UNIVERSITY OF BIRMINGHAM
University of Birmingham Research at Birmingham

Tight bounds for planar strongly connected Steiner subgraph with fixed number of terminals (and extensions)

Chitnis, Rajesh; Feldmann, Andreas Emil ; Hajiaghayi, MohammadTaghi; Marx, Dániel

DOI:
10.1137/18M122371X

License:
None: All rights reserved

Document Version
Peer reviewed version

Citation for published version (Harvard):
Chitnis, R, Feldmann, AE, Hajiaghayi, M & Marx, D 2020, 'Tight bounds for planar strongly connected Steiner subgraph with fixed number of terminals (and extensions)', *SIAM Journal on Computing*, vol. 49, no. 2, pp. 318–364. https://doi.org/10.1137/18M122371X

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
© 2020, Society for Industrial and Applied Mathematics

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 10. May. 2024

Tight Bounds for Planar Strongly Connected Steiner Subgraph with Fixed Number of Terminals (and Extensions)*

Rajesh Chitnis† Andreas Emil Feldmann‡ MohammadTaghi Hajiaghayi§
Dániel Marx¶

Abstract

Given a vertex-weighted directed graph $G = (V,E)$ and a set $T = \{t_1, t_2, \ldots t_k\}$ of k terminals, the objective of the STRONGLY CONNECTED STEINER SUBGRAPH (SCSS) problem is to find a vertex set $H \subseteq V$ of minimum weight such that $G[H]$ contains a $t_i \to t_j$ path for each $i \neq j$. The problem is NP-hard, but Feldman and Ruhl [FOCS '99; SICOMP '06] gave a novel $n^{O(k)}$ algorithm for the SCSS problem, where n is the number of vertices in the graph and k is the number of terminals. We explore how much easier the problem becomes on planar directed graphs.

- Our main algorithmic result is a $2^{O(k)} \cdot n^{O(\sqrt{k})}$ algorithm for planar SCSS, which is an improvement of a factor of $O(\sqrt{k})$ in the exponent over the algorithm of Feldman and Ruhl.
- Our main hardness result is a matching lower bound for our algorithm: we show that planar SCSS does not have an $f(k) \cdot n^{o(\sqrt{k})}$ algorithm for any computable function f, unless the Exponential Time Hypothesis (ETH) fails.

To obtain our algorithm, we first show combinatorially that there is a minimal solution whose treewidth is $O(\sqrt{k})$, and then use the dynamic-programming based algorithm for finding bounded-treewidth solutions due to Feldmann and Marx [ICALP '16]. To obtain the lower bound matching the algorithm, we need a delicate construction of gadgets arranged in a grid-like fashion to tightly control the number of terminals in the created instance.

The following additional results put our upper and lower bounds in context:

- Our $2^{O(k)} \cdot n^{O(\sqrt{k})}$ algorithm for planar directed graphs can be generalized to graphs excluding a fixed minor. Additionally, we can obtain this running time for the problem of finding an optimal planar solution even if the input graph is not planar.
- In general graphs, we cannot hope for such a dramatic improvement over the $n^{O(k)}$ algorithm of Feldman and Ruhl: assuming ETH, SCSS in general graphs does not have an $f(k) \cdot n^{o(k/\log k)}$ algorithm for any computable function f.

*A preliminary version of this paper appeared in SODA 2014 [19]

†School of Computer Science, University of Birmingham. Part of this work was done while at the University of Maryland, USA and University of Warwick, UK (supported by ERC grant 2014-CoG 647557). Email: rajeshchitnis@gmail.com

‡Department of Applied Mathematics, Charles University, Prague, Czechia. Supported by project CE-ITI (GAČR no. P202/12/G061) of the Czech Science Foundation, and by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE/SCI/004). Email: feldmann.a.e@gmail.com

§Department of Computer Science , University of Maryland at College Park, USA. Supported in part by NSF CAREER award 1053605, ONR YIP award N000141110662, DARPA/AFRL award FA8650-11-1-7162 and a University of Maryland Research and Scholarship Award (RASA). Email: hajiagha@cs.umd.edu

¶Max Planck Institute for Informatics, Saarbrücken, Germany. Supported by ERC Starting Grant PARAMTIGHT (No. 280152) and ERC Consolidator Grant SYSTEMATICGRAPH (No. 725978). Email: dmarx@mpi-inf.mpg.de

- Feldman and Ruhl generalized their $n^{O(k)}$ algorithm to the more general DIRECTED STEINER NETWORK (DSN) problem; here the task is to find a subgraph of minimum weight such that for every source s_i there is a path to the corresponding terminal t_i. We show that, assuming ETH, there is no $f(k) \cdot n^{o(k)}$ time algorithm for DSN on acyclic planar graphs.

All our lower bounds hold for the edge-unweighted version, while the algorithm works for the more general vertex-(un)weighted version.

1 Introduction

The STEINER TREE (ST) problem is one of the earliest and most fundamental problems in combinatorial optimization: given an undirected graph $G = (V, E)$ and a set $T \subseteq V$ of terminals, the objective is to find a tree of minimum size which connects all the terminals. The ST problem is believed to have been first formally defined by Gauss in a letter in 1836, and the first combinatorial formulation is attributed independently to Hakimi [43] and Levin [52] in 1971. The ST problem is known to be NP-complete, and was in fact part of Karp's original list [49] of 21 NP-complete problems. In the directed version of the ST problem, called DIRECTED STEINER TREE (DST), we are also given a root vertex r and the objective is to find a minimum size arborescence which connects the root r to each terminal from T. An easy reduction from SET COVER shows that the DST problem is also NP-complete.

Steiner-type problems arise in the design of networks. Since many networks are symmetric, the directed versions of Steiner-type problems were mostly of theoretical interest. However, in recent years, it has been observed [65, 67] that the connection cost in various networks such as satellite or radio networks are not symmetric. Therefore, directed graphs form the most suitable model for such networks. In addition, Ramanathan [65] also used the DST problem to find low-cost multicast trees, which have applications in point-to-multipoint communication in high bandwidth networks. We refer the interested reader to Winter [69] for a survey on applications of Steiner problems in networks.

In this paper we consider two well-studied Steiner-type problems in directed graphs, namely the STRONGLY CONNECTED STEINER SUBGRAPH and the DIRECTED STEINER NETWORK problems. In the (vertex-unweighted) STRONGLY CONNECTED STEINER SUBGRAPH (SCSS) problem, given a directed graph $G = (V, E)$ and a set $T = \{t_1, t_2, \ldots, t_k\}$ of k terminals, the objective is to find a set $S \subseteq V$ of minimum size such that $G[S]$ contains a $t_i \to t_j$ path for each $1 \leq i \neq j \leq k$. Thus, just as DST, the SCSS problem is another directed version of the ST problem, where all terminals need to be connected to each other. The (vertex-unweighted) DIRECTED STEINER NETWORK (DSN) problem generalizes both DST and SCSS: given a directed graph $G = (V, E)$ and a set $T = \{(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)\}$ of k pairs of terminals, the objective is to find a set $S \subseteq V$ of minimum size such that $G[S]$ contains an $s_i \to t_i$ path for each $1 \leq i \leq k$. We first describe the known results for both SCSS and DSN before stating our results and techniques.

1.1 Previous work

Since both DSN and SCSS are NP-complete, one can try to design polynomial-time approximation algorithms for these problems. An α-approximation for DST implies a 2α-approximation for SCSS as follows: fix a terminal $t \in T$ and take the union of the solutions of the DST instances $(G, t, T \setminus t)$ and $(G_{\text{rev}}, t, T \setminus t)$, where G_{rev} is the graph obtained from G by reversing the orientations of all edges. The best known approximation ratio in polynomial time for SCSS is k^{ε} for any $\varepsilon > 0$ [14]. A result of Halperin and Krauthgamer [44] implies SCSS has no $\Omega(\log^{2-\varepsilon} n)$-approximation for any $\varepsilon > 0$, unless NP has quasi-polynomial Las Vegas algorithms. For the more general DSN problem, the best approximation ratio known is $n^{2/3+\varepsilon}$ for any $\varepsilon > 0$. Berman et al. [4] showed that DSN has no $\Omega(2^{\log^{1-\varepsilon} n})$-approximation for any $0 < \varepsilon < 1$, unless NP has quasi-polynomial time algorithms.

Rather than finding approximate solutions in polynomial time, one can look for exact solutions in time that is still better than the running time obtained by brute force algorithms. For (unweighted versions of) both the SCSS and DSN problems, brute force can be used to check in time $n^{O(p)}$ if a solution of size at most p exists: one can go through all sets of size at most p. A more efficient algorithm would have runtime $f(p) \cdot n^{O(1)}$, where f is some computable function depending only on p. A problem is said to be *fixed-parameter tractable* (FPT) with a particular parameter p if it admits such an algorithm; see [23, 28, 37, 62] for more background on FPT algorithms. A natural parameter for our considered problems is the number k of terminals or terminal pairs; with this parameterization, it is not even clear if there is a polynomial-time algorithm for every fixed k, much less if the problem is FPT. It is known that STEINER TREE on undirected graphs is FPT parameterized by the number k of terminals: the classical algorithm of Dreyfus and Wagner [29] solves the problem in time $3^k \cdot n^{O(1)}$. The running time was recently improved to $2^k \cdot n^{O(1)}$ by Björklund et al. [5]. The same algorithms work for DIRECTED STEINER TREE as well.

For the SCSS and DSN problems, we cannot expect fixed-parameter tractability: Guo et al. [42] showed that SCSS is W[1]-hard parameterized by the number of terminals k, and DSN is W[1]-hard parameterized by the number of terminal pairs k. In fact, it is not even clear how to solve these problems in polynomial time for small fixed values of the number k of terminals/pairs. The case of $k = 1$ in DSN is the well-known shortest path problem in directed graphs, which is known to be polynomial time solvable. For the case $k = 2$ in DSN, an $O(n^5)$ algorithm was given by Li et al. [53] which was later improved to $O(mn + n^2 \log n)$ by Natu and Fang [61]. The question regarding the existence of a polynomial time algorithm for DSN when $k = 3$ was open. Feldman and Ruhl [35] solved this question by giving an $n^{O(k)}$ algorithm for DSN, where k is the number of terminal pairs. They first designed an $n^{O(k)}$ algorithm for SCSS, where k is the number of terminals, and used it as a subroutine in the algorithm for the more general DSN problem.

1.2 Our results and techniques

Given the amount of attention the planar version of Steiner-type problems received from the viewpoint of approximation (see, e.g., [2, 3, 11, 26, 32]) and the availability of techniques for parameterized algorithms on planar graphs (see, e.g., [6, 27, 40, 50, 59]), it is natural to explore SCSS and DSN restricted to planar graphs[1]. In general, one can have the expectation that the problems restricted to planar graphs become easier, but sophisticated techniques might be needed to exploit planarity. In particular, a certain *square root phenomenon* was observed for a wide range of algorithmic problems: the exponent of the running time can be improved from $O(k)$ to $O(\sqrt{k})$ (or to $O(\sqrt{k} \log k)$) and lower bounds indicate that this improvement is essentially best possible [1, 24, 38, 39, 50, 51, 55, 58–60, 63]. Our main algorithmic result is also an improvement of this form:

Theorem 1.1. *An instance (G, T) of the vertex-weighted* STRONGLY CONNECTED STEINER SUBGRAPH *problem with $|G| = n$ and $|T| = k$ can be solved in $2^{O(k)} \cdot n^{O(\sqrt{k})}$ time, when the underlying undirected graph of G is planar.*

This algorithm presents a major improvement over the Feldman-Ruhl algorithm for SCSS in general graphs which runs in $n^{O(k)}$ time. A preliminary version of this paper [19] by a subset of the authors contained a complicated algorithm with a worse running time of $2^{O(k \cdot \log k)} \cdot n^{O(\sqrt{k})}$. It relied on modifying the Feldman-Ruhl token game, and then using the excluded grid theorem for planar graphs followed by treewidth-based techniques. We briefly give some intuition behind this algorithm and the original $n^{O(k)}$ algorithm of Feldman-Ruhl. The algorithm of Feldman-Ruhl for SCSS is based on defining a game with $2k$ tokens and costs associated with the moves of the tokens such that the minimum cost of the game is equivalent to the minimum cost of a solution of the SCSS problem; then the minimum cost of the game can be computed by exploring a state space of size $n^{O(k)}$. The $2^{O(k \cdot \log k)} \cdot n^{O(\sqrt{k})}$ algorithm was obtained by generalizing the Feldman-Ruhl

[1] Planarity for directed graph problems refers to the underlying undirected graph being planar

token game via introducing *supermoves*, which are sequences of certain types of moves. The generalized game still has a state space of $n^{O(k)}$, but it has the advantage that we can now give a bound of $O(k)$ on the number of supermoves required for the game (such a bound is not possible for the original version of the game). This gives an $O(k)$-sized summary of the token game, and hence has treewidth $O(\sqrt{k})$. However, this summary is "unlabeled", i.e., we do not explicitly know which vertices occur where in the summary. Guessing by brute force requires $n^{O(k)}$ time, and the improvement to $2^{O(k\cdot\log k)}\cdot n^{O(\sqrt{k})}$ is obtained by using an embedding theorem of Klein and Marx [50].

Unlike the $2^{O(k\cdot\log k)}\cdot n^{O(\sqrt{k})}$ algorithm of [19], the $2^{O(k)}\cdot n^{O(\sqrt{k})}$ algorithm from Theorem 1.1 does not depend on the Feldman-Ruhl algorithm. It is conceptually much simpler: first we show combinatorially (see Lemma 2.2) that there is a minimal solution whose treewidth is $O(\sqrt{k})$, and then use the dynamic-programming based algorithm for finding bounded-treewidth solutions for DSN due to Feldmann and Marx [36, Theorem 5]. The simplicity of our new approach also allows transparent generalizations in two directions:

- From planar to H-minor-free graphs: we may use the excluded grid minor theorem for H-minor-free graphs [25] instead of the excluded grid minor theorem for planar graphs [66] to prove the existence of a minimal solution of treewidth $O(\sqrt{k})$, which again implies a $2^{O(k)}\cdot n^{O(\sqrt{k})}$ time algorithm.

- Between restricted inputs and restricted solutions: our algorithm only exploits the H-minor-freeness of an optimum solution, and not of the whole input graph. Thus, only the existence of one optimum H-minor-free solution in an otherwise unrestricted input graph is enough to show that some optimum solution (which might not necessarily be H-minor-free) can be found in $2^{O(k)}\cdot n^{O(\sqrt{k})}$ time[2].

Can we get a better speedup in planar graphs than the improvement from $O(k)$ to $O(\sqrt{k})$ in the exponent of n? Our main hardness result matches our algorithm: it shows that $O(\sqrt{k})$ is best possible under the Exponential Time Hypothesis (ETH).

Theorem 1.2. *The edge-unweighted version of the SCSS problem is W[1]-hard parameterized by the number of terminals k, even when the underlying undirected graph is planar. Moreover, under ETH, the SCSS problem on planar graphs cannot be solved in $f(k)\cdot n^{o(\sqrt{k})}$ time where f is any computable function, k is the number of terminals and n is the number of vertices in the instance.*

This also answers the question of Guo et al. [42], who showed the W[1]-hardness of these problems on general graphs and left the fixed-parameter tractability status on planar graphs as an open question. Recall that ETH has the consequence that n-variable 3SAT cannot be solved in time $2^{o(n)}$ [45, 46]. There are relatively few parameterized problems that are W[1]-hard on planar graphs [7, 13, 33, 58]. The reason for the scarcity of such hardness results is mainly because for most problems, the fixed-parameter tractability of finding a solution of size k in a planar graph can be reduced to a bounded-treewidth problem by standard layering techniques. However, in our case the parameter k is the number of terminals, hence such a simple reduction to the bounded-treewidth case does not seem to be possible. Our reduction is from the GRID TILING problem formulated by Marx [56, 58] (see also [23]), which is a convenient starting point for parameterized reductions for planar problems. For our reduction we need to construct two types of gadgets, namely the connector gadget and main gadget, which are then arranged in a grid-like structure (see Figure 2). The main technical part of the reduction is the structural result regarding the existence and construction of particular types of connector gadgets and main gadgets (Lemma 3.3 and Lemma 3.6). Interestingly, the construction of the connector gadget poses a greater challenge: here we exploit in a fairly delicate way the fact that the $t_i \rightsquigarrow t_j$

[2]This can be thought of as an intermediate requirement between the two extremes of either forcing the input graph itself to be H-minor-free versus the other extreme of finding an optimum solution which is H-minor-free. There has been some recent work in this direction [17, 18]

and the reverse $t_j \rightsquigarrow t_i$ paths appearing in the solution subgraph might need to share edges to reduce the weight.

We present additional results that put our algorithm and lower bound for SCSS in a wider context. Given our speedup for SCSS in planar graphs, one may ask if it is possible to get any similar speedup in general graphs. Our next result shows that the $n^{O(k)}$ algorithm of Feldman-Ruhl is almost optimal in general graphs:

Theorem 1.3. *Under ETH, the edge-unweighted version of the SCSS problem cannot be solved in time $f(k) \cdot n^{o(k/\log k)}$ where f is an arbitrary computable function, k is the number of terminals and n is the number of vertices in the instance.*

Our proof of Theorem 1.3 is similar to the W[1]-hardness proof of Guo et al. [42]. They showed the W[1]-hardness of SCSS on general graphs parameterized by the number k of terminals by giving a reduction from k-CLIQUE. However, this reduction uses "edge selection gadgets" and since a k-clique has $\Theta(k^2)$ edges, the parameter is increased at least to $\Theta(k^2)$. Combining with the result of Chen et al. [15] regarding the non-existence of an $f(k) \cdot n^{o(k)}$ algorithm for k-Clique under ETH, this gives a lower bound of $f(k) \cdot n^{o(\sqrt{k})}$ for SCSS on general graphs. To avoid the quadratic blowup in the parameter and thereby get a stronger lower bound, we use the PARTITIONED SUBGRAPH ISOMORPHISM (PSI) problem as the source problem of our reduction. For this problem, Marx [57] gave a $f(k) \cdot n^{o(k/\log k)}$ lower bound under ETH, where $k = |E(G)|$ is the number of edges of the subgraph G to be found in graph H. The reduction of Guo et al. [42] from CLIQUE can be turned into a reduction from PSI which uses only $|E(G)|$ edge selection gadgets, and hence the parameter is $\Theta(|E(G)|)$. Then the lower bound of $f(k) \cdot n^{o(k/\log k)}$ transfers from PSI to SCSS. A natural question is whether we can close the $O(\log k)$ factor in the exponent: however, our reduction is from the PSI problem and the best known lower bound for PSI also has such a gap [57]. Note that there are many other parameterized problems for which the only known way of proving almost tight lower bounds is by a similar reduction from PSI, and hence an $O(\log k)$ gap appears for these problems as well [8–10, 12, 18, 20–22, 31, 34, 41, 47, 48, 54, 60, 64].

Even though Feldman and Ruhl were able to generalize their $n^{O(k)}$ time algorithm from SCSS to DSN, we show that, surprisingly, such a generalization is not possible for our $2^{O(k)} \cdot n^{O(\sqrt{k})}$ time algorithm for planar SCSS.

Theorem 1.4. *The edge-unweighted version of the* DIRECTED STEINER NETWORK *problem is W[1]-hard parameterized by the number k of terminal pairs, even when the input is restricted to planar directed acyclic graphs (planar DAGs). Moreover, there is no $f(k) \cdot n^{o(k)}$ time algorithm for any computable function f, unless the ETH fails.*

This implies that the Feldman-Ruhl algorithm for DSN is optimal, even on planar directed acyclic graphs. As in our lower bound for planar SCSS, the proof is by a reduction from an instance of the $k \times k$ GRID TILING problem. However, unlike in the reduction to SCSS where we needed $O(k^2)$ terminals, the reduction to DSN needs only $O(k)$ pairs of terminals (see Figure 6). Since the parameter blowup is linear, the $f(k) \cdot n^{o(k)}$ lower bound for GRID TILING from [56] transfers to DSN.

Remark: All our hardness results (Theorem 1.2, Theorem 1.3 and Theorem 1.4) are presented for weighted-edge versions with polynomially-bounded integer weights (including edges with weight zero). By splitting each edge of weight W into W edges of weight one, all the results also hold for the unweighted-edge version. Our algorithm (Theorem 1.1) is presented for the weighted-vertex version. Appendix A shows that the unweighted-vertex version is more general than the weighted-edge version. Hence all our lower bounds also hold for the (un)weighted-vertex version too.

Finally, instead of parameterizing by the number of terminals, we can consider parameterization by the number of edges/vertices of the solution. Let us briefly and informally discuss this parameterization. Note that the number of terminals is a lower bound on the number of edges/vertices of the solution (up to

a factor of 2 in the case of DSN parameterized by the number of edges), thus fixed-parameter tractability could be easier to obtain by parameterizing with the number of edges/vertices. However, our lower bound for SCSS on general graphs (as well as the W[1]-hardness of Guo et al. [42]) actually proves hardness also with these parameterizations, making fixed-parameter tractability unlikely. On the other hand, it follows from standard techniques that both SCSS and DSN are FPT on planar graphs when parameterizing by the number k of edges/vertices in the solution. The main argument here is that the solution is fully contained in the k-neighborhood of the terminals, whose number is at most $2k$. It is known that the k-neighborhood of $O(k)$ vertices in a planar graph has treewidth $O(k)$, and thus one can use standard techniques on bounded-treewidth graphs (dynamic programming or Courcelle's Theorem). Alternatively, at least in the unweighted case, one can formulate the problem as a first order formula of size depending only on k and then invoke the result of Frick and Grohe [40] stating that such problems are FPT. Therefore, as fixed-parameter tractability is easy to establish on planar graphs, the challenge here is to obtain optimal dependence on k. One would expect that sub-exponential dependence on k (e.g., $2^{O(\sqrt{k})}$ or $k^{O(\sqrt{k})}$) should be possible at least for SCSS, but this is not yet fully understood even for undirected STEINER TREE [63]. A slightly different parameterization is to consider the number k of *non-terminal* vertices in the solution, which can be much smaller than the number of terminals. This leads to problems of somewhat different flavour, see e.g. [30, 48].

1.3 Further related work

Subsequent to the conference version [19] of this paper, there have been several related results. Chitnis et al. [16] considered a variant of SCSS with only 2 terminals but with a requirement of multiple paths. Formally, in the 2-SCSS-(k_1,k_2) problem we are given two vertices s,t and the goal is to find a min weight subset $H \subseteq E(G)$ such that H has k_1,k_2 paths from $s \rightsquigarrow t, t \rightsquigarrow s$, respectively. The objective function is given by $\text{cost}(H) = \sum_{e \in H} \phi(e) \cdot \text{cost}(e)$ where $\phi(e)$ is the maximum number of times e appears on $s \rightsquigarrow t$ paths and $t \rightsquigarrow s$ paths. Chitnis et al. [16] showed that the 2-SCSS-$(k,1)$ problem can be solved in $n^{O(k)}$ time for any $k \geq 1$, and has a $f(k) \cdot n^{o(k)}$ lower bound under ETH.

Suchý [68] introduced a generalization of DST and SCSS called the q-ROOT STEINER TREE (q-RST) problem. In this problem, given a set of q roots and a set of k leaves, the task is to find a minimum-cost network where the roots are in the same strongly connected component and every leaf can be reached from every root. Generalizing the token game of Feldman and Ruhl [35], Suchý [68] designed a $2^{O(q)} \cdot n^{O(k)}$ algorithm for q-RST.

Recently, Chitnis et al. [18] considered the SCSS and DSN problems on bidirected graphs: these are directed graphs with the guarantee that for every edge (u,v) the reverse edge (v,u) exists and has the same weight. They showed that on bidirected graphs, the DSN problem stays W[1]-hard parameterized by k but SCSS becomes FPT (while still being NP-hard). In fact, under ETH, no $f(k)n^{o(k/\log k)}$ time algorithm for DSN on bidirected graphs exists, and thus the problem is essentially as hard as for general directed graphs. For bidirected planar graphs however, Chitnis et al. [18] show that DSN can be solved in $2^{O(k^{3/2}\log k)}n^{O(\sqrt{k})}$, which is in contrast to Theorem 1.4. Some FPT approximability and inapproximability results for SCSS and DSN were also shown in [17, 18].

