AN ABSTRACT OF THE DISSERTATION OF

Hung Le for the degree of Doctor of Philosophy in Computer Science presented on May 24, 2018.

Title: Structural Results and Approximation Algorithms in Minor-free Graphs

Abstract approved: ______________________________

Glencora Borradaile

Planarity has been successfully exploited to design faster and more accurate approximation algorithms for many graph optimization problems. The celebrated theorem of Kuratowski [66] completely characterizes planar graphs as those excluding K_5 and $K_{3,3}$ as minors. Kuratowski's theorem allows one to generalize planar graphs to H-minor-free graphs: those that exclude a fixed graph H as a minor. The deep results of Robertson and Seymour [82] reveal many hidden structures in H-minor-free graphs, that have been used extensively in algorithmic designs. Relying on these structures, we design (i) an (efficient) polynomial time approximation scheme (PTAS) for two different variants of the traveling salesperson problem (TSP) and (ii) simple local search PTASes for r-dominating set and feedback vertex set problems. We then present several results concerning structures of planar graphs. Specifically, we make progresses on two conjectures on existence of large induced forests in planar graphs.

©Copyright by Hung Le
May 24, 2018
All Rights Reserved

Structural Results and Approximation Algorithms in Minor-free Graphs

by

Hung Le

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented May 24, 2018
Commencement June 2018

Doctor of Philosophy dissertation of Hung Le presented on May 24, 2018.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Hung Le, Author

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Cora Borradaile who has been a very patient and supportive advisor. This thesis is not possible without her guidance throughout the years. I have learned so many things, both research and non-research related, from conversations with her.

I have been so lucky to be a collaborator of many amazing individuals. The collaboration with Christian Christian Wulff-Nilsen has sparked ideas in my best papers. I have learned many techniques from hours brainstorming with Amir Nayyeri and Baigong Zheng on many problems. I would like to thank all of you and hope that we can have more fruitful collaborations in the future.

I would like to thank Christian Wulff-Nilse, Amir Nayyeri , Mike Rosulek, Eugene Zhang and Clayton Petsche for serving on my committee. I also want to thank all faculty and students of the theory group for creating a warm and scholarly environment.

Thank you to my friends in the GRE group, in particular Chi, Huyen Ga, Tan, Trang, Tuan, Tuyen. They inspired me a lot to pursue PhD in the U.S. Special thanks to Duc Le, who helped connecting me and Cora in the first place, and to my friends in Corvallis who have made this rainy town memorable.

Last but definitely not the least, I would like to thank my family for always standing by my side. Thanks to my wife, Thuong, who makes my PhD not only about research, but also about love. Thank you for taking care of our little buddy, Viet Le, when I am not at home. And thanks to Viet who deserves all credits for improving our life in so many ways. I would like to express my sincere gratitude to my parents and sister, who put so much trust in me and support me in every decision I made.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

Figure | Page

LIST OF FIGURES (Continued)

Chapter 1: Introduction

Planarity has been used to design faster algorithms for many polynomial time solvable problems and to design more accurate approximation algorithms for many NP-hard optimization problems. Several representative examples are shortest path [49], network flow [19], maximum matching [77], TSP [60] and independent set [72]. We can think of planar graphs as graphs that can be drawn on a plane (or equivalently on the surface of a 3D sphere) without edge crossings. This definition of planar graphs can be extended to a more general class of graphs, called *bounded-genus graphs*, where we can draw the graphs on a surface of bounded genus without edge crossings. The non-crossing drawing allows us to generalize many algorithmic results in planar graphs to bounded-genus graphs [25, 18].

Kuratowski [66] characterized planar graphs in terms of their excluded minors; a graph is planar if it excludes K_5 and $K_{3,3}$ as minors.[1] We can characterize bounded-genus graphs in the same way, but the set of excluded minors are typically large. The minor-exclusion characterization allows us to define a more general class of graphs, called *H-minor-free graphs*, that exclude a fixed graph H as a minor. H-minor-free graphs are much broader than bounded-genus graphs: one representative example is $K_{3,\ell}$; when ℓ is the part of the input, $K_{3,\ell}$ cannot be embedded in any bounded-genus surface [81] but excludes K_5 as a minor. Figure 1.1 illustrates the containment between different classes of graphs.[2]

A natural research question is: can we generalize known algorithmic results in planar graphs and bounded-genus graphs to H-minor-free graphs? There have been several successful efforts toward answering this question. The deletion decomposition and contraction decomposition frameworks by Demaine, Hajiaghayi and Kawarabayashi [37, 38] extend PTASes for many unweighted graph problems in planar graphs to H-minor-free graphs. However, there has been little progress on extending PTASes for the most interesting edge-weighted connectivity problems, including: TSP, subset TSP, Steiner tree

[1] The definition of minor is given in Section 1.1.
[2] Bounded treewidth and bounded pathwidth graphs are defined in Section 1.1.

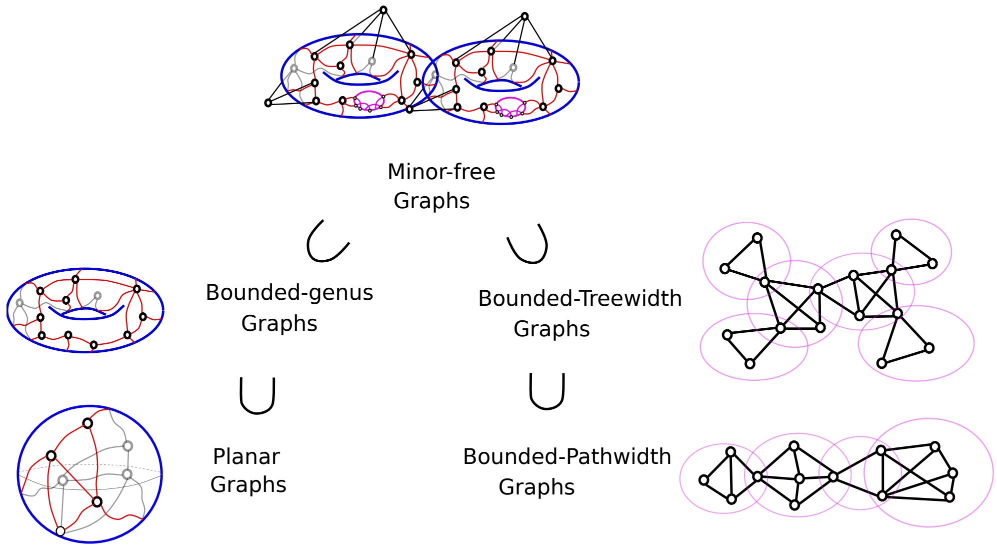

Figure 1.1: Relationship between different classes of of graphs

and survivable network design. In planar graphs, PTASes depend on the construction of spanners which depend heavily on planar embedding. However, in H-minor-free graphs, such non-crossing embeddings are unavailable. The Robertson and Seymour decomposition theorem only provides a *weak embedding* where we allow crossing on some part of the embedding of graphs. Toward addressing this difficulty, we present in Chapter 2 spanner constructions for TSP and subset TSP problems that does not depend on such weak embeddings of the input graph. Our result implies an efficient PTAS for TSP problem and the first PTAS for the subset TSP problem in H-minor-free graphs and opens a new possibility to construct embedding-free spanners for other connectivity problems in H-minor-free graphs.

Another emerging research direction for designing PTASes for H-minor-free graphs is local search. Previous PTASes for H-minor-free graphs use the deletion and contraction decomposition frameworks of Demaine, Hajiaghayi and Kawarabayashi [37, 38] that in turn, rely on the Robertson and Seymour decomposition algorithm. However, the running time to obtain this decomposition has an enormous dependency[3] [55] on the size of the minor. Local search PTASes [78, 26] are conceptually simple and the algorithms

[3]Even when $|V(H)| = 5$, the constant is bigger than the size of the universe.

do not require a Robertson-Seymour decomposition or indeed any property of the input graphs; only the analysis does. However, the PTASes obtained by local search are non-efficient in the theoretical sense; making local search PTASes efficient is an intersting research direction. In Chapter 3, we present local search PTASes for the r-dominating set problem, even when r is part of the input, and for the feedback vertex set (FVS) problem. Previous local search PTASes only apply to problems that have "local properties" [26, 46] while FVS has a global constraint. Thus, to analyze the local search algorithm for FVS, we introduce several new ideas and we believe our ideas could be useful elsewhere.

In Chapter 4, we present our work concerning the existence of a large induced forest in planar graphs. Albertson and Berman [4] conjectured that any planar graph has an induced forest of at least half of the vertices. The conjecture, if true, would provide an alternate proof for the fact that any planar graph has an independent set containing at least a quarter of vertices; this fact is only known by the four color theorem whose proof is computer-assisted. The Albertson-Berman Conjecture is known to be true in the case of outer-planar graphs [54] and very sparse planar graphs [5]. We prove the Albertson-Berman Conjecture for two-outerplanar graphs. In the same vein, we study the Akiyama and Watanabe conjecture: any bipartite planar graph has an induced forest containing at least $\frac{5}{8}$ of its vertices. We present our results on a stronger version of the Akiyama-Watanabe Conjecture in Section 4.2.

1.1 Preliminaries

Standard graph terminologies Let G be a finite, simple graph. We denote the vertex set and the edge set of G by $V(G)$ and $E(G)$, respectively. We use n and m to denote the number of vertices and edges of G. The order of G is the number of vertices, denoted by $|G|$. Let X be a subset of vertices of G. We define $G[X]$ to be the subgraph of G induced by X. Two special graphs we consider in this proposal are K_ℓ, the complete graph on ℓ vertices, and $K_{p,q}$, the *complete bipartite graph* with p vertices on one side and q vertices on the other side.

A *walk* W of length d in G is an alternating sequence of vertices and edges $\{v_1, e_1, v_2, \ldots, e_d, v_{d+1}\}$ such that v_i, v_{i+1} are the endpoints of e_i, $1 \leq i \leq d$. We call W a *closed walk* if $v_1 = v_{d+1}$. If no vertex of W is repeated twice, we call W a *path*. In this case, we denote the subpath of W between u and v by $W[u, v]$. Let W_1, W_2 be two walks of

G such that the last vertex of W_1 is the first vertex of W_2. We define the composition of W_1 and W_2, denoted by $W_1 \circ W_2$, to be the walk obtained by identifying the last vertex of W_1 and the first vertex of W_2.

Let S be a connected subgraph of G. By $w_G(S)$, we denote the total edge weight of S. We define the diameter of S, denoted by $\mathrm{diam}(S)$, to be $\mathrm{diam}(S) \stackrel{\mathrm{def}}{=} \max_{u,v \in S} d_S(u,v)$. A shortest path D in S where $w_G(D) = \mathrm{diam}(S)$ is called a *diameter path* of S.

Polynomial time approximation scheme A polynomial time approximation scheme (PTAS) is an algorithm, which is given a fixed error parameter ϵ, can find a $(1 \pm \epsilon)$-approximate solution for an optimization problem in polynomial time. A PTAS is efficient (an EPTAS) if its running time is $2^{f(\epsilon)} n^{O(1)}$ for some function $f(\cdot)$ of ϵ that is independent of n.

Tree decomposition A *tree decomposition* of a graph G is a pair $(\mathcal{T}, \mathcal{X})$ where $\mathcal{X}$ is a family of subsets of V, called *bags*, and $\mathcal{T}$ is a tree whose nodes are bags in $\mathcal{X}$ such that:

(i) $\cup_{X \in \mathcal{X}} X = V(G)$.

(ii) For every edge $uv \in E$, there is a bag $X \in \mathcal{X}$ that contains both u and v.

(iii) For every $u \in V$, the set of bags containing u induces a (connected) subtree of $\mathcal{T}$.

The *width* of $(\mathcal{T}, \mathcal{X})$ is $\max_{X \in \mathcal{X}}(|X| - 1)$ and the *treewidth* of G is the minimum width over all possible tree decompositions of G.

A *path decomposition* is a tree decomposition where the underlying tree is a path. *Pathwidth* of G is the minimum width over all possible path decompositions of G

Graph minors We define the *contraction* of an edge e, denoted by G/e, as the graph obtained from G by identifying endpoints of e and removing e from the graph. We can naturally extend the contraction to a set of edges, say X, to be the graph, denoted by G/X, obtained by contracting every edge in X. A graph H is called a *minor* of G if H can be obtained from G by a sequence of edge deletion, vertex deletion and edge contraction operations. A graph is called *H-minor-free* if it excludes a fixed graph H as a minor. Graph G is *planar* if we can draw G on a plane (or equivalently on a surface of a sphere) without edge crossing. Kuratowski [66] showed the following relationship between planarity and excluded minors:

Theorem 1.1.1 (Kuratowski [66]). *A graph is planar if and only if it excludes K_5 and $K_{3,3}$ as minors.*

A family of graphs $\mathcal{G}$ is *minor-closed* if for every graph $K \in \mathcal{G}$, every minor of K is also in $\mathcal{G}$. Wagner's conjecture said that any minor-closed family excludes a finite set of minors. The conjecture was proved by the work of Robertson and Seymour [83] that spanned over 20 years.

Theorem 1.1.2 (Robertson and Seymour [83]). *Any minor-closed family of graphs excludes a finite set of minors.*

Over the course of proving the Wagner's conjecture, Robertson and Seymour [82] proved a structural decomposition of H-minor-free graphs. Such a decomposition has numerous algorithmic consequences [38]. However, the constant involved in the Robertson-Seymour decomposition is galactic [71, 55] in the size of the minor. In some cases, we can avoid such a dependency by relying the sparsity of minor-free graphs.

Lemma 1.1.3. *(Mader [75]) An H-minor-free graph of n vertices has $O(\sigma_H n)$ edges where $\sigma_H = |V(H)|\sqrt{\log |V(H)|}$.*

We note that Mader's bound is tight as shown by Thomason [88].

1.2 Bibliography

Results presented in Section 2.2 of Chapter 2 are based on the joint work with Glencora Borradaile and Christian Wulff-Nilsen [24]. Results presented in Section 2.3 of Chapter 2 are based on a manuscript in submission at the time of writing [69]. Results presented in Chapter 3 are from a joint work with Baigong Zheng [70]. Results presented in Section 4.2 of Chapter 4 are from a joint work with Glencora Borradaile and Melissa Sherman-Bennett [67]. Results in Section 4.3 of Chapter 4 are based on [68].

Chapter 2: The Traveling Salesperson Problem

The Traveling Salesperson Problem (TSP) is a classic combinatorial optimization that has been studies since 1930s. TSP problem has found numerous practical applications such as vehicle routing, scheduling, computer wiring [76]. It is in Karp's list of NP-hard problems [56] and remains NP-hard even in planar graphs [50]. In this chapter, we consider the TSP problem in H-minor-free graphs and its subset version. We state the problems here:

TSP problem Given an edge-weighted graph G, find a shortest tour that goes through every vertex of G.

Subset TSP problem Given an edge-weighted graph G and a set of terminal T, find a shortest tour that goes through every vertex of T.

Before going into details of each problem, we describe a general framework to design PTAS for these problems in H-minor-free graphs.

2.1 A framework for designing PTASes for connectivity problems in H-minor-free graphs

Graphs in this section are edge-weighted and H-minor-free. Klein [60] was the first to design a linear time PTAS for the TSP problem in planar graphs. His algorithm then becomes a general framework for designing a PTAS for many connectivity problems in planar graphs such as subset TSP [61], Steiner tree [21], survivable network design [19], Steiner forest [13] and in H-minor-free graphs [24, 69]. Let $\mathscr{P}$ be the connectivity minimization problem. The framework consists of four steps.

Spanner step: Find a *light spanner* for $\mathscr{P}$. (The definition of light spanners will be given below.)

Contraction decomposition step: Partition edges of the spanner into k sets $S_1, S_2, \ldots, S_k$ such that contracting any set would result in a graphs of treewidth at most $O_H(k)$

where k is a constant depending on ϵ. Let $S_{\min}$ be the set of edges of minimum weight.

Dynamic programming step: Solve $\mathscr{P}$ optimally on graph $G/S_{\min}$ (of treewidth at most $O_H(k)$) by dynamic programming.

Lifting step: Lift the the solution found in the dynamic programming step to a solution in the original graph G. Some edges in $S_{\min}$ may be added to the solution of G in this step.

In most cases, the lifting step is straightforward. We now describes other steps in more details.

Light spanners: A light spanner for $\mathscr{P}$ is a subgraph K of G that can be found in polynomial time and satisfies two properties:

Near-optimality property: K must contain a feasible solution for $\mathscr{P}$ of length at most $(1+\epsilon)$OPT where OPT is the weight of the optimal solution for $\mathscr{P}$ in G. *Shortness property:* the weight of H must be bounded by $f(\epsilon)$ where $f(\epsilon)$ is a constant that depends on ϵ only.

