the ones from the weighted, undirected case, and orientations are determined analogously.

References

[ADD+93] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9:81–100, 1993.

[BL76] Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. *J. of Computer and System Sciences*, 13:335–379, 1976.

[Cai94] Leizhen Cai. $\mathcal{NP}$-completeness of minimum spanner problems. *Discrete Applied Math.*, 48:187–194, 1994.

[CC95] Leizhen Cai and D.G. Corneil. Tree spanners. *SIAM J. Discrete Math.*, 8(3):359–387, 1995.

[HT74] John E. Hopcroft and Robert E. Tarjan. Efficient planarity testing. *J. of the Association for Computing Machinery*, 21:549–568, 1974.

[HY64] S.L. Hakimi and S.-T. Yau. Distance matrix of a graph and its realizability. *Q. J. Mech. appl. Math.*, 22:305–317, 1964.

[LS93] A.L. Liestman and Thomas Shermer. Grid spanners. *Networks*, 23:123–133, 1993.

[Man83] Anthony Mansfield. Determining the thickness of graphs is $\mathcal{NP}$-hard. *Math. Proc. Camb. Phil. Soc.*, 93:9–23, 1983.

[PS89] David Peleg and Alejandro A. Schäffer. Graph spanners. *J. of Graph Theory*, 13:99–116, 1989.

[PU87] David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. In *Proceedings 1987 6th ACM Symposium on Principles of Dist. Comp., Vancouver*, pages 77–85, 1987.

[Soa92] José Soares. Graph spanners: a survey. *Congressus Numerantium*, 89:225–238, 1992.