Pattern graphs and DSN: The set of pairs $\{(s_i,t_i) \ : \ i \in [k]\}$ in the input of DSN can be interpreted as a directed (unweighted) pattern graph on a set $R = \bigcup_{i=1}^{k} \{s_i,t_i\}$ of terminals. For a graph class $\mathcal{H}$, the $\mathcal{H}$-DSN problem takes as input a directed graph $H \in \mathcal{H}$ on vertex set R and the goal is to find a minimum cost subgraph $N \subseteq E(G)$ such that N has an $s \rightsquigarrow t$ path for each $(s,t) \in E(H)$. Thus for a fixed class $\mathcal{H}$ of pattern graphs, the $\mathcal{H}$-DSN problem is a restricted special case of the general DSN problem, and it is possible that $\mathcal{H}$-DSN is FPT (for example, if $\mathcal{H}$ is the class of out-stars). Feldmann and Marx [36] gave a complete dichotomy for which graph classes the $\mathcal{H}$-DSN problem is FPT or W[1]-hard parameterized by $|R|$.

Given an instance of DSN with the pattern graph $H = (R,A)$ on the terminal set R with $|A| = k$, the algorithm of Feldman and Ruhl [35] runs in $n^{O(k)}$ time. The $f(k) \cdot n^{o(k)}$ lower bound under ETH for DSN

in this paper (Theorem 1.4) has $|A| = O(|R|)$. Hence, for the parameter $|R|$ we have a lower bound of $f(|R|) \cdot n^{o(|R|)}$ and an upper bound of $n^{O(|R|^2)}$ (since $|A| = O(|R|^2)$ in the worst case). Recently, Eiben et al. [31] essentially closed this gap by showing a lower bound of $f(|R|) \cdot n^{o(|R|^2/\log|R|)}$ under ETH for DSN. They also gave an algorithm for DSN on bounded genus graphs: for graphs of genus g, the algorithm runs in $f(|R|) \cdot n^{O_g(|R|)}$ time where $O_g(\cdot)$ hides constants depending only on g.

2 Improved algorithm for SCSS on planar graphs

In this section we describe the proof to Theorem 1.1, i.e., we present an algorithm to solve SCSS on planar graphs in $2^{O(k)} \cdot n^{O(\sqrt{k})}$ time. The definitions of some of the graph-theoretic notions used in this section such as treewidth and minors are deferred to Appendix B to maintain the flow of presentation. The key is to analyze the structure of *edge-minimal solutions*, i.e., subgraphs of the input graph G (induced by some set $S \subseteq V$) containing all terminals for which no edge can be removed without also removing all $s \rightsquigarrow t$ paths for some terminal pair (s,t). We show that for an edge-minimal solution M of the SCSS problem there is a vertex set $W \subseteq V(M)$ of size $O(k)$ such that, after removing W from M, each component has constant treewidth. More formally, we define a *W_M-component* as a connected component of the (underlying undirected) graph induced by $V(M) \setminus W$ in M, and prove the following.

Lemma 2.1. *For any edge-minimal solution M to the edge-weighted SCSS problem there is a set of at most $9k$ vertices $W \subseteq V(M)$ for which every W_M-component has treewidth at most 2.*

We defer the proof of Lemma 2.1 to Section 2.1. First, we see how we can use Lemma 2.1 to bound the treewidth of the minimal solution M.

Lemma 2.2. *If an edge-minimal solution M to edge-weighted SCSS is planar (or excludes some fixed minor), then its treewidth is $O(\sqrt{k})$.*

Proof. By the planar grid theorem [66], there is a constant c_{Planar} such that any planar graph G with treewidth $c_{\text{Planar}} \cdot \omega$ has a $\omega \times \omega$ grid minor. If the treewidth of M is at least $c_{\text{Planar}} \cdot \lceil 20\sqrt{k} \rceil$, then it follows that M has a $\lceil 20\sqrt{k} \rceil \times \lceil 20\sqrt{k} \rceil$ grid minor M'. It is easy to see that M' contains at least $\lfloor \frac{\lceil 20\sqrt{k} \rceil}{3} \rfloor \cdot \lfloor \frac{\lceil 20\sqrt{k} \rceil}{3} \rfloor$ (pairwise vertex-disjoint) grids of size 3×3. For each $k \geq 1$, one can easily verify that $\lfloor \frac{\lceil 20\sqrt{k} \rceil}{3} \rfloor \geq \lceil 4\sqrt{k} \rceil$, and hence the number of pairwise vertex-disjoint 3×3 grids is at least $\lceil 4\sqrt{k} \rceil \cdot \lceil 4\sqrt{k} \rceil \geq 4\sqrt{k} \cdot 4\sqrt{k} = 16k$. By Lemma 2.1, there is a set of vertices W of size $9k$ whose deletion makes every W_M-component have treewidth at most 2. Since $16k > 9k$, it follows that W does not contain a vertex from at least one of the (pairwise vertex-disjoint) $16k$ grid minors of size 3×3 in M. Hence, there is a W_M-component, which contains a 3×3 grid minor, and hence has treewidth at least 3, which is a contradiction.

For the case when the input graph is H-minor-free for some fixed graph H, we can instead use the excluded grid-minor theorem of Demaine and Hajiaghayi [25]: for any fixed graph H, there is a constant c_H (which depends only on $|H|$) such that any H-minor-free graph of treewidth at least $c_H \cdot \omega$ has a $\omega \times \omega$ grid as a minor. □

To prove Theorem 1.1, which is restated below, we invoke an algorithm of [36] to find the optimum solution of bounded treewidth. The algorithm of [36] is designed for the edge-weighted version, and we state below the corresponding statement for the more general unweighted vertex version (so that it may also be of future use).

Theorem 2.3. ***(generalization of [36, Theorem 5])*** *If there is an optimum solution to an instance on k terminals of the vertex-weighted version of SCSS which has treewidth at most ω, then an optimum solution*[3] *can be found in $2^{O(k+\omega\cdot\log\omega)}\cdot n^{O(\omega)}$ time.*

Proof. In the given graph G, we start by subdividing each edge by adding a non-terminal vertex of weight 0 (note that this does not increase the treewidth). Let us call these vertices we have added as dummy vertices, and the graph obtained at this point be G^*. Note that each dummy vertex has in-degree one and out-degree one. Now we reduce the vertex-weighted version of SCSS to the edge-weighted version, using a standard reduction: substitute each non-terminal vertex $u \in G$ of weight W with two new non-terminal vertices u^- and u^+ and an edge (u^-, u^+) of the same weight W. Every edge that had u as its head will now have u^- as its head instead, and every edge that had u as its tail will now have u^+ as its tail. We set the weight of all these edges to be zero. Let the graph obtained after these modifications be G'.

Consider an optimum solution S for the vertex-weighted version of SCSS, and without loss of generality we can assume that S is minimal under vertex deletions (if it is not, then make it minimal by deleting unnecessary vertices). Let $S' = (S \cap T) \cup \{\{u^-, u^+\} : u \in S \cap (V \setminus T)\}$. We now show that the induced graph $G'[S']$ is an edge-minimal solution (with same weight as that of S) for the edge-weighted version of SCSS: we do this by showing that deletion of any edge from $G'[S']$ creates a non-terminal source or a non-terminal sink which contradicts the fact that S was a vertex-minimal solution for vertex-weighted version of SCSS. The construction of the graph G' from G implies that any edge e in G must be of either of the following two types:

- Without loss of generality[4], the edge is (y, v^-) for some dummy vertex y and some non-terminal $v \in G$ in which case deleting this edge makes the non-terminal y to be a sink.
- The edge is (z^-, z^+) for some non-terminal $z \in G$ in which case deleting this edge makes the non-terminal z^+ a source and the non-terminal z^- a sink.

Note that G' is not necessarily planar (or H-minor-free) even if G is. However, the treewidth of $G'[S']$ is at most twice the treewidth of $G[S]$ since we can simply replace each non-terminal vertex u in the bags of the tree decomposition of N by the two vertices u^- and u^+. Feldmann and Marx [36, Theorem 5] showed that if the optimum solution to an instance on k terminals of the edge-weighted version of SCSS has treewidth ω, then it can be found in $2^{O(k+\omega\cdot\log\omega)}\cdot n^{O(\omega)}$ time. Hence, the claimed running time for the vertex-weighted version follows. □

Finally, we are now ready to prove Theorem 1.1

Theorem 1.1. *An instance (G,T) of the vertex-weighted* STRONGLY CONNECTED STEINER SUBGRAPH *problem with $|G| = n$ and $|T| = k$ can be solved in $2^{O(k)}\cdot n^{O(\sqrt{k})}$ time, when the underlying undirected graph of G is planar (or more generally, H-minor-free for any fixed graph H).*

Proof. Consider a subgraph M of G induced by the optimum solution $S \subseteq V$, which is also minimal, i.e., no edge of M can be removed without destroying the connectivity between some terminal pair (s,t). By Lemma 2.2 we know that the treewidth of M is $O(\sqrt{k})$. Hence, the claimed running time follows from Theorem 2.3. □

Note that Lemma 2.2 only used the planarity (or H-minor-freeness) of M, and not of the input graph. Hence, the algorithm of Theorem 1.1 also works for the weaker restriction of finding an optimal planar (or H-minor-free) solution in an otherwise unrestricted input graph, rather than finding an optimal solution in a planar (or H-minor-free respectively) graph. It only remains to prove Lemma 2.1, which is done in the next section.

[3] Not necessarily the same optimum solution as the one mentioned in the first part of this theorem. For example, the actual optimum found by this algorithm might have treewidth much larger than ω.

[4] The other case is the edge being (v^+, y) for some dummy vertex y and some non-terminal $v \in G$

2.1 Proof of Lemma 2.1

Fix an arbitrary terminal $r \in T$. It is easy to see (observed for example by Feldman and Ruhl [35]) that any minimal SCSS solution M is the union of an in-arborescence $A_{\texttt{in}}$ and an out-arborescence $A_{\texttt{out}}$, both having the same root $r \in T$ and only terminals as leaves, since every terminal of T can be reached from r, and conversely every terminal can reach r in M. We construct the set W_M by including three different kinds of vertices. First, W_M contains every *branching point* of $A_{\texttt{in}}$ and $A_{\texttt{out}}$, i.e. every vertex with in-degree at least 2 in $A_{\texttt{in}}$ and every vertex with out-degree at least 2 in $A_{\texttt{out}}$. Since $A_{\texttt{in}}$ and $A_{\texttt{out}}$ are arborescences with at most k leaves (the terminals), they each have at most k branching points. Secondly, W_M contains all terminals of T, which adds another k vertices to the set W_M. The third kind of vertices in W_M is the following. Note that every component of the intersection of $A_{\texttt{in}}$ and $A_{\texttt{out}}$ forms a path (possibly consisting only of a single vertex), since every vertex of $A_{\texttt{in}}$ has out-degree at most 1, while every vertex of $A_{\texttt{out}}$ has in-degree at most 1. We call such a component a *shared path*. If a shared path contains a branching point or a terminal, we add the endpoints of the shared path to W_M. For a branching point or terminal v on such a shared path, we can map the endpoints of the shared path to v. This maps at most two endpoints of shared paths to each branching point or terminal, so that the number of vertices of the third kind in W_M is at most $6k$ (as there are k terminals and at most $2k$ branching points). Thus the total size of W_M is at most $9k$.

We claim that every W_M-component consists of at most two interacting paths, one from $A_{\texttt{in}}$ and one from $A_{\texttt{out}}$. More formally, consider a $u \to v$ path P of $A_{\texttt{in}}$ such that u and v are either terminals or branching points of $A_{\texttt{in}}$, and such that no internal vertex of P is a terminal or branching point of $A_{\texttt{in}}$. We call any such path P an *essential path of* $A_{\texttt{in}}$. Note that we ignore the branching points of $A_{\texttt{out}}$ in this definition, and that the edge set of the arborescence $A_{\texttt{in}}$ is the disjoint union of the edge sets of its essential paths. Analogously we define the *essential paths of* $A_{\texttt{out}}$ as those $u \to v$ paths P in $A_{\texttt{out}}$ for which u and v are terminals or branching points of $A_{\texttt{out}}$, and no internal vertices of P are of such a type.

Claim 2.4. *Every W_M-component contains edges of at most two essential paths, one from A_{in} and one from A_{out}.*

Proof. Any vertex at which two essential paths of the same arborescence intersect is a terminal or branching point. These vertices are in W_M and therefore not contained in any W_M-component. Thus if a W_M-component H contains at least two essential paths then they either coincide on every edge of H, in which case the claim is clearly true, or H contains the endpoint v of a shared path, i.e., there are two essential paths, one from each arborescence, that both contain vertex v. We will show that there is only one pair of essential paths that can meet at an endpoint of a shared path in H, from which the claim follows.

In order to prove this, we label every essential $u \to v$ path P of $A_{\texttt{in}}$ with those terminals $T_P \subseteq T$ that can reach the start vertex of P in the in-arborescence, i.e. $t \in T_P$ if and only if there exists a $t \to u$ path in $A_{\texttt{in}}$. Note that no two essential paths of $A_{\texttt{in}}$ can have the same label. We also label any essential $u \to v$ path Q of $A_{\texttt{out}}$ analogously, by setting the label $T_Q \subseteq T$ to be the terminals which can be reached from the end vertex of Q in the out-arborescence, i.e. there is a $v \to t$ path in $A_{\texttt{out}}$ if and only if $t \in T_Q$. Even though no two essential paths of an individual arborescence have the same label, there can be pairs of essential paths from $A_{\texttt{in}}$ and $A_{\texttt{out}}$ with the same label. Let P and Q be essential paths of $A_{\texttt{in}}$ and $A_{\texttt{out}}$, respectively. We prove that if P and Q meet at an endpoint v of a shared path, then $v \in W_M$ or $T_P = T_Q$.

Assume this is not the case so that $v \notin W_M$ and $T_P \neq T_Q$. Let I be the shared path in the intersection of $A_{\texttt{in}}$ and $A_{\texttt{out}}$ for which v is an endpoint. If u is the other endpoint of I, assume w.l.o.g. that I is a $u \to v$ path (the other case is symmetric). If there were any branching points or terminals on I then $v \in W_M$, since v would then be one of the third kind of vertices in W_M. As this is not the case, I lies in the intersection of P and Q, there are edges $e_v \in E(P)$ and $f_v \in E(Q)$ leaving v such that $e_v \notin E(A_{\texttt{out}})$ and $f_v \notin E(A_{\texttt{in}})$, and there are edges $e_u \in E(P)$ and $f_u \in E(Q)$ entering u such that $e_u \notin A_{\texttt{out}}$ and $f_u \notin E(A_{\texttt{in}})$.

As $T_P \neq T_Q$ there is a terminal t contained in one of the two sets but not the other. Consider the case

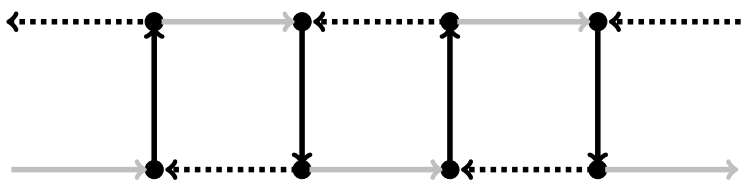

Figure 1: Let H be a W_M-component M such that at least two shared paths of M intersect with H. By Claim 2.4, H contains edges of two essential paths P and Q of $A_{\texttt{in}}$ and $A_{\texttt{out}}$, respectively. Claim 2.5 shows that H looks roughly as shown in the figure: let P, Q be denoted by light+black paths and dotted+black paths respectively. Then the black paths are exactly the shared paths. Note though that a shared path may have length 0.

when $t \in T_Q \setminus T_P$, i.e. there is a $v \to t$ path in $A_{\texttt{out}}$ but no $t \to u$ path in $A_{\texttt{in}}$. The latter implies that e_v cannot be reached from t in $A_{\texttt{in}}$, as the $u \to v$ path I contains no branching point of $A_{\texttt{in}}$. The in-arborescence $A_{\texttt{in}}$ does however contain a $t \to r$ path from t to the root r. Since $e_v \notin E(A_{\texttt{out}})$, this means that the root r can be reached from v through the $v \to t$ path of $A_{\texttt{out}}$ and the $t \to r$ path without passing through e_v. Hence e_v can safely be removed without making the solution M infeasible. This contradicts the minimality of M.

In case $t \in T_P \setminus T_Q$ a symmetric argument shows that the edge f_u is redundant in M, which again contradicts its minimality. We have thus shown that P and Q are the only essential paths that meet in any endpoint of a shared path in the W_M-component H. Hence H consists of exactly these two paths P and Q, and the claim follows. ⌟

Consider the case when there is at most one shared path of M that intersects with a W_M-component H. Since by Claim 2.4, H consists of at most two essential paths, it is easy to see that in this case H is a tree, and thus its treewidth is 1. If at least two shared paths of M intersect with H, by Claim 2.4, H contains edges of two essential paths P and Q of $A_{\texttt{in}}$ and $A_{\texttt{out}}$ respectively. To show that in this case the treewidth of H is at most 2, we need the following observation on P and Q:

Claim 2.5. *Let $I_1, \ldots, I_h$ be the connected components in the intersection of P and Q, ordered in the way that P visits them, i.e. for any $i \in \{1, \ldots, h-1\}$ there is a subpath of P with prefix I_i and suffix I_{i+1}. The path Q visits the shared paths in the opposite order, i.e. for any $i \in \{1, \ldots, h-1\}$ there is a subpath of Q with suffix I_i and prefix I_{i+1}.*

Proof. Assume this is not the case, so that there is an index $i \in \{1, \ldots, h-1\}$ such that both P and Q contain subpaths with prefix I_i and suffix I_{i+1}. This means that there are edges $e \in E(P) \setminus E(Q)$ and $f \in E(Q) \setminus E(P)$ which share the last vertex u of I_i. Hence Q contains a $u \to v$ subpath Q' to the first vertex v of I_{i+1}, which does not contain the edge e, and also P contains a $u \to v$ subpath P', which does not contain the edge f. As Q, and therefore also Q', contains no branching point of $A_{\texttt{out}}$, any terminal reachable from u through Q' in $A_{\texttt{out}}$ is also reachable from u via the detour P'. We can therefore remove edge $f \in E(A_{\texttt{out}})$ without violating the feasibility of M. This however contradicts its minimality. ⌟

Claim 2.5 implies that the structure of H is roughly as shown in Figure 1 (the black edges shown in Figure 1 correspond to paths of length 0 or more, while the light and dotted edges correspond to paths of length at least 1). If we contract each path of length at least 1 to a path of length 1, then the resulting graph is planar and all vertices belong to the outer face. Such graphs are called outerplanar graphs. In other words, H is a subdivision of an outerplanar graph. Lemma B.4 shows that treewidth of subdivisions of outerplanar graphs is at most 2, which proves Lemma 2.1.

3 W[1]-hardness for SCSS in planar graphs

The goal of this section is to prove Theorem 1.2. We reduce from the GRID TILING problem[5] introduced by Marx [56]:

> $k \times k$ **GRID TILING**
> *Input* : Integers k, n, and k^2 non-empty sets $S_{i,j} \subseteq [n] \times [n]$ where $1 \leq i, j \leq k$
> *Question*: For each $1 \leq i, j \leq k$ does there exist an entry $\gamma_{i,j} \in S_{i,j}$ such that
>
> - If $\gamma_{i,j} = (x, y)$ and $\gamma_{i,j+1} = (x', y')$ then $x = x'$.
> - If $\gamma_{i,j} = (x, y)$ and $\gamma_{i+1,j} = (x', y')$ then $y = y'$.

Under ETH [45, 46], it was shown by Chen et al. [15] that k-CLIQUE[6] does not admit an algorithm running in time $f(k) \cdot n^{o(k)}$ for any computable function f. There is a simple reduction [23, Theorem 14.28] from k-CLIQUE to $k \times k$ GRID TILING implying the same runtime lower bound for the latter problem. To prove Theorem 1.2, we give a reduction which transforms the problem of $k \times k$ GRID TILING into a planar instance of SCSS with $O(k^2)$ terminals. We design two types of gadgets: the *connector gadget* and the *main gadget*. The reduction from GRID TILING represents each cell of the grid with a copy of the main gadget, with a connector gadget between main gadgets that are adjacent either horizontally or vertically (see Figure 2).

The proof of Theorem 1.2 is divided into the following steps: In Section 3.1, we first introduce the connector gadget and Lemma 3.3 states the existence of a particular type of connector gadget. In Section 3.2, we introduce the main gadget and Lemma 3.6 states the existence of a particular type of main gadget. Section 3.3 describes the construction of the planar instance (G^*, T^*) of SCSS. The two directions implying the reduction from GRID TILING to SCSS are proved in Section 3.4 and Section 3.5 respectively. Using Lemmas 3.3 and 3.6 as a blackbox, we prove Theorem 1.2 in Section 3.6. The proofs of Lemmas 3.3 and Lemma 3.6 are given in Sections 4 and 5 respectively.

3.1 Existence of connector gadgets

A connector gadget CG_n is a directed (embedded) planar graph with $O(n^2)$ vertices and positive integer weights[7] on its edges. It has a total of $2n+2$ distinguished vertices divided into the following 3 types:

- The vertices p, q are called *internal-distinguished* vertices
- The vertices $p_1, p_2, \ldots, p_n$ are called *source-distinguished* vertices
- The vertices $q_1, q_2, \ldots, q_n$ are called *sink-distinguished* vertices

Let $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$. The vertices $P \cup Q$ appear in the clockwise order $p_1, \ldots, p_n, q_n, \ldots, q_1$ on the boundary of the gadget. In the connector gadget CG_n, every vertex in P is a source and has exactly one outgoing edge. Also every vertex in Q is a sink and has exactly one incoming edge.

Definition 3.1. *An edge set $E' \subseteq E(CG_n)$ satisfies the* **connectedness** *property if each of the following four conditions hold for the graph $CG_n[E']$:*

[5] The GRID TILING problem has been defined in two (symmetrical) ways in the literature: either the first coordinate or the second coordinate remains the same in a row. Here, we follow the notation of [56], but the other definition also appears in some places (e.g. [23]).

[6] The k-CLIQUE problem asks whether there is a clique of size $\geq k$.

[7] Weights are polynomial in n.

1. *p can be reached from some vertex in P*
2. *q can be reached from some vertex in P*
3. *p can reach some vertex in Q*
4. *q can reach some vertex in Q*

Definition 3.2. *An edge set E' satisfying the connectedness property* ***represents*** *an integer $i \in [n]$ if in E' the only outgoing edge from P is the one incident to p_i and the only incoming edge into Q is the one incident to q_i.*

The next lemma shows we can construct a particular type of connector gadget:

Lemma 3.3. *Given an integer n one can construct in polynomial time a connector gadget CG_n and an integer C_n^* such that the following two properties hold* [8]*:*

1. *For every $i \in [n]$, there is an edge set $E_i \subseteq E(CG_n)$ of weight C_n^* such that E_i satisfies the connectedness property and represents i. Note that, in particular, E_i contains a $p_i \rightsquigarrow q_i$ path (via p or q).*
2. *If there is an edge set $E' \subseteq E(CG_n)$ such that E' has weight at most C_n^* and E' satisfies the connectedness property, then E' has weight exactly C_n^* and it represents some $i \in [n]$.*

3.2 Existence of main gadgets

A main gadget MG is a directed (embedded) planar graph with $O(n^3)$ vertices and positive integer weights on its edges. It has $4n$ distinguished vertices given by the following four sets:

- The set $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of *left-distinguished* vertices.
- The set $R = \{r_1, r_2, \ldots, r_n\}$ of *right-distinguished* vertices.
- The set $T = \{t_1, t_2, \ldots, t_n\}$ of *top-distinguished* vertices.
- The set $B = \{b_1, b_2, \ldots, b_n\}$ of *bottom-distinguished* vertices.