Contraction decomposition: A contraction decomposition, given any integer k, partitions the edge set of G into k subsets such that contracting any subset would result in a graph of treewidth at most $O_H(k)$. A contraction decomposition algorithm for planar graphs was found by Klein [60]. Klein's result was generalized to bounded genus graphs by Demaine, Hajiaghayi and Mohar [39] and to H-minor-free graphs by Demaine, Hajiaghayi and Kawarabayashi [37].

Dynamic programming: Dynamic programming is a standard tool for solving many optimization problems in bounded treewdith graphs optimally [15]. Many connectivity problems, such as TSP, subset TSP, Steiner tree can be solved optimally in treewidth-tw graphs in time $2^{O(\mathrm{tw}\log \mathrm{tw})}n^{O(1)}$ times using standard dynamic programming. However, in some case, we need to design an algorithm with running time at most $2^{O(\mathrm{tw})}n^{O(1)}$. Fortunately, there are several advanced techniques for general graphs [34, 16, 48] and H-minor-free graphs [41] that helps us achieve this goal. We will use one of these techniques in designing PTAS for the susbet TSP problem in Section 2.3.

```
GreedySpanner(G(V, E), t)
    S ← (V, ∅).
    Sort edges of E in non-decreasing order of weights.
    For each edge xy ∈ E in sorted order
        if t · w(xy) < d_S(x, y)
            E(S) ← E(S) ∪ {e}
    return S
```

Figure 2.1: The greedy spanner algorithm.

2.2 The Traveling Salesperson Problem

In this section, we present an EPTAS for the TSP problem in H-minor-free graphs. Among four steps in the PTAS framework, we already describe three last steps in Section 2.1. Thus, we only discuss the spanner step in this section.

Let MST be the minimum spanning tree of G. We use $d_G(x, y)$ to denote the shortest path distance between two vertices x and y. We drop the subscript when the graph is clear from the context. A subgraph S of G is a $(1+\epsilon)$-*spanner*, or simply called *spanner*, if $d_S(x, y) \leq (1 + \epsilon)d_G(x, y)$. The *lightness* of S is the ratio $\frac{w(S)}{w(\mathrm{MST})}$. S is *light* if $w(S) \leq f(\epsilon)w(\mathrm{MST})$ where $f(\epsilon)$ is a function depending on ϵ only.

Peleg and Schäffer [79] introduced t-spanners of graphs as a way to sparsify the graphs while approximately preserving the pairwise distances between vertices. A t-spanner can be found by the greedy algorithm in Figure 2.1

We refer to t-spanners obtained by the greedy algorithm when $t = 1 + \epsilon$ *greedy spanners*. Light spanners are obtained by Althöfer, Das, Dobkin, Joseph and Soares [7] for planar graphs and by Grigni [51] for bounded-genus graphs. But light spanners for H-minor-free graphs are unknown. In 2002, Grigni and Sissokho [53] showed that the greedy algorithm gives a spanner of lightness $O(\frac{\sigma_H \log n}{\epsilon})$[1] and left the problem of removing the $\log n$ factor in the lightness as an open problem. (Recall that $\sigma_H = |V(H)|\sqrt{\log |V(H)|}$.) Ten years later, Grigni and Hung [52] show that it is possible to remove $\log n$ factor if the input graph has bounded *pathwidth*, a special case of H-minor-free graphs (see Figure 1.1), and conjectured that it is possible to do so for H-minor-free graphs.

Since the analysis of greedy spanners for planar graphs and bounded-genus graphs heavily uses the non-crossing embedding of graphs, it is natural to expect that a similar

[1]Grigni and Sissokho's spanner, though is not light, was used with the contraction decomposition framework of Demaine, Hajiaghayi and Kawarabayashi to give a (non-efficient) PTAS for TSP.

analysis works for H-minor-free graphs. The main difficulty is that H-minor-free graphs don't have such non-crossing embedding. Robertson and Seymour [82] decomposition theorem only provides a weak embedding: one can decompose an H-minor-free graph into a collection of *nearly embeddable* subgraphs glued together in a tree-like way.

In a joint work Glencora Borradaile [23], we use Robertson and Seymour's decomposition theorem to show that if bounded-treewidth graphs have light spanners, then H-minor-free graphs also have light spanners(see the relationship between two classes of graphs in Figure 1.1). In the algorithmic world, bounded-treewidth graphs are much easier to deal with than H-minor-free graphs since many NP-hard problems can be solved in polynomial time [15] in bounded treewidth graphs. Thus, our result conceptually reduces the difficulty of Grigni and Hung Conjecture significantly. However, the conjecture remains open for bounded treewidth graphs. In this section, we present a proof of Grigni and Hung Conjecture.

Theorem 2.2.1. *Any H-minor-free graph G has an $(1+\epsilon)$-spanner of lightness $O\left(\frac{\sigma_H}{\epsilon^3}\log\frac{1}{\epsilon}\right)$.*

Using the PTAS framework in Section 2.1, we obtain the following theorem from Theorem 2.2.1:

Theorem 2.2.2. *The TSP problem in H-minor-free graphs admits a PTAS with running time $2^{O(\mathrm{poly}(\frac{1}{\epsilon}))}n^O(1)$.*

Theorem 2.2.2 generalizes Klein's EPTAS for TSP in planar graphs [60] and Borradaile, Demaine and Tazari's EPTAS for TSP in bounded genus graphs [18]. It improves the previous result by Demaine, Hajiaghayi and Kawarabayashi [38] who used Grigni and Sissokho's spanner to give a PTAS for TSP in H-minor-free graphs with running time $n^{O(\mathrm{poly}(\frac{1}{\epsilon}))}$.

2.2.1 Overview of the proof of Theorem 2.2.1

We will show that greedy spanner has lightness $O\left(\frac{\sigma_H}{\epsilon^3}\log\frac{1}{\epsilon}\right)$. Our approach is based on the iterative clustering approach and does not require Robertson and Seymour decomposition. Iterative clustering was first used by Awerbuch, Luby, Goldberg and Plotkin [9] to decompose a graph into subgraphs of low-diameter and low-chromatic number and then, by Elkin and Peleg [45] to find *hybrid spanners* of graphs. Recently, Chechick and

Wulff-Nilsen [30] used iterative clustering to construct light $(1+2k)(1+\epsilon)$-spanners for general graphs. We tailor the construction of Chechick and Wulff-Nilsen to our setting.

We first partition spanner edges into $\log\frac{1}{\epsilon}$ sets[2], and then the total weight of each set is bounded separately; this induces the $\log\frac{1}{\epsilon}$ factor in the lightness. Each set consists of spanner edges in an exponential scale of many *levels*.

First, a non-negative credit $c(\epsilon)$ is assigned to each MST edge of unit weight; $c(\epsilon)\log\frac{1}{\epsilon}$ is also the lightness of the spanner. In each level, clusters are constructed iteratively from clusters of the previous level; level-1 clusters are constructed directly from the MST. An invariant is maintained that each cluster must have some amount of credit to pay for spanner edges in their level. Credits of level-1 clusters are taken directly from MST edges. Credits of level-i clusters are taken from credits of clusters of level $i-1$ and MST edges connecting those lower-level clusters. However, to pay for the spanner edges, level-i clusters cannot take all credits from level-$(i-1)$ clusters. Instead, it is guaranteed that on average, each level-$(i-1)$ cluster has a non-trivial amount of credit left to pay for level-i spanner edges. Since the input graph is H-minor-free, each level $i-1$ cluster on average must pay for only a constant number of level-i spanner edges.

2.2.2 Assigning credits to MST edges

Let $w_0 = \frac{w(\mathrm{MST})}{n-1}$ be the average weight of an MST edge. We first bound the total weight of edges that have weight at most w_0.

Claim 2.2.3. *Let L_S be the set of edges of S of weight at most w_0. Then, $w(L_S) \leq 2\sigma_H w(\mathrm{MST})$.*

Proof. $w(L_S) \leq w_0|L_S| \leq w_0\sigma_H \cdot n = \frac{w(\mathrm{MST})}{n-1}\sigma_H \cdot n \leq 2\sigma_H w(\mathrm{MST})$. □

We now focus on bounding the total weight of edges of weight at least w_0 in S. We subdivide and allocate credits to MST edges such that every MST edge has weight at most w_0 and at least $c(\epsilon)w_0$ credits where $c(\epsilon)$ is a constant that only depends on ϵ and will be specified later. We will guarantee that the total allocated credit is $O(c(\epsilon))w(\mathrm{MST})$ where $O(c(\epsilon))$ is also the lightness of the spanner. First, we subdivide every MST edge e of weight more than w_0 into $\lceil\frac{w(e)}{w_0}\rceil$ new edges with equal weights summing up to $w(e)$;

[2] Here, log denotes the base 2 logarithm.

note that each new edge has weight at most w_0. Letting S' be the new graph, we have $w(\mathrm{MST}(S')) = w(\mathrm{MST})$. We then allocate $c(\epsilon)w_0$ credits to each MST edge of S'.

Claim 2.2.4. *The total credit allocated to the* MST *edges of S' is at most* $2c(\epsilon)w(\mathrm{MST})$.

Proof. The total credits assigned to MST edges of S' is:

$$\begin{aligned} c(\epsilon)w_0|E(\mathrm{MST}(S'))| &\leq c(\epsilon)w_0 \sum_{e\in\mathrm{MST}} \lceil \frac{w(e)}{w_0} \rceil \leq c(\epsilon)w_0 \left(\sum_{e\in\mathrm{MST}} (\frac{w(e)}{w_0} + 1) \right) \\ &= c(\epsilon)w(\mathrm{MST}) + c(\epsilon)w_0(n-1) = 2c(\epsilon)w(\mathrm{MST}) \end{aligned} \tag{2.1}$$

□

2.2.3 Iterative Clustering

Let $J_0 = \{e \in S', w_0 < w(e) \leq \frac{2w_0}{\epsilon}\}$. We first bound the weight of J_0 and pay for edges in J_0 separately. The purpose is to simplify the base case in the inductive amortized argument that we present below.

Claim 2.2.5. $w(J_0) \leq \frac{4\sigma_H}{\epsilon} w(\mathrm{MST})$.

Proof.

$$w(J_0) = \sum_{e\in J_0, w(e)>w_0} w(e) \leq \sigma_H \cdot n\frac{2w_0}{\epsilon} \leq \frac{4\sigma_H}{\epsilon} w(\mathrm{MST})$$ □

Let $I_\epsilon = \lceil \log \frac{1}{\epsilon} \rceil$ and $I_n = \lceil \log n \rceil$. Note that the longest distance between any two vertices in S' is at most $n \cdot w_0$. We partition the spanner edges (of weight at least w_0) of S' into $I_n \cdot I_\epsilon$ sets $\{\Pi_i^j, 0 \leq i \leq I_n - 1, 0 \leq j \leq I_\epsilon - 1\}$ where each edge $e \in \Pi_i^j$ has weight in the range $(\frac{2^j}{\epsilon^i} w_0, \frac{2^{j+1}}{\epsilon^i} w_0]$. For each $0 \leq j \leq I_\epsilon - 1$, let

$$S_j = \bigcup_{i=0}^{I_n - 1} \Pi_i^j \tag{2.2}$$

Lemma 2.2.6. *For each $0 \leq j \leq I_\epsilon - 1$, there is a set of spanner edges B such that $w(B) = O(\frac{1}{\epsilon^2} w(\mathrm{MST}))$ and $w(S_j \setminus B) \leq \frac{\sigma_H}{\epsilon^3} w(\mathrm{MST})$.*

It is not hard to see that Lemma 2.2.6 directly imply Theorem 2.2.1. Thus, we only focus on proving Lemma 2.2.6 for a fixed j. We refer to edges of Π_i^j as edges in *level i* (Equation 2.2). Let $\ell_i = \frac{2^{j+1}}{\epsilon^i} w_0$. We construct a set of clusters, which are subgraphs of S', for each level and guarantee inductively two diameter-credits invariants:

DC1 A cluster of level i of diameter k has at least $c(\epsilon) \cdot \max\{k, \frac{\ell_i}{2}\}$ credits.

DC2 A cluster of level i has diameter at most $g\ell_i$ for some constant $g > 2$ (specified later).

A cluster of level i, say C_i, is the union of a subset of clusters in level $i-1$ connected by MST and level-i spanner edges. Clusters of level $i-1$ are referred to as *ϵ-clusters*. To satisfy DC1, we assign the credits from ϵ-clusters in C_i and the MST edges connecting the ϵ-clusters to C_i. However, we need to group ϵ-clusters in such a way that there are some extra ϵ-clusters whose credits are not needed to maintain DC1 for C_i. We will use credits of these extra ϵ-clusters to pay for level-i spanner edges incident to every ϵ-cluster in C_i. The credit lower bound $c\ell_i/2$ (DC1) helps us achieve the goal.

To guarantee the diameter-credit invariants for level 0, we greedily break the MST into components (level-0 clusters) of diameter at least ℓ_0 and at most $4\ell_0$. Recall $\ell_0 = 2^{j+1}w_0 \leq \frac{2w_0}{\epsilon}$. To guarantee DC1, we use the credits of MST edges in the longest path of each cluster. Since the credit of each MST edge is at least its length, DC1 is satisfied. Invariant DC2 follows directly from the construction. Note that we have already accounted for the weight of spanner edges of E_0 in Claim 2.2.5.

2.2.3.1 Constructing higher level clusters

We construct clusters of level i from the ϵ-clusters of level $i-1$. We assume that the stretch of the spanner is $1 + s\epsilon$ for some constant s (independent of ϵ) that we will pick sufficiently big to make our claims below hold. Furthermore, we assume that ϵ is bounded from above by a sufficiently small positive constant. We call vertices of $V(S') \setminus V(S)$ *virtual vertices*. We call a cluster *virtual* of it only contains virtual vertices and *non-virtual* otherwise. Let $\mathcal{K}(\mathcal{C}_\epsilon, E_i)$ be the multigraph obtained by taking the subgraph of G consisting of ϵ-clusters and spanner edges in E_i and contracting each ϵ-cluster into a single vertex. Let $\ell = \ell_i$.

Lemma 2.2.7. *$\mathcal{K}(\mathcal{C}_\epsilon, E_i)$ is a simple graph.*

Proof. Recall ϵ-clusters have diameter at most $g\epsilon\ell$ by the diameter-credit invariants. Recall edges in E_i have weight in range $(\ell/2, \ell]$. Thus, when ϵ is sufficiently small, $\mathcal{K}$ has no self-loops. To show that $\mathcal{K}$ has no parallel edges, we assume there are such two xy and uv where $w(xy) < w(uv)$. Let C_u, C_v be two ϵ-clusters that contain u and v, respectively. We further assume, w.l.o.g, that $x \in C_u, y \in C_v$. Then the u-to-v path P_{uv} from u to x inside C_u, edge xy and then y to v inside C_v has length at most $w(xy)+2g\epsilon\ell$, which is at most $(1+4g\epsilon)w(uv)$ since $w(uv) \geq \ell/2$. Thus, by choosing $s \geq 4g$, edge uv is not added to the spanner by the greedy algorithm. Thus, $\mathcal{K}$ is simple. □

We say an ϵ-cluster *high-degree* if it is incident to at least $\Delta_{g,\epsilon} = \frac{3g}{\epsilon}$ ϵ-clusters in $\mathcal{K}(\mathcal{C}_\epsilon, E_i)$. We construct clusters in five phases:

Phase 1: High-degree ϵ-clusters This phase has three steps. The main purpose is to guarantee that every high-degree ϵ-cluster and its neighbors are grouped into clusters. Initially, every ϵ-cluster of $\mathcal{C}_\epsilon$ is unmarked.

(Step 1) Let $X \in \mathcal{C}_\epsilon$ be a high-degree ϵ-cluster such that all of its neighbor ϵ-clusters in $\mathcal{K}(\mathcal{C}_\epsilon, E_i)$ are unmarked. We form a new level-i cluster from X, its neighbors and its incident level-i spanner edges. We then mark X, its neighbors and repeat this step.

(Step 2) For each unmarked high-degree ϵ-cluster Y, there must be another ϵ-cluster, say Z, that is marked in Step 1. Let C be the level-i cluster formed in Step 1 that contains Z. We augment C by Y, its neighbors and its corresponding level-i spanner edges. We then mark Y and its neighbors and repeat this step.

(Step 3) Let Y be an unmarked low-degree ϵ-cluster that has a (marked) high-degree neighbor, say Z. By construction in Step 2, Z must be marked in Step 1. Let C be the level-i cluster that contains Z. We augment C by Y and its incident level-i spanner edge between Y and Z.

See Figure 2.2 for an illustration of a level-i cluster constructed in Phase 1.

Phase 2: Low-degree, branching ϵ-clusters Let $\mathcal{F}$ be a maximal forest whose nodes are the ϵ-clusters that remain unmarked after Phase 1 and whose edges are MST edges between pairs of such ϵ-clusters. Let $\mathcal{T}$ be a subtree of $\mathcal{F}$. We say an ϵ-cluster X *$\mathcal{T}$-branching* if it has degree at least 3 in $\mathcal{T}$.