The distinguished vertices appear in the (clockwise) order $t_1, \ldots, t_n, r_1, \ldots, r_n, b_n, \ldots, b_1, \ell_n, \ldots, \ell_1$ on the boundary of the gadget. In the main gadget MG, every vertex in $L \cup T$ is a source and has exactly one outgoing edge. Also each vertex in $R \cup B$ is a sink and has exactly one incoming edge.

Definition 3.4. *An edge set $E' \subseteq E(MG)$ satisfies the* **connectedness** *property if each of the following four conditions hold for the graph $MG[E']$:*

1. *There is a directed path from some vertex in L to $R \cup B$*
2. *There is a directed path from some vertex in T to $R \cup B$*
3. *Some vertex in R can be reached from $L \cup T$*
4. *Some vertex in B can be reached from $L \cup T$*

[8] We use the notation C_n^* to emphasize that C^* depends only on n.

Definition 3.5. *An edge set $E' \subseteq E(MG)$* ***represents*** *a pair $(i,j) \in [n] \times [n]$ if each of the following five conditions holds:*

- *The only edge of E' leaving L is the one incident to ℓ_i*
- *The only edge of E' entering R is the one incident to r_i*
- *The only edge of E' leaving T is the one incident to t_j*
- *The only edge of E' entering B is the one incident to b_j*
- *E' contains an $\ell_i \rightsquigarrow r_i$ path and an $t_j \rightsquigarrow b_j$ path*

The next lemma shows we can construct a particular type of main gadget:

Lemma 3.6. *Given a subset $S \subseteq [n] \times [n]$, one can construct in polynomial time a main gadget MG_S and an integer M_n^* such that the following two properties hold* [9]*:*

1. *For every $(i,j) \in S$ there is an edge set $E_{i,j} \subseteq E(MG_S)$ of weight M_n^* such that $E_{i,j}$ represents (i,j). Note that the last condition of Definition 3.5 implies that $E_{i,j}$ satisfies the connectedness property.*
2. *If there is an edge set $E' \subseteq E(MG_S)$ such that E' has weight at most M_n^* and satisfies the connectedness property, then E' has weight exactly M_n^* and represents some $(i,j) \in S$.*

3.3 Construction of the SCSS instance

In order to prove Theorem 1.2, we reduce from the GRID TILING problem. The following assumption will be helpful in handling some of the border cases of the gadget construction. We may assume that $1 < x, y < n$ holds for every $(x,y) \in S_{i,j}$: indeed, we can increase n by two and replace every (x,y) by $(x+1, y+1)$ without changing the problem. This is just a minor technical modification[10] which is introduced to make some of the arguments easier in Section 5 cleaner.

Given an instance $(k, n, \{S_{i,j} \,:\, i,j \in [k]\})$ of GRID TILING, we construct an instance (G^*, T^*) of SCSS the following way (see Figure 2):

- We introduce a total of k^2 main gadgets and $2k(k+1)$ connector gadgets.
- For every set $S_{i,j}$ in the GRID TILING instance, we construct a main gadget $MG_{i,j}$ using Lemma 3.6 for the subset $S_{i,j}$.
- Half of the connector gadgets have the same orientation as in Figure 3 (with the p_i vertices on the top side and the q_i vertices on the bottom side), and we call them HCG to denote *horizontal connector gadgets*[11]. The other half of the connector gadgets are rotated anti-clockwise by 90 degrees with respect to the orientation of Figure 3, and we call them VCG to denote *vertical connector gadgets*. The internal-distinguished vertices of the connector gadgets are shown in Figure 2.
- For each $1 \le i, j \le k$, the main gadget $MG_{i,j}$ is surrounded by the following four connector gadgets:

[9] We use the notation M_n^* to emphasize that M^* depends only on n, and not on the set S.

[10] For the interested reader, what this modification does is to ensure no shortcut edge added in Section 5.1 has either endpoint on the unbounded face of the planar embedding of the main gadget provided in Figure 4. This helps to streamline the proofs by avoiding the need to have to consider any special cases.

[11] The horizontal connector gadgets are so called because they connect things horizontally as seen by the reader.

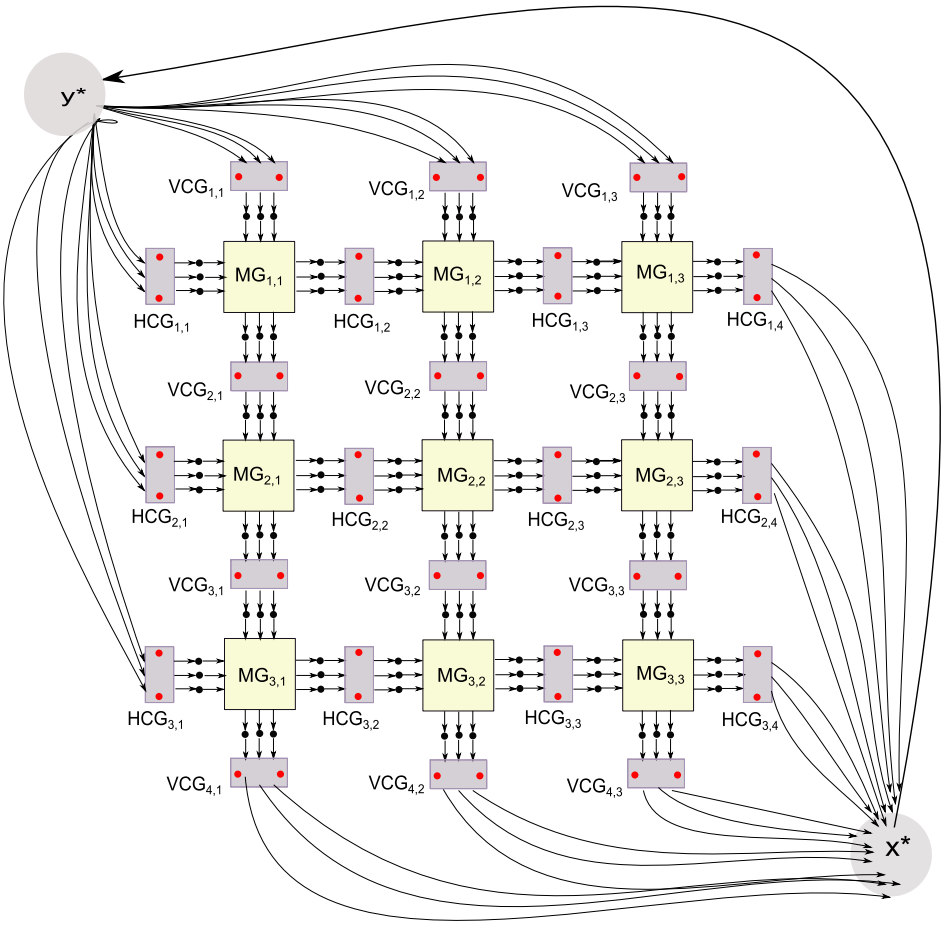

Figure 2: An illustration of the reduction from GRID TILING to SCSS on planar graphs.

1. The *vertical connector gadgets* $VCG_{i,j}$ is on the top and $VCG_{i+1,j}$ is on the bottom. Identify (or glue together) each sink-distinguished vertex of $VCG_{i,j}$ with the top-distinguished vertex of $MCG_{i,j}$ of the same index. Similarly identify each source-distinguished vertex of $VCG_{i+1,j}$ with the bottom-distinguished vertex of $MCG_{i,j}$ of the same index.
2. The *horizontal connector gadgets* $HCG_{i,j}$ is on the left and $HCG_{i,j+1}$ is on the right. Identify (or glue together) each sink-distinguished vertex of $HCG_{i,j}$ with the left-distinguished vertex of $MCG_{i,j}$ of the same index. Similarly, identify each source-distinguished vertex of $HCG_{i,j+1}$ with the right-distinguished vertex of $MCG_{i,j}$ of the same index.

- We introduce two special vertices x^*, y^* and add an edge (x^*, y^*) of weight 0.
- For each $1 \leq i \leq k$, add an edge of weight 0 from y^* to each source-distinguished vertex of the vertical connector gadget $VCG_{1,i}$.
- For each $1 \leq j \leq k$, add an edge of weight 0 from y^* to each source-distinguished vertex of the horizontal connector gadget $HCG_{j,1}$.
- For each $1 \leq i \leq k$, add an edge of weight 0 from each sink-distinguished vertex of the vertical connector gadget $VCG_{k+1,i}$ to x^*.

- For each $1 \leq j \leq k$, add an edge of weight 0 from each sink-distinguished vertex of the horizontal connector gadget $HCG_{j,k+1}$ to x^*.
- For each $i \in [k], j \in [k+1]$, denote the two internal-distinguished vertices of $HCG_{i,j}$ by $\{p^h_{i,j}, q^h_{i,j}\}$
- For each $i \in [k+1], j \in [k]$, denote the two internal-distinguished vertices of $VCG_{i,j}$ by $\{p^v_{i,j}, q^v_{i,j}\}$
- The set of terminals T^* for the SCSS instance on G^* is $\{x^*, y^*\} \cup \{p^h_{i,j}, q^h_{i,j} \mid 1 \leq i \leq k+1, 1 \leq j \leq k\} \cup \{p^v_{i,j}, q^v_{i,j} \mid 1 \leq i \leq k, 1 \leq j \leq k+1\}$.
- We note that the total number of terminals is $|T^*| = 4k(k+1)+2 = O(k^2)$
- The edge set of G^* is a disjoint union of
  - Edges of main gadgets
  - Edges of horizontal connector gadgets
  - Edges of vertical connector gadgets
  - Edges from y^* to source-distinguished vertices of the vertical connector gadgets $VCG_{1,i}$ (for each $i \in [k]$), and from y^* to source-distinguished vertices of horizontal connector gadgets $HCG_{j,1}$ (for each $j \in [k]$)
  - Edges from sink-distinguished vertices of the vertical connector gadgets $VCG_{k+1,i}$ (for each $i \in [k]$) to x^*, and from sink-distinguished vertices of horizontal connector gadgets $HCG_{j,k+1}$ (for each $j \in [k]$) to x^*
  - The edge (x^*, y^*)

Define the following quantity:

$$W^*_n = k^2 \cdot M^*_n + 2k(k+1) \cdot C^*_n. \tag{1}$$

In the next two sections, we show that GRID TILING has a solution if and only if the SCSS instance (G^*, T^*) has a solution of weight at most W^*_n.

3.4 GRID TILING has a solution $\Rightarrow$ SCSS has a solution of weight $\leq W^*_n$

Lemma 3.7. *If the* GRID TILING *instance* $(k, n, \{S_{i,j} \,:\, i, j \in [k]\})$ *has a solution, then the SCSS instance* (G^*, T^*) *has a solution of weight at most* W^*_n.

Proof. Since GRID TILING has a solution, for each $1 \leq i, j \leq k$ there is an entry $(x_{i,j}, y_{i,j}) = \gamma_{i,j} \in S_{i,j}$ such that

- For every $i \in [k]$, we have $x_{i,1} = x_{i,2} = x_{i,3} = \ldots = x_{i,k} = \alpha_i$
- For every $j \in [k]$, we have $y_{1,j} = y_{2,j} = y_{3,j} = \ldots = y_{k,j} = \beta_j$

We build a solution E^* for the SCSS instance (G^*, T^*) and show that it has weight at most W^*_n. In the edge set E^*, we take the following edges:

1. The edge (x^*, y^*) which has weight 0.
2. For each $j \in [k]$ the edge of weight 0 from y^* to the source-distinguished vertex of $VCG_{1,j}$ of index β_j, and the edge of weight 0 from the sink-distinguished vertex of $VCG_{k+1,j}$ of index β_j to x^*.

3. For each $i \in [k]$ the edge of weight 0 from y^* to the source-distinguished vertex of $HCG_{i,1}$ of index α_i, and the edge of weight 0 from the sink-distinguished vertex of $HCG_{i,k+1}$ of index α_i to x^*

4. For each $1 \leq i, j \leq k$ for the main gadget $MG_{i,j}$, use Lemma 3.6(1) to generate a solution $E^M_{i,j}$ which has weight M^*_n and represents (α_i, β_j).

5. For each $1 \leq i \leq k$ and $1 \leq j \leq k+1$ for the horizontal connector gadget $HCG_{i,j}$, use Lemma 3.3(1) to generate a solution $E^{HC}_{i,j}$ of weight C^*_n which represents α_i.

6. For each $1 \leq j \leq k$ and $1 \leq i \leq k+1$ for the vertical connector gadget $VCG_{i,j}$, use Lemma 3.3(1) to generate a solution $E^{VC}_{i,j}$ of weight C^*_n which represents β_j.

The weight of E^* is $k^2 \cdot M^*_n + k(k+1) \cdot C^*_n + k(k+1) \cdot C^*_n = W^*_n$. It remains to show that E^* is a solution for the SCSS instance (G^*, T^*). Since we have already picked up the edge (x^*, y^*), it is enough to show that for any terminal $t \in T^* \setminus \{x^*, y^*\}$, both $t \rightsquigarrow x^*$ and $y^* \rightsquigarrow t$ paths exist in E^*. Then for any two terminals t_1, t_2, there is a $t_1 \rightsquigarrow t_2$ path given by $t_1 \rightsquigarrow x^* \to y^* \rightsquigarrow t_2$.

We now show the existence of both a $t \rightsquigarrow x^*$ path and a $y^* \rightsquigarrow t$ path in E^* when t is a terminal in a vertical connector gadget. Without loss of generality, let t be the terminal $p^v_{i,j}$ for some $1 \leq i \leq k, 1 \leq j \leq k+1$.

- Existence of $p^v_{i,j} \rightsquigarrow x^*$ path in E^*: By Lemma 3.3(1), the terminal $p^v_{i,j}$ can reach the sink-distinguished vertex of $VCG_{i,j}$ which has the index β_j. This vertex is the top-distinguished vertex of the index β_j of the main gadget $MG_{i,j}$. By Definition 3.5, there is a path from this vertex to the bottom-distinguished vertex of the index β_j of the main gadget $MG_{i,j}$. However this vertex is exactly the same as the source-distinguished vertex of the index β_j of $VCG_{i+1,j}$. By Lemma 3.3(1), the source-distinguished vertex of the index β_j of $VCG_{i+1,j}$ can reach the sink-distinguished vertex of the index β_j of $VCG_{i+1,j}$. This vertex is exactly the top-distinguished vertex of $MG_{i+1,j}$. Continuing in this way we can reach the source-distinguished vertex of the index β_j of $VCG_{k+1,j}$. By Lemma 3.3(1), this vertex can reach the sink-distinguished vertex of the index β_j of $VCG_{k+1,j}$. But E^* also contains an edge (of weight 0) from this sink-distinguished vertex to x^*, and hence there is a $p^v_{i,j} \rightsquigarrow x^*$ path in E^*.

- Existence of $y^* \rightsquigarrow p^v_{i,j}$ path in E^*: Recall that E^* contains an edge (of weight 0) from y^* to the source-distinguished vertex of the index β_j of $VCG_{1,j}$. If $i = 1$, then by Lemma 3.3(1), there is a path from this vertex to $p^v_{1,j}$. If $i \geq 2$, then by Lemma 3.3(1), there is a path from source-distinguished vertex of the index β_j of $VCG_{1,j}$ to the sink-distinguished vertex of the index β_j of $VCG_{1,j}$. But this is the top-distinguished vertex of $MG_{1,j}$ of the index β_j. By Definition 3.5, from this vertex we can reach the bottom-distinguished vertex of the index β_j of $MG_{1,j}$. However, this vertex is exactly the source-distinguished vertex of index β_j of $VCG_{2,j}$. Continuing this way we can reach the source-distinguished vertex of the index β_j of $VCG_{i,j}$. By Lemma 3.3(1), from this vertex we can reach $p^v_{i,j}$. Hence there is a $y^* \rightsquigarrow p^v_{i,j}$ path in E^*.

The arguments when t is a terminal in a horizontal connector gadget are similar, and we omit the details here. □

3.5 SCSS has a solution of weight $\leq W^*_n \Rightarrow$ GRID TILING has a solution

First we show that the following preliminary claim:

Claim 3.8. *Let E' be any solution to the SCSS instance (G^*, T^*). Then*

- *E' restricted to each connector gadget satisfies the connectedness property (see Definition 3.1).*

- *E' restricted to each main gadget satisfies the connectedness property (see Definition 3.4).*

Proof. First we show that the edge set E' restricted to each connector gadget satisfies the connectedness property. Consider a horizontal connector gadget $HCG_{i,j}$ for some $1 \leq j \leq k+1, 1 \leq i \leq k$. This gadget contains two terminals: $p^h_{i,j}$ and $q^h_{i,j}$. The only incoming edges from $G^* \setminus HCG_{i,j}$ into $HCG_{i,j}$ are incident onto the source-distinguished vertices of $HCG_{i,j}$, and the only outgoing edges from $HCG_{i,j}$ into $G^* \setminus HCG_{i,j}$ are incident on the sink-distinguished vertices of $HCG_{i,j}$. Since E' is a solution of the SCSS instance (G^*, T^*) it follows that E^* contains a path from $p^h_{i,j}$ to the terminals in $T^* \setminus \{p^h_{i,j} \cup q^h_{i,j}\}$. Since the only outgoing edges from $HCG_{i,j}$ into $G^* \setminus HCG_{i,j}$ are incident on the sink-distinguished vertices of $HCG_{i,j}$, it follows that $p^h_{i,j}$ can reach some sink-distinguished vertex of $HCG_{i,j}$ in the solution E'. The other three conditions of Definition 3.1 can be verified similarly, and hence E' restricted to each main connector satisfies the connectedness property.

Next we argue that E' restricted to each main gadget satisfies the connectedness property. Consider a main gadget $MG_{i,j}$. Since E' is a solution for the SCSS instance (G^*, T^*) it follows that the terminal $p^h_{i,j}$ from $HCG_{i,j}$ is able to reach other terminals of T^*. However, the only outgoing edges from $HCG_{i,j}$ into $G^* \setminus HCG_{i,j}$ are incident on the sink-distinguished vertices of $HCG_{i,j}$. Moreover, each sink-distinguished vertex of $HCG_{i,j}$ is identified with a left-distinguished vertex of $MG_{i,j}$ of the same index. Hence, these outward paths from $p^h_{i,j}$ to other terminals of T^* must continue through the left-distinguished vertices of $MG_{i,j}$. However, the only outgoing edges from $MG_{i,j}$ into $G^* \setminus MG_{i,j}$ are incident on the right-distinguished vertices or bottom-distinguished vertices of $HCG_{i,j}$. Hence, some left-distinguished vertex of $MG_{i,j}$ can reach some vertex in the set given by the union of right-distinguished and bottom-distinguished vertices of $MG_{i,j}$. Hence the first condition of Definition 3.4 is satisfied. Similarly it can be shown the other three conditions of Definition 3.4 also hold, and hence E' restricted to each main gadget satisfies the connectedness property. □

Now we are ready to prove the following lemma:

Lemma 3.9. *If the SCSS instance (G^*, T^*) has a solution E'' of weight at most W^*_n, then the* GRID TILING *instance $(k, n, \{S_{i,j} : i, j \in [k]\})$ has a solution.*

Proof. By Claim 3.8, the edge set E'' restricted to any connector gadget satisfies the connectedness property and the edge set E'' restricted to any main gadget satisfies the connectedness property. Let $\mathcal{C}$ and $\mathcal{M}$ be the sets of connector and main gadgets respectively. Recall that $|\mathcal{C}| = 2k(k+1)$ and $|\mathcal{M}| = k^2$. Recall that we have defined W^*_n as $k^2 \cdot M^*_n + 2k(k+1)C^*_n$. Let $\mathcal{C}' \subseteq \mathcal{C}$ be the set of connector gadgets that have weight at most C^*_n in E''. By Lemma 3.3(2), each connector gadget from the set $\mathcal{C}'$ has weight exactly C^*_n. Since all edge-weights in connector gadgets are positive integers, each connector gadget from the set $\mathcal{C} \setminus \mathcal{C}'$ has weight at least $C^*_n + 1$. Similarly, let $\mathcal{M}' \subseteq \mathcal{M}$ be the set of main gadgets which have weight at most M^*_n in E''. By Lemma 3.6(2), each main gadget from the set $\mathcal{M}'$ has weight exactly M^*_n. Since all edge-weights in main gadgets are positive integers, each main gadget from the set $\mathcal{M} \setminus \mathcal{M}'$ has weight at least $M^*_n + 1$. As any two gadgets are pairwise edge-disjoint, we have

$$\begin{aligned}
W^*_n &= k^2 \cdot M^*_n + 2k(k+1)C^*_n \\
&\geq |\mathcal{M} \setminus \mathcal{M}'| \cdot (M^*_n + 1) + |\mathcal{M}'| \cdot M^*_n + |\mathcal{C} \setminus \mathcal{C}'| \cdot (C^*_n + 1) + |\mathcal{C}'| \cdot C^*_n \\
&= |\mathcal{M}| \cdot M^*_n + |\mathcal{C}| \cdot C^*_n + |\mathcal{M} \setminus \mathcal{M}'| + |\mathcal{C} \setminus \mathcal{C}'| \\
&= k^2 \cdot M^*_n + 2k(k+1) \cdot C^*_n + |\mathcal{M} \setminus \mathcal{M}'| + |\mathcal{C} \setminus \mathcal{C}'| \\
&= W^*_n + |\mathcal{M} \setminus \mathcal{M}'| + |\mathcal{C} \setminus \mathcal{C}'|.
\end{aligned}$$

This implies $|\mathcal{M} \setminus \mathcal{M}'| = 0 = |\mathcal{C} \setminus \mathcal{C}'|$. However, we had $\mathcal{M}' \subseteq \mathcal{M}$ and $\mathcal{C}' \subseteq \mathcal{C}$. Therefore, $\mathcal{M}' = \mathcal{M}$ and $\mathcal{C}' = \mathcal{C}$. Hence in E'', each connector gadget has weight C^*_n and each main gadget has weight M^*_n. From Lemma 3.3(2) and Lemma 3.6(2), we have

- For each vertical connector gadget $VCG_{i,j}$, the restriction of the edge set E'' to $VCG_{i,j}$ represents an integer $\beta_{i,j} \in [n]$ where $i \in [k+1], j \in [k]$.
- For each horizontal connector gadget $HCG_{i,j}$, the restriction of the edge set E'' to $HCG_{i,j}$ represents an integer $\alpha_{i,j}$ where $i \in [k], j \in [k+1]$.
- For each main gadget $MG_{i,j}$, the restriction of the edge set E'' to $MG_{i,j}$ represents an ordered pair $(\alpha'_{i,j}, \beta'_{i,j}) \in S_{i,j}$ where $i, j \in [k]$.

Consider the main gadget $MG_{i,j}$ for any $1 \leq i, j \leq k$. We can make the following observations:

- $\underline{\beta_{i,j} = \beta'_{i,j}}$: By Lemma 3.3(2) and Definition 3.2, the terminal vertices in $VCG_{i,j}$ can exit the vertical connector gadget only via the unique edge entering the sink-distinguished vertex of index $\beta_{i,j}$. By Lemma 3.6(2) and Definition 3.5, the only edge in E'' incident to any top-distinguished vertex of $MG_{i,j}$ is the unique edge leaving the top-distinguished vertex of the index $\beta'_{i,j}$. Hence if $\beta_{i,j} \neq \beta'_{i,j}$ then the terminals in $VCG_{i,j}$ will not be able to exit $VCG_{i,j}$ and reach other terminals.
- $\underline{\beta'_{i,j} = \beta_{i+1,j}}$: By Lemma 3.3(2) and Definition 3.2, the unique edge entering $VCG_{i+1,j}$ is the edge entering the source-distinguished vertex of the index $\beta_{i+1,j}$. By Lemma 3.6(2) and Definition 3.5, the only edge in E'' incident to any bottom-distinguished vertex of $MG_{i,j}$ is the unique edge entering the bottom-distinguished vertex of index $\beta'_{i,j}$. Hence if $\beta'_{i,j} \neq \beta_{i+1,j}$, then the terminals in $VCG_{i+1,j}$ cannot be reached from the other terminals.
- $\underline{\alpha_{i,j} = \alpha'_{i,j}}$: By Lemma 3.3(2) and Definition 3.2, the paths starting at the terminal vertices in $HCG_{i,j}$ can leave the horizontal connector gadget only via the unique edge entering the sink-distinguished vertex of index $\alpha_{i,j}$. By Lemma 3.6(2) and Definition 3.5, the only edge in E'' incident to any left-distinguished vertex of $MG_{i,j}$ is the unique edge leaving the left-distinguished vertex of the index $\alpha'_{i,j}$. Hence if $\alpha_{i,j} \neq \alpha'_{i,j}$ then the terminals in $HCG_{i,j}$ will not be able to reach other terminals.
- $\underline{\alpha'_{i,j} = \alpha_{i,j+1}}$: By Lemma 3.3(2) and Definition 3.2, the unique edge entering $HCG_{i,j+1}$ is the edge entering the source-distinguished vertex of index $\alpha_{i,j+1}$. By Lemma 3.6(2) and Definition 3.5, the only edge in E'' incident to any right-distinguished vertex of $MG_{i,j}$ is the unique edge entering the right-distinguished vertex of index $\alpha'_{i,j}$. Hence if $\alpha'_{i,j} \neq \alpha_{i,j+1}$, then the terminals in $HCG_{i,j+1}$ cannot be reached from the other terminals.