Let $\mathcal{P}$ be a path of ϵ-clusters connected by MST edges. We define the diameter of $\mathcal{P}$, denoted by $\mathrm{diam}(\mathcal{P})$, to be the diameter of the subgraph of S' formed by edges inside ϵ-clusters and MST edges connecting ϵ-clusters of $\mathcal{P}$. We define *effective diameter* of $\mathcal{P}$, denoted by $\mathrm{ediam}(\mathcal{P})$, to be the diameters of ϵ-clusters in $\mathcal{P}$. We define the effective diameter of a subtree $\mathcal{T}$ of $\mathcal{F}$ to be the effective diameter of the diameter path of $\mathcal{T}$.

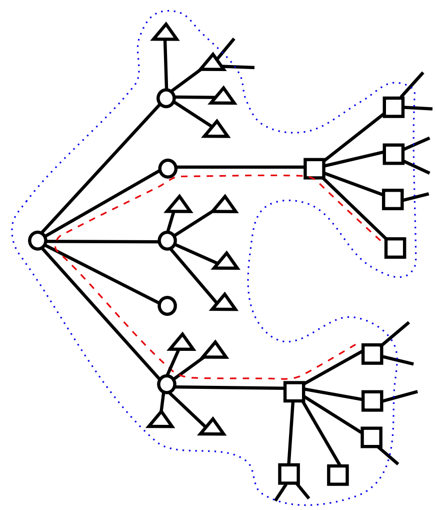

Figure 2.2: A cluster C formed in Phase 1 is enclosed in the dotted blue curve. Round vertices, square vertices and triangular vertices are grouped in Step 1, Step 2 and Step 3, respectively. The diameter path $\mathcal{D}$ is highlighted by the dashed red curve.

This phase has two steps. The purpose is to group every $\mathcal{F}$-branching vertices of high-diameter trees into clusters. By construction in Phase 1, vertices in this phase are low-degree.

(Step 1) Let $\mathcal{T}$ be a minimal subtree of $\mathcal{F}$ of effective diameter at least 2ℓ and at most 4ℓ that has a $\mathcal{T}$-branching ϵ-cluster, say X. We form a new level-i cluster from ϵ-clusters and MST edges of $\mathcal{T}$. We repeat this step until it no longer applies. After Step 1, every component of $\mathcal{F}$ is tree of effective diameter most 2ℓ or is a path of effective diameter at least 4ℓ.

(Step 2) We call a path of $\mathcal{F}$ *high-diameter* if it has effective diameter at least 4ℓ. Let Y be an ϵ-cluster in a high-diameter path which is $\mathcal{F}$-branching before Step 1. That is, all but at most two neighbors of Y are removed from $\mathcal{F}$ in Step 1. Let Z be a removed neighbor of Y (in $\mathcal{F}$) and e be the MST edge between Y and Z. Let C be the level-i cluster in Step 1 that contains Z. We augment C with Y and e. We then remove Y from $\mathcal{F}$ and repeat. (See Figure 2.3.)

Phase 3: ϵ-clusters in high diameter paths. Let e be a spanner edge in E_i whose endpoints, x and y are in high-diameter cluster paths, $\mathcal{P}$ and $\mathcal{Q}$, respectively, where it may be that $\mathcal{P} = \mathcal{Q}$. Let X and Y be the ϵ-clusters containing x and y, respectively.

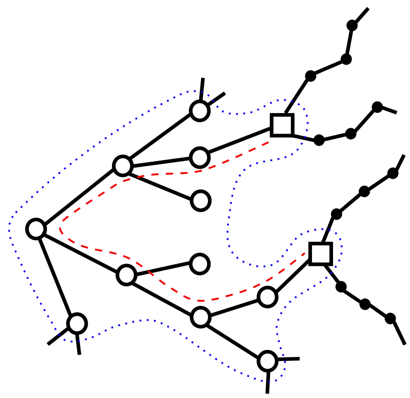

Figure 2.3: A cluster C formed in Phase 2 is enclosed in the dotted blue curve. Round vertices and square vertices are grouped in Step 1 and Step 2, respectively. The diameter path $\mathcal{D}$ is highlighted by the dashed red curve.

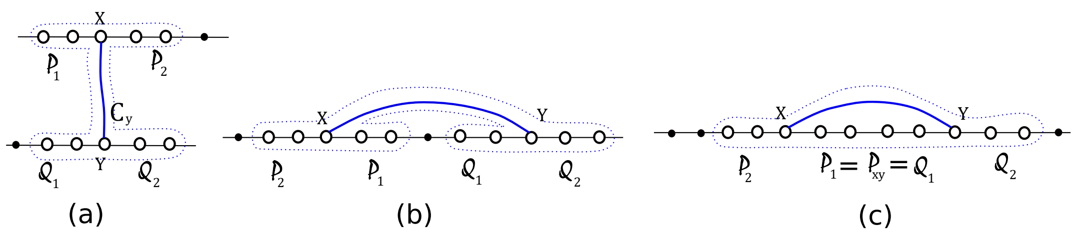

Figure 2.4: Three different forms that a cluster (enclosed in the dotted red curves) in Phase 2 can take. The solid blue line is the spanner edge e. (a) e connects ϵ-clusters in different clusters paths, (b) e connects ϵ-cluster in the same path and $\mathcal{P}_1$ and $\mathcal{Q}_1$ are disjoint and (c) e connects ϵ-cluster in the same path and $\mathcal{P}_1$ and $\mathcal{Q}_1$ are overlapped. In case (c), we redefine $\mathcal{P}_1 = \mathcal{Q}_1 = \mathcal{P}_{xy}$.

We only proceed with this phase if the two affix cluster subpaths of $\mathcal{P}$ ending at X have effective diameter at least 2ℓ (likewise for $\mathcal{Q}$). Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the two minimal subpaths of $\mathcal{P}$ ending at X that have effective diameter at least 2ℓ. Likewise define $\mathcal{Q}_1$ and $\mathcal{Q}_2$. We group ϵ-clusters and MST edges of $\mathcal{P}_1 \cup \mathcal{Q}_1 \cup \mathcal{P}_2 \cup \mathcal{Q}_2$ and e as a new level-i cluster. See Figure 2.4 for an illustration of the different forms this cluster can take.

Phase 4: Low diameter components. Let $\mathcal{F}$ be the set of trees (and paths) remaining of effective diameter at most 4ℓ. By construction, each component $\mathcal{T}'$ of $\mathcal{F}$ has a MST edge, say e, to a level-i cluster constructed in previous phases, say C. We augment C by $\mathcal{T}'$ and e.

Phase 5: Remaining high diameter paths. Let $\mathcal{P}$ be a cluster path of effective diameter at least 4ℓ. We greedily break $\mathcal{P}$ into subpaths of effective diameter at least 2ℓ and at most 4ℓ. If any affix of $\mathcal{P}$, say $\mathcal{P}'$, has a MST edge, say e, to a level-i cluster constructed in previous phases, say C, we augment C with $\mathcal{P}'$ and e. We then make each remaining cluster subpath of $\mathcal{P}$ into a new level-i cluster.

This completes the cluster construction for level i.

2.2.3.2 Showing diameter-credit invariant DC2

Observation 2.2.8. *For any cluster path $\mathcal{P}$, $\mathrm{diam}(\mathcal{P}) \le 2\mathrm{ediam}(\mathcal{P})$.*

Proof. The observation follows from the fact that ϵ-clusters have diameter at least w_0 (by construction of the base case) which is at least the weight of edges connecting them in $\mathcal{P}$. □

By construction, each level-i cluster constructed in Phase 4 is a cluster path of effective diameter at most 2ℓ. By Observation 2.2.8, we have:

Claim 2.2.9. *Level-i clusters constructed in Phase 5 have diameter at most 4ℓ.*

Claim 2.2.10. *Level-i clusters have diameter at most 53ℓ when ϵ is smaller than $\frac{1}{g}$.*

Proof. Let C be a level-i cluster that is initially formed in Phase 1,2 or 3. By construction, C may be augmented in Phases 4 and 5. Let C' and C'' be the augmented clusters of C after Phase 4 and Phase 5, respectively. It could be that $C = C' = C''$. C' is obtained from C by attaching trees of effective diameter at most 4ℓ via MST edges. C'' is obtained from C' by attaching paths of effective diameter at most 2ℓ via MST edges. Recall each MST edge has length at most w_0. By Observation 2.2.8, we have:

$$\mathrm{diam}(C') \le \mathrm{diam}(C) + 16\ell + 2w_0 \qquad \text{and} \qquad \mathrm{diam}(C'') \le \mathrm{diam}(C') + 8\ell + 2w_0 \quad (2.3)$$

By invariant (DC2) for level $i-1$, ϵ-clusters have diameter at most $g\epsilon\ell$. Thus, if C is constructed in Phase 1, $\mathrm{diam}(C) \le 6\ell + 7\epsilon g\ell \le 13\ell$ when ϵ is smaller than $\frac{1}{g}$ (see Figure 2.2).

If C is constructed in Phase 2, by Observation 2.2.8, after Step 1, $\mathrm{diam}(C) \le 8\ell$. Since in Step 2, C is augmented by ϵ-clusters via MST edges, after Step 2, $\mathrm{diam}(C) \le 8\ell + 2w_0 + 2g\epsilon\ell \le 12\ell$ ($\ell \ge w_0$ by construction of the base case).

If C is constructed in Phase 3, we have:

$$\mathrm{diam}(C) \leq \mathrm{diam}(\mathcal{P}_1) + \mathrm{diam}(\mathcal{P}_2) + \mathrm{diam}(\mathcal{Q}_1) + \mathrm{diam}(\mathcal{Q}_2) + \ell(e)$$

Since $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2$ are minimal, each has effective diameter at most $2\ell + g\epsilon\ell$. Thus, $\mathrm{diam}(C) \leq 4(4\ell + 2g\epsilon\ell) + \ell = 17\ell + 8g\epsilon\ell \leq 25\ell$.

Thus, in any case, $\mathrm{diam}(C) \leq 25\ell$. By Equation (2.3), $\mathrm{diam}(C'') \leq 49\ell + 4w_0 \leq 53\ell$. □

Thus, by Claim 2.2.10, we can choose $g = 53$.

2.2.3.3 Showing diameter-credit invariant DC1

We define $cr(\mathcal{X})$ to be the total credits of a set of ϵ-clusters $\mathcal{X}$. Recall all high-degree ϵ-clusters are grouped in Phase 1. Thus, ϵ-clusters involved in later phases have at most $\Delta_{g,\epsilon}$ ($\Delta_{g,\epsilon} = \frac{3g}{\epsilon}$) incident level-$i$ spanner edges.

Clusters originating in Phase 5: Let C be a level-i cluster formed in Phase 5. We call C a *long cluster* if it has at least $\frac{2g}{\epsilon} + 1$ ϵ-clusters and a *short cluster* otherwise. We have:

Claim 2.2.11. *A long cluster can both maintain invariant DC1 and pay for its incident spanner edges when* $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$.

Proof. Let $\mathcal{X}$ be a set of any $\frac{2g}{\epsilon}$ ϵ-clusters of C. By invariant DC1 for level $i-1$, we have:

$$cr(\mathcal{X}) \geq \frac{2g}{\epsilon} c(\epsilon)\ell/2 = c(\epsilon)g\ell$$

which is at least $c(\epsilon) \cdot \max(\mathrm{diam}(C), \ell/2)$ since $\mathrm{diam}(C) \leq g\ell$ as shown in Claim 2.2.9 (since $g = 53$). Thus, credits of $\mathcal{X}$ are enough to maintain DC1 for C.

Since C is a long cluster, there is at least one ϵ-cluster, say Y, not in $\mathcal{X}$. By DC1 for level $i-1$, Y has at least $c(\epsilon)\ell/2$ credits. Since there are at most:

$$\Delta_{g,\epsilon} \cdot \left(\frac{2g}{\epsilon} + 1\right) = O(\frac{g^2}{\epsilon^2})$$

level-i spanner edges incident to ϵ-clusters in $\mathcal{X} \cup \{Y\}$, Y's credits are enough to pay for those spanner edges when $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$.

For each ϵ-cluster $z \in C \setminus (\mathcal{X} \cup \{Y\})$, we use z's credit to pay for the spanner edges incident to z. Since z is incident to at most $\Delta_{g,\epsilon}$ level-i spanner edges, this amount of credit is sufficient when $c(\epsilon) = \Omega(\frac{\Delta_{g,\epsilon}}{\epsilon}) = \Omega(\frac{g}{\epsilon^2})$ by invariant DC1 for level $i-1$,. $\square$

Claim 2.2.12. *The credits of ϵ-clusters and* MST *edges connecting ϵ-clusters of each short cluster C are enough to maintain invariant DC1 for C.*

Proof. We abuse notation by letting MST(C) be the set of MST edges in C that connects its ϵ-clusters. Since C is a cluster path, we have:

$$\mathrm{diam}(C) \leq \sum_{X_\epsilon \in C} \mathrm{diam}(X_\epsilon) + \sum_{e \in \mathrm{MST}(C)} w(e)$$

By invariant DC1 for level $i-1$, $cr(X_\epsilon) \geq c(\epsilon) \cdot \mathrm{diam}(X_\epsilon)$ and since each MST edge has credit at least $c(\epsilon)$ times its length, the claim follows. $\square$

A short cluster may need to use all the credits of ϵ-clusters and MST edges to maintain DC1, hence, it many not have extra credit to pay for any incident level-i spanner edges. In this case, we need to use credits of other level-i clusters to pay for those spanner edges. We call a short cluster *internal* if it is not an affix of a long path $\mathcal{P}$ in Phase 5.

Observation 2.2.13. *There is no level-i spanner edge e that has both endpoints in internally short clusters.*

Proof. If there is such an edge e, it would be grouped into a level-i cluster in Phase 3. $\square$

Thus, a level-i spanner edge incident to an internally short cluster can be paid by the level-i cluster that contains the other endpoint of e. However, if a short cluster is not internal, we must find a way to pay for its incident spanner edges. Recall after Phase 2, every cluster path of effective diameter at least 4ℓ must have an MST edge from one of its endpoint ϵ-clusters to a level-i cluster. By construction in Phase 5, if a short cluster is an affix of $\mathcal{P}$, called a *short affix cluster*, the other affix of $\mathcal{P}$, called the *sibling affix*, must have an MST edge to a cluster originating in the first three phases and thus augments it. (The only exception is when there is no level-i clusters after the first three phases and we will handle this case at the end of this paper.) Thus, we can use the credit of ϵ-clusters of the sibling affices to pay for incident spanner edges of affix short clusters. To that end, we analyze clusters originally constructed in the first three phases.

Clusters originating in Phase 1: Consider the induced subgraph $\mathcal{H}$ of $\mathcal{K}(\mathcal{C}_\epsilon, E_i)$ on the set of ϵ-clusters involved in Phase 1. Since G is H-minor-free, then $\mathcal{H}$ is also H-minor-free. Thus $|E(\mathcal{H})| \leq O(|V(H)|\sqrt{\log|V(H)|})|V(\mathcal{H})|$.

Let C be a level-i cluster constructed in Phase 1 and C', C'' be the augmentation of C in Phase 4 and 5, respectively. By construction, C has at least $\frac{3g}{\epsilon}$ ϵ-clusters. Let $\mathcal{Z}, \mathcal{Z}'$ be two disjoint subsets of ϵ-custers of C such that $|\mathcal{Z}| = \frac{2g}{\epsilon}$ and $|\mathcal{Z}'| = \frac{g}{\epsilon}$. By invariant (DC1) for level $i-1$, we have:

$$cr(\mathcal{Z}) \geq \frac{2g}{\epsilon}\frac{c(\epsilon)\epsilon\ell}{2} = c(\epsilon)g\ell$$

which is at least $c(\epsilon) \cdot \max\{\text{diam}(C''), \ell/2\}$. Thus, credits of ϵ-clusters in $\mathcal{Z}$ are sufficient to maintain invariant (DC1). We then redistribute credits of ϵ-clusters in $\mathcal{Z}'$ to ϵ-clusters in $\mathcal{Z} \cup \mathcal{Z}'$. On average, each has at least :

$$(\frac{g}{\epsilon}\frac{c(\epsilon)\epsilon\ell}{2})/(\frac{3g}{\epsilon}) = c(\epsilon)\epsilon\ell/6$$

credits. Note that other ϵ-clusters in $C''' \setminus (\mathcal{Z}_1 \cup \mathcal{Z}_2)$ have at least $c(\epsilon)\epsilon\ell/2$ credits each.

Let $\mathcal{R}$ be a short affix cluster in Phase 5, that has the sibling $\mathcal{S}$ in C''. Let X be an ϵ-cluster in $\mathcal{S}$. Note that X is low-degree. We use credits of X to pay for level-i spanner edges incident to ϵ-clusters in $\mathcal{R} \cup \{X\}$. Since $\mathcal{R}$ is short, there are at most:

$$\Delta_{g,\epsilon}(\frac{2g}{\epsilon} + 1) = \Omega(\frac{g^2}{\epsilon^2})$$

Thus, credits of X (at least $c(\epsilon)\epsilon\ell/2$) is sufficient when $c(\epsilon) = \frac{g^2}{\epsilon^3}$.