We claim that for $1 \leq i, j \leq k$, the entries $(\alpha'_{i,j}, \beta'_{i,j}) \in S_{i,j}$ form a solution for the GRID TILING instance. For this we need to check two conditions:

- $\underline{\alpha'_{i,j} = \alpha'_{i,j+1}}$: This holds because $\alpha_{i,j} = \alpha'_{i,j} = \alpha_{i,j+1} = \alpha'_{i,j+1}$.
- $\underline{\beta'_{i,j} = \beta'_{i+1,j}}$: This holds because $\beta_{i,j} = \beta'_{i,j} = \beta_{i+1,j} = \beta'_{i+1,j}$.

This completes the proof of the lemma. □

3.6 Proof of Theorem 1.2

Finally we are ready to prove Theorem 1.2 which is restated below:

Theorem 1.2. *The edge-unweighted version of the SCSS problem is W[1]-hard parameterized by the number of terminals k, even when the underlying undirected graph is planar. Moreover, under the ETH, the SCSS problem on planar graphs cannot be solved in $f(k) \cdot n^{o(\sqrt{k})}$ time where f is any computable function, k is the number of terminals and n is the number of vertices in the instance.*

Proof. Each connector gadget has $O(n^2)$ vertices and G^* has $O(k^2)$ connector gadgets. Each main gadget has $O(n^3)$ vertices and G^* has $O(k^2)$ main gadgets. It is easy to see that the graph G^* has $O(n^3k^2) = \text{poly}(n,k)$ vertices. Moreover, the graph G^* can be constructed in $\text{poly}(n+k)$ time: recall that each connector gadget (Lemma 3.3) and main gadget (Lemma 3.6) can be constructed in polynomial time. Each main gadget and connector gadget is planar, and any two gadgets are pairwise edge-disjoint. Moreover, the 0-weight edges incident on x^* or y^* do not affect planarity (see Figure 2 for a planar embedding). Hence, G^* is planar.

It is known [23, Theorem 14.28] that $k \times k$ GRID TILING is W[1]-hard parameterized by k, and under ETH cannot be solved in $f(k) \cdot n^{o(k)}$ for any computable function f. Combining the two directions from Section 3.4 and Section 3.5, we get a parameterized reduction from $k \times k$ GRID TILING to a planar instance of SCSS with $O(k^2)$ terminals. Hence, it follows that SCSS on planar graphs is W[1]-hard and under ETH cannot be solved in $f(k) \cdot n^{o(\sqrt{k})}$ time for any computable function f. □

This shows that the $2^{O(k)} \cdot n^{O(\sqrt{k})}$ algorithm for SCSS on planar graphs given in Theorem 1.1 is asymptotically optimal.

4 Proof of Lemma 3.3: constructing connector gadgets

We prove Lemma 3.3 in this section, by constructing a connector gadget satisfying the specifications of Section 3.1.

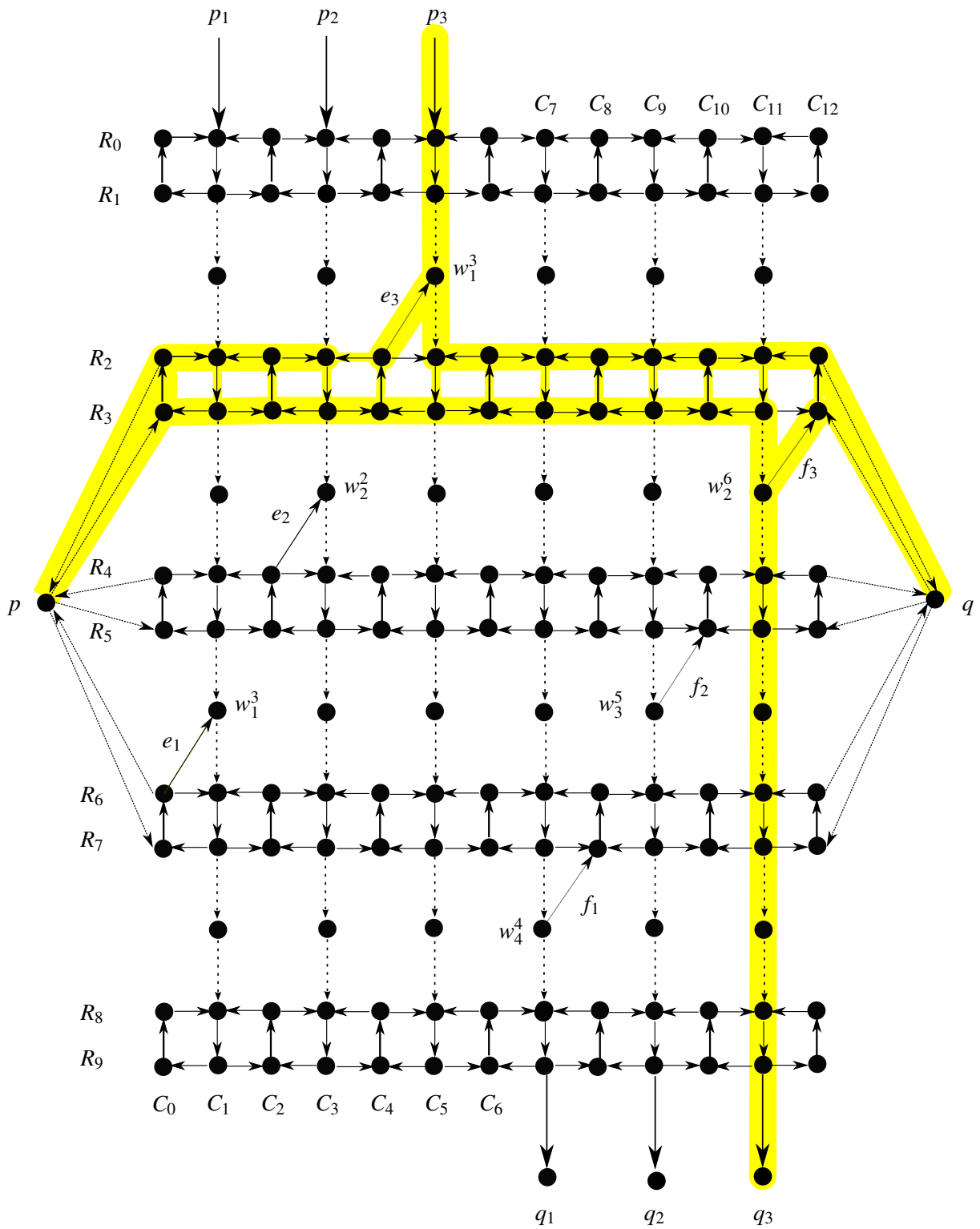

Figure 3: The connector gadget for $n = 3$. A set of edges representing 3 is highlighted in the figure.

4.1 Different types of edges in connector gadget

Before proving Lemma 3.3, we first describe the construction of the connector gadget in more detail (see Figure 3). The connector gadget has $2n+4$ rows denoted by $R_0, R_1, R_2, \ldots, R_{2n+3}$ and $4n+1$ columns denoted by $C_0, C_1, \ldots, C_{4n}$. Let us denote the vertex at the intersection of row R_i and column C_j by v_i^j. We now describe the different kinds of edges present in the connector gadget.

1. **Source Edges**: For each $i \in [n]$, there is an edge (p_i, v_0^{2i-1}). These edges are together called source edges.

2. **Sink Edges**: For each $i \in [n]$, there is an edge $(v_{2n+3}^{2n+2i-1}, q_i)$. These edges are together called sink edges.

3. **Terminal Edges**: The union of the sets of edges incident to the terminals p or q are called terminal edges. The set of edges incident on p is $\{(p, v_{2i+1}^0 \ : \ i \in [n])\} \cup \{(v_{2i}^0, p \ : \ i \in [n])\}$. The set of edges incident on q is $\{(q, v_{2i+1}^{4n} \ : \ i \in [n])\} \cup \{(v_{2i}^{4n}, q \ : \ i \in [n])\}$.

4. **Inrow Edges**:

   - Inrow Up Edges: For each $0 \le i \le n+1$, we call the $\uparrow$ edges connecting vertices of row R_{2i+1} to R_{2i} as inrow up edges. Explicitly, this set of edges is given by $\{(v_{2i+1}^{2j}, v_{2i}^{2j}) \ : \ 0 \le j \le 2n\}$.
   - Inrow Down Edges: For each $0 \le i \le n+1$, we call the $\downarrow$ edges connecting vertices of row R_{2i} to R_{2i+1} as inrow down edges. Explicitly, this set of edges is given by $\{(v_{2i}^{2j-1}, v_{2i+1}^{2j-1}) \ : \ 1 \le j \le 2n\}$.
   - Inrow Left Edges: For each $0 \le i \le 2n+3$, we call the $\leftarrow$ edges connecting vertices of row R_i as inrow left edges. We explicitly list the set of inrow left edges for even-numbered and odd-numbered rows below:
     - For each $0 \le i \le n+1$, the set of inrow left edges for the row R_{2i} is given by $\{(v_{2i}^{2j}, v_{2i}^{2j-1}) \ : \ j \in [2n]\}$
     - For each $0 \le i \le n+1$, the set of inrow left edges for the row R_{2i+1} is given by $\{(v_{2i+1}^{2j-1}, v_{2i+1}^{2j-2}) \ : \ j \in [2n]\}$
   - Inrow Right Edges: For each $0 \le i \le 2n+3$, we call the $\rightarrow$ edges connecting vertices of row R_i as inrow right edges. We explicitly list the set of inrow right edges for even-numbered and odd-numbered rows below:
     - For each $0 \le i \le n+1$, the set of inrow right edges for the row R_{2i} is given by $\{(v_{2i}^{2j-2}, v_{2i}^{2j-1}) \ : \ j \in [2n]\}$
     - For each $0 \le i \le n+1$, the set of inrow right edges for the row R_{2i+1} is given by $\{(v_{2i+1}^{2j-1}, v_{2i+1}^{2j}) \ : \ j \in [2n]\}$

5. **Interrow Edges**: For each $i \in [n+1]$ and each $j \in [2n]$, we subdivide the edge $(v_{2i-1}^{2j-1}, v_{2i}^{2j-1})$ by introducing a new vertex w_i^j and adding the edges (v_{2i-1}^{2j-1}, w_i^j) and (w_i^j, v_{2i}^{2j-1}). All these edges are together called interrow edges. Note that there is a total of $4n(n+1)$ interrow edges.

6. **Shortcuts**: There are $2n$ shortcut edges, namely $e_1, e_2, \ldots, e_n$ and $f_1, f_2, \ldots, f_n$. They are drawn as follows:

   - The edge e_i is given by $(v_{2n-2i+2}^{2i-2}, w_{n-i+1}^i)$.
   - The edge f_i is given by $(w_{n-i+2}^{n+i}, v_{2n-2i+3}^{2n+2i})$.

4.2 Assigning weights in the connector gadget

Fix the quantity $B = 18n^2$. We assign weights to the edges as follows

1. For $i \in [n]$, the source edge (p_i, v_0^{2i-1}) has weight $B^5 + (n-i+1)$.
2. For $i \in [n]$, the sink edge $(v_{2n+3}^{2n+2i-1}, q_i)$ has weight $B^5 + i$.
3. Each terminal edge has weight B^4.
4. Each inrow up edge has weight B^3.
5. Each interrow edge has weight $\frac{B^2}{2}$ each.
6. Each inrow right edge has weight B.
7. For each $i \in [n]$, the shortcut edge e_i has weight $n \cdot i$.
8. For each $j \in [n]$, the shortcut edge f_j has weight $n(n-j+1)$.
9. Each inrow left edge and inrow down edge has weight 0.

Now we define the quantity C_n^* stated in statement of Lemma 3.3:

$$C_n^* = 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + (4n-2)B + (n+1)^2. \tag{2}$$

In the next two sections, we prove the two statements of Lemma 3.3.

4.3 For every $i \in [n]$, there is a solution E_i of weight C_n^* that satisfies the connectedness property and represents i

Let E_i be the union of the following sets of edges:

- Select the edges (p_i, v_0^{2i-1}) and $(v_{2n+3}^{2n+2i-1}, q_i)$. This incurs a weight of $B^5 + (n-i+1) + B^5 + i = 2B^5 + (n+1)$.
- The two terminal edges $(p, v_{2n-2i+3}^0)$ and $(v_{2n-2i+2}^0, p)$. This incurs a weight of $2B^4$.
- The two terminal edges $(q, v_{2n-2i+3}^{4n})$ and $(v_{2n-2i+2}^{4n}, q)$. This incurs a weight of $2B^4$.
- All $2n$ inrow right edges and $2n$ inrow left edges which occur between vertices of $R_{2n-2i+2}$. This incurs a weight of $2n \cdot B$ since each inrow left edge has weight 0 and each inrow right edge has weight B.
- All $2n$ inrow right edges and $2n$ inrow left edges which occur between vertices of $R_{2n-2i+3}$. This incurs a weight of $2n \cdot B$ since each inrow left edge has weight 0 and each inrow right edge has weight B.
- All the $2n+1$ inrow up edges that are between vertices of $R_{2n-2i+2}$ and $R_{2n-2i+3}$. These edges are given by $(v_{2n-2i+3}^{2j}, v_{2n-2i+2}^{2j})$ for $0 \le j \le 2n$. This incurs a weight of $(2n+1)B^3$.
- All $2n$ inrow down edges that occur between vertices of row $R_{2n-2i+2}$ and row $R_{2n-2i+3}$. This incurs a weight of 0, since each inrow down edge has weight 0.
- The vertically downward $v_0^{2i-1} \rightsquigarrow v_{2n-2i+3}^{2i-1}$ path P_1 formed by interrow edges and inrow down edges, and the vertically downward $v_{2n-2i+2}^{2n+2i-1} \rightsquigarrow v_{2n+3}^{2n+2i-1}$ path P_2 formed by interrow edges and inrow down edges. These two paths together incur a total weight of $(n+1)B^2$, since the inrow down edges have weight 0.

- The edges e_i and f_i. This incurs a weight of $n \cdot i + n(n-i+1) = n(n+1)$.

Finally, **remove** the two inrow right edges $(v^{2i-2}_{2n-2i+2}, v^{2i-1}_{2n-2i+2})$ and $(v^{2n+2i-1}_{2n-2i+3}, v^{2n+2i}_{2n-2i+3})$ from E_i. This saves a weight of $2B$. From the above paragraph and Equation 2 it follows that the total weight of E_i is exactly C^*_n. Note that even though we removed the edge $(v^{2i-2}_{2n-2i+2}, v^{2i-1}_{2n-2i+2})$ we can still travel from $v^{2i-2}_{2n-2i+2}$ to $v^{2i-1}_{2n-2i+2}$ in E_i using the edge e_i as follows: take the path $v^{2i-2}_{2n-2i+2} \to w^i_{n-i+1} \to v^{2i-1}_{2n-2i+2}$. Similarly, even though we removed the edge $(v^{2n+2i-1}_{2n-2i+3}, v^{2n+2i}_{2n-2i+3})$ we can still travel from $v^{2n+2i-1}_{2n-2i+3}$ to $v^{2n+2i}_{2n-2i+3}$ in E_i using the edge f_i as follows: take the path $v^{2n+2i-1}_{2n-2i+3} \to w^{n+i}_{n-i+2} \to v^{2n+2i}_{2n-2i+3}$.

It remains to show that E_i satisfies the connectedness property and it represents i. It is easy to see E_i represents i since the only edge in E_i which is incident to P is the edge leaving p_i. Similarly, the only edge in E_i incident to Q is the one entering q_i. We show that the connectedness property holds as follows (recall Definition 3.1):

1. There is a $p_i \rightsquigarrow p$ path in E_i by starting with the source edge leaving p_i and then following downward path P_1 from $v^{2i-1}_0 \rightsquigarrow v^{2i-1}_{2n-2i+3}$. Then travel towards the left from $v^{2i-1}_{2n-2i+3}$ to p by using inrow left, inrow up and inrow down edges from rows $R_{2n-2i+2}$ and $R_{2n-2i+3}$. Finally, use the edge $(v^0_{2n-2i+2}, p)$

2. For the existence of a $p_i \rightsquigarrow q$ path in E_i, we have seen above that there is a $p_i \rightsquigarrow v^{2i-1}_{2n-2i+3}$ path. Then travel towards the right from $v^{2i-1}_{2n-2i+2}$ to q by using inrow right, inrow up and inrow down edges from rows $R_{2n-2i+2}$ and $R_{2n-2i+3}$ to reach the vertex $v^{4n}_{2n-2i+2}$. The only potential issue is that the inrow right edge $(v^{2n+2i-1}_{2n-2i+3}, v^{2n+2i}_{2n-2i+3})$ is missing in E_i: however this is not a problem since we have the path $v^{2n+2i-1}_{2n-2i+3} \to w^{n+i}_{n-i+2} \to v^{2n+2i}_{2n-2i+3}$ in E_i. Finally, use the edge $(v^{4n}_{2n-2i+2}, q)$.

3. For the existence of a $p \rightsquigarrow q_i$ path in E_i, first use the edge $(p, v^0_{2n-2i+3})$. Then travel towards the right by using inrow up, inrow right and inrow down edges from rows $R_{2n-2i+2}$ and $R_{2n-2i+3}$ to reach the vertex $v^{2n+2i-1}_{2n-2i+2}$. The only potential issue is that the inrow right edge $(v^{2i-2}_{2n-2i+2}, v^{2i-1}_{2n-2i+2})$ is missing in E_i: however this is not a problem since we have the path $v^{2i-2}_{2n-2i+2} \to w^i_{n-i+1} \to v^{2i-1}_{2n-2i+2}$ in E_i. Then take the downward path P_2 from $v^{2n+2i-1}_{2n-2i+2}$ to $v^{2n+2i-1}_{2n+3}$. Finally, use the sink edge $(v^{2n+2i-1}_{2n+3}, q_i)$ incident to q_i.

4. For the existence of a $q \rightsquigarrow q_i$ path in E_i, first use the terminal edge $(q, v^{4n}_{2n-2i+3})$. Then travel towards the left by using inrow up, inrow left and inrow down edges from rows $R_{2n-2i+2}$ and $R_{2n-2i+3}$ until you reach the vertex $v^{2n+2i-1}_{2n-2i+2}$. Then take the downward path P_2 from $v^{2n+2i-1}_{2n-2i+2}$ to $v^{2n+2i-1}_{2n+3}$. Finally, use the sink edge $(v^{2n+2i-1}_{2n+3}, q_i)$ incident to q_i.

Therefore, E_i indeed satisfies the connectedness property.

4.4 E' satisfies the connectedness property and has weight at most $C^*_n \Rightarrow E'$ represents some $\beta \in [n]$ and has weight exactly C^*_n

Next we show that if a set of edges E' satisfies the connectedness property and has weight at most C^*_n, then in fact the weight of E' is exactly C^*_n and it represents some $\beta \in [n]$. We do this via the following series of claims and observations.

Claim 4.1. *E' contains exactly one source edge and one sink edge.*

Proof. Since E' satisfies the connectedness property it must contain at least one source edge and at least one sink edge. Without loss of generality, suppose that there are at least two source edges in E'. Then the weight of E' is a least the sum of the weights of these two source edges plus the weight of at least one sink edge.

Thus if E' contains at least two source edges, then its weight is at least $3B^5$. However, from Equation 2 we get that

$$\begin{aligned}
C_n^* &= 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + (4n-2)B + (n+1)^2 \\
&\leq 2B^5 + 4n \cdot B^4 + 3n \cdot B^4 + 2n \cdot B^4 + 4n \cdot B^4 + 4n \cdot B^4 \\
&\leq 2B^5 + 17n \cdot B^4 \\
&< 3B^5,
\end{aligned}$$

since $B = 18n^2 > 17n$. □

Thus we know that E' contains exactly one source edge and exactly one sink edge. Let the source edge be incident to $p_{i'}$ and the sink edge be incident to $q_{j'}$.

Claim 4.2. *E' contains exactly four terminal edges.*

Proof. Since E' satisfies the connectedness property, it must contain at least one incoming and one outgoing edge for both p and q. Therefore, we need at least four terminal edges. Suppose we have at least five terminal edges in E'. We already know that the source and sink edges contribute at least $2B^5$ to weight of E', hence the weight of E' is at least $2B^5 + 5B^4$. However, from Equation 2, we get that

$$\begin{aligned}
C_n^* &= 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + (4n-2)B + (n+1)^2 \\
&\leq 2B^5 + 4B^4 + 3n \cdot B^3 + 2n \cdot B^3 + 4n \cdot B^3 + 4n \cdot B^3 \\
&= 2B^5 + 4B^4 + 13n \cdot B^3 \\
&< 2B^5 + 5B^4,
\end{aligned}$$

since $B = 18n^2 > 13n$. □

Hence we know that E' contains exactly four terminal edges.

Claim 4.3. *E' contains exactly $2n+1$ inrow up edges, one from each column C_{2i} for $0 \leq i \leq 2n$.*

Proof. Observe that for each $1 \leq j \leq 2n-1$, the inrow up edges in column C_{2j} form a cut between vertices from columns C_{2j-1} and C_{2j+1}. Since E' must have a $p_{i'} \rightsquigarrow p$ path, we need to use at least one inrow up edge from each of the columns $C_0, C_2, \ldots, C_{2i'-2}$. Since E' must have a $p_{i'} \rightsquigarrow q$, path we need to use at least one inrow up edge from each of the columns $C_{2i'}, C_{2i'+2}, \ldots, C_{4n}$. Hence E' has at least $2n+1$ inrow up edges, as we require at least one inrow up edge from each of the columns $C_0, C_2, \ldots, C_{4n}$.

Suppose E' contains at least $2n+2$ inrow up edges. We already know that E' has a contribution of $2B^5 + 4B^4$ from source, sink, and terminal edges. Hence the weight of E' is at least $2B^5 + 4B^4 + (2n+2)B^3$. However, from Equation 2, we get that

$$\begin{aligned}
C_n^* &= 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + (4n-2)B + (n+1)^2 \\
&\leq 2B^5 + 4B^4 + (2n+1)B^3 + 2n \cdot B^2 + 4n \cdot B^2 + 4n \cdot B^2 \\
&= 2B^5 + 4B^4 + (2n+1)B^3 + 10n \cdot B^2 \\
&< 2B^5 + 4B^4 + (2n+2)B^3,
\end{aligned}$$

since $B = 18n^2 > 10n$. □

Therefore, we know that E' contains exactly one inrow edge per column C_{2i} for every $0 \le i \le 2n$. By Claim 4.2, we know that exactly two terminal edges incident to p are selected in E'. Observe that the terminal edge leaving p should be followed by an inrow up edge, and similarly, the terminal edge entering p follows an inrow up edge. Since we select exactly one inrow up edge from column C_0, it follows that the two terminal edges in E' incident to p must be incident to the rows $R_{2\ell+1}$ and $R_{2\ell}$ respectively for some $\ell \in [n]$. Similarly, the two terminal edges in E' incident to q must be incident to the rows $R_{2\ell'+1}$ and $R_{2\ell'}$ for some $\ell' \in [n]$. We summarize this in the following claim:

Observation 4.4. *There exist integers $\ell, \ell' \in [n]$ such that*

- *the only two terminal edges in E' incident to p are $(p, v^0_{2\ell+1})$ and $(v^0_{2\ell}, p)$, and*
- *the only two terminal edges in E' incident to q are $(q, v^{4n}_{2\ell'+1})$ and $(v^{4n}_{2\ell'}, q)$.*

Definition 4.5. *For $i \in [n+1]$, we call the $4n$ interrow edges which connect vertices from row R_{2i-1} to vertices from R_{2i} as Type(i) interrow edges. We can divide the Type(i) interrow edges into $2n$ "pairs" of adjacent interrow edges given by (v^{2j-1}_{2i-1}, w^j_i) and (w^j_i, v^{2j-1}_{2i}) for each $1 \le j \le 2n$*

Note that there are a total of $n+1$ types of interrow edges.

Claim 4.6. *E' contains a pair of interrow edges of Type(r) for each $r \in [n+1]$. Moreover, these two edges are the only interrow edges of Type(r) chosen in E'.*

Proof. First we show that E' contains at least one pair of interrow edges of each type. Observation 4.4 implies that we cannot avoid using interrow edges of any type by, for example, going into p via an edge from some R_{2i} and then exiting p via an edge to some R_{2j+1} for any $j > i$ (similarly for q). By the connectedness property, set E' contains a $p_{i'} \rightsquigarrow p$ path P_1. By Observation 4.4, the only edge entering p is $(v^0_{2\ell}, p)$. Hence E' must contain at least one pair of interrow edges of Type(r) for $1 \le r \le \ell$ since the only way to travel from row R_{2r-1} to R_{2r} (for each $r \in [\ell]$) is by using a pair of interrow edges of Type(r) . Similarly E' contains a $p \rightsquigarrow q_i$ path and the only outgoing edge from p is $(p, v^0_{2\ell+1})$. Hence E' must contain at least one pair of interrow edges of Type(r) for each $\ell+1 \le r \le n+1$ since the only way to travel from row R_{2r-1} to R_{2r} is by using a pair of interrow edges of Type(r). Therefore, the edge set E' contains at least one pair of interrow edges of each Type(r) for $1 \le r \le n+1$.

Next we show that E' contains exactly two interrow edges of Type(r) for each $r \in [n+1]$. Suppose E' contains at least three interrow edges of some Type(r) for some $r \in [n+1]$. Since weight of each interrow edge is $B^2/2$, this implies E' gets a weight of at least $(n+1+\frac{1}{2}) \cdot B^2$ from the interrow edges. We have already seen E' has contribution of $2B^5 + 4B^4 + (2n+1)B^3$ from source, sink, terminal, and inrow up edges. Hence the weight of E' is at least $2B^5 + 4B^4 + (2n+1)B^3 + (n+1+\frac{1}{2}) \cdot B^2$. However, from Equation 2, we get that

$$\begin{aligned}
C^*_n &= 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + (4n-2)B + (n+1)^2 \\
&\le 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + 4n \cdot B + 4n \cdot B \\
&= 2B^5 + 4B^4 + (2n+1)B^3 + (n+1)B^2 + 8n \cdot B \\
&< 2B^5 + 4B^4 + (2n+1)B^3 + (n+1+\frac{1}{2})B^2,
\end{aligned}$$

since $\frac{B}{2} = 9n^2 > 8n$. Hence, E' contains exactly two interrow edges of Type(r) for each $r \in [n+1]$. □

Claim 4.7. *For each $r \in [n+1]$, let the unique pair of interrow edges in E' (guaranteed by Claim 4.6) of Type(r) belong to column $C_{2\ell_r-1}$. If the unique source and sink edges in E' (guaranteed by Claim 4.1) are incident to $p_{i'}$ and $q_{j'}$, respectively, then we have $i' \le \ell_1 \le \ell_2 \le \ldots \le \ell_{n+1} \le n+j'$.*

Proof. Observation 4.4 implies the only way to get from row R_{2i-1} to R_{2i} is to use a pair of interrow edges of Type(i). By Claim 4.6, we use exactly one pair of interrow edges of each type. Recall that there is a walk $P = p_{i'} \rightsquigarrow p \rightsquigarrow q_{j'}$ in E', and this walk must use each of these interrow edges.

First we show that $\ell_1 \geq i'$. Suppose $\ell_1 < i' \leq n$. Since we use the source edge incident to $p_{i'}$, we must reach vertex $v_0^{2i'-1}$. Since $i' > \ell_1$, to use the pair of interrow edges to travel from $v_1^{2\ell_1-1}$ to $v_2^{2\ell_1-1}$, the walk P must contain a $v_0^{2i'-1} \rightsquigarrow v_1^{2\ell_1-1}$ subwalk P'. By the construction of the connector gadget this subwalk P' must use the inrow up edge $(v_1^{2i'-2}, v_0^{2i'-2})$. Now the walk P has to reach column $C_{2n+2j'-1}$ from column $C_{2\ell_1-1}$, and so it must use another inrow edge from column $C_{2i'-2}$ (between rows R_{2i} and R_{2i+1} for some $i \geq 1$), which contradicts Claim 4.3.

Now we prove $\ell_{n+1} \leq n + j'$. Suppose to the contrary that $\ell_{n+1} > n + j'$. Then by reasoning similar to that of above one can show that at least two inrow up edges from column $C_{2n+2j'}$ are used, which contradicts Claim 4.3.

Finally suppose there exists $r \in [n]$ such that $\ell_r > \ell_{r+1}$. We consider the following three cases:

- $\underline{\ell_{r+1} < \ell_r \leq n}$: Then by using the fact that there is a $p_{i'} \rightsquigarrow q_{j'}$ walk in E' we get at least two inrow up edges are used from column $C_{2\ell_r-2}$, which contradicts Claim 4.3.
- $\underline{n < \ell_r \leq n + j'}$: Then we need to use at least two inrow up edges from column $C_{2\ell_r-2}$, which contradicts Claim 4.3.
- $\underline{\ell_r > n + j'}$: Then we need to use at least two inrow up edges from column $C_{2n+2j'}$, which contradicts Claim 4.3.

□

Claim 4.8. *E' contains at most two shortcut edges.*

Proof. For the proof we will use Claim 4.7. We will show that we can use at most one e-shortcut. The proof for f-shortcut is similar.

Suppose we use two e-shortcuts viz. e_x and e_y such that $x > y$. Note that it makes sense to include a shortcut into E' only if we use the interrow edge that continues it. Hence $\ell_x = x$ and $\ell_y = y$. By Claim 4.7, we have $y = \ell_y \geq \ell_x = x$, which is a contradiction. □

Claim 4.9. *E' contains exactly $4n-2$ inrow right edges.*

Proof. Since E' contains a $p \rightsquigarrow q_{j'}$ path, it follows that E' has a path connecting some vertex from the column C_i to some vertex from column C_{i+1} for each $0 \leq i \leq 2n+2j'-2$. Since E' contains a $p_{i'} \rightsquigarrow q$ path, it follows that E' has a path connecting some vertex from the column C_j to some vertex from the column C_{j+1} for each $2i'-1 \leq j \leq 4n-1$.

Since $2n+2j'-2 \geq 2n$ and $2i'-1 \leq 2n$, it follows that for each $0 \leq i \leq 4n-1$ the solution E' must contain a path connecting some vertex from column C_i to some vertex from column C_{i+1}. Each such path has to either be a path of one which must be an inrow right edge, or a path of two edges consisting of a shorcut and an interrow edge. But Claim 4.8 implies E' contains at most two shortcuts. Therefore, E' contains at least $4n-2$ inrow right edges. Suppose E' contains at least $4n-1$ inrow right edges. We have already seen the contribution of source, sink, terminal, inrow up and interrow edges is $2B^5+4B^4+(2n+1)B^3+(n+1)B^2$. If E' contains at least $4n-1$ inrow right edges, then the weight of E' is at least $2B^5+4B^4+(2n+1)B^3+(n+1)B^2+(4n-1)B$.

However, from Equation 2, we get that

$$\begin{aligned} C_n^* &= 2B^5+4B^4+(2n+1)B^3+(n+1)B^2+(4n-2)B+(n+1)^2 \\ &= 2B^5+4B^4+(2n+1)B^3+(n+1)B^2+(4n-2)B+4n^2 \\ &< 2B^5+4B^4+(2n+1)B^3+(n+1)B^2+(4n-1)B, \end{aligned}$$

since $B = 18n^2 > 4n^2$. $\square$

From Claim 4.8 and the proof of Claim 4.9, we can make the following observation:

Observation 4.10. *E' contains exactly two shortcuts.*

Let the shortcuts used be $e_{i''}$ and $f_{j''}$. Recall that Claim 4.1 implies that at most one edge incident to P and at most one edge incident to Q is used in E'. Therefore, if we show that $i' = j'$, then it follows that E' represents $\beta = i' = j'$.

Claim 4.11. *The following inequalities hold:*

- *$i'' \geq i'$ and $j'' \leq j'$*
- *$i'' \geq j''$*

Proof. To use the shortcut $e_{i''}$, we need to use the lower half of a pair of interrow edges from column $C_{2i''-1}$. Claim 4.7 implies $i' \leq \ell_1$ and the pairs of interrow edges used are monotone from left to right. Hence $i'' \geq i'$. Similarly, to use the shortcut $f_{j''}$, we need to use the upper half of an interrow edge from Column $C_{2n+2j''-1}$. Claim 4.7 implies $n + j' \geq \ell_{n+1} \geq n + j''$. Hence $j'' \leq j'$.

Since we use the shortcut $e_{i''}$ it follows that $\ell_{n-i''+1} = i''$. Similarly, since we use the shortcut $f_{j''}$ it follows that $\ell_{n-j''+2} = n + j''$. As $1 \leq i'', j'' \leq n$ it follows that $n + j'' > i''$. By mono tonicity of the ℓ-sequence shown in Claim 4.7, we have $n - j'' + 2 > n - i'' + 1$, i.e., $i'' \geq j''$. $\square$

Theorem 4.12. *The weight of E' is exactly C_n^*, and E' represents some integer $\beta \in [n]$.*

Proof. As argued above it is enough to show that $i' = j'$. We have already seen E' has the following contribution to its weight:

- The source edge incident to $p_{i'}$ has weight $B^5 + (n - i' + 1)$ by Claim 4.1.
- The sink edge incident to $q_{j'}$ has weight $B^5 + j'$ by Claim 4.1.
- The terminal edges incur weight $4B^4$ by Claim 4.2.
- The inrow up edges incur weight $(2n+1)B^3$ by Claim 4.3.
- The interrow edges incur weight $(n+1)B^2$ by Claim 4.6.
- The inrow right edges incur weight $(4n-2)B$ by Claim 4.9.
- The shortcut $e_{i''}$ incurs weight $n \cdot i''$ and $f_{j''}$ incurs weight $n(n - j'' + 1)$ by Claim 4.10.

Thus we already have a weight of

$$C^{**} = (2B^5 + (n - i' + j' + 1)) + 4B^4 + (2n+1)B^3 + (n+1)B^2 + (4n-2)B + n(n - j'' + i'' + 1) \quad (3)$$

Observe that adding any edge of non-zero weight to E' (other than the ones mentioned above) increases the weight C^{**} by at least B, since Claim 4.8 does not allow us to use any more shortcuts. Equation 2 and Equation 3 imply $C^{**} + B - C_n^* = B - n(i' - j') - (j'' - i'') \geq 20n^3 - n(i' - j') - (j'' - i'') \geq 0$, since $i', i'', j', j'' \in [n]$. This implies that the weight of E' is exactly C^{**}. We now show that in fact $C^{**} - C_n^* \geq 0$, which will imply that $C^{**} = C_n^*$. From Equation 2 and Equation 3, we have $C^{**} - C_n^* = (j' - i') + n(i'' - j'')$. We now show that this quantity is non-negative. Recall that from Claim 4.11, we have $i'' \geq j''$.

- If $i'' > j''$ then $n(i''-j'') \geq n$. Since $j', i' \in [n]$, we have $j'-i' \geq 1-n$. Therefore, $(j'-i')+n(i''-j'') \geq n+(1-n) = 1$
- If $i'' = j''$ then by Claim 4.11 we have $i' \leq i'' = j'' \leq j'$. Hence $(j'-i') \geq 0$ and so $(j'-i')+n(i''-j'') \geq 0$.

Therefore $C^{**} = C_n^*$, i.e., E' has weight exactly C_n^*. However $C_n^* = C^{**}$ implies

$$j'-i'+n(i''-j'') = 0 \tag{4}$$

Since $i', j' \in [n]$ we have $n-1 \geq j'-i' \geq 1-n$. If $i'' \neq j''$ then $n(i''-j'') \geq n$ and hence $j'-i'+n(i''-j'') \geq (1-n)+n \geq 1$. Contradiction. Hence, we have $j'' = i''$ and therefore Equation 4 implies $j' = i'$, i.e, E' is represented by $i' = j' \in [n]$. $\square$

5 Proof of Lemma 3.6: constructing the main gadget

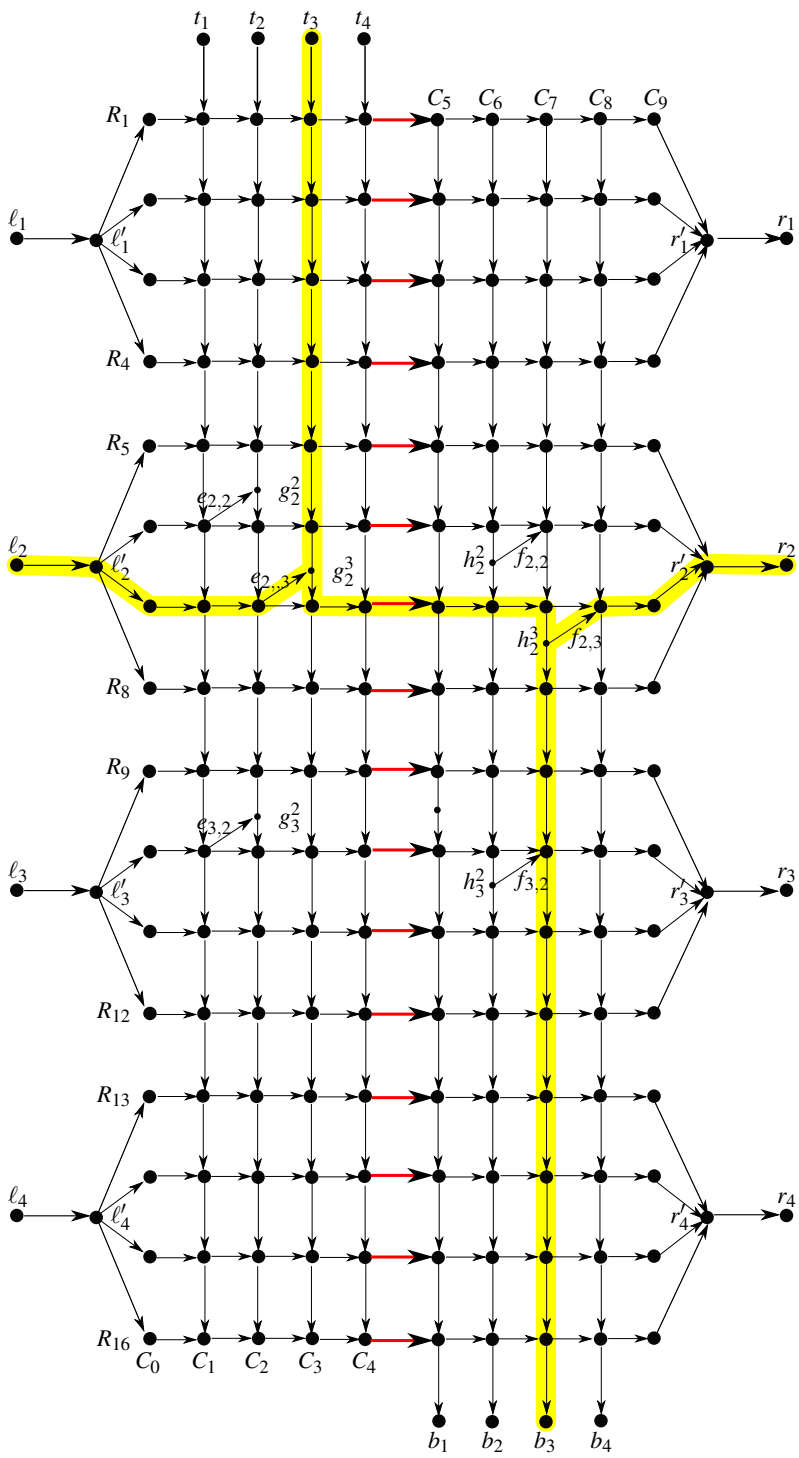

Figure 4: The main gadget (for $n = 4$) representing the set $\{(2,2),(2,3),(3,2)\}$. The highlighted edges represent the pair $(2,3)$.

We prove Lemma 3.6 in this section, by constructing a main gadget satisfying the specifications of Section 3.2. Recall that, as discussed at the start of Section 3.3, we may assume that $1 < x, y < n$ holds for every $(x, y) \in S_{i,j}$.

5.1 Different types of edges in main gadget

Before proving Lemma 3.6, we first describe the construction of the main gadget in more detail (see Figure 4). The main gadget has n^2 rows denoted by $R_1, R_2, \ldots, R_{n^2}$ and $2n+1$ columns denoted by $C_0, C_1, \ldots, C_{2n+1}$. Let us denote the vertex at intersection of row R_i and column C_j by v_i^j. We now describe the various different kinds of edges in the main gadget.

1. **Left Source Edges**: For every $i \in [n]$, the edge (ℓ_i, ℓ'_i) is a left source edge.
2. **Right Sink Edges**: For every $i \in [n]$, the edge (r'_i, r_i) is a right sink edge.
3. **Top Source Edges**: For every $i \in [n]$, the edge (t_i, v_1^i) is a top source edge.
4. **Bottom Sink Edges**: For every $i \in [n]$, the edge $(v_{n^2}^{n+i}, b_i)$ is a bottom sink edge.
5. **Source Internal Edges**: This is the set of n^2 edges of the form (ℓ'_i, v_j^0) for $i \in [n]$ and $n(i-1)+1 \leq j \leq n \cdot i$. We number the source internal edges from top to bottom, i.e., the edge (ℓ'_i, v_j^0) is called the jth source internal edge, where $i \in [n]$ and $n(i-1)+1 \leq j \leq n \cdot i$.
6. **Sink Internal Edges**: This is the set of n^2 edges of the form (v_j^{2n+1}, r'_i) for $i \in [n]$ and $n(i-1)+1 \leq j \leq n \cdot i$. We number the sink internal edges from top to bottom, i.e., the edge (v_j^{2n+1}, r'_i) is called the jth sink internal edge, where $i \in [n]$ and $n(i-1)+1 \leq j \leq n \cdot i$.
7. **Bridge Edges**: This is the set of n^2 edges of the form (v_i^n, v_i^{n+1}) for $1 \leq i \leq n^2$. We number the bridge edges from top to bottom, i.e., the edge (v_i^n, v_i^{n+1}) is called the ith bridge edge. These edges are shown in red color in Figure 4.
8. **Inrow Right Edges**: For each $i \in [n^2]$ we call the $\rightarrow$ edges (except the bridge edge (v_i^n, v_i^{n+1})) connecting vertices of row R_i as inrow right edges. Formally, the set of inrow right edges of row R_i are given by $\{(v_i^j, v_i^{j+1}) \ : \ 0 \leq j \leq n-1\} \bigcup \{(v_i^j, v_i^{j+1}) \ : \ n+1 \leq j \leq 2n\}$
9. **Interrow Down Edges**: For each $i \in [n^2-1]$ we call the $2n \downarrow$ edges connecting vertices of row R_i to R_{i+1} as interrow down edges. The $2n$ interrow edges between row R_i and R_{i+1} are (v_i^j, v_{i+1}^j) for each $1 \leq j \leq 2n$.
10. **Shortcut Edges**: There are $2|S|$ shortcut edges, namely $e_1, e_2, \ldots, e_{|S|}$ and $f_1, f_2, \ldots, f_{|S|}$. The shortcut edge for a $(x, y) \in S$ for some $1 < x, y < n$ is defined the following way:
    - Introduce a new vertex g_x^y at the middle of the edge $(v_{n(x-1)+y-1}^y, v_{n(x-1)+y}^y)$ to create two new edges $(v_{n(x-1)+y-1}^y, g_x^y)$ and $(g_x^y, v_{n(x-1)+y}^y)$. Then the edge $e_{x,y}$ is $(v_{n(x-1)+y}^{y-1}, g_x^y)$.
    - Introduce a new vertex h_x^y at the middle of the edge $(v_{n(x-1)+y}^{n+y}, v_{n(x-1)+y+1}^{n+y})$ to create two new edges $(v_{n(x-1)+y}^{n+y}, h_x^y)$ and $(h_x^y, v_{n(x-1)+y+1}^{n+y})$. Then the edge $f_{x,y}$ is $(h_x^y, v_{n(x-1)+y}^{n+y+1})$.

5.2 Assigning weights in the main gadget

Define $B = 11n^2$. We assign weights to the edges as follows:

1. Each left source edge has weight B^4.
2. Each right sink edge has weight B^4.
3. For every $1 \le i \le n$, the ith top source edge (t_i, v_1^i) has weight B^4.
4. For every $1 \le i \le n$, the ith bottom sink edge $(v_{n^2}^{n+i}, b_i)$ has weight B^4.
5. For each $i \in [n^2]$, the ith bridge edge (v_i^n, v_i^{n+1}) has weight B^3.
6. For each $i \in [n^2]$, the ith source internal edge has weight $B^2(n^2 - i)$.
7. For each $j \in [n^2]$, the jth sink internal edge has weight $B^2 \cdot j$.
8. Each inrow right edge has weight $3B$.
9. For each $(x, y) \in S$, both the shortcut edges $e_{x,y}$ and $f_{x,y}$ have weight B each.
10. Each interrow down edge that does not have a shortcut incident to it has weight 2. If an interrow edge is split into two edges by the shortcut incident to it, then we assign a weight 1 to each of the two parts.

Now we define the quantity M_n^* stated in Lemma 3.6:

$$M_n^* = 4B^4 + B^3 + B^2n^2 + B(6n-4) + 2(n^2-1). \tag{5}$$

Next we are ready to prove the statements of Lemma 3.6.

5.3 For every $(x, y) \in S$, there is a solution $E_{x,y}$ of weight M_n^* that represents (x, y)

For $(x, y) \in S \subseteq [n] \times [n]$ define $z = n(x-1) + y$. Let $E_{x,y}$ be the union of the following sets of edges:

- The xth left source edge and x^{th} right sink edge. This incurs a weight of $2B^4$.
- The yth top source edge and the yth bottom sink edge. This incurs a weight of $2B^4$.
- The zth bridge edge. This incurs a weight of B^3.
- The zth source internal edge and z^{th} sink internal edge. This incurs a weight of B^2n^2.
- All inrow right edges from row R_z except (v_z^{y-1}, v_z^y) and (v_z^{n+y}, v_z^{n+y+1}). This incurs a weight of $3B \cdot (2n-2)$.
- The shortcut edges $e_{x,y}$ and $f_{x,y}$. This incurs a weight of $2B$.
- The vertically downward path $v_1^y \to v_2^y \to \ldots \to v_z^y$ formed by interrow down edges in column C_y. This incurs a weight of $2(z-1)$.
- The vertically downward path $v_z^{n+y} \to v_{z+1}^{n+y} \to \ldots \to v_{n^2}^{n+y}$ formed by interrow down edges in column C_{n+y}. This incurs a weight of $2(n^2 - z)$.