For remaining ϵ-clusters, say Y, of C'', if Y is in $C'' \setminus C$, we use its credit to pay for its incident edges. Y's credit is sufficient when $c(\epsilon) = \Omega(\frac{g}{\epsilon^2})$. If Y is in C, we use its credits to pay for spanner edges in $\mathcal{H}$. Thus Y's credit is sufficient when $c(\epsilon) = \Omega(\frac{\sigma_H}{\epsilon})$ since on average, Y is incident to at most $\sigma_H$level-i spanner edges. We have:

Claim 2.2.14. *Clusters originating in Phase 1 can maintain invariant (DC1) and pay for incident level-i spanner edges if* $c(\epsilon) = \Omega(\max(\frac{\sigma_H}{\epsilon}, \frac{g^2}{\epsilon^3}))$.

Clusters originating in Phase 2+3 Let C be a level-i cluster constructed in Phase 2 or 3. Let C' and C'' be the augmentations of C in Phase 4 and 5, respectively. Let D be the diameter path of the spanner given by edges and vertices in C''. Let $\mathcal{D}$ be the walk obtained from D by contracting each maximal subpath of D that is inside an ϵ-cluster of C'' to a single vertex.

Definition 2.2.15 (Canonical pair)**.** *Let $\mathcal{S} \subseteq C \cup \mathcal{D}$ be a subset of ϵ-clusters of C'' such that $|\mathcal{S}| \leq \frac{2g}{\epsilon}$ and the credits of ϵ-clusters in $\mathcal{S}$ and* MST *edges of C'' are sufficient to maintain invariant DC1 for C''. Let Y be an ϵ-cluster of C that is not in $\mathcal{S}$. We call $(\mathcal{S}, Y)$ a* canonical pair *of C''.*

Note that we do not claim the existence of canonical pairs. Indeed, the main goal of this subsection is to prove that a canonical pair exists for C'' since its existence implies that $C'' \setminus \mathcal{S} \neq \emptyset$. Thus, we can use credits of ϵ-clusters in $C'' \setminus \mathcal{S}$ to pay for level-i spanner edges incident to ϵ-clusters of C'' and ϵ-clusters of short affix clusters hat have sibling affices in C''.

Claim 2.2.16. *If C'' has a canonical pair $(\mathcal{S}, Y)$, we then can pay for every level-i spanner edge that is incident to ϵ-clusters of C'' and ϵ-clusters of short affix clusters that have sibling affices in C'' using credits of ϵ-clusters in $C'' \setminus \mathcal{S}$ when $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$.*

Proof. Let $\mathcal{R}$ be a set of ϵ-clusters that contains every ϵ-cluster in $\mathcal{S} \cup \{Y\}$ and affix short clusters in Phase 5 whose sibling affices contain an ϵ-cluster of $\mathcal{D}$. Recall that C' is augmented by attaching paths of ϵ-clusters via MST edges. Thus, $\mathcal{R}$ contains at most two short clusters as a result of Phase 5 (see Figure 2.5).

Since $|\mathcal{S}| \leq \frac{2g}{\epsilon}$ and each short cluster has at most $\frac{2g}{\epsilon}$ ϵ-clusters, $|\mathcal{R}| = O(\frac{g}{\epsilon})$. Since each ϵ-cluster is incident to at most $\Delta_{g,\epsilon}$ level-i spanner edges, ϵ-clusters in $\mathcal{R}$ are incident to at most $O(\frac{g^2}{\epsilon^2})$ level-i spanner edges. Recall that each level-i spanner edge has length at most ℓ. By invariant DC1 for level $i-1$, Y has at least $c(\epsilon)\epsilon\ell/2$ credits. Thus, by choosing $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$, Y's credit is sufficient to pay for every spanner edge incident to ϵ-clusters in $\mathcal{R}$.

For each ϵ-cluster z in $C'' \setminus \mathcal{R}$, the credit of z is sufficient to pay for incident level-i spanner edges incident to z. However, we also need to pay for short affix clusters in Phase 5, whose siblings augment C' in Phase 5. To afford this, we use half the credit of each ϵ-cluster in $C \setminus \mathcal{R}$ (of value at least $c(\epsilon)\epsilon\ell/4$ by invariant DC1 for level $i-1$) to pay

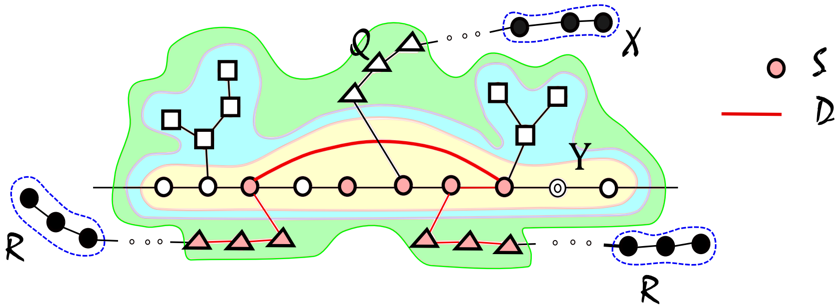

Figure 2.5: Clusters C, C' and C'' are enclosed by yellow-shaded, cyan-shaded and green-shaded regions, respectively. Round vertices, square vertices and triangular vertices are in C, C', C'' respectively. The red path is the cluster walk $\mathcal{D}$. Shaded ϵ-clusters are in $\mathcal{S}$ and Y marked with double-circle ϵ-cluster. Short affix clusters in Phase 5 are enclosed by dotted blue curves. $\mathcal{R}$ contains $\mathcal{S} \cup \{Y\}$ and two (annotated) short affix clusters that have two corresponding sibling affices in $\mathcal{D}$.

for level-i spanner edges incident to it. Since each ϵ-cluster is incident to at most $\Delta_{g,\epsilon}$ level-i spanner edges, this credit is sufficient when $c(\epsilon) \geq \frac{4\Delta_{g,\epsilon}}{\epsilon} = \Omega(\frac{g}{\epsilon^2})$.

Since C' is augmented by attaching cluster paths via MST edges, an affix cluster not in $\mathcal{R}$ has its sibling in a subset of $C'' \setminus \mathcal{R}$. For each short affix cluster $\mathcal{X}$ (see Figure 2.5) whose sibling affix, say $\mathcal{Q}$, is in C'', we use the remaining half of the credits of the ϵ-clusters of $\mathcal{Q}$ to pay for the level-i spanner edges incident to $\mathcal{X}$. Note that X is incident to at most $\frac{2g}{\epsilon}\Delta_{g,\epsilon}$ level-i spanner edges. Since $\mathrm{ediam}(\mathcal{Q}) \geq \ell$, $cr(\mathcal{Q}) \geq c(\epsilon)\ell$ by invariant DC1 for level $i-1$. Thus, half the credit of $\mathcal{Q}$ is sufficient when $c(\epsilon) \geq \frac{2g}{\epsilon}\Delta_{g,\epsilon} = \frac{g^2}{\epsilon^2}$. □

By Claim 2.2.16, it remains to show that C'' has a canonical pair $(\mathcal{S}, Y)$. Let $\mathcal{X}$ be a set of ϵ-clusters. We define a subset of $\mathcal{X}$ as follows:

$$\lfloor \mathcal{X} \rfloor_{2g/\epsilon} = \begin{cases} \mathcal{X} & \text{if } |\mathcal{X}| \leq {}^{2g}/_{\epsilon} \\ \text{any subset of } {}^{2g}/_{\epsilon}\, \epsilon\text{-clusters of } \mathcal{X} & \text{otherwise} \end{cases}$$

Claim 2.2.17. *If C is constructed in Phase 2, then C'' has a canonical pair.*

Proof. Recall C is a tree of ϵ-clusters. That implies C'' is also a tree of ϵ-clusters that are connected by MST edges. Thus, $\mathcal{D}$ is a simple path. Since C contains a branching ϵ-cluster X, there must be at least one neighbor ϵ-cluster of X that is not in $\mathcal{D}$. Let Y be an arbitrary neighbor ϵ-cluster in C of X and $\mathcal{S} = \lfloor \mathcal{D} \rfloor_{2g/\epsilon}$. By definition, $|\mathcal{S}| \leq \frac{2g}{\epsilon}$.

It remains to show that credits of ϵ-clusters of $\mathcal{S}$ and MST edges of D is sufficient to guarantee invariant DC1 for C''. Suppose that $\mathcal{S}$ contains at least $\frac{2g}{\epsilon}$ ϵ-clusters. By invariant DC1 for level $i-1$, $cr(S) \geq c(\epsilon)g\ell \geq c(\epsilon)\max(\mathrm{diam}(C''), \ell/2)$ which is enough to maintain invariant DC1. Thus, we can assume that $\mathcal{S}$ contains every ϵ-cluster of $\mathcal{D}$. Since $\mathcal{D}$ consists of ϵ-clusters and MST edges only, we have:

$$\mathrm{diam}(\mathcal{D}) \leq \sum_{X_\epsilon \in \mathcal{D}} \mathrm{diam}(X_\epsilon) + \sum_{e \in \mathrm{MST}(\mathcal{D})} w(e)$$

Thus, credits of ϵ-clusters of $\mathcal{S}$ and MST edges of $\mathcal{D}$ are sufficient to maintain DC1. $\square$

We now consider the case when C is constructed in Phase 3. Recall C consists of four paths $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2$ that are not necessarily distinct and a single spanner edge e (see Figure 2.4).

Claim 2.2.18. *If the four paths $\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2$ are distinct, then C'' has a canonical pair.*

Proof. Let $\mathcal{F} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{Q}_1, \mathcal{Q}_2\}$. By construction in Phase 3, C is an acyclic graph of ϵ-clusters connected by MST edges and a single spanner edge e. Thus, $\mathcal{D}$ is a simple path. That implies at most two paths, say $\mathcal{P}'$ and $\mathcal{Q}'$, among four paths in $\mathcal{F}$ share ϵ-clusters with $\mathcal{D}$. Let other two paths of $\mathcal{F}$ be $\mathcal{P}''$ and $\mathcal{Q}''$. Let Y be an arbitrary ϵ-cluster of $\mathcal{Q}''$ and

$$\mathcal{S} = \lfloor \mathcal{D} \cup \mathcal{P}' \cup \mathcal{Q}' \cup \mathcal{P}'' \rfloor_{2g/_\epsilon}$$

If $\mathcal{S} = \frac{2g}{\epsilon}$, then $cr(\mathcal{S}) \geq c(\epsilon)g\ell$ by invariant DC1 for level $i-1$. Hence, credits of ϵ-clusters in $\mathcal{S}$ are sufficient to maintain DC1 for C'' since $\mathrm{diam}(C'') \leq g\ell$ as shown in the previous section.

Thus, we can assume that $\mathcal{S} < \frac{2g}{\epsilon}$. In this case, $\mathcal{S} = \mathcal{D} \cup \mathcal{P}' \cup \mathcal{Q}' \cup \mathcal{P}''$. If $\mathcal{D}$ does not contain the spanner edge e, using the same argument in Claim 2.2.17, we can show that credits of ϵ-clusters and MST edges of $\mathcal{D}$ are sufficient to maintain invariant DC1 for C''. Otherwise, we assign credits of $\mathcal{P}''$ to e. Since $\mathrm{ediam}(\mathcal{P}'') \geq \ell \geq w(e)$, by invariant DC1 for level $i-1$, $cr(\mathcal{P}'') \geq c(\epsilon)\mathrm{ediam}(\mathcal{P}'') \geq c(\epsilon)w(e)$. Thus e is assigned credit of at least $c(\epsilon)$ times its length. We then use credits of ϵ-clusters and edges of $\mathcal{D}$ to maintain DC1. The rest of the proof is similar to Claim 2.2.17. $\square$

We assume that $\mathcal{P}_1 = \mathcal{Q}_1 = \mathcal{P}_{xy}$. In this case, C contains a unique cycle, which is $\{e\} \cup \mathcal{P}_{xy}$. We first prove that $\mathcal{D}$ is a path when ϵ is sufficiently small.

Claim 2.2.19. *$\mathcal{D}$ is a path if ϵ is smaller than $\frac{1}{2g}$.*

Proof. If $\mathcal{D}$ is not simple, it contains a cycle $\mathcal{C}_{xy}$. Let u and v be two vertices of the same ϵ-cluster, say X_ϵ, such that D enters and leaves $\mathcal{C}_{xy}$ at u and v, respectively. Then, the subpath D_{uv} between u and v of D must contain edge e of length at least $\ell/2$. However, we can shortcut D_{uv} through X_ϵ by a path of length at most $\mathrm{diam}(X_\epsilon) \leq g\epsilon\ell$ by DC2. For $\epsilon < \frac{1}{2g}$, the shortcut has length smaller than $w(D_{uv})$, contradicting that D is a shortest path. □

Observation 2.2.20. *$\mathcal{P}_{xy} \nsubseteq \mathcal{D}$.*

Proof. For otherwise, $\mathcal{D}$ could be shortcut through e at a cost of

$$\begin{aligned}
&\leq \underbrace{\mathrm{diam}(C_x) + \mathrm{diam}(C_y) + w(e)}_{\text{cost of shortcut}} - \underbrace{(\mathrm{diam}(\mathcal{P}_{xy}) - \mathrm{diam}(C_x) - \mathrm{diam}(C_y))}_{\text{lower bound on diameter}} \\
&\leq w(e) + 4g\epsilon\ell - (1 + s\epsilon)w(e) \quad \text{(by the stretch condition for } e) \\
&\leq 4g\epsilon\ell - s\epsilon\ell/2 \quad (\text{since } w(e) \geq \ell/2)
\end{aligned}$$

This change in cost is negative for $s \geq 8g + 1$. □

Claim 2.2.21. *C'' has a canonical pair.*

Proof. Let Y be an ϵ-cluster of $\mathcal{P}_{xy} \setminus \mathcal{D}$. Y exists by Observation 2.2.20. We define:

$$\mathcal{S} = \lfloor \mathcal{D} \cup \mathcal{P}_2 \cup \mathcal{Q}_2 \cup \mathcal{P}_{xy} \setminus \{Y\} \rfloor_{2g/\epsilon}$$

If $|\mathcal{S}| = \frac{2g}{\epsilon}$, then the total credit of ϵ-clusters in $\mathcal{S}$ is at least $c(\epsilon)g\ell$ by invariant DC1 for level $i-1$. Thus credits of ϵ-clusters in $\mathcal{S}$ is sufficient to maintain invariant DC1 for C''. That implies C'' has a canonical pair.

Otherwise, $\mathcal{S} = \mathcal{D} \cup \mathcal{P}_2 \cup \mathcal{Q}_2 \cup \mathcal{P}_{xy} \setminus \{Y\}$. If $\mathcal{D}$ does not contain the spanner edge e, then by the same argument in Claim 2.2.17, we can argue that credits of ϵ-clusters and MST edges in $\mathcal{D}$ is enough to maintain invariant DC1 for C''. Thus, we can assume that $\mathcal{D}$ contains e. We consider two cases:

1. $\mathcal{D}$ contains an internal ϵ-cluster of $\mathcal{P}_{xy}$. Since $\mathcal{D}$ is a path by Claim 2.2.19, its does not contain any internal ϵ-cluster of at least one of two paths $\mathcal{P}_2, \mathcal{Q}_2$, w.l.o.g., say $\mathcal{P}_2$. Since $\mathrm{ediam}(\mathcal{P}_2) \geq \ell$, by invariant DC1 for level $i-1$, the total credit of ϵ-clusters in $\mathcal{P}_2$ is at least $c(\epsilon)\ell$ which is at least $c(\epsilon)w(e)$. Thus, by assigning credits of $\mathcal{P}_2$ to e, every edge of $\mathcal{D}$ has credit at least $c(\epsilon)$ times it length. Thus, credits of ϵ-clusters and edges of $\mathcal{D}$ are enough to maintain DC1 for C''.