From the above paragraph and Equation 5, it follows the total weight of $E_{x,y}$ is exactly M_n^*. Note that we did not include two inrow right edges, (v_z^{y-1}, v_z^y) and (v_z^{n+y}, v_z^{n+y+1}), from row R_z in $E_{x,y}$. However, we can mimic the function of both these inrow right edges using shortcut edges and interrow down edges in $E_{x,y}$ as follows:

- We can still travel from v_z^{y-1} to v_z^y in $E_{x,y}$ as follows: take the path $(v_z^{y-1} \to g_x^y \to v_z^y)$.
- We can still travel from $(v_z^{n+y}$ to $v_z^{n+y+1})$ in $E_{x,y}$ via the path $(v_z^{n+y} \to h_{x,y} \to v_z^{n+y+1})$.

The following observation follows from the previous paragraph:

Observation 5.1. *In $E_{x,y}$ we can reach v_z^j from v_z^i for any $2n+1 \geq j \geq i \geq 0$.*

It remains to show that $E_{x,y}$ represents $(x,y) \in S$. It is easy to see that the first four conditions of Definition 3.5 are satisfied since the definition of $E_{x,y}$ itself gives the following:

- In $E_{x,y}$ the only outgoing edge from L is the one incident to ℓ_x
- In $E_{x,y}$ the only incoming edge to R is the one incident to r_x
- In $E_{x,y}$ the only outgoing edge from T is the one incident to t_y
- In $E_{x,y}$ the only incoming edge to B is the one incident to b_y

We now show that the last condition of Definition 3.5 is also satisfied by $E_{x,y}$:

1. There is a $\ell_x \rightsquigarrow r_x$ path in $E_{x,y}$ obtained by taking the edges in the following order:
   - the left source edge (ℓ_x, ℓ'_x),
   - the source internal edge (ℓ'_x, v_z^0),
   - the horizontal path $v_z^0 \to v_z^1 \to \dots v_z^n$ given by Observation 5.1,
   - the bridge edge (v_z^n, v_z^{n+1}),
   - the horizontal path $v_z^{n+1} \to v_z^{n+2} \to \dots v_z^{2n+1}$ given by Observation 5.1,
   - the sink internal edge (v_z^{2n+1}, r'_x), and
   - the right sink edge (r'_x, r_x).
2. There is a $t_y \rightsquigarrow b_y$ path in $E_{x,y}$ obtained by taking the edges in the following order:
   - the top source edge (t_y, v_1^y),
   - the downward path $v_1^y \to v_2^y \to \dots v_z^y$ given by interrow down edges in column C_y,
   - the horizontal path $v_z^y \to v_z^{y+1} \to \dots v_z^n$ given by Observation 5.1,
   - the bridge edge (v_z^n, v_z^{n+1}),
   - the horizontal path $v_z^{n+1} \to v_z^{n+2} \to \dots v_z^{n+y}$ given by Observation 5.1,
   - the downward path $v_z^{n+y} \to v_{z+1}^{n+y} \to \dots v_{n^2}^{n+y}$ given by interrow down edges in column C_{n+y}, and
   - the bottom sink edge $(v_{n^2}^{n+y}, b_y)$.

Therefore, $E_{x,y}$ has weight M_n^* and represents (x,y).

5.4 E' satisfies the connectedness property and has weight at most $M_n^* \Rightarrow E'$ represents some $(\alpha, \beta) \in S$ and has weight exactly M_n^*

In this section we show that if a set of edges E' satisfies the connectedness property and has weight M_n^*, then it represents some $(\alpha, \beta) \in S$. We do this via the following series of claims and observations.

Claim 5.2. *E' contains*

- *exactly one left source edge,*
- *exactly one right sink edge,*
- *exactly one top source edge, and*
- *exactly one bottom sink edge.*

Proof. Since E' satisfies the connectedness property, it must contain at least one edge of each of the above types. Without loss of generality, suppose we have at least two left source edges in E'. Then the weight of the edge set E' is greater than or equal to the sum of weights of these two left source edges and the weight of a right sink edge, the weight of a top source edge, and the weight of a bottom sink edge. Thus if E' contains at least two left source edges, then its weight is at least $5B^4$. However, from Equation 5, we get that

$$\begin{aligned} M_n^* &= 4B^4 + B^3 + B^2n^2 + B(6n-4) + 2(n^2-1) \\ &\leq 4B^4 + n \cdot B^3 + n \cdot B^3 + 6n \cdot B^3 + 2n \cdot B^3 \\ &= 4B^4 + 10n \cdot B^3 \\ &< 5B^4, \end{aligned}$$

since $B = 11n^2 > 10n$. □

Therefore, we can set up the following notation:

- Let $i_L \in [n]$ be the unique index such that the left source edge in E' is incident to ℓ_{i_L}.
- Let $i_R \in [n]$ be the unique index such that the right sink edge in E' is incident to r_{i_R}.
- Let $i_T \in [n]$ be the unique index such that the top source edge in E' is incident to t_{i_T}.
- Let $i_B \in [n]$ be the unique index such that the bottom sink edge in E' is incident to b_{i_B}.

Claim 5.3. *The edge set E' contains exactly one bridge edge.*

Proof. To satisfy the connectedness property, we need at least one bridge edge, since the bridge edges form a cut between the top-distinguished vertices and the right-distinguished vertices as well as between the top-distinguished vertices and the bottom-distinguished vertices. Suppose that the edge set E' contains at least two bridge edges. This contributes a weight of $2B^3$. We already have a contribution on $4B^4$ to weight of E' from Claim 5.2. Therefore, the weight of E' is at least $4B^4 + 2B^3$. However, from Equation 5, we get that

$$\begin{aligned} M_n^* &= 4B^4 + B^3 + B^2n^2 + B(6n-4) + 2(n^2-1) \\ &\leq 4B^4 + B^3 + B^2n^2 + 6n \cdot B + 2n^2 \\ &\leq 4B^4 + B^3 + B^2n^2 + 6n^2B^2 + 2n^2B^2 \\ &= 4B^4 + B^3 + 9B^2n^2 \\ &< 4B^8 + 2B^3, \end{aligned}$$

since $B = 11n^2 > 9n^2$. □

Let the index of the unique bridge edge in E' (guaranteed by Claim 5.3) be $\gamma \in [n^2]$. The connectedness property implies that we need to select at least one source internal edge incident to ℓ'_{i_L} and at least one sink internal edge incident to r'_{i_R}. Let the index of the source internal edge incident to ℓ'_{i_L} be j_L and the index of the sink internal edge incident to r'_{i_R} be j_R.

Claim 5.4. *$i_L = i_R$ and $j_L = j_R = \gamma$.*

Proof. By the connectedness property, there is a path from ℓ_{i_L} to some vertex in $r_{i_R} \cup b_{i_B}$. The paths starts with $\ell_{i_L} \to \ell'_{i_L} \to v^1_{j_L}$ and has to use the γth bridge edge. By the construction of the main gadget (all edges are either downwards or towards the right), this path can never reach any row R_ℓ for $\ell < j_L$. Therefore, $\gamma \geq j_L$. By similar logic, we get $j_R \geq \gamma$. Therefore $j_R \geq j_L$.

If $j_R > j_L$, then the weight of the source internal edge and the sink internal edge is $B^5(n^2 - j_L + j_R) \geq B^5(n^2+1)$. We already have a contribution of $4B^4 + B^3$ to the weight of E' from Claim 5.2 and Claim 5.3. Therefore, the weight of E' is at least $4B^4 + B^3 + B^2(n^2+1)$. However, from Equation 5, we get that

$$\begin{aligned}
M^*_n &= 4B^4 + B^3 + B^2n^2 + B(6n-4) + 2(n^2-1) \\
&\leq 4B^4 + B^3 + B^2n^2 + 6n \cdot B + 2n^2 \\
&\leq 4B^4 + B^3 + B^2n^2 + 6n^2 \cdot B + 2n^2 \cdot B \\
&= 4B^4 + B^3 + B^2n^2 + 8n^2 \cdot B \\
&< 4B^4 + B^3 + B^2(n^2+1),
\end{aligned}$$

since $B = 11n^2 > 8n^2$. Hence $j_R = j_L = \gamma$. Observing that $i_L = \lceil \frac{j_L}{n} \rceil$ and $i_R = \lceil \frac{j_R}{n} \rceil$, we obtain $i_L = i_R$. □

Let $i_L = i_R = \alpha$ and $\gamma = n(\alpha - 1) + \beta$. We will now show that E' represents the pair (α, β). By Definition 3.5, we need to prove the following four conditions:

1. The only left source edge in E' is the one incident to ℓ_α and the only right sink edge in E' is the one incident to r_α.
2. The pair (α, β) is in S.
3. The only top source edge in E' is the one incident to t_β and the only bottom sink edge in E' is the one incident to b_β.
4. E' has an $\ell_\alpha \rightsquigarrow r_\alpha$ path and an $t_\beta \rightsquigarrow b_\beta$ path.

The first statement above follows from Claim 5.2 and Claim 5.4. We now continue with the proof of the other three statements mentioned above:

Claim 5.5. *E' contains exactly $2n-2$ inrow right edges, all of them from row R_γ. As a corollary, we get that there are two shortcuts incident to row R_γ, i.e., $(\alpha, \beta) \in S$ and also that E' uses both these shortcuts.*

Proof. Note that by the construction of the main gadget, there can be at most two shortcut edges incident on the vertices of row R_γ.

Claim 5.4 implies $j_L = j_R = \gamma$. Hence the $\ell_\alpha \rightsquigarrow r_\alpha \cup b_{i_B}$ path in E' contains a $v^0_\gamma \rightsquigarrow v^n_\gamma$ subpath P_1. By the construction of the main gadget, we cannot reach an upper row from a lower row. Hence this subpath P_1 must be the path $v^0_\gamma \to v^2_\gamma \to \ldots \to v^n_\gamma$. This path P_1 can at most use the unique shortcut edge incident to row R_γ and column C_β to replace an inrow right edge. Hence P_1 uses at least $n-1$ inrow right edges, with equality only if R_γ has a shortcut incident to it.

Similarly, the $\ell_\alpha \cup t_{i_T} \rightsquigarrow r_\alpha$ path in E' contains a $v^{n+1}_\gamma \rightsquigarrow v^{2n+1}_\gamma$ subpath P_2. By the construction of the main gadget, we cannot reach an upper row from a lower row. Hence this subpath P_2 must be the path

$v_\gamma^{n+1} \to v_\gamma^{n+2} \to \ldots \to v_\gamma^{2n+1}$. This path P_2 can at most use the unique shortcut edge incident to row R_γ and column $C_{n+\beta}$ to replace an inrow right edge. Hence P_2 uses at least $n-1$ inrow right edges, with equality only if R_γ has a shortcut incident to it.

Clearly, the sets of inrow edges used by P_1 and P_2 are disjoint, and hence E' contains at least $2n-2$ inrow right edges from row R_γ. Suppose E' contains at least $2n-1$ inrow right edges. Then it incurs a weight of $3B \cdot (2n-1)$. From Claim 5.2, Claim 5.3 and Claim 5.4 we already have a contribution of $4B^4 + B^3 + B^2n^2$. Therefore the weight of E' is at least $4B^4 + B^3 + B^2n^2 + 3B \cdot (2n-1)$.

However, from Equation 5, we get that

$$\begin{aligned} M_n^* &= 4B^4 + B^3 + B^2n^2 + B(6n-4) + 2(n^2-1) \\ &\leq 4B^4 + B^3 + B^2n^2 + B(6n-4) + 2n^2 \\ &< 4B^4 + B^3 + B^2n^2 + 3B \cdot (2n-1), \end{aligned}$$

since $B = 11n^2 > 2n^2$. Therefore, E' can only contain at most $2n-2$ inrow right edges. Hence there must be two shortcut edges incident to row R_γ, which are both used by E'. Since $\gamma = n(\alpha-1)+\beta$, the fact that row R_γ has shortcut edges incident to it implies $(\alpha,\beta) \in S$. □

To prove the third claim it is sufficient to show that $i_T = i_B = \beta$, since Claim 5.2 implies E' contains exactly one top source edge and exactly one bottom sink edge. Note that the remaining budget left for the weight of E' is at most $2(n^2-1)$.

Claim 5.6. $i_T = i_B = \beta$

Proof. Recall that the only bridge edge used is the one on row R_γ. Moreover, the bridge edges form a cut between T and $R \cup B$. Hence, to satisfy the connectedness property it follows that the $t_{i_T} \rightsquigarrow r_\alpha \cup b_{i_B}$ path in E' contains a $v_1^{i_T} \rightsquigarrow v_\gamma^n$ subpath P_3. By Claim 5.5, all inrow right edges are only from row R_γ. As the only remaining budget is $2(n^2-1)$, we cannot use any other shortcuts or inrow right edges since $B = 11n^2 > 2(n^2-1)$. Therefore, P_3 contains another $v_1^{i_T} \to v_\gamma^{i_T}$ subpath P_3'. If $i_T \neq \beta$, then P_3' incurs weight $2(\gamma-1)$. Note that we also pay a weight of 1 to use half of the interrow edge when we use the shortcut edge (which we have to use due to Claim 5.5) which is incident to row R_γ and column C_β.

Similarly, the $\ell_\alpha \cup t_{i_T} \rightsquigarrow b_{i_B}$ path in E' contains a $v_\gamma^{n+1} \rightsquigarrow v_{n^2}^{n+i_B}$ subpath P_4'. By Claim 5.5, all inrow horizontal edges are only from row R_γ. As the only remaining budget is $2(n^2-1)$, we cannot use any other shortcuts or inrow right edges. Therefore, P_4 contains another $v_\gamma^{n+i_B} \rightsquigarrow v_{n^2}^{n+i_B}$ subpath P_4'. If $i_B \neq \beta$, then P_4' incurs weight $2(n^2-\gamma)$. Note that we also pay a weight of 1 to use (half of) the interrow edge when we use the shortcut edge (which we have to use due to Claim 5.5) which is incident to row R_γ and column $C_{n+\beta}$.

Suppose without loss of generality that $i_T \neq \beta$. Then P_3' incurs a weight of $2(\gamma-1)$, and the half interrow edge used incurs an additional weight of 1. In addition, path P_4' incurs a weight of $2(n^2-\gamma)$. Hence the total weight incurred is $2(\gamma-1)+1+2(n^2-\gamma) = 2(n^2-1)+1$ which is greater than our allowed budget. Hence $i_T = \beta$. It can be shown similarly that $i_B = \beta$. □

Claim 5.7. *E' has an $\ell_\alpha \rightsquigarrow r_\alpha$ path and an $t_\beta \rightsquigarrow b_\beta$ path.*

Proof. First we show that E' has an $\ell_\alpha \rightsquigarrow r_\alpha$ path by taking the following edges (in order)

- The path $\ell_\alpha \to \ell_\alpha' \to v_\gamma^0$ which exists since $i_L = \alpha$ and $j_L = \gamma$
- The $v_\gamma^0 \rightsquigarrow v_\gamma^n$ path P_1 guaranteed in proof of Claim 5.5
- The bridge edge $v_\gamma^n \to v_\gamma^{n+1}$ guaranteed by Claim 5.3
- The $v_\gamma^{n+1} \rightsquigarrow v_\gamma^{2n+1}$ path P_2 guaranteed in proof of Claim 5.5

- The path $r_\alpha \leftarrow r'_\alpha \leftarrow v_\gamma^{2n+1}$ which exists since $i_R = \alpha$ and $j_R = \gamma$

Next we show that E' has an $t_\beta \rightsquigarrow b_\beta$ path by taking the following edges (in order)

- The edge $t_\beta \rightarrow v_1^\beta$ which exists since $i_T = \beta$
- The $v_1^\beta \rightsquigarrow v_\gamma^n$ path P_3 guaranteed in proof of Claim 5.6
- The bridge edge $v_\gamma^n \rightarrow v_\gamma^{n+1}$ guaranteed by Claim 5.3
- The $v_\gamma^{n+1} \rightsquigarrow v_{n^2}^{n+\beta}$ path P_4 guaranteed in proof of Claim 5.6
- The edge $b_\beta \leftarrow v_{n^2}^{n+\beta}$ which exists since $i_B = \beta$

□

Claim 5.2, Claim 5.4, Claim 5.5, Claim 5.6 and Claim 5.7 together imply that E' represents $(\alpha, \beta) \in S$ (see Definition 3.5). We now show that weight of E' is exactly M_n^*.

Lemma 5.8. *Weight of E' is exactly M_n^**

Proof. Claim 5.2 contributes a weight of $4B^4$ to E'. Claim 5.3 contributes a weight of B^3 to E'. From the proof of Claim 5.4, we can see that E' incurs weight B^2n^2 from the source internal edge and sink internal edge. Claim 5.5 implies that E' contains exactly $2n-2$ inrow right edges from row R_γ and also both shortcuts incident to row R_γ. This incurs a cost of $3B(2n-2)+2B = B(6n-4)$. By arguments similar to that in the proof of Claim 5.6, E' contains at least $(\gamma-1)$ interrow edges from column C_β and at least $(n^2-\gamma)$ interrow edges from column $C_{n+\beta}$. Therefore, we have weight of $E' \geq 4B^4 + B^3 + B^2n^2 + B\cdot(6n-4) + 2(\gamma-1) + 2(n^2-\gamma) = 4B^4 + B^3 + B^2n^2 + B\cdot(6n-4) + 2(n^2-1) = M_n^*$. Hence the weight of E' is exactly M_n^*. □

This completes the proof of the second statement of Lemma 3.6.

6 W[1]-hardness for SCSS in general graphs

The main goal of this section is to prove Theorem 1.3. We note that the reduction of Guo et al. [42] gives a reduction from MULTICOLORED CLIQUE which builds an equivalent instance of STRONGLY CONNECTED STEINER SUBGRAPH with quadratic blowup in the number of terminals. Hence using the reduction of Guo et al. [42] only an $f(k)\cdot n^{o(\sqrt{k})}$ algorithm for SCSS can be ruled out under ETH. We are able to improve upon this hardness by using the PARTITIONED SUBGRAPH ISOMORPHISM (PSI) problem introduced by Marx [57]. Our reduction is also slightly simpler than the one given by Guo et al.

PARTITIONED SUBGRAPH ISOMORPHISM (PSI)
Input : Undirected graphs $G = (V_G = \{g_1, g_2, \ldots, g_\ell\}, E_G)$ and $H = (V_H, E_H)$, and a partition of V_H into disjoint subsets $H_1, H_2, \ldots, H_\ell$
Question: Is there an injection $\phi : V_G \rightarrow V_H$ such that

1. For every $i \in [\ell]$ we have $\phi(g_i) \in H_i$.
2. For every edge $\{g_i, g_j\} \in E_G$ we have $\{\phi(g_i), \phi(g_j)\} \in E_H$.

The PSI problem is so-called because the vertices of H are partitioned into parts: one part corresponding to every vertex of G. Marx [57] showed the following hardness result:

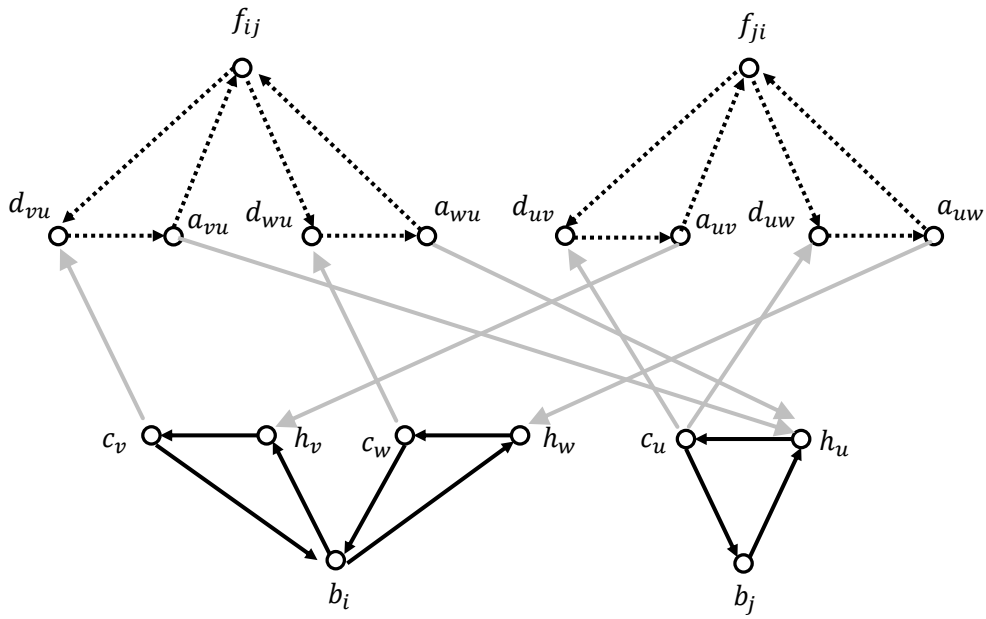

Figure 5: An illustration of the reduction from PSI to SCSS described in Theorem 1.3 for the special case when $V_G = \{g_1, g_2\}, E_G = g_1 - g_2$ and H is a path on three vertices $v - u - w$ with $H_1 = \{v, w\}$ and $H_2 = \{u\}$.

Theorem 6.1. *Unless ETH fails,* PARTITIONED SUBGRAPH ISOMORPHISM *cannot be solved in time* $f(r) \cdot n^{o(r/\log r)}$ *where* f *is any computable function,* r *is the number of edges in* G *and* n *is the number of vertices in* H.

By giving a reduction from PARTITIONED SUBGRAPH ISOMORPHISM to STRONGLY CONNECTED STEINER SUBGRAPH where $k = O(|E_G|)$ we will obtain a $f(k) \cdot n^{o(k/\log k)}$ hardness for SCSS under the ETH, where k is the number of terminals. Consider an instance (G, H) of PARTITIONED SUBGRAPH ISOMORPHISM. We now build an instance (G^*, T^*) of STRONGLY CONNECTED STEINER SUBGRAPH as follows:

- $B = \{b_i \mid i \in [\ell]\}$
- $C = \{c_v \mid v \in V_H\}$
- $H = \{h_v \mid v \in V_H\}$
- $D = \{d_{uv} \cup d_{vu} \mid \{u, v\} \in E_H\}$
- $A = \{a_{uv} \cup a_{vu} \mid \{u, v\} \in E_H\}$
- $F = \{f_{ij} \mid 1 \leq i, j \leq \ell \mid g_i g_j \in E_G\}$
- $V^* = B \cup C \cup H \cup D \cup A \cup F$
- $E_1 = \{(c_v, b_i) \mid v \in H_i, 1 \leq i \leq \ell\}$

- $E_2 = \{(b_i, h_v) \mid v \in H_i, 1 \leq i \leq \ell\}$
- $E_3 = \{(h_v, c_v) \mid v \in V_H\}$
- $E_4 = \{(c_v, d_{vu}) \mid \{u,v\} \in E_H\}$
- $E_5 = \{(a_{vu}, h_u) \mid \{u,v\} \in E_H\}$
- $E_6 = \{(d_{vu}, a_{vu}) \mid \{u,v\} \in E_H\}$
- $E_7 = \{(f_{ij}, d_{vu}) \cup (a_{vu}, f_{ij}) \mid \{u,v\} \in E_H; v \in H_i; u \in H_j; 1 \leq i,j \leq \ell\}$
- $E^* = E_1 \cup E_2 \cup E_3 \cup E_4 \cup E_5 \cup E_6 \cup E_7$
- The set of terminals is $T^* = B \cup F$.

This completes the construction of the graph $G^* = (V^*, E^*)$. An illustration of the construction for a small graph is given in Figure 5. In the instance of PARTITIONED SUBGRAPH ISOMORPHISM we can assume the graph G is connected, otherwise we can solve the problem for each connected component. Therefore, we have that $k = |T| = \ell + 2|E_G| = O(|E_G|)$. For ease of argument, we distinguish the different types of edges of G^* as follows (see Figure 5):

- Edges of $E_1 \cup E_2 \cup E_3$ are denoted using black edges
- Edges of $E_4 \cup E_5$ are denoted using light/gray edges
- Edges of $E_6 \cup E_7$ are denoted using dotted edges

We now show two lemmas which complete the reduction from PARTITIONED SUBGRAPH ISOMORPHISM to STRONGLY CONNECTED STEINER SUBGRAPH .