2. $\mathcal{D}$ does not contain any internal ϵ-cluster of $\mathcal{P}_{xy}$. We have:

$$\begin{aligned}
&\mathrm{diam}(\mathcal{P}_{xy} \setminus \{C_x, C_y\}) \\
&\geq \mathrm{diam}(\mathcal{P}_{xy}) - \mathrm{diam}(C_x) - \mathrm{diam}(C_y) \\
&\geq (1+s\epsilon)w(e) - \mathrm{diam}(C_x) - \mathrm{diam}(C_y) \quad \text{(by the stretch condition)} \\
&\geq w(e) + s\epsilon\ell/2 - 2g\epsilon\ell \quad \text{(by bounds on } w(e) \text{ and DC2)} \\
&\geq w(e) + g\epsilon\ell \quad \text{(for } s \geq 8g+1\text{, as previously required)}
\end{aligned} \tag{2.4}$$

The credit of the MST edges and ϵ-clusters of $\mathcal{P}_{xy} \setminus \{C_x, C_y\}$ is at least:

$$\begin{aligned}
&c(\epsilon) \cdot (\mathrm{MST}(\mathcal{P}_{xy} \setminus \{C_x, C_y\}) + \mathrm{ediam}(\mathcal{P}_{xy} \setminus \{C_x, C_y\})) \\
&\geq c(\epsilon) \cdot \mathrm{diam}(\mathcal{P}_{xy} \setminus \{C_x, C_y\}) \\
&\geq c(\epsilon)(w(e) + g\epsilon\ell) \qquad \square
\end{aligned}$$

Since $\mathrm{diam}(Y) \leq g\epsilon\ell$ by invariant DC2 for level $i-1$, the total credit of ϵ-clusters of $\mathcal{P}_{xy} \setminus \{C_x, C_y, Y\}$ and MST edges of $\mathcal{P}_{xy} \setminus \{C_x, C_y\}$ is at least $c(\epsilon) \cdot w(e)$. Thus, by assigning this credit to e, we can argue that credits of ϵ-clusters and edges of $\mathcal{D}$ are enough to maintain DC1 for C''.

2.2.3.4 No Phase 1,2 or 3 clusters

We now deal with the case when there are no level-i clusters formed in Phase 1,2 and 3.

Observation 2.2.22. *There is no level-i cluster formed in Phase 1, 2 and 3 if and only if (i) the tree $\mathcal{T}$ of ϵ-clusters is a path and (ii) every spanner edge is incident to an ϵ-cluster in an affix of $\mathcal{T}$ having effective diameter at most 2ℓ.*

By Claim 2.2.11, we only need to pay for spanner edges incident to short affix clusters of $\mathcal{T}$. Since short clusters have at most $\frac{2g}{\epsilon}$ ϵ-clusters, there are at most $\frac{4g\Delta_{g,\epsilon}}{\epsilon}$ such spanner edges, that we assign to set B (Lemma 2.2.6). Below, we show that $w(B) \leq O(\frac{g^2}{\epsilon^2}) \cdot w(\text{MST})$ across all levels, implying Lemma 2.2.6.

Claim 2.2.23. $w(B) = O(\frac{g^2}{\epsilon^2}) \cdot w(\text{MST})$.

Proof. We have:

$$\begin{aligned}
\frac{4g\Delta_{g,\epsilon}}{\epsilon}\sum_i \ell_i &\leq \frac{4g\Delta_{g,\epsilon}}{\epsilon}\ell_{\max}\sum_i \epsilon^i, \text{ where } \ell_{\max} = \max_{e\in S}\{w(e)\} \\
&\leq \frac{4g\Delta_{g,\epsilon}}{\epsilon} w(\text{MST})\sum_i \epsilon^i \qquad (2.5)\\
&\leq \frac{4g\Delta_{g,\epsilon}}{\epsilon} w(\text{MST})\frac{1}{1-\epsilon} = \frac{12g^2}{\epsilon^2}\cdot w(\text{MST})
\end{aligned}$$

□

2.3 The Subset Traveling Salesperson Problem

In this section, we give the first polynomial time approximation scheme for the subset Traveling Salesperson Problem (subset TSP) in H-minor-free graphs. Our main technical contribution is a polynomial time algorithm that, given an edge-weighted H-minor-free graph G and a set of k terminals T, finds a subgraph of G with weight at most $O_H(\text{poly}(\frac{1}{\epsilon})\log k)$ times the weight of the minimum Steiner tree for T that preserves pairwise distances between terminals up to $(1+\epsilon)$ factor. This is the first such spanner for H-minor-free graphs. Given this spanner, we use the PTAS framework in Section 2.1 to obtain a PTAS for the subset TSP problem. Our PTAS generalizes PTASes for the same problem by Klein [61] for planar graphs and by Borradaile, Demaine and Tazari [18] for bounded genus graphs.

Theorem 2.3.1. *For any fixed $\epsilon > 0$, there is a polynomial time algorithm that, given an edge-weighted H-minor-free graph G and a set of k terminals T in G, finds a tour that visits every terminal of T at least once whose length is at most $(1+\epsilon)$ times the length of the optimal tour.*

The precise running time of our algorithm in Theorem 2.3.1 is $n^{O_H(\text{poly}(\frac{1}{\epsilon}))}$. .

Light spanners A light spanner for subset TSP, called a *light subset spanner*, is a subgraph S that satisfies two conditions: (i) $d_G(x,y) \leq d_S(x,y) \leq (1+\epsilon)d_S(x,y)$ for every $x, y \in T$ and (ii) $w(S) \leq f(\epsilon)w(\mathrm{ST})$ where ST is an optimal Steiner tree that spans T and $f(\epsilon)$ is a constant, called *lightness*, that depends on ϵ only.

Klein [61] was the first to give a polynomial time construction of light subset spanners for planar graphs. Klein's construction can be generalized to bounded genus graphs via the cutting technique, as noted by Borradaile, Demaine and Tazari [18]. Borradaile, Demaine and Tazari [18] conjectured that by using Robertson and Seymour's decomposition [82], it is possible to extend Klein's construction to minor-free graphs. However, this direction has not been fruitful. The main difficulty is that known subset spanner constructions rely on a charging argument based on non-crossing embeddings of input graphs. It is unclear how the charging argument can be modified at the presence of crossings. Thus, even in bounded treewidth graphs, which are normally regarded as easy instances of minor-free graphs, it is unknown whether light subset spanners exist.

In this thesis, we present a different path toward constructing a subset spanner with small weight. First, we introduce the *ℓ-close spanner* problem that captures the difficulty of constructing light subset spanners. An ℓ-close spanner for a terminal set T is a subgraph that almost preserves distances between terminal pairs whose shortest paths have weight at most ℓ. We show how to construct ℓ-close spanners of small weight based on two ingredients: (1) single-source spanners for general graphs that generalize Klein's planar single-source spanners [61] and (2) shortest path separators [1] for H-minor-free graphs. Our ℓ-close spanner construction is inspired by the terminal path cover construction for planar graphs of Cheung, Goranci and Henzinger [31]. An $O_H(\log k)$ factor is introduced into the lightness of ℓ-close spanners. Second, we show a lightness-preserving reduction from constructing light subset spanners to constructing light ℓ-close spanners. Our reduction is inspired by the analysis of greedy spanners presented in Section 2.2. Since an $O(\log k)$ factor is introduced in the first step, the overall lightness of our spanner is $O_H(\log k \,\mathrm{poly}(\frac{1}{\epsilon}))$. For bounded treewidth graphs, we are able to remove $O(\log k)$ factor based on the recent work of Krauthgamer, Nguyễn, and Zondiner [65] in constructing terminal distance preserving minors.

Theorem 2.3.2. *Let T be a subset of k vertices of an H-minor-free graph G. Let* ST *be a minimum Steiner tree of G for T. There is a polynomial time algorithm that can* *find a subgraph S of G such that:*

1. $d_G(x,y) \le d_H(x,y) \le (1+\epsilon)d_G(x,y)$ *for every two distinct terminals* $x,y \in T$.
2. $w(S) = O_H(\mathrm{poly}(\frac{1}{\epsilon})\log k)w(\mathrm{ST})$.

where $O_H(.)$ *hides the dependency of the constant on* $|H|$. *Furthermore, if* G *has treewidth* tw, *then* $w(S) = O(\mathrm{poly}(\frac{1}{\epsilon})\mathrm{tw}^5)w(\mathrm{ST})$.

Using the standard dynamic program for bounded treewidth graphs [15], we can find an optimal solution for subset TSP in treewidth-tw graphs in $2^{O(\mathrm{tw}\log\mathrm{tw})}n^{O(1)}$ time. Since our spanner has a $O(\log k)$ factor in the lightness is not a light spanner as in the description of the PTAS framework in Section 2.1, we need a dynamic program of running time $2^{O(\mathrm{tw})}n^{O(1)}$. We apply the rank based method [16] to achieve this result.

Theorem 2.3.3. *There is a* $2^{O(\mathrm{tw})}n^{O(1)}$*-time algorithm that can solve subset TSP optimally in graphs of treewidth at most* tw.

We present the proof of Theorem 2.3.3 in Section 2.3.5.

2.3.1 Subset spanners

Let $w_G : E(G) \mapsto \mathbb{R}^+$ be the weight function on edges of G. When the graph is clear from context, we would drop the subscript in the weight function. We say an edge-weighted graph H is a *strict minor* of G if (i) H is a minor of G, (ii) $V(H) \subseteq V(G)$ and (iii) for every edge $e \in H$ with two endpoints x, y, $w_H(e) = d_G(x,y)$.

Given a terminal set T of a graph G, Krauthgamer, Nguyễn, and Zondiner [65] showed that G can be compressed by applying the minor transformation such that the distance between every pair of terminals is preserved.

Lemma 2.3.4 (Theorem 2.1 [65])**.** *Let* T *be a set of* k *terminals in a graph* G. *There is a strict minor* G' *of* G *such that (i)* $T \subseteq V(G')$, *(ii)* $V(G') = O(k^4)$ *and* $E(G') = O(k^4)$ *and (iii)* $d_{G'}(x,y) = d_G(x,y)$ *for every two distinct terminals* $x,y \in T$. *Furthermore,* G' *can be found in polynomial time.*

If G has bounded treewidth, Krauthgamer, Nguyễn, and Zondiner [65] showed a stronger version of Lemma 2.3.4.

Lemma 2.3.5. *Let T be a set of k terminals in a graph G of treewidth at most* tw*. There is a strict minor G' of G such that (i) $T \subseteq V(G')$, (ii) $V(G') = O(\mathrm{tw}^3 k)$ and (iii) $d_{G'}(x,y) = d_G(x,y)$ for every two distinct terminals $x, y \in T$. Furthermore, we can find G' in polynomial time.*

Lemma 2.3.5 allows us to obtain a stronger bound on the weight of the subset spanner in bounded treewidth graphs as stated in Theorem 2.3.2.

2.3.2 Subset spanner construction overview

By Lemma 2.3.4, we can assume w.l.o.g that G only has $O(k^4)$ vertices since we can find a subset spanner for terminals in the compressed graph of G and then decompress the subset spanner by replacing each edge by a shortest path between the edge's endpoints in G. Thus, the $\log n$ factor incurred in the weight of our subset spanner construction below can be reduced to $\log k$. We say two terminals x, y are *ℓ-close* if $d_G(x,y) \le \ell$.

Definition 2.3.6 (ℓ-close spanners)**.** *Given a graph G and a set of terminals T, a subgraph S of G is an* ℓ-close spanner *for T if for every two distinct ℓ-close terminals $x, y \in T$, $d_G(x,y) \le d_S(x,y) \le (1+\epsilon)d_G(x,y)$.*

Our first major contribution is to show that one can obtain an ℓ-close spanner of small weight in H-minor-free graphs. Since there are at most $O(k^2)$ terminal pairs, one can trivially obtain a spanner of weight at most $O(k^2\ell)$ by adding in a shortest path for each ℓ-close terminal pair. However, in our problem, we need an ℓ-close spanner of smaller weight. By exploiting H-minor-freeness, we can replace a factor k by a factor $\log n$. We also show a stronger result for graphs of treewidth at most tw.

Theorem 2.3.7. *Given an H-minor-free graph G of n vertices, a terminal set T of size k and a positive parameter ℓ, there is a polynomial time algorithm that can find an ℓ-close spanner S for T with weight at most $O_H(\ell k \log n \,\mathrm{poly}(\frac{1}{\epsilon}))$. Furthermore, if G has treewidth at most* tw*, then $w(S) = O(\mathrm{tw}^5 \ell k)$.*

When G has treewidth at most tw, Lemma 2.3.5 tells us that shortest paths between terminals in bounded treewidth graphs share many edges. Thus, by carefully choosing a set of shortest paths between terminal pairs, we can obtain an ℓ-close spanner of weight at most $O(k\ell)$ from such paths. One may ask whether we can apply Lemma 2.3.4 to obtain an ℓ-close spanner with small weight for minor-free graphs. In our construction, to obtain an ℓ-close spanner with (nearly) constant lightness, we need a strict minor of (nearly) linear size. Lemma 2.3.4 only gives us an ℓ-close spanner of lightness $O(k^3)$, which is worst than the trivial spanner that includes all pairwise shortest paths.

A natural idea to deal with H-minor-free graphs is extending Lemma 2.3.5 to H-minor-free graphs. However, a negative result by Krauthgamer, Nguyễn, and Zondiner [65] showed that it is impossible to do so, even in planar graphs. Formally, they showed that any minor must have at least $\Omega(k^2)$ Steiner vertices[3] to preserve pairwise distances of k terminals exactly. Even in the approximate setting where one seeks to approximately preserve terminal distances up to $(1+\epsilon)$ factor, the best known approximate terminal distance preserving minors for planar graphs have $\Omega(k^2 \operatorname{poly}(\log k)/\epsilon^2)$ Steiner vertices [31].

Inspired by the construction of the terminal path cover for planar graphs by Cheung, Goranci and Henzinger [31] that was in turn inspired by the construction of distance oracles for H-minor-free graphs by Kawarabayashi, Klein and Sommer [57], we propose an ℓ-close spanner construction based on single-source spanners. Instead of bounding the number of Steiner vertices as in previous papers [57, 31], we bound the weight of the spanner.

Our second major contribution is a reduction from the problem of constructing a subset spanner to that of constructing an ℓ-close spanner.

Theorem 2.3.8. *Given an H-minor-free graph G of n vertices and a terminal set T of size k. If for any given ℓ and any subset $T' \subseteq T$, there is an ℓ-close spanner for T' with weight at most $O(\tau(\epsilon, k, n)|T|'\ell)$, then G has a subset spanner with weight at most $O(\operatorname{poly}(\frac{1}{\epsilon})\tau(\epsilon, k, n))w(\mathrm{ST})$ where $\tau(\epsilon, k, n)$ is a function of ϵ, k, n.*

Theorem 2.3.2 follows from Theorem 2.3.8 since $\tau(\epsilon, k, n) = O(\log n \operatorname{poly}(\frac{1}{\epsilon}))$ when G is H-minor-free and $\tau(\epsilon, k, n) = O(\mathrm{tw}^5)$ when G has treewidth at most tw (Theorem 2.3.7). By Lemma 2.3.4, we can further improve the $\log n$ factor to $\log k$.

Our reduction is based on the iterative super-clustering technique that we use to show that greedy $(1 + \epsilon)$-spanners in H-minor-free graphs are light in Section 2.2.

[3]Vertices in $V(G) \setminus T$ are called *Steiner vertices*.

2.3.3 Constructing ℓ-close spanners

In this section, we prove Theorem 2.3.7. We first show how to construct ℓ-close spanners for bounded treewidth graphs.

2.3.3.1 Proof of Theorem 2.3.7 for bounded treewidth graphs

Suppose that G has treewidth at most tw. Let G' be a strict minor of G as stated in Lemma 2.3.5. We remove from $E(G')$ every edge e that has $w_{G'}(e) > \ell$ since no shortest paths between terminals in G' of weight at most ℓ can use e. Since G' has treewidth at most tw (it is a minor of G), $|E(G')| \leq \mathrm{tw}|V(G')|$ (see Kloks [63]). Since $|V(G')| = O(\mathrm{tw}^3 k)$, $|E(G')| = O(\mathrm{tw}^4 k)$. Thus, $w_{G'}(E(G')) = O(\mathrm{tw}^4 \ell k)$. Let S be the subgraph of G obtained by replacing every edge of $E(G')$ by a shortest path in G between its endpoints. We have $w_G(S) \leq w_{G'}(E(G')) = O(\mathrm{tw}^4 \ell k)$.

We now show that $d_G(x, y) = d_S(x, y)$ for every two distinct ℓ-close terminals $x, y \in T$. Let P' be a shortest path between x and y in G'. Let $\{e_1, e_2, \ldots, e_t\}$ be the set of edges in P', where t is the number of edges of P'. Since $d_{G'}(x, y) = d_G(x, y) \leq \ell$, no edges in P' are removed. Replace each edge e_i in P' by a shortest path P_i in G between its endpoints. Let $P = P_1 \circ P_2 \circ \ldots \circ P_t$. Since P is a walk between x, y in S, we have $w_S(P) \geq d_S(x, y)$. By construction of S, $w_S(P) = w_{G'}(P') = d_{G'}(x, y)$ and by Lemma 2.3.5, $d_G(x, y) = d_{G'}(x, y)$. Thus, $w_S(P) = d_G(x, y)$. That implies $d_S(x, y) = d_G(x, y)$.

2.3.3.2 Proof of Theorem 2.3.7 for H-minor-free graphs

Our starting point is the construction of single-source spanners for planar graphs by Klein (Theorem 4.1 [61]). We show that Klein's single source-spanner has small weight even without planarity.

Lemma 2.3.9. *Let p be a vertex and P be a shortest path in a graph G. Let $y_0 \in P$ be such that $d_G(p, y_0) = d_G(p, P)$. Let $R = d_G(p, P)$. Fix an endpoint of P to be its left-most vertex. Let $\{y_1, \ldots, y_I\} \subseteq V(P)$ be a maximal set of vertices such that y_i is the closest point to the right of y_{i-1} such that:*

$$(1 + \epsilon) d_G(p, y_i) < d_G(p, y_{i-1}) + d_P(y_{i-1}, y_i) \qquad 1 \leq i \leq I \tag{2.6}$$

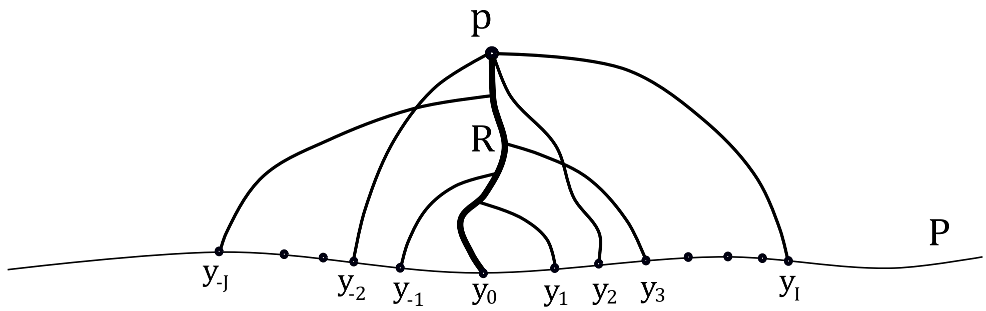

Figure 2.6: A single-source spanner constructed by Klein's algorithm. The thick path is R.

We symmetrically define a maximal set of points $(y_{-1}, y_{-2}, \ldots, y_{-J})$ *to the left of* y_0 *on* P*. Let* $\mathcal{Q} = \{Q_{-J}, Q_{-J+1}, \ldots, Q_{-1}, Q_0, Q_1, \ldots, Q_I\}$ *be a set of shortest paths where* Q_i *is a shortest* p*-to-*y_i *path in* G*,* $-J \leq i \leq I$*. Then, we have:*

(1) $d_{\mathcal{Q}\cup P}(p, q) \leq (1+\epsilon) d_G(p, q)$ *for every* $q \in P$*.*

(2) $w(\mathcal{Q}) \leq 8\epsilon^{-2} R$*.*

(3) $I \leq 8\epsilon^{-2}$ *and* $J \leq 8\epsilon^{-2}$*.*

(4) $d_P(y_0, y_I) \leq 4\epsilon^{-1} R$ *and* $d_P(y_{-J}, y_0) \leq 4\epsilon^{-1} R$*.*

Proof. See Figure 2.6 for an illustration. Property (1) follows directly from the maximality of the set of points $y_{-J}, \ldots, y_0, \ldots, y_I$. We now show property (4). By symmetry, it is sufficient to show that:

$$d_P(y_0, y_I) \leq 4\epsilon^{-1} R \tag{2.7}$$

Suppose otherwise. Then, there exists $\ell \in \{0, \ldots, I-1\}$ such that $d_P(y_0, y_\ell) \leq 4\epsilon^{-1} R$ and $d_P(y_0, y_{\ell+1}) > 4\epsilon^{-1} R$. We have:

$$\begin{aligned}
(1+\epsilon) d_G(p, y_{\ell+1}) &\geq (1+\epsilon)(d_G(y_0, y_{\ell+1}) - d_G(p, y_0)) \qquad \text{(by triangle inequality)} \\
&= (1+\epsilon)(d_G(y_0, y_{\ell+1}) + d_G(p, y_0)) - 2(1+\epsilon) d_G(p, y_0) \\
&\geq (d_G(y_0, y_{\ell+1}) + d_G(p, y_0)) + \epsilon d_G(y_0, y_{\ell+1}) - 2(1+\epsilon) d_G(p, y_0) \\
&= (d_G(y_0, y_\ell) + d_G(p, y_0)) + d_P(y_\ell, y_{\ell+1}) + \epsilon d_G(y_0, y_{\ell+1}) - 2(1+\epsilon) d_G(p, y_0) \\
&\geq d_G(p, y_\ell) + d_P(y_\ell, y_{\ell+1}) + \epsilon d_G(y_0, y_{\ell+1}) - 2(1+\epsilon) d_G(p, y_0) \\
&> d_G(p, y_\ell) + d_P(y_\ell, y_{\ell+1}) + 4R - 2(1+\epsilon) R \qquad \text{(since } \epsilon d_P(y_0, y_{\ell+1}) > 4R\text{)} \\
&\geq d_G(p, y_\ell) + d_P(y_\ell, y_{\ell+1}) \qquad \text{(since } \epsilon < 1\text{)}
\end{aligned}$$

contradicting Equation (2.6). Thus, no such ℓ exists.

To prove (3), we use the argument in the proof of Theorem 4.1 of Klein [61], that we elaborate here for completeness.

$$\begin{aligned}
w(Q_I) &< (1+\epsilon)^{-1}(w(Q_{I-1}) + d_P(y_{I-1}, y_I)) \\
&\leq (1+\epsilon)^{-1}w(Q_{I-1}) + d_P(y_{I-1}, y_I) \\
&< (1-\epsilon/2)w(Q_{I-1}) + d_P(y_{I-1}, y_I) \qquad \text{(since } \epsilon < 1\text{)} \\
&\leq w(Q_{I-1}) - \frac{\epsilon R}{2} + d_P(y_{I-1}, y_I) \\
&\leq w(Q_0) - I\frac{\epsilon R}{2} + d_P(y_0, y_I) \\
&= (1-\epsilon I/2)R + d_P(y_0, y_I) \\
&\leq (1-\epsilon I/2)R + 4\epsilon^{-1}R \qquad \text{(by Equation (2.7))}
\end{aligned} \tag{2.8}$$

Since $w(Q_I) \geq R$, by Equation (2.8), we have $I < 8\epsilon^{-2}$. By a similar argument, we can show that $J < 8\epsilon^{-2}$.

To prove (2), we sum both sides of Equation (2.6) for every $1 \leq i \leq I$.

$$\begin{aligned}
(1+\epsilon)(w(Q_0) + \ldots + w(Q_I)) &\leq (w(Q_0) + \ldots + w(Q_I)) - w(Q_I) + d_P(y_0, y_I) \\
&\leq (w(Q_0) + \ldots + w(Q_I)) + 4\epsilon^{-1}R \qquad \text{(by (2.7))}
\end{aligned} \tag{2.9}$$

That implies $w(Q_0) + \ldots + w(Q_I) \leq 4\epsilon^{-2}R$. By a symmetric argument, we can show that $w(Q_{-J}) + \ldots + w(Q_0) \leq 4\epsilon^{-2}R$. □

Let SSSpanner(G, P, p, ϵ) be the subgraph of Lemma 2.3.9. We can also obtain a generalization of Klein's bipartite spanner (Theorem 5.1 [61]) for non-planar graphs from Lemma 2.3.9.

Corollary 2.3.10. *Let W be a walk and P be a shortest path in a graph G. We denote by R the distance between W and P. That is $R = \min_{v\in W} d_G(v, P)$. Then, there is a subgraph H of G such that:*

1. *For every $p \in W, q \in P$, $d_{H\cup P}(p, q) \leq (1+\epsilon)d_G(p, q)$.*
2. *$w(H) \leq O(\epsilon^{-3})w(W) + O(\epsilon^{-2})R$.*

Proof. Let $W = \{v_0, v_1, \ldots, v_r\}$ where r is the length of W. Note that there may be a vertex that appears multiple times along W. We define a sequence of vertices $Y = \{y_0 = v_0, y_1, \ldots, y_I\}$ along W as follows: (i) $y_0 = v_0$ and (ii) y_i is a closet vertex after y_{i-1} such that:

$$d_W(y_i, y_{i-1}) > \epsilon d_G(y_i, P) \tag{2.10}$$

For each y_i, let $\mathcal{Q}_i \leftarrow \textsc{SSSpanner}(G, P, y_i, \epsilon)$. $\mathcal{Q}_i$ is a collection of shortest paths with source y_i. Let $H = \mathcal{Q}_0 \cup \ldots \cup \mathcal{Q}_I$. We first bound the weight of H. Let $R_i = d_G(y_i, P)$. By Equation (2.10), we have:

$$\sum_{i=1}^{I} R_i \leq \epsilon^{-1} d_W(y_0, y_I) = \epsilon^{-1} w(W)$$

Since $R_0 \leq w(W) + R$, we have:

$$\sum_{i=0}^{I} R_i \leq (\epsilon^{-1} + 1) w(W) + R \tag{2.11}$$

By (2) of Lemma 2.3.9, we have:

$$w(H) \leq \sum_{i=0}^{I} w(\mathcal{Q}_i) \leq 8\epsilon^{-2} \sum_{i=0}^{I} R_i \tag{2.12}$$

From Equation (2.11) and Equation (2.12), we obtain the desired upper bound on the weight of H.

We now show property (1). If $p \in Y$, then property (1) is satisfied by construction and Lemma 2.3.9. Thus, we can assume that $p \notin Y$. Let ℓ be such that $p \in W[y_\ell, y_{\ell+1}]$. (If $\ell = I$, we define $y_{\ell+1}$ to be the endpoint of W after y_ℓ). Since $p \notin Y$, by Equation (2.10), $d_W(p, y_\ell) < \epsilon d_G(p, P)$ which is at most $\epsilon d_G(p, q)$. Let M be a path from p to q that consists of $W[p, y_\ell]$ and a shortest y_ℓ-to-q path in $H \cup P$. We have:

$$
\begin{aligned}
w(M) &\le w(d_W(p, y_\ell)) + d_{H \cup P}(y_\ell, q) \\
&\le w(d_W(p, y_\ell)) + (1+\epsilon) d_G(y_\ell, q) \\
&\le w(d_W(p, y_\ell)) + (1+\epsilon)(d_G(y_\ell, p) + d_G(p, q)) \qquad \text{(by triangle inequailty)} \\
&\le (2+\epsilon) d_W(y_\ell, p) + (1+\epsilon) d_G(p, q) \qquad (d_G(p, y_\ell) \le d_W(p, y_\ell)) \\
&< (2+\epsilon)\epsilon d_G(p, q) + (1+\epsilon) d_G(p, q) \qquad (d_W(p, y_\ell) < \epsilon d_G(p, q)) \\
&\le (1+4\epsilon) d_G(p, q) \qquad (\text{since } \epsilon < 1)
\end{aligned}
\tag{2.13}
$$

By setting $\epsilon' = 4\epsilon$ we have property (1). □

Claim 2.3.11. *Let P be a shortest path of an edge-weighted graph G. Let $\mathcal{Q} = \{Q_1, \ldots, Q_r\}$ be a set of shortest paths in G such that $Q_i \cap P \neq \emptyset$ and $w(Q_i) \le \ell$, for every $1 \le i \le r$. We denote the endpoints of each Q_i by s_i and t_i. Let k be the number of distinct endpoints of $\mathcal{Q}$. There is a subgraph H of G with weight at most $O(k\epsilon^{-2}\ell)$ such that $d_H(s_i, t_i) \le (1+\epsilon) d_G(s_i, t_i)$ for every $1 \le i \le r$.*

Proof. We first delete every edge of G of length more than ℓ since no path in $\mathcal{Q}$ can contain such an edge. Let $X = \{x_1, x_2, \ldots, x_k\}$ be the set of endpoints of all paths in $\mathcal{Q}$. Let $R_j = d_G(x_j, P)$ and y_j be the closet vertex of x_j in P. Since $Q_i \cap P \neq \emptyset$ and $w(Q_i) \le \ell$ for every i, $d_G(x_j, P) \le \ell$ for every $1 \le j \le k$. For each j, let $\mathcal{Q}_j \leftarrow$ SSSpanner(G, P, x_j, ϵ). Let P_j be a minimal subpath of P that contains every vertex of distance (in P) at most $4\epsilon^{-1}\ell$ from y_j. Since P_j has no edge of length more than ℓ, $w(P_j) \le (8\epsilon^{-1} + 2)\ell$. Since $|R_j| \le \ell$, by (4) of Lemma 2.3.9, we have:

Observation 2.3.12. *P_j contains all endpoints on P of paths in $\mathcal{Q}_j$.*

Recall paths in $\mathcal{Q}_j$ share endpoint x_j. Let:

$$H = \bigcup_{j=1}^{k} (\mathcal{Q}_j \cup P_j) \tag{2.14}$$

We first bound the weight of H. For any j, $1 \le j \le k$, by (2) of Lemma 2.3.9,

$$w(\mathcal{Q}_j) \le O(\epsilon^{-2}) R_j \le O(\epsilon^{-2}\ell)$$

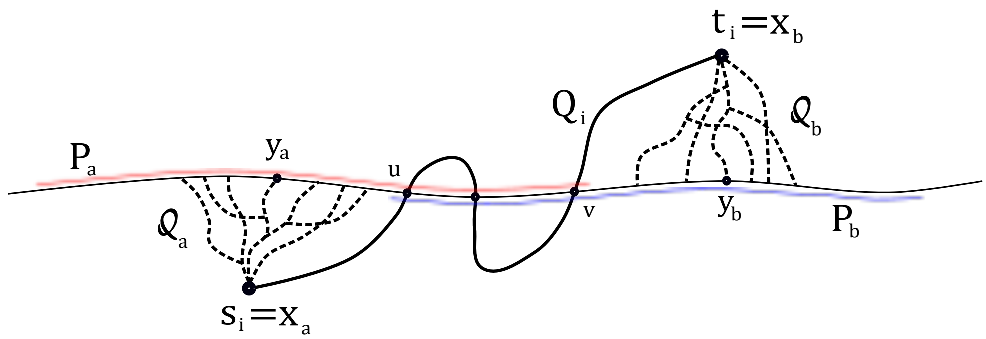

Figure 2.7: Shortest path P is the thin black curve and shortest path Q_i between two terminals s_i, t_i is the thick black curve. P_a and P_b are highlighted red and blue, respectively.

Thus, $w(H) \leq O(k\epsilon^{-2})\ell$.

We now show that $d_H(s_i) \leq (1+\epsilon)d_G(s_i,t_i)$ for any $1 \leq i \leq r$. Let u and v be the first vertex and the last vertex (from s_i) in $Q_i \cap P$, respectively. Suppose that $x_a = s_i$ and $x_b = t_i$ for some a, b, $1 \leq a, b \leq k$ (see Figure 2.7). Since $d_P(y_a, u) = d_G(y_a, u) \leq d_G(s_i, y_a) + d_G(y_a, u) \leq 2\ell$ which is at most $(4\epsilon^{-1}+1)\ell$ when $\epsilon < 1$. Thus, $u \in P_a$. Similarly, we can show that $v \in P_a$. That implies:

Observation 2.3.13. *Subpath $P[u,v]$ of P is a subgraph of H.*

By a similar argument, we can show that u, v both are in P_b (see Figure 2.7). By (1) of Lemma 2.3.9 and Observation 2.3.12, we have:

$$d_H(s_i,u) \leq (1+\epsilon)d_G(s_i,u) \qquad d_H(v,t_i) \leq d_G(v,t_i) \tag{2.15}$$

Since P is a shortest path of G, $w(P[u,v]) = w(Q_i)[u,v]$ and both have length at most ℓ. Thus, we have:

$$\begin{aligned} d_H(s_i,t_i) &\leq d_H(s_i,u) + d_H(u,v) + d_H(v,t_i) \\ &= d_H(s_i,u) + w(Q_i[u,v]) + d_H(v,t_i) \qquad \text{(by Observation 2.3.13)} \\ &\leq (1+\epsilon)d_G(s_i,u) + w(Q_i[u,v]) + (1+\epsilon)d_G(v,t_i) \qquad \text{(by Equation (2.15))} \\ &= (1+\epsilon)w(Q_i[s_i,u]) + w(Q_i[u,v]) + (1+\epsilon)w(Q_i[v,t_i]) \\ &\leq (1+\epsilon)w(Q_i[s_i,t_i]) = (1+\epsilon)d_G(s_i,t_i) \end{aligned}$$

□

For any two paths P and Q, we say P *crosses* Q if $P \cap Q \neq \emptyset$. We say P crosses a set of paths $\mathcal{Q}$ if there exists a path $Q \in \mathcal{Q}$ such that P crosses Q. By Claim 2.3.11, we have:

Lemma 2.3.14. *Let $\mathcal{P}$ be a set of shortest paths in an edge-weighted graph G. Let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_r\}$ be another set of shortest paths in G such that Q_i crosses $\mathcal{P}$ and $w(Q_i) \leq \ell$, for every $1 \leq i \leq r$. We denote the endpoints of each Q_i by s_i and t_i. Let k be the number of distinct endpoints of $\mathcal{Q}$. There is a subgraph H of G with weight at most $O(k\epsilon^{-2}\ell|\mathcal{P}|)$ such that $d_H(s_i, t_i) \leq (1+\epsilon)d_G(s_i, t_i)$ for every $1 \leq i \leq r$. Furthermore, H can be found in polynomial time.*

Proof. Fix an ordering of paths $P_1, P_2, \ldots, P_h$ in $\mathcal{P}$ where $h = |\mathcal{P}|$. For each path P_j, $1 \leq j \leq h$, let $\mathcal{Q}_j$ be the set of paths in $\mathcal{Q}$ such that each path in $\mathcal{Q}$ crosses P_j and does not cross any P_i for all $i < j$. Let H_j be the subgraph of G obtained by applying Claim 2.3.11 with parameters $G, P_j, \mathcal{Q}_j, \epsilon$ and ℓ. Let $H = \cup_{j=1}^{h} H_j$. Then, $w(H) \leq \sum_{i=1}^{h} w(H_i) = O(k\epsilon^{-2}\ell|\mathcal{P}|)$. The stretch guarantee of H follows directly from Claim 2.3.11. □

Let PTPSpanner$(G, \mathcal{P}, \mathcal{Q}, \ell, \epsilon)$ (PTP means path-to-path.) be the subgraph of Lemma 2.3.14. We use this to construct an ℓ-close spanner S as stated in Theorem 2.3.7. (See Figure 2.8.) The input to EllCloseSpanner$(G, T, \mathcal{Q}, \ell, \epsilon)$ consists of an edge-weighted H-minor-free graph G, a set of terminals T, a set of shortest paths $\mathcal{Q} = \{Q_1, \ldots, Q_h\}$ between ℓ-close terminals in T and the stretch parameter ϵ. The algorithm makes use of the following shortest path separator for H-minor-free graphs by Abraham and Gavoille [1].