Lemma 6.2. *If the instance (G,H) of* PARTITIONED SUBGRAPH ISOMORPHISM *answers YES then the instance (G^*, T^*) of* STRONGLY CONNECTED STEINER SUBGRAPH *has a solution of size $\leq 3\ell + 10|E_G|$.*

Proof. Suppose the instance (G,H) of PARTITIONED SUBGRAPH ISOMORPHISM answers YES and let ϕ be the injection from $V_G \to V_H$. Then we claim the following set M' of $3\ell + 10|E_G|$ edges forms a solution for the STRONGLY CONNECTED STEINER SUBGRAPH instance:

- $M_1 = \{(h_{\phi(g_i)}, c_{\phi(g_i)}) \mid i \in [\ell]\}$
- $M_2 = \{(b_i, h_{\phi(g_i)}) \mid i \in [\ell]\}$
- $M_3 = \{(c_{\phi(g_i)}, b_i) \mid i \in [\ell]\}$
- $M_4 = \{(c_{\phi(g_i)}, d_{\phi(g_i)\phi(g_j)}) \cup (d_{\phi(g_i)\phi(g_j)}, a_{\phi(g_i)\phi(g_j)}) \cup (a_{\phi(g_i)\phi(g_j)}, h_{\phi(g_j)}) \mid g_i g_j \in E_G; 1 \leq i,j \leq \ell\}$.
- $M_5 = \{(f_{ij}, d_{\phi(g_i)\phi(g_j)}) \cup (a_{\phi(g_i)\phi(g_j)}, f_{ij}) \mid g_i g_j \in E_G; 1 \leq i,j \leq \ell\}$.
- $M' = M_1 \cup M_2 \cup M_3 \cup M_4 \cup M_5$

First consider $i \neq j$ such that $g_i g_j \in E_G$. Then there is a $b_i \rightsquigarrow b_j$ path in M', namely $b_i \to h_{\phi(g_i)} \to c_{\phi(g_i)} \to d_{\phi(g_i)\phi(g_j)} \to a_{\phi(g_i)\phi(g_j)} \to h_{\phi(g_j)} \to c_{\phi(g_j)} \to b_j$. Generalizing this and observing G is connected we can see any two terminals in B are strongly connected. Now consider two terminals f_{ij} and b_q such that $1 \leq i,j,q \leq \ell$. The existence of the terminal f_{ij} implies $g_i g_j \in E_G$ and hence $\phi(g_i)\phi(g_j) \in E_H$. There is a path in M' from f_{ij} to b_q: use the path $f_{ij} \rightsquigarrow d_{\phi(g_i)\phi(g_j)} \to a_{\phi(g_i)\phi(g_j)} \to h_{\phi(g_j)} \to c_{\phi(g_j)} \to b_j$ followed by the $b_j \rightsquigarrow b_q$ path

(which was shown to exist above). Similarly there is a path in M' from b_q to f_{ij}: use the $b_q \rightsquigarrow b_i$ path (which was shown to exist above) followed by the path $b_i \rightsquigarrow h_{\phi(g_i)} \to c_{\phi(g_i)} \to d_{\phi(g_i)\phi(g_j)} \to a_{\phi(g_i)\phi(g_j)} \to f_{ij}$. Hence each terminal in B can reach every terminal in F and vice versa. Finally consider any two terminals f_{ij} and f_{st} in F: the terminal f_{ij} can first reach b_i and we have seen above that b_i can reach any terminal in F. This shows M' forms a solution for the STRONGLY CONNECTED STEINER SUBGRAPH instance. □

Lemma 6.3. *If the instance (G^*, T^*) of* STRONGLY CONNECTED STEINER SUBGRAPH *has a solution of size $\leq 3\ell + 10|E_G|$ then the instance (G, H) of* PARTITIONED SUBGRAPH ISOMORPHISM *answers YES.*

Proof. Let X be a solution of size $3\ell + 10|E_G|$ for the instance (G^*, T^*) of SCSS. Consider a terminal $f_{ij} \in F$. The only out-neighbors of f_{ij} are vertices from D, and hence X must contain an edge (f_{ij}, d_{vu}) such that $v \in H_i$ and $u \in H_j$. However the only neighbor of d_{vu} is a_{vu}, and hence X has to contain this edge as well. Finally, X must also contain one incoming edge into f_{ij} since we desire strong connectivity. So for each terminal f_{ij}, we need three "private" dotted edges in the sense that every terminal in F needs three such edges in any optimum solution. This uses up $6|E_G|$ of the budget since $|F| = 2|E_G|$. Referring to Figure 5, we can see any $f_{ij} \in F$ needs two "private" light edges in X: one edge coming out of some vertex in A and some edge going into a vertex of D. This uses up $4|E_G|$ more from the budget leaving us with only 3ℓ edges.

Consider b_i for $i \in [\ell]$. First we claim that X must contain at least three black edges for b_i to have incoming and outgoing paths to the other terminals. The only outgoing edge from b_i is to vertices of H, and hence we need to pick an edge (b_i, h_v) such that $v \in H_i$. Since the only out-neighbor of h_v is c_v, it follows that X must pick this edge as well. Additionally, X also needs to contain at least one incoming edge into b_i to account for incoming paths from other terminals to b_i. So each b_i needs to have at least three edges selected in order to have incoming and outgoing paths to other terminals. Moreover, all these edges are clearly "private", i.e., different for each b_i. But as seen in the previous paragraph, our remaining budget was at most 3ℓ. Hence X selects exactly three such edges for each b_i. We now claim that once X contains the edges (b_i, h_v) and h_v, c_v such that $v \in H_i$ then X must also contain the edge (c_v, b_i). Suppose not, and for incoming towards b_i the solution X selects the edge (c_w, b_i) for some $w \in H_i$ such that $w \neq v$. Then since h_w is the only neighbor of c_w, the solution X would be forced to select this edge as well. This implies that at least four edges have been selected for b_i, which is a contradiction. So for every $i \in [\ell]$, there is a vertex $v_i \in H_i$ such that the edges $(b_i, h_{v_i}), (h_{v_i}, c_{v_i})$ and (c_{v_i}, b_i) are selected in the solution for the STRONGLY CONNECTED STEINER SUBGRAPH instance. Further these are the only black edges in X corresponding to b_i (refer to Figure 5). It also follows for each $f_{ij} \in F$, the solution X contains exactly three of the dotted edges (we argued above that each f_{ij} needs three dotted edges, and the budget now implies that this is the maximum we can allow).

Define $\phi : V_G \to V_H$ by $\phi(g_i) = v_i$ for each $i \in [\ell]$. Since $v_i \in H_i$ and the sets $H_1, H_2, \ldots, H_\ell$ form a disjoint partition of V_H, it follows that the function ϕ is an injection. Consider any edge $g_i g_j \in E_G$. We have seen above that the solution X contains exactly three dotted edges per $f_{ij} \in F$. Suppose for $f_{ij} \in F$ the solution X contains the edges $(f_{ij}, d_{vu}), (d_{vu}, a_{vu})$ and (a_{vu}, f_{ij}) for some $v \in H_i, u \in H_j$. The only incoming path for f_{ij} is via d_{vu}. Also the only outgoing path from b_i is via c_{v_i}. If $v_i \neq v$ then we will need two other dotted edges to reach f_{ij}, which is a contradiction since have already picked the allocated budget of three such edges. Hence, $v_i = v$. Similarly, it follows that $v_j = u$. Finally, the existence of the vertex d_{vu} implies $vu \in E_H$, i.e., $\phi(g_i)\phi(g_j) \in E_H$. □

6.1 Proof of Theorem 1.3

Finally, we are now ready to prove Theorem 1.3 which is restated below:

Theorem 1.3. *Under ETH, the edge-unweighted version of the SCSS problem cannot be solved in time $f(k) \cdot n^{o(k/\log k)}$ where f is any computable function, k is the number of terminals and n is the number of vertices in the instance.*

Proof. Lemma 6.2 and Lemma 6.3 together give a parameterized reduction from PSI to SCSS. Observe that the number of terminals k of the SCSS instance is $|B \cup F| = |V_G| + 2|E_G| = O(|E_G|)$ since we had the assumption that G is connected. The number of vertices in the SCSS instance is $|V^*| = |V_G| + 2|V_H| + 4|E_H| + 2|E_G| = O(|E_H|)$. Therefore from Theorem 6.1 we can conclude that under ETH there is no $f(k) \cdot n^{o(k/\log k)}$ algorithm for SCSS where n is the number of vertices in the graph and k is the number of terminals. □

7 W[1]-hardness for DSN in planar DAGs

The main goal of this section is to prove Theorem 1.4 which is restated below

Theorem 1.4. *The edge-unweighted version of the* DIRECTED STEINER NETWORK *problem is W[1]-hard parameterized by the number k of terminal pairs, even when the input is restricted to planar directed acyclic graphs (DAGs). Moreover, there is no $f(k) \cdot n^{o(k)}$ algorithm for any computable function f, unless the ETH fails.*

Note that this shows that the $n^{O(k)}$ algorithm of Feldman-Ruhl is asymptotically optimal. To prove Theorem 1.4, we reduce from the GRID TILING problem introduced by Marx [57].

$k \times k$ **GRID TILING**
Input : Integers k, n, and k^2 non-empty sets $S_{i,j} \subseteq [n] \times [n]$ where $1 \leq i, j \leq k$
Question: For each $1 \leq i, j \leq k$ does there exist an entry $s_{i,j} \in S_{i,j}$ such that

- If $s_{i,j} = (x, y)$ and $s_{i,j+1} = (x', y')$ then $x = x'$.
- If $s_{i,j} = (x, y)$ and $s_{i+1,j} = (x', y')$ then $y = y'$.

Consider an instance $(k, n, \{S_{i,j} : 1 \leq i, j \leq k\})$ of GRID TILING. We now build an instance (G, T) of edge-weighted DIRECTED STEINER NETWORK as shown in Figure 6. Set $T = \{(a_i, b_i) \cup (c_i, d_i) \ : i \in [k]\}$, i.e., we have $2k$ terminal pairs. We introduce k^2 red gadgets where each gadget is an $n \times n$ grid. Set the weight of each black edge to 2.

Definition 7.1. *An $a_i \rightsquigarrow b_i$* canonical *path is a path from a_i to b_i which starts with a blue edge coming out of a_i, then follows a horizontal path of black edges and finally ends with a blue edge going into b_i. Similarly, a $c_j \rightsquigarrow d_j$* canonical *path is a path from c_j to d_j which starts with a blue edge coming out of c_j, then follows a vertically downward path of black edges and finally ends with a blue edge going into d_j.*

There are n edge-disjoint $a_i \rightsquigarrow b_i$ canonical paths: let us call them $P_i^1, P_i^2, \ldots, P_i^n$ as viewed from top to bottom. They are named using magenta color in Figure 6. Similarly we call the canonical paths from c_j to d_j as $Q_j^1, Q_j^2, \ldots, Q_j^n$ when viewed from left to right. For each $i \in [k]$ and $\ell \in [n]$ we assign a weight of $\Delta(n+1-\ell)$ and $\Delta\ell$ to the first and last blue edges of P_i^ℓ, respectively. Similarly for each $j \in [k]$ and $\ell \in [n]$ we assign a weight of $\Delta(n+1-\ell)$ and $\Delta\ell$ to the first and last blue edges of Q_j^ℓ, respectively. Thus the total weight of the first and the last blue edges on any canonical path is exactly $\Delta(n+1)$. The idea is to choose Δ large enough such that in any optimum solution the paths between the terminals will be exactly the canonical paths. We will see later that $\Delta = 5n^2$ will suffice for this purpose. Any canonical path consists of the following set of edges:

- Two blue edges (which sum up to $\Delta(n+1)$)
- $(k+1)$ black edges not inside the gadgets
- $(n-1)$ black edges inside each gadget

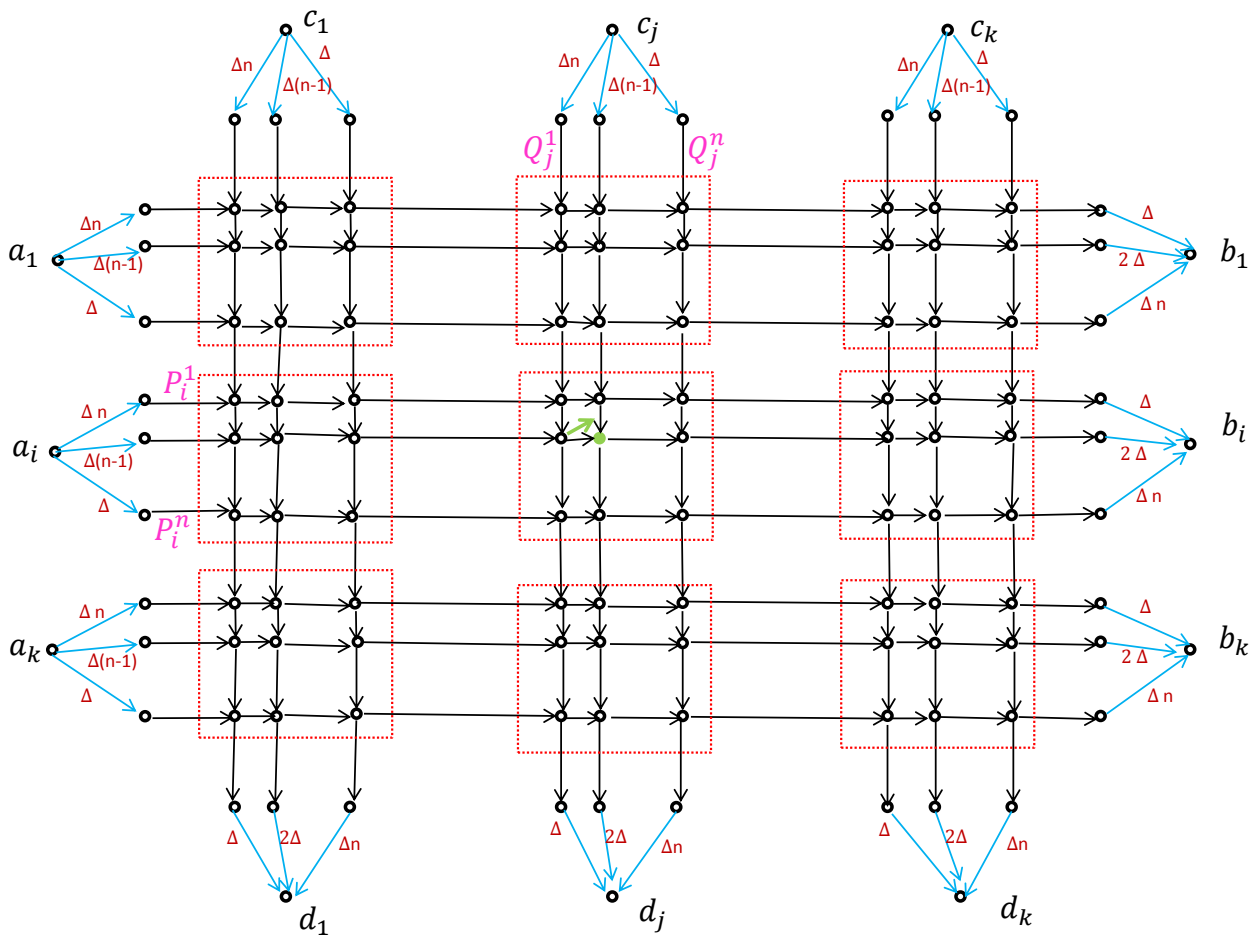

Figure 6: The instance of DIRECTED STEINER NETWORK created from an instance of GRID TILING.

Since the number of gadgets each canonical path visits is k and the weight of each black edge is 2, we have the total weight of any canonical path is $\Delta(n+1)+2(k+1)+2k(n-1)$.

Intuitively the k^2 gadgets correspond to the k^2 sets in the GRID TILING instance. Let us denote by $G_{i,j}$ the gadget which contains all vertices which lie on the intersection of any $a_i \rightsquigarrow b_i$ path and any $c_j \rightsquigarrow d_j$ path. If $(x,y) \in S_{i,j}$ then we color green the vertex in the gadget $G_{i,j}$ which is the unique intersection of the canonical paths P_i^x and Q_j^y. Then we add a shortcut as shown in Figure 7. The idea is if both the $a_i \rightsquigarrow b_i$ path and $c_j \rightsquigarrow d_j$ path pass through the green vertex then the $a_i \rightsquigarrow b_i$ path can save a weight of 1 by using the green edge and a vertical edge to reach the green vertex, instead of paying a weight of 2 to use the horizontal edge reaching the green vertex. It is easy to see that there is a solution (without using green edges) for the DSN instance of weight $B^* = 2k\Big(\Delta(n+1)+2(k+1)+2k(n-1)\Big)$: each terminal pair just uses a canonical path and these canonical paths are pairwise edge-disjoint.

The following assumption will be helpful in handling some of the border cases of the gadget construction. We may assume that $1 < \min\{x,y\}$ holds for every $(x,y) \in S_{i,j}$: indeed, we can increase n by one and replace every (x,y) by $(x+1,y+1)$ without changing the problem. Hence, no green vertex can be in the first row or first column of any gadget. Combining this fact with the orientation of the edges we get the only gadgets which can intersect any $a_i \rightsquigarrow b_i$ path are $G_{i,1}, G_{i,2}, \ldots, G_{i,k}$. Similarly the only gadgets which can intersect any $c_j \rightsquigarrow d_j$ path are $G_{1,j}, G_{2,j}, \ldots, G_{k,j}$. This completes the construction of the instance (G,T) of DIRECTED

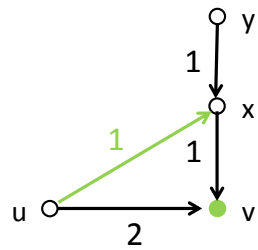

Figure 7: Let u, v be two consecutive vertices on the canonical path P_i^ℓ. Let v be on the canonical path $Q_j^{\ell'}$ and let y be the vertex preceding it on this path. If v is a green vertex then we subdivide the edge (y,v) by introducing a new vertex x and adding two edges (y,x) and (x,v) of weight 1. We also add an edge (u,x) of weight 1. The idea is if both the edges (y,v) and (u,v) were being used initially then now we can save a weight of 1 by making the horizontal path choose (u,x) and then we get (x,v) for free, as it is already being used by the vertical canonical path.

STEINER NETWORK.

Lemmas 7.2 and 7.6 below prove that the reduction described above is indeed a correct reduction from GRID TILING to DSN.

Lemma 7.2. *If the instance* $(k, n, \{S_{i,j} : 1 \le i,j \le k\})$ *of* GRID TILING *has a solution then the instance* (G,T) *of* DIRECTED STEINER NETWORK *has a solution of weight at most* $B^* - k^2$.

Proof. For each $1 \le i, j \le k$ let $s_{i,j} \in S_{i,j}$ be the entry in the solution of the GRID TILING instance. Therefore for every $i \in k$ we know that each of the k entries $s_{i,1}, s_{i,2}, \ldots, s_{i,k}$ have the same first coordinate α_i. Similarly for every $j \in [k]$ each of the k vertices $s_{1,j}, s_{2,j}, \ldots, s_{k,j}$ has the same second coordinate γ_j. For each $j \in [k]$ we use the canonical path $Q_j^{\gamma_j}$ to satisfy the terminal for (c_j, d_j). For each $i \in [k]$, we essentially use the canonical path $P_i^{\alpha_i}$ with the following modifications: for each $j \in [k]$, take the shortcut green edge (as shown in Figure 7) when we encounter the green vertex (this is guaranteed to happen since $(\alpha_i, \gamma_j) = s_{i,j} \in S_{i,j}$) in $G_{i,j}$ at intersection of the canonical paths $P_i^{\alpha_i}$ and $Q_j^{\gamma_j}$. Hence, overall we save a total of k^2: a saving of one per gadget. Thus, we have produced a solution for the instance (G,T) of weight $2k\Big(\Delta(n+1) + 2(k+1) + 2k(n-1)\Big) - k^2 = B^* - k^2$. □

We now prove the other direction which is more involved. First we show some preliminary claims:

Claim 7.3. *Any optimum solution for* (G,T) *contains a* $c_j \rightsquigarrow d_j$ *canonical path for each* $j \in [k]$.

Proof. Suppose to the contrary that there is an optimum solution N for (G,T) which does not contain a canonical $c_j \rightsquigarrow d_j$ path for some $j \in [k]$. From the orientation of the edges, we know that there is a $c_j \rightsquigarrow d_j$ path in N that starts with the blue edge from Q_j^ℓ and ends with a blue edge from $Q_j^{\ell'}$ for some $\ell' > \ell$. We create a new set of edges N' from N as follows:

- Add all those edges of Q_j^ℓ which were not present in N. In particular, we add the last blue edge of Q_j^ℓ since $\ell' > \ell$
- Delete the last blue edge of $Q_j^{\ell'}$.

It is easy to see that N' is also a solution for (G,T): this is because N' contains the canonical path Q_j^ℓ to satisfy the pair (c_j, d_j), and the last (blue) edge of any $c_j \rightsquigarrow d_j$ canonical path cannot be on any $a_i \rightsquigarrow b_i$ path for any $i \in [k]$. Changing the last blue edge saves us $(\ell' - \ell)\Delta \le \Delta = 5n^2$. However we have to be careful

since we added some edges to the solution. But these edges are the internal (black) edges of Q_j^ℓ, and their weight is $\leq 2(k+1)+2k(n-1) = 2kn+2 < 5n^2 = \Delta$ since $1 \leq k \leq n$. Therefore we are able to create a new solution N' whose weight is less than that of an optimum solution N, which is a contradiction. □

Definition 7.4. *An $a_i \rightsquigarrow b_i$ path is called an* almost canonical *path if its first and last edges are blue edges from the same $a_i \rightsquigarrow b_i$ canonical path.*

Hence, an $a_i \rightsquigarrow b_i$ almost canonical path looks very similar to an $a_i \rightsquigarrow b_i$ canonical path, except it can replace some of the horizontal black edges by green edges and vertical black edges as shown in Figure 7. However, note that by definition, an almost canonical path must however end on the same horizontal level on which it began. The proof of the next claim is very similar to that of Claim 7.3.

Claim 7.5. *Any optimum solution for DSN contains an $a_i \rightsquigarrow b_i$* almost canonical *path for every $i \in [k]$.*

Proof. Suppose to the contrary that there is an optimum solution N which does not contain an almost canonical $a_i \rightsquigarrow b_i$ path for some $i \in [k]$. Hence, the $a_i \rightsquigarrow b_i$ path in N starts and ends at different levels. From the orientation of the edges, we know that there is a $a_i \rightsquigarrow b_i$ path in the optimum solution that starts with the blue edge from P_i^ℓ and ends with a blue edge from $P_i^{\ell'}$ for some $\ell' > \ell$ (note that the construction in Figure 7 does not allow any $a_i \rightsquigarrow b_i$ path to climb onto an upper level).

We create a new set of edges N' from N as follows:

- Add all those edges of P_i^ℓ which were not present in N. Note that in particular, we add the last blue edge of P_i^ℓ since $\ell' > \ell$.
- Delete the last blue edge of $P_i^{\ell'}$.

It is easy to see that N' is also a solution for (G,T): this is because N' contains the canonical path P_i^ℓ to satisfy the pair (a_i, b_i), and the last (blue) edge of any $a_i \rightsquigarrow b_i$ canonical path cannot be on any $c_j \rightsquigarrow d_j$ path for any $j \in [k]$. Changing the last edge saves us $(\ell' - \ell)\Delta \leq \Delta = 5n^2$. But we have to careful since we also added some edges to the solution. The total weight of edges added is $\leq 2(k+1)+2k(n-1) = 2kn+2 < 5n^2 = \Delta$ since $1 \leq k \leq n$. So we are able to create a new solution N' whose weight is less than that of an optimum solution N, which is a contradiction. □

Lemma 7.6. *If the instance (G,T) of* Directed Steiner Network *has a solution of weight at most $B^* - k^2$ then the instance $(k, n, \{S_{i,j} : 1 \leq i,j \leq k\})$ of* Grid Tiling *has a solution.*

Proof. Consider any optimum solution X. By Claim 7.3 and Claim 7.5 we know that X has an $a_i \rightsquigarrow b_i$ almost canonical path and a $c_j \rightsquigarrow d_j$ canonical path for every $1 \leq i,j \leq k$. Moreover these set of $2k$ paths form a solution for DSN. Since any optimum solution is minimal, X is the union of these $2k$ paths: one for each terminal pair. For each $i,j \in [k]$ let the $a_i \rightsquigarrow b_i$ almost canonical path in X be $\overline{P}_i^{\alpha_i}$ and the $c_j \rightsquigarrow d_j$ canonical path in X be $Q_j^{\gamma_j}$.

The $a_i \rightsquigarrow b_i$ almost canonical path $\overline{P}_i^{\alpha_i}$ and $c_j \rightsquigarrow d_j$ canonical path $Q_j^{\gamma_j}$ in X intersect in a unique vertex in the gadget $G_{i,j}$. If each $a_i \rightsquigarrow b_i$ path was canonical instead of almost canonical, then the weight of X would have been exactly B^*. However we know that weight of X is at most $B^* - k^2$. It is easy to see any $a_i \rightsquigarrow b_i$ almost canonical path and any $c_j \rightsquigarrow d_j$ canonical path can have at most one edge in common: the edge which comes vertically downwards into the green vertex (see Figure 7). There are k^2 gadgets, and there is at most one edge per gadget which is used for two paths in X. Hence for each gadget $G_{i,j}$ there is exactly one edge which is used by both the $a_i \rightsquigarrow b_i$ almost canonical path and the $c_j \rightsquigarrow d_j$ canonical path in X. So the endpoint of each of these common edges must be a green vertex, i.e., $(\alpha_i, \gamma_j) \in S_{i,j}$ for each $i,j \in [k]$. □

7.1 Proof of Theorem 1.4

Finally, we are now ready to prove Theorem 1.4 which is restated below:

Theorem 1.4. *The edge-unweighted version of the* DIRECTED STEINER NETWORK *problem is W[1]-hard parameterized by the number k of terminal pairs, even when the input is restricted to planar directed acyclic graphs (DAGs). Moreover, there is no $f(k) \cdot n^{o(k)}$ algorithm for any computable function f, unless the ETH fails.*

Proof. Given an instance $(k, n, \{S_{i,j} : 1 \leq i, j \leq k\})$ of GRID TILING, we use the reduction described earlier in this section to build an instance (G, T) of edge-weighted DIRECTED STEINER NETWORK (see Figure 6 for an illustration). It is easy to see that the total number of vertices in G is $O(n^2k^2)$ and moreover can be constructed in $\text{poly}(n, k)$ time. Each grid is planar (green shortcut edges do not destroy planarity), and the grids are arranged again in a grid-like manner. Figure 6 actually gives a planar embedding of G. Moreover, it is not hard to observe that G is a DAG.

It is known [23, Theorem 14.28] that $k \times k$ GRID TILING is W[1]-hard parameterized by k, and under ETH cannot be solved in $f(k) \cdot n^{o(k)}$ for any computable function f. Combining the two directions from Lemma 7.2 and Lemma 7.6, we get a parameterized reduction from $k \times k$ GRID TILING to an instance of DSN which is a planar DAG and has $O(k)$ terminal pairs. Hence, it follows that DSN on planar DAGs is W[1]-hard and under ETH cannot be solved in $f(k) \cdot n^{o(k)}$ time for any computable function f. □

Note that Theorem 1.4 shows that the $n^{O(k)}$ algorithm of Feldman-Ruhl [35] for DSN is asymptotically optimal.

References

[1] Aboulker, P., Brettell, N., Havet, F., Marx, D., Trotignon, N.: Coloring graphs with constraints on connectivity. Journal of Graph Theory 85(4), 814–838 (2017), https://doi.org/10.1002/jgt.22109

[2] Bateni, M., Chekuri, C., Ene, A., Hajiaghayi, M.T., Korula, N., Marx, D.: Prize-collecting Steiner Problems on Planar Graphs. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011. pp. 1028–1049 (2011), https://doi.org/10.1137/1.9781611973082.79

[3] Bateni, M., Hajiaghayi, M.T., Marx, D.: Approximation Schemes for Steiner Forest on Planar Graphs and Graphs of Bounded Treewidth. J. ACM 58(5), 21:1–21:37 (2011), https://doi.org/10.1145/2027216.2027219

[4] Berman, P., Bhattacharyya, A., Makarychev, K., Raskhodnikova, S., Yaroslavtsev, G.: Approximation algorithms for spanner problems and directed steiner forest. Inf. Comput. 222, 93–107 (2013), https://doi.org/10.1016/j.ic.2012.10.007

[5] Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Fourier meets möbius: fast subset convolution. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007. pp. 67–74 (2007), https://doi.org/10.1145/1250790.1250801

[6] Bodlaender, H.L., Fomin, F.V., Lokshtanov, D., Penninkx, E., Saurabh, S., Thilikos, D.M.: (Meta) Kernelization. J. ACM 63(5), 44:1–44:69 (2016), https://doi.org/10.1145/2973749

[7] Bodlaender, H.L., Lokshtanov, D., Penninkx, E.: Planar Capacitated Dominating Set Is $W[1]$-Hard. In: Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers. pp. 50–60 (2009), https://doi.org/10.1007/978-3-642-11269-0_4

[8] Bonnet, É., Giannopoulos, P., Lampis, M.: On the Parameterized Complexity of Red-Blue Points Separation. In: 12th International Symposium on Parameterized and Exact Computation, IPEC 2017, September 6-8, 2017, Vienna, Austria. pp. 8:1–8:13 (2017), https://doi.org/10.4230/LIPIcs.IPEC.2017.8

[9] Bonnet, É., Miltzow, T.: Parameterized Hardness of Art Gallery Problems. In: 24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark. pp. 19:1–19:17 (2016), https://doi.org/10.4230/LIPIcs.ESA.2016.19

[10] Bonnet, É., Sikora, F.: The Graph Motif problem parameterized by the structure of the input graph. Discrete Applied Mathematics 231, 78–94 (2017), https://doi.org/10.1016/j.dam.2016.11.016

[11] Borradaile, G., Klein, P.N., Mathieu, C.: An $O(n \log n)$ approximation scheme for Steiner tree in planar graphs. ACM Trans. Algorithms 5(3), 31:1–31:31 (2009), https://doi.org/10.1145/1541885.1541892

[12] Bringmann, K., Kozma, L., Moran, S., Narayanaswamy, N.S.: Hitting set for hypergraphs of low vc-dimension. In: 24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark. pp. 23:1–23:18 (2016), https://doi.org/10.4230/LIPIcs.ESA.2016.23

[13] Cai, L., Fellows, M.R., Juedes, D.W., Rosamond, F.A.: The Complexity of Polynomial-Time Approximation. Theory Comput. Syst. 41(3), 459–477 (2007), https://doi.org/10.1007/s00224-007-1346-y

[14] Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S., Li, M.: Approximation Algorithms for Directed Steiner Problems. J. Algorithms 33(1), 73–91 (1999), https://doi.org/10.1006/jagm.1999.1042

[15] Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. J. Comput. Syst. Sci. 72(8), 1346–1367 (2006), https://doi.org/10.1016/j.jcss.2006.04.007

[16] Chitnis, R., Esfandiari, H., Hajiaghayi, M.T., Khandekar, R., Kortsarz, G., Seddighin, S.: A Tight Algorithm for Strongly Connected Steiner Subgraph on Two Terminals with Demands. Algorithmica 77(4), 1216–1239 (2017), https://doi.org/10.1007/s00453-016-0145-8

[17] Chitnis, R., Feldmann, A.E.: FPT Inapproximability of Directed Cut and Connectivity Problems. To appear in IPEC (2019)

[18] Chitnis, R., Feldmann, A.E., Manurangsi, P.: Parameterized Approximation Algorithms for Bidirected Steiner Network Problems. In: 26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland. pp. 20:1–20:16 (2018), https://doi.org/10.4230/LIPIcs.ESA.2018.20

[19] Chitnis, R.H., Hajiaghayi, M., Marx, D.: Tight Bounds for Planar Strongly Connected Steiner Subgraph with Fixed Number of Terminals (and Extensions). In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014. pp. 1782–1801 (2014), https://doi.org/10.1137/1.9781611973402.129

[20] Curticapean, R., Dell, H., Marx, D.: Homomorphisms are a good basis for counting small subgraphs. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017. pp. 210–223 (2017), https://doi.org/10.1145/3055399.3055502

[21] Curticapean, R., Marx, D.: Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014. pp. 130–139 (2014), https://doi.org/10.1109/FOCS.2014.22

[22] Curticapean, R., Xia, M.: Parameterizing the permanent: Genus, apices, minors, evaluation mod 2k. In: IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015. pp. 994–1009 (2015), https://doi.org/10.1109/FOCS.2015.65

[23] Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015), https://doi.org/10.1007/978-3-319-21275-3

[24] Demaine, E.D., Fomin, F.V., Hajiaghayi, M.T., Thilikos, D.M.: Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. J. ACM 52(6), 866–893 (2005), https://doi.org/10.1145/1101821.1101823

[25] Demaine, E.D., Hajiaghayi, M.T.: Graphs excluding a fixed minor have grids as large as treewidth, with combinatorial and algorithmic applications through bidimensionality. In: Proceedings of the

Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005. pp. 682–689 (2005), http://dl.acm.org/citation.cfm?id=1070432.1070528

[26] Demaine, E.D., Hajiaghayi, M.T., Klein, P.N.: Node-Weighted Steiner Tree and Group Steiner Tree in Planar Graphs. ACM Trans. Algorithms 10(3), 13:1–13:20 (2014), https://doi.org/10.1145/2601070

[27] Demaine, E.D., Hajiaghayi, M.: The Bidimensionality Theory and Its Algorithmic Applications. Comput. J. 51(3), 292–302 (2008), https://doi.org/10.1093/comjnl/bxm033

[28] Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science, Springer (1999), https://doi.org/10.1007/978-1-4612-0515-9

[29] Dreyfus, S.E., Wagner, R.A.: The steiner problem in graphs. Networks 1(3), 195–207 (1971), https://doi.org/10.1002/net.3230010302

[30] Dvorák, P., Feldmann, A.E., Knop, D., Masarík, T., Toufar, T., Veselý, P.: Parameterized Approximation Schemes for Steiner Trees with Small Number of Steiner Vertices. In: 35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France. pp. 26:1–26:15 (2018), https://doi.org/10.4230/LIPIcs.STACS.2018.26

[31] Eiben, E., Knop, D., Panolan, F., Suchý, O.: Complexity of the Steiner Network Problem with Respect to the Number of Terminals. In: 36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany. pp. 25:1–25:17 (2019), https://doi.org/10.4230/LIPIcs.STACS.2019.25

[32] Eisenstat, D., Klein, P.N., Mathieu, C.: An efficient polynomial-time approximation scheme for Steiner forest in planar graphs. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012. pp. 626–638 (2012), https://doi.org/10.1137/1.9781611973099.53

[33] Enciso, R., Fellows, M.R., Guo, J., Kanj, I.A., Rosamond, F.A., Suchý, O.: What Makes Equitable Connected Partition Easy. In: Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers. pp. 122–133 (2009), https://doi.org/10.1007/978-3-642-11269-0_10

[34] Eppstein, D., Lokshtanov, D.: The Parameterized Complexity of Finding Point Sets with Hereditary Properties. In: 13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland. pp. 11:1–11:14 (2018), https://doi.org/10.4230/LIPIcs.IPEC.2018.11

[35] Feldman, J., Ruhl, M.: The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals. SIAM J. Comput. 36(2), 543–561 (2006), https://doi.org/10.1137/S0097539704441241

[36] Feldmann, A.E., Marx, D.: The Complexity Landscape of Fixed-Parameter Directed Steiner Network Problems. A preliminary version appeared in ICALP 2016. CoRR abs/1707.06808 (2017), http://arxiv.org/abs/1707.06808

[37] Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series, Springer (2006), https://doi.org/10.1007/3-540-29953-X

[38] Fomin, F.V., Kolay, S., Lokshtanov, D., Panolan, F., Saurabh, S.: Subexponential Algorithms for Rectilinear Steiner Tree and Arborescence Problems. In: 32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA. pp. 39:1–39:15 (2016), https://doi.org/10.4230/LIPIcs.SoCG.2016.39

[39] Fomin, F.V., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Subexponential Parameterized Algorithms for Planar and Apex-Minor-Free Graphs via Low Treewidth Pattern Covering. In: IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, New Brunswick, New Jersey, USA. pp. 515–524 (2016), https://doi.org/10.1109/FOCS.2016.62

[40] Frick, M., Grohe, M.: Deciding first-order properties of locally tree-decomposable structures. J. ACM 48(6), 1184–1206 (2001), https://doi.org/10.1145/504794.504798

[41] Guo, J., Hartung, S., Niedermeier, R., Suchý, O.: The Parameterized Complexity of Local Search for TSP, More Refined. Algorithmica 67(1), 89–110 (2013), https://doi.org/10.1007/s00453-012-9685-8

[42] Guo, J., Niedermeier, R., Suchý, O.: Parameterized Complexity of Arc-Weighted Directed Steiner Problems. SIAM J. Discrete Math. 25(2), 583–599 (2011), https://doi.org/10.1137/100794560

[43] Hakimi, S.L.: Steiner's problem in graphs and its implications. Networks 1(2), 113–133 (1971), https://doi.org/10.1002/net.3230010203

[44] Halperin, E., Krauthgamer, R.: Polylogarithmic inapproximability. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA. pp. 585–594 (2003), https://doi.org/10.1145/780542.780628

[45] Impagliazzo, R., Paturi, R.: On the Complexity of k-SAT. J. Comput. Syst. Sci. 62(2), 367–375 (2001), https://doi.org/10.1006/jcss.2000.1727

[46] Impagliazzo, R., Paturi, R., Zane, F.: Which Problems Have Strongly Exponential Complexity? J. Comput. Syst. Sci. 63(4), 512–530 (2001), https://doi.org/10.1006/jcss.2001.1774

[47] Jansen, K., Kratsch, S., Marx, D., Schlotter, I.: Bin packing with fixed number of bins revisited. J. Comput. Syst. Sci. 79(1), 39–49 (2013), https://doi.org/10.1016/j.jcss.2012.04.004

[48] Jones, M., Lokshtanov, D., Ramanujan, M.S., Saurabh, S., Suchý, O.: Parameterized Complexity of Directed Steiner Tree on Sparse Graphs. SIAM J. Discrete Math. 31(2), 1294–1327 (2017), https://doi.org/10.1137/15M103618X

[49] Karp, R.M.: Reducibility Among Combinatorial Problems. In: Proceedings of a Symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. pp. 85–103 (1972), https://doi.org/10.1007/978-1-4684-2001-2_9

[50] Klein, P.N., Marx, D.: Solving Planar k-Terminal Cut in $\mathrm{O}(n^{c\sqrt{k}})$ Time. In: Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I. pp. 569–580 (2012), https://doi.org/10.1007/978-3-642-31594-7_48

[51] Klein, P.N., Marx, D.: A subexponential parameterized algorithm for Subset TSP on planar graphs. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014,

Portland, Oregon, USA, January 5-7, 2014. pp. 1812–1830 (2014), https://doi.org/10.1137/1.9781611973402.131

[52] Levin, A.: Algorithm for the shortest connection of a group of graph vertices. Soviet Math. Dokl. 12, 1477–1481 (1971), http://dx.doi.org/10.4086/toc.2010.v006a005

[53] Li, C., McCormick, S.T., Simchi-Levi, D.: The point-to-point delivery and connection problems: complexity and algorithms. Discrete Applied Mathematics 36(3), 267–292 (1992), https://doi.org/10.1016/0166-218X(92)90258-C

[54] Lokshtanov, D., Ramanujan, M.S., Saurabh, S., Zehavi, M.: Parameterized Complexity and Approximability of Directed Odd Cycle Transversal. To appear in SODA 2020. CoRR abs/1704.04249 (2017), http://arxiv.org/abs/1704.04249

[55] Lokshtanov, D., Saurabh, S., Wahlström, M.: Subexponential Parameterized Odd Cycle Transversal on Planar Graphs. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India. pp. 424–434 (2012), https://doi.org/10.4230/LIPIcs.FSTTCS.2012.424

[56] Marx, D.: On the Optimality of Planar and Geometric Approximation Schemes. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings. pp. 338–348 (2007), https://doi.org/10.1109/FOCS.2007.50

[57] Marx, D.: Can You Beat Treewidth? Theory of Computing 6(1), 85–112 (2010), https://doi.org/10.4086/toc.2010.v006a005

[58] Marx, D.: A Tight Lower Bound for Planar Multiway Cut with Fixed Number of Terminals. In: Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I. pp. 677–688 (2012), https://doi.org/10.1007/978-3-642-31594-7_57

[59] Marx, D., Pilipczuk, M., Pilipczuk, M.: On Subexponential Parameterized Algorithms for Steiner Tree and Directed Subset TSP on Planar Graphs. In: 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018. pp. 474–484 (2018), https://doi.org/10.1109/FOCS.2018.00052

[60] Marx, D., Pilipczuk, M.: Optimal Parameterized Algorithms for Planar Facility Location Problems Using Voronoi Diagrams. In: Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings. pp. 865–877 (2015), https://doi.org/10.1007/978-3-662-48350-3_72

[61] Natu, M., Fang, S.: The Point-to-point Connection Problem - Analysis and Algorithms. Discrete Applied Mathematics 78(1-3), 207–226 (1997), https://doi.org/10.1016/S0166-218X(97)00010-3

[62] Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006), https://doi.org/10.1093/ACPROF:OSO/9780198566076.001.0001

[63] Pilipczuk, M., Pilipczuk, M., Sankowski, P., van Leeuwen, E.J.: Subexponential-Time Parameterized Algorithm for Steiner Tree on Planar Graphs. In: 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 - March 2, 2013, Kiel, Germany. pp. 353–364 (2013), https://doi.org/10.4230/LIPIcs.STACS.2013.353

[64] Pilipczuk, M., Wahlström, M.: Directed Multicut is W[1]-hard, Even for Four Terminal Pairs. TOCT 10(3), 13:1–13:18 (2018), https://doi.org/10.1145/3201775

[65] Ramanathan, S.: Multicast tree generation in networks with asymmetric links. IEEE/ACM Trans. Netw. 4(4), 558–568 (1996), https://doi.org/10.1109/90.532865

[66] Robertson, N., Seymour, P.D., Thomas, R.: Quickly Excluding a Planar Graph. J. Comb. Theory, Ser. B 62(2), 323–348 (1994), https://doi.org/10.1006/jctb.1994.1073

[67] Salama, H.F., Reeves, D.S., Viniotis, Y.: Evaluation of Multicast Routing Algorithms for Real-Time Communication on High-Speed Networks. IEEE Journal on Selected Areas in Communications 15(3), 332–345 (1997), https://doi.org/10.1109/49.564132

[68] Suchý, O.: On Directed Steiner Trees with Multiple Roots. In: Graph-Theoretic Concepts in Computer Science - 42nd International Workshop, WG 2016, Istanbul, Turkey, June 22-24, 2016, Revised Selected Papers. pp. 257–268 (2016), https://doi.org/10.1007/978-3-662-53536-3_22

[69] Winter, P.: Steiner problem in networks: A survey. Networks 17(2), 129–167 (1987), https://doi.org/10.1002/net.3230170203

A Vertex-unweighted versions are more general than edge-weighted versions with integer weights

In this section, for both the SCSS and DSN problems we show that the edge-weighted version (with polynomially-bounded integer weights) can be solved using the vertex-unweighted version. Hence all our hardness results from Theorem 1.2, Theorem 1.3 and Theorem 1.4 hold for the vertex-(un)weighted versions as well.

We give a formal proof for the DSN problem; the proof for the SCSS problem is similar. Consider an instance $I_1 = (G,T)$ of edge-weighted DSN with integer weights where $T = \{(s_i,t_i) \mid i \in [k]\}$. Replace each edge of weight ℓ by $n\ell$ internal vertices where $|G| = n$. Let the new graph be G'. Consider the instance I_2 of vertex-unweighted version where the set of terminals is the same as in I_1.

Theorem A.1. *The instance I_1 of edge-weighted DSN has a solution of weight at most C if and only if the instance I_2 of vertex-unweighted DSN has a solution with at most $Cn+n$ vertices.*

Proof. Suppose there is a solution E_1 for I_1 of weight at most C. For each edge in E_1 pick all its internal vertices and two endpoints in E_2. Clearly E_2 is a solution for I_2. The number of vertices in E_2 is $Cn+\gamma$ where γ is the number of vertices of G incident to the edges in E_1. Since $\gamma \leq n$ we are done.

Suppose there is a (vertex-minimal) solution E_2 for I_2 having at most $Cn+n$ vertices. For any edge $e \in G$ of weight c we need to pick all the cn internal vertices (plus the two endpoints of e) in E_2 if we actually want to use e in a solution for I_1. So for every edge $e \in E$ we know that E_2 contains either all or none of the internal vertices obtained after splitting up e according to its weight in G. Let the set of edges of G all of whose internal vertices are in E_2 be $E_1 = \{e_1, e_2, \ldots, e_r\}$ and their weights be $c_1, c_2, \ldots, c_r$ respectively. Since E_2 is a solution for I_2 it follows that E_1 is a solution for I_1. Let S be the union of set of endpoints of the edges in E_1. Therefore $Cn+n \geq |S| + n(\sum_{i=1}^{r} c_i)$. Since $|S| \geq 1$ we have $C \geq \sum_{i=1}^{r} c_i$, i.e., E_1 has weight at most C. □

Note that the above reduction works even in the case when the edges have zero weight[12]: in this case we simply won't be adding any internal vertices.

[12] We mention this explicitly because some of the reductions in this paper do have edges with zero weight

B Treewidth and Minors

Definition B.1. ***(treewidth)*** *Let G be a given undirected graph. Let $\mathcal{T}$ be a tree and $B : V(\mathcal{T}) \to 2^{V(G)}$. The pair $(\mathcal{T}, B)$ is called as a* tree decomposition *of an undirected graph G is a tree $\mathcal{T}$ in which every vertex $x \in V(\mathcal{T})$ has an assigned set of vertices $B_x \subseteq V(G)$ (called a bag) such that the following properties are satisfied:*

- ***(P1)****: $\bigcup_{x \in V(\mathcal{T})} B_x = V(G)$.*
- ***(P2)****: For each $\{u, v\} \in E(G)$, there exists an $x \in V(\mathcal{T})$ such that $u, v \in B_x$.*
- ***(P3)****: For each $v \in V(G)$, the set of vertices of $\mathcal{T}$ whose bags contain v induce a connected subtree of $\mathcal{T}$.*

The width *of the tree decomposition $(\mathcal{T}, B)$ is $\max_{x \in V(\mathcal{T})} |B_x| - 1$. The treewidth of a graph G, usually denoted by $\boldsymbol{tw}(G)$, is the minimum width over all tree decompositions of G.*

Definition B.2. ***(minor)*** *Let G, H be undirected graphs. Then H is called a minor of G if H can be obtained from G by deleting edges, deleting vertices and by contracting edges.*

Definition B.3. ***(subdivision)*** *Let G be an undirected graph. An edge $e = u - v$ is subdivided by adding a new vertex w and the edges $u - w$ and $v - w$. An undirected graph H is called a subdivision of G if H can be obtained from G by subdividing edges of G.*

Lemma B.4. *Subdivisions of outerplanar graphs have treewidth at most* 2.

Proof. Outerplanar graphs are known to be a subclass of series parallel graphs, and hence have treewidth at most 2. To prove this lemma, it is enough to show that subdividing one edge of an outerplanar does not increase the treewidth. Let G be an outerplanar graph, and $(\mathcal{T}, B)$ be a tree-decomposition of G of width at most 2. Let $e = u - v$ be an edge in G which is subdivided by adding a vertex w and the edges $u - w$ and $v - w$. We now build a tree decomposition for the resulting graph G'. We add only one vertex to $V(\mathcal{T})$: by property **(P2)**, there exists $t \in V(\mathcal{T})$ such that $u, v \in B_t$. Add a new vertex t' and set $B_{t'} = \{u, v, w\}$. Make t' adjacent only to t. It is easy to check that $V(\mathcal{T}) \cup \{t'\}$ is a tree-decomposition for G' of treewidth at most 2. □