Lemma 2.3.15 (Theorem 1 [1]). *For every connected H-minor-free graph G of n vertices, there is a family of γ sets of paths $\Omega = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_\gamma\}$ of G such that:*

1. $\sum_{i=1}^{\gamma} |\mathcal{P}_i| = O_H(1)$.
2. *$\mathcal{P}_1$ is a set of shortest paths of G and $\mathcal{P}_i$ is a set of shortest paths of $G \setminus V(\cup_{j<i}\mathcal{P}_i)$ for $i \geq 2$.*
3. *Connected components of $G \setminus V(\Omega)$ have size at most $n/2$.*

We represent the execution of EllCloseSpanner$(G, T, \mathcal{Q}, \ell, \epsilon)$ by a recursion tree $\mathcal{T}$ where each node represents a recursive call on a subgraph, say K of G, and its child

```
EllCloseSpanner(G, T, Q, ℓ, ε)
    if |T| ≤ 1 return ∅
    S ← ∅
    P_0 ← ∅; Ω ← {P_1, ..., P_γ} as in Lemma 2.3.15
    for i ← 1 to γ
        G_i ← G \ (∪_{j=0}^{i-1} P_j)
        Q_i ← the set of paths in Q that cross P_i
        S ← S ∪ PTPSpanner(G_i, P_i, Q_i, ℓ, ε)
        Q ← Q \ Q_i
    for each component G' of G \ V(Ω)
        T' ← T ∩ V(G')
        Q' ← remaining paths in Q with both endpoints in T'
        S ← S ∪ EllCloseSpanner(G', T', Q', ℓ, ε)
    return S
```

Figure 2.8: An ℓ-spanner construction algorithm.

nodes are recursive calls on connected components of $K \setminus \Omega_K$. Here Ω_K is a shortest-path separator of K as in Lemma 2.3.15. The root node of $\mathcal{T}$ is a call on G. Since the size of child graphs in recursive calls is at most half the size of the parent graph, $\mathcal{T}$ has depth $O(\log n)$.

We note that in each recursive call $\textsc{EllCloseSpanner}(G', T', \mathcal{Q}', \ell, \epsilon)$ in the algorithm in Figure 2.8, paths in $\mathcal{Q}'$ are shortest paths of G' since they are shortest paths in G. Observe that none of the paths in $\mathcal{Q}'$ of the second **for** loop contains a vertex of $V(\Omega)$ since any path of $\mathcal{Q}$ that crosses at least one set of paths in Ω will be removed in the first **for** loop.

We now bound the total weight of $S \stackrel{\text{def}}{=} \textsc{EllCloseSpanner}(G, T, \mathcal{Q}, \ell, \epsilon)$. Consider i-th iteration in the first **for** loop in the algorithm in Figure 2.8. We have:

Observation 2.3.16. *$\mathcal{Q}_i$ is a set of shortest paths in G_i.*

By Lemma 2.3.14 and (1) of Lemma 2.3.15, the total weight of S after the first for loop is at most:

$$O(k\epsilon^{-2}\ell \sum_{i=1}^{\gamma} |\mathcal{P}_i|) = O_H(k\epsilon^{-2}\ell)$$

That implies at each level of $\mathcal{T}$, the weight of the returned subgraph of each node is $O_H(k\epsilon^{-2}\ell)$ plus the weight of the subgraphs returned from recursive calls. Since the depth of $\mathcal{T}$ is $O(\log n)$, $w(S) \leq O_H(k\epsilon^{-2}\ell \log n)$.

To complete the proof of Theorem 2.3.7, we need to show that $d_S(x, y) \leq (1 + \epsilon)d_G(x, y)$ for every two distinct ℓ-close terminals $x, y \in \mathcal{T}$. Let $Q_{x,y}$ be the shortest path between x, y in $\mathcal{Q}$. By triangle inequality, we can assume that $Q_{x,y}$ contains no other terminals except x and y. Since the algorithm only stops after each component of G contains at most one terminal, $Q_{x,y}$ must be removed from $\mathcal{Q}$ at some node of $\mathcal{T}$, say τ. More precisely, $Q_{x,y}$ is removed in some iteration, say i, in the first for loop of τ. By Observation 2.3.16 and Lemma 2.3.14, we have:

$$d_S(x, y) \leq (1 + \epsilon)d_{G_i}(x, y) = (1 + \epsilon)d_G(x, y)$$

That implies at each level of $\mathcal{T}$, the weight of the returned subgraph of each node is $O_H(k\epsilon^{-2}\ell)$ plus the weight of the subgraphs returned from recursive calls. Since the depth of $\mathcal{T}$ is $O(\log n)$, $w(S) \leq O_H(k\epsilon^{-2}\ell \log n)$.

To complete the proof of Theorem 2.3.7, we need to show that $d_S(x, y) \leq (1 + \epsilon)d_G(x, y)$ for every two distinct ℓ-close terminals $x, y \in \mathcal{T}$. Let $Q_{x,y}$ be the shortest path between x, y in $\mathcal{Q}$. By triangle inequality, we can assume that $Q_{x,y}$ contains no other terminals except x and y. Since the algorithm only stops after each component of G contains at most one terminal, $Q_{x,y}$ must be removed from $\mathcal{Q}$ at some node of $\mathcal{T}$, say τ. More precisely, $Q_{x,y}$ is removed in some iteration, say i, in the first for loop of τ. By Observation 2.3.16 and Lemma 2.3.14, we have:

$$d_S(x, y) \leq (1 + \epsilon)d_{G_i}(x, y) = (1 + \epsilon)d_G(x, y)$$

2.3.4 A lightness-preserving reduction to constructing ℓ-close spanners

In this section, we give a proof of Theorem 2.3.8.

2.3.4.1 Setup

We first find a constant approximation (in linear time [87]) of the optimal Steiner tree (ST) of G for terminal set T. Let $\mathcal{Q}$ be the set of shortest paths between all terminal pairs of T such that none of them contains a terminal except its endpoints. By the triangle inequality, it suffices to construct a spanner for paths in $\mathcal{Q}$. Note that $|\mathcal{Q}| \leq \frac{k(k-1)}{2}$. Let $n_0 = \max(n, \frac{k(k-1)}{2})$ and $w_0 = \frac{w(\mathrm{ST})}{n_0}$. We construct a subset spanner S in multiple steps. Firstly, we add to S every path in $\mathcal{Q}$ of length at most w_0. The total weight of added paths is at most:

$$\sum_{Q \in \mathcal{Q}, w(Q) \leq w_0} w(Q) \leq \frac{k(k-1)}{2} \frac{w(\mathrm{ST})}{n_0} \leq w(\mathrm{ST}) \tag{2.16}$$

Thus, we can assume that paths in $\mathcal{Q}$ have length at least w_0. Recall paths in $\mathcal{Q}$ have length at most $w(\mathrm{ST})$. Let $J = \lceil \log(1/\epsilon) \rceil$ and $I = \lceil \log_{1/\epsilon} n_0 \rceil$.

Path hierarchies For a fixed i, j where $1 \leq j \leq J, 0 \leq i \leq I$, we define:

$$\Pi_i^j = \left\{ Q \in \mathcal{Q} : \frac{2^{j-1}}{\epsilon^i} w_0 < w(Q) \leq \frac{2^j}{\epsilon^i} w_0 \right\}$$

For a fixed j, $1 \leq j \leq J$, we define a hierarchy of paths:

$$\mathcal{H}_j = \cup_{i=0}^{I} \Pi_i^j \tag{2.17}$$

We refer to paths in Π_j^i as *level-i paths* of hierarchy $\mathcal{H}_j$. We will find a low weight spanner for shortest paths in each hierarchy separately.

Assigning credits to ST edges We guarantee that every edge of ST has weight at most w_0 and has at least $c(\epsilon) w_0$ credits while the total allocated credit is small. We first subdivide every edge e of weight at least w_0 into $\lceil \frac{w(e)}{w_0} \rceil$ edges of equal weight. We call subdividing vertices *virtual vertices*. We then allocate $c(\epsilon) w_0$ credits to each edge (now of weight at most w_0) of ST. The total allocated credit is:

$$\begin{aligned}\sum_{e \in \mathrm{ST}} (\frac{w(e)}{w_0} + 1) c(\epsilon) w_0 &= c(\epsilon) w(\mathrm{ST}) + c(\epsilon) w_0 |E(\mathrm{ST})| \\ &\leq c(\epsilon) w(\mathrm{ST}) + c(\epsilon) \frac{w(\mathrm{ST})}{\max(n, k(k-1)/2)} (n-1) \\ &\leq 2c(\epsilon) w(\mathrm{ST})\end{aligned} \tag{2.18}$$

Thus, we can think of $c(\epsilon)$ as an asymptotic upper bound on the weight of the spanner S that we are going to build. The total number of virtual vertices of ST is at most:

$$\sum_{e \in \mathrm{ST}} (\frac{w(e)}{w_0} + 1) \leq \frac{w(\mathrm{ST})}{w_0} + |V(\mathrm{ST}) - 1| = O(\max(n, k(k-1)/2)) \tag{2.19}$$

Thus, after subdividing long edges of ST, the total number of vertices of the graph is still polynomial. The main result of this section is showing that with a reasonable choice of $c(\epsilon)$, credits of ST edges are enough to pay for a low-weight spanner of paths in each hierarchy.

Theorem 2.3.17. *Let $\tau(\epsilon, k, n)$ be the parameter in the assumption of Theorem 2.3.8. Let $c(\epsilon) = \Theta(\mathrm{poly}(\frac{1}{\epsilon})\tau(\epsilon, k, n))$. For a fixed j, $1 \leq j \leq J$, there is a set of shortest paths $\mathcal{B}_j \subseteq \mathcal{H}_j$ and a subgraph S_j of G such that:*

1. *$\sum_{Q \in B_j} w(Q) \leq O(\epsilon^{-2}) w(\mathrm{ST})$.*
2. *$w(S_j) \leq O(c(\epsilon)) w(\mathrm{ST})$.*
3. *For every path $Q \in \mathcal{H}_j$, $w(Q) \leq d_{S_j \cup \mathcal{B}_j}(x, y) \leq (1+\epsilon) w(Q)$ where x, y are Q's endpoints.*

Both S_i and $\mathcal{B}_j$ can be found in polynomial time.

Theorem 2.3.17 immediately implies Theorem 2.3.8 since we only have $J = O(\log \frac{1}{\epsilon})$ hierarchies. Herein, we focus on proving Theorem 2.3.17. For simplicity of presentation, we drop the index j and refer to $\mathcal{H}_j$, S_j, $\mathcal{B}_j$ and Π_i^j as $\mathcal{H}$, S, $\mathcal{B}$ and Π_i, respectively. We call $\mathcal{B}$ the *holding bag*. We will use the iterative clustering technique which is described in Section 2.2.

We first add all edges of ST to S. We will build a hierarchy of clusters corresponding to the hierarchy of paths $\mathcal{H}$ where each cluster is a connected subgraph of S. That is, for each level i of $\mathcal{H}$, we construct a set of clusters $\mathcal{C}_i$. Level-0 clusters are subtrees of ST and level-i clusters are constructed from level-$(i-1)$ clusters and ST edges.

Recall paths in Π_i have length at most $\ell_i \stackrel{\mathrm{def}}{=} \frac{2^j}{\epsilon^i} w_0$. Let T' be the set of terminals that are endpoints of paths in Π_i. For each cluster $C \in \mathcal{C}_{i-1}$ that contains at least one terminal in T', we designate one terminal to be its centers. Let T'' be the set of centers. We construct an $O(\ell_i)$-close spanner, say K, for T'' and add all edges of K to S. We can show that K is an $(1+\epsilon)$-spanner for paths in Π_i.

To pay for edges of K, we will inductively maintain an invariant that each level-$(i-1)$ cluster has credits proportional to its diameter. (Level-0 clusters take credits of

ST edges and higher-level clusters take credits of lower-level clusters and unused credits of some ST edges connecting them.) Our cluster construction will guarantee that after maintaining credit invariant of level-i clusters, each level-$(i-1)$ cluster will have some credits left that are sufficient to pay for edges of K.

2.3.4.2 Cluster invariants

Recall that $\ell_i = \frac{2^j}{\epsilon^i} w_0$ is the maximum length of level-i paths. We maintain the following invariants for level-i clusters:

(I1) Each level-i cluster has diameter at most $g\ell_i$ where $g = 125$.

(I2) Each level-i cluster of diameter d has at least $c(\epsilon)\max(d, \ell_i/2)$ credits.

Unlike clusters used in the spanner setting, clusters in this setting are not necessarily vertex-disjoint. That introduces various technical complications in the cluster construction. Specifically, we use credits of ST edges in each cluster to both maintain invariant (I2) and pay for spanner edges. However, one ST edge can be shared by multiple clusters and credits of each ST edge can only be used at most once. To resolve this issue, after cluster construction for each level, we will maintain a set of ST edges whose credits have not been used so far. Specifically, we maintain a *cluster tree* $\mathcal{ST}_i(\mathcal{V}_i, \mathcal{E}_i)$ whose vertices are level-i clusters and whose edges are ST edges connecting two vertices in the two corresponding clusters. $\mathcal{ST}_i(\mathcal{V}_i, \mathcal{E}_i)$ satisfies the following invariant:

(I3) Credits of edges of $\mathcal{ST}_i(\mathcal{V}_i, \mathcal{E}_i)$ have not been used in the construction of level-i or lower level clusters.

To simplify the argument for the base case, we add ST to S and add every level-0 path to the holding bag $\mathcal{B}$. The total weight of paths added to $\mathcal{B}$ is at most:

$$\sum_{Q \in \Pi_0^j} w(Q) \leq (2\epsilon^{-1}) \frac{k(k-1)}{2} \frac{w(\mathrm{ST})}{n_0} \leq O(\epsilon^{-1}) w(\mathrm{ST}) \tag{2.20}$$

To construct level-0 clusters, we break ST into subtrees of diameter at least ℓ_0 and at most $6\ell_0$ as follows: breaking a longest path of ST, say P, into subpaths of diameter at least ℓ_0 and at most $2\ell_0$, removing vertices of P from ST and repeat. Let Γ_0 be the collection of the paths. Remaining components of $\mathrm{ST} \setminus V(\Gamma_0)$ are trees of diameter at most ℓ_0. Let T' be such a tree. By construction, T' has an ST edge, say e, to a path, say P of Γ_0. We then augment T' and e to P. Since $w(e) \leq w_0$, after this step, Γ_0 contains trees of diameter at most $4\ell_0 + 2w_0 \leq 6\ell_0$. Then, each tree in Γ_0 serves as a level-0 cluster. Note that clusters in Γ_0 are vertex-disjoint subgraphs of S.

We now show that clusters in $\mathcal{C}_0$ satisfy three invariants. Invariants (I1) is directly implied by the construction. To maintain invariant (I2), we take credits of ST edges in the diameter path of each tree in Γ_0. Let $\mathcal{ST}_0(\mathcal{V}_0, \mathcal{E}_0)$ be the cluster tree obtained from ST by contracting each tree of Γ_0 into a single vertex. Since credits of ST edges outside level-0 clusters are unused, $\mathcal{ST}_0(\mathcal{V}_0, \mathcal{E}_0)$ satisfies invariant (I3).

For simplicity of presentation, we would guarantee that the spanner that we are going to construct has $(1 + s \cdot \epsilon)$ stretch where $s \stackrel{\text{def}}{=} 16g + 1 = 2001$. We can obtain a stretch $(1 + \epsilon')$ by simply setting $\epsilon' = s \cdot \epsilon$. We also assume ϵ is sufficiently smaller than 1 since a subset spanner of stretch $(1 + \epsilon)$ is also a subset spanner of stretch $(1 + 2\epsilon)$ with the same asymptotic weight.

2.3.4.3 Constructing higher level clusters and spanners

In this section, we show how to construct level-i clusters, for $i \geq 1$, from level-$(i-1)$ clusters. To simplify the presentation, we will drop the index i. That is, $\Pi = \Pi_i$ and $\ell = \ell_i$. We refer to clusters in level $(i-1)$ as *ϵ-clusters* since their diameter is an ϵ-fraction of the diameter of level-i clusters. By invariant (I1) for level $i-1$, ϵ-clusters have diameter at most $g\epsilon\ell$. Recall that $\ell = \frac{2^j}{\epsilon^i} w_0$. Let Q be a path in Π, that we call a Π-*path*. We observe that:

Observation 2.3.18. *There is no Π-path that has both endpoints in the same ϵ-cluster when $\epsilon < \frac{1}{g}$.*

Proof. Let Q be a Π-path with both endpoints, say x and y, in the same ϵ-cluster. By invariant (I1) for level $i-1$, there is a path between x and y of length at most $g\epsilon\ell < \ell$; contradicting that Q is a shortest path. $\square$

Π-path removal: We say two Π-paths are *parallel* if their endpoints are both in the same two ϵ-clusters. For each maximal set of parallel Π-paths, we only keep one Π-path of minimum length and remove other paths from Π. We apply this removal process to all maximal subsets of parallel paths of Π. We then remove every Π-path Q from Π such that the distance between two endpoints of Q in S (constructed so far) is at most $(1+s\cdot\epsilon)w(Q)$, since there is already an $(1+\epsilon)$-stretch path between Q's endpoints in S.

Constructing a spanner for Π

Note that paths in Π have length at least $\ell/2$ and at most ℓ. Since ϵ-clusters are non-disjoint, a terminal can be contained in many different ϵ-clusters. For each terminal $t \in T$, we designate an (arbitrary) ϵ-cluster containing t to be its *primary ϵ-cluster*. We say that an ϵ-cluster C is *incident* to Q if C is a primary ϵ-cluster of at least one of Q's endpoints. By Observation 2.3.18, Q has exactly two incident ϵ-clusters.

We call an ϵ-cluster X a Π-*neighbor* of an ϵ-cluster Y if X and Y are incident to the same Π-path. We say an ϵ-cluster has *high-degree* if it has at least $\frac{3g}{\epsilon}$ Π-neighbors and *low-degree* otherwise. For each low-degree ϵ-cluster X, we add to spanner S all Π-paths incident to X.

Let $\mathcal{C}_\epsilon$ be the set of all high-degree ϵ-clusters. For each $X \in \mathcal{C}_\epsilon$, we designate a terminal to be its center. Note that X must have a terminal since it is incident to a Π-path. Let T' be the set of centers of ϵ-clusters in $\mathcal{C}_\epsilon$. Since each terminal has exactly one primary ϵ-cluster, $T' = |\mathcal{C}_\epsilon|$. Let $K \leftarrow \textsc{EllCloseSpanner}(G, T', \mathcal{Q}', 3\ell, \epsilon)$ where $\mathcal{Q}'$ is the maximal set of shortest paths of length at most 3ℓ between terminals in T'. K is a (3ℓ)-close spanner for T'. By the assumption of Theorem 2.3.8, we have:

$$w(K) = O(\tau(\epsilon,k,n)\ell|T'|) = O(\tau(\epsilon,k,n)\ell|\mathcal{C}_\epsilon|) \tag{2.21}$$

We then add every edge of K to S. It remains to show the stretch guarantee for paths in Π.

Claim 2.3.19. *For every Π-path P, there is a path between two endpoints of P in S of length at most $(1+s\cdot\epsilon)w(P)$ when $\epsilon < \frac{1}{g}$.*

Proof. We first assume that P survives after the Π-path removal step. If P is incident to a low-degree ϵ-cluster, then, it is added to S; the lemma trivially holds. Otherwise, P is incident to two high-degree ϵ-clusters, say C_x and C_y. Let $x \in C_x, y \in C_y$ be P's endpoints. Let x_1, y_1 be two centers of C_x, C_y, respectively. Let R be the x_1-to-y_1

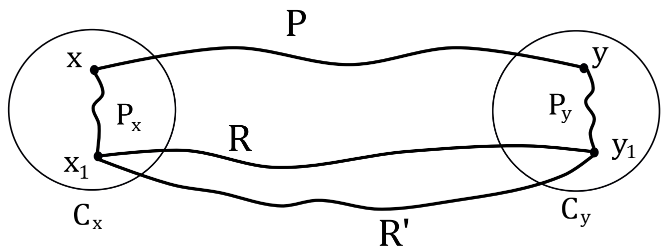

Figure 2.9: Two circles represents ϵ-clusters C_x and C_y with centers x and y, respectively. $R(R')$ is a shortest x_1-to-y_1 path in G (S).

shortest path in G. Let P_x be a shortest x-to-x_1 path in C_x and P_y be a shortest y-to-y_1 path in C_y (see Figure 2.9). We have:

$$w(R) \leq w(P_x) + w(P_y) + w(P) \leq 2g\epsilon\ell + \ell \leq 3\ell \tag{2.22}$$

Since K is a (3ℓ)-close spanner of T', there is a shortest path, say R', between x_1 and y_1 in K such that $w(R') \leq (1+\epsilon)w(R)$. Furthermore, we have:

$$\begin{aligned} w(R) &\leq w(P) + w(P_x) + w(P_y) \\ &\leq w(P) + 2g\epsilon\ell \qquad \text{(by (I1) for level } i-1) \\ &\leq (1+4g\epsilon)w(P) \qquad \text{(since } w(P) \geq \ell/2) \end{aligned} \tag{2.23}$$

Thus, by Equation (2.23), we have:

$$\begin{aligned} w(R') &\leq (1+\epsilon)(1+4g\epsilon)w(P) \\ &\leq (1+(8g+1)\epsilon)w(P) \qquad \text{(since } \epsilon < 1) \end{aligned} \tag{2.24}$$

Let $P' \stackrel{\text{def}}{=} P_x \circ R' \circ P_y$ be an x-to-y walk in S. We have:

$$\begin{aligned} w(P') &\leq w(R') + w(P_x) + w(P_y) \\ &\leq w(R') + 2g\epsilon\ell \qquad \text{(by (I1) for level } i-1) \\ &\leq (1+(8g+1)\epsilon)w(P) + 2g\epsilon\ell \qquad \text{(by Equation (2.24))} \\ &\leq (1+(12g+1)\epsilon)w(P) \qquad \text{(since } w(P) \geq \ell/2) \\ &\leq (1+s\cdot\epsilon)w(P) \qquad \text{(since } s > 12g+1) \end{aligned} \tag{2.25}$$

It remains to consider the case P is removed during the Π-path removal step. There are two subcases: (1) P is removed because there is another Π-path parallel to P and has smaller weight or (2) the stretch between two endpoints of P is already smaller than $(1 + s \cdot \epsilon)$. The latter case immediately implies the lemma. In the former case, let x, y be P's endpoints. Let P_0 be the path parallel to P that survives after Π-path removal step. Note that $w(P_0) \leq w(P)$.

As shown in the second-last line of Equation (2.25), that there is a path P_0' in S of length at most $(1 + (12g + 1)\epsilon)w(P_0)$ between P_0's endpoints, say x_0, y_0. Let C_x (C_y) be the ϵ-cluster that contains x (y) and x_0 (y_0). Then, an x-to-y walk in S consisting of a shortest x-to-x_0 path in C_x, P_0', and a shortest y_0-to-y path in C_y has length at most:

$$
\begin{aligned}
(1 + (12g + 1)\epsilon)w(P_0) + 2g\epsilon\ell &\leq (1 + (12g + 1)w(P) + 2g\epsilon\ell \\
&\leq (1 + (16g + 1)\epsilon)w(P) \qquad \text{(since } w(P) \geq \ell/2\text{)} \\
&\leq (1 + s \cdot \epsilon)w(P) \qquad \text{(since } s = 16g + 1\text{)}
\end{aligned}
$$

□

Constructing clusters

To simplify the notation, we use $\mathcal{ST}(\mathcal{V}, \mathcal{E})$ to denote the cluster tree of level $i-1$, that is, vertices of $\mathcal{V}$ correspond to ϵ-clusters and edges in $\mathcal{E}$ are ST edges connecting ϵ-clusters. Recall that in the spanner construction step, every Π-path incident to low-degree ϵ-clusters is added to S and every Π-path incident to two high-degree ϵ-clusters has an $(1 + \epsilon)$-approximate shortest path in S. Let $\mathcal{E}'$ be the set of edges between vertices in $\mathcal{V}$ where each edge in $\mathcal{E}'$ corresponds to a Π-path Q connecting its incident ϵ-clusters or Q's approximate shortest path in S if both endpoint ϵ-clusters of Q have high degree. We call edges of $\mathcal{E}'$ Π-*edges*. We denote the graph, called *cluster graph*, with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \cup \mathcal{E}'$ by $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$. Observe that $\mathcal{ST}(\mathcal{V}, \mathcal{E})$ is a spanning tree of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$. We use bold lowercase letters to denote vertices and edges of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$.

Let $\kappa(.)$ be the function that maps each vertex $\mathbf{v} \in \mathcal{V}$ to the corresponding ϵ-cluster and each edge $\mathbf{e} \in \mathcal{E} \cup \mathcal{E}'$ to the corresponding ST edge or paths. For each subgraph $\mathcal{S}$ of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$, we denote the corresponding subgraph of S by $\kappa(\mathcal{S})$, where:

$$\kappa(\mathcal{S}) = (\cup_{\mathbf{v} \in \mathcal{S}} \kappa(\mathbf{v})) \bigcup (\cup_{\mathbf{e} \in \mathcal{S}} \kappa(\mathbf{e}))$$

Observation 2.3.20. *$\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$ is a simple graph when $\epsilon < \frac{1}{4g+2}$*

Proof. Suppose otherwise. We first observe that there are no two parallel Π-edges in $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$ since otherwise, one of them would be removed during the Π-path removal process. Thus, the only possibility left is a Π-edge $\mathbf{e}'$ parallel to an $\mathcal{ST}$ edge $\mathbf{e}$. Let $\mathbf{x}$, $\mathbf{y}$ be the endpoints of $\mathbf{e}$. Let x and x' (y and y') be the endpoints of $\kappa(e)$ and $\kappa(e')$ in $\kappa(\mathbf{x})$ ($\kappa(\mathbf{y})$), respectively. Recall $\ell = \ell_i = \frac{2^j}{\epsilon^i} w_0 > \frac{w_0}{\epsilon}$ since $i \geq 1$. Thus, an x-to-y walk W in S that consists of a shortest x'-to-x in $\kappa(\mathbf{x})$, $\kappa(\mathbf{e})$, a shortest y-to-y' in $\kappa(\mathbf{y})$ has length at most:

$$2g\epsilon\ell + w_0 \leq (2g+1)\epsilon\ell < \ell/2$$

when $\epsilon < \frac{1}{4g+2}$. Thus, $w(W) \leq w(\kappa(\mathbf{e}'))$; contradicting that $\kappa(\mathbf{e}')$ is a shortest path between x and y in S. $\square$

We define a weight function $\boldsymbol{\omega} : \mathcal{V} \cup \mathcal{E} \cup \mathcal{E}' \to \mathbb{R}$ where $\boldsymbol{\omega}(\mathbf{v}) = \mathrm{diam}(\kappa(\mathbf{v}))$ for each vertex $\mathbf{v} \in \mathcal{V}$ and $\boldsymbol{\omega}(\mathbf{e}) = w(\kappa(\mathbf{e}))$ for each edge $\mathbf{e} \in \mathcal{E} \cup \mathcal{E}'$. Let $\mathcal{P}$ be a path of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$. We define $\mathcal{P}$'s weight, denoted by $\boldsymbol{\omega}(\mathcal{P})$, to be its total vertex and edge weights.

Recall that high-degree ϵ-cluster is incident to at least $\frac{3g}{\epsilon}$ Π-paths. We call the corresponding vertex $\kappa^{-1}(X) \in \mathcal{V}$ of a high-degree ϵ-cluster X a *high-degree vertex*. Instead of constructing level-i clusters directly, we will construct a set of connected subgraphs Γ of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$. Each subgraph $\mathcal{S} \in \Gamma$ will then define a level-i cluster $\kappa(\mathcal{S})$. The construction proceeds in four phases. The construction is similar to the cluster construction in Section 2.2.

Phase 1: High-degree vertices This phase has three steps. The main purpose is to guarantee that every high-degree vertex and its Π-neighbors are grouped into subgraphs. Initially, every vertex of $\mathcal{V}$ is unmarked.

(Step 1) Let $\mathbf{x} \in \mathcal{V}$ be a high-degree vertex such that all of its Π-neighbors are unmarked. We form a new subgraph $\mathcal{S}$ from $\mathbf{x}$, its Π-neighbors and the connecting Π-edges. We then mark every vertex of $\mathcal{S}$, add $\mathcal{S}$ to Γ and repeat this step.

(Step 2) For each unmarked high-degree vertex $\mathbf{y}$, there must be a Π-neighbor, say $\mathbf{z}$ that is marked in Step 1. Let $\mathcal{S} \in \Gamma$ be the subgraph formed in Step 1 that contains $\mathbf{z}$. We augment $\mathcal{S}$ by $\mathbf{y}$, its unmarked Π-neighbors and the connecting Π-edges. We then mark $\mathbf{y}$, its Π-neighbors and repeat this step.

(Step 3) Let $\mathbf{y}'$ be an unmarked low-degree vertex that has a high-degree Π-neighbor $\mathbf{z}'$. By construction in Step 2, $\mathbf{z}'$ must be marked in Step 1. Let $\mathcal{S}$ be the subgraph in Γ

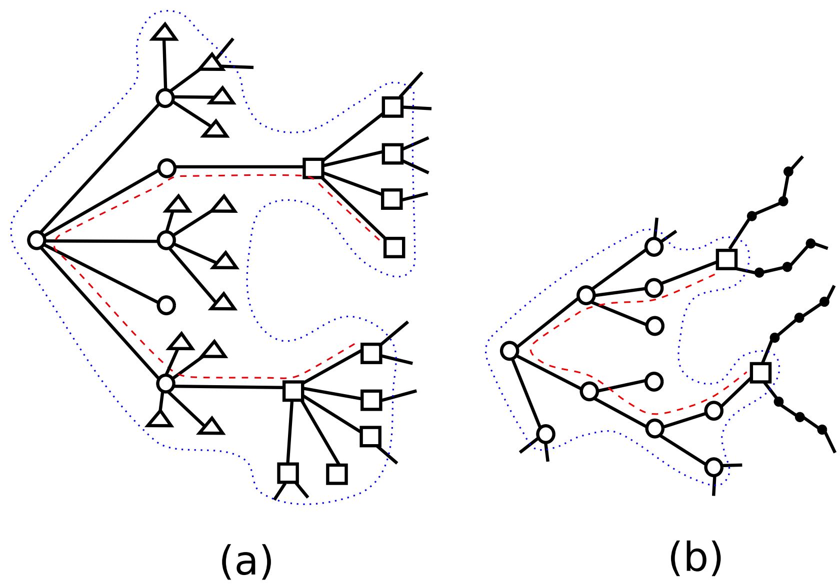

Figure 2.10: (a) A Phase-1 subgraph in Γ is enclosed in dotted blue curve. Round vertices, square vertices and triangular vertices are grouped in Step 1, Step 2 and Step 3, respectively. The diameter path $\mathcal{D}$ is highlighted by a dashed red curve. (b) A Phase-2 subgraph in Γ is enclosed by the dotted blue curve. Round vertices are grouped in Step 1 and square vertices are added in Step 2. The diameter path $\mathcal{D}$ of $\mathcal{T}$ is highlighted by a dashed red curve.

that contains $\mathbf{z}'$. We augment $\mathcal{S}$ by $\mathbf{y}'$ and its incident Π-edge between $\mathbf{y}'$ and $\mathbf{z}'$.

See Figure 2.10(a) for an illustration of subgraphs constructed in Phase 1.

Phase 2: Low-degree, branching vertices Let $\mathcal{F}$ be the forest of $\mathcal{ST}(\mathcal{V}, \mathcal{E})$ obtained by removing vertices marked in Phase 1. We say a vertex $\mathbf{v}$ $\mathcal{F}$-*branching* if it has degree at least 3 in $\mathcal{F}$. Let $\mathcal{P}$ be a path of $\mathcal{F}$. We define *effective diameter* of $\mathcal{P}$ to be the total vertex weight of $\mathcal{P}$. We then define effective diameter of a subtree of $\mathcal{F}$ to be the maximum effective diameter over all paths of the tree. This phase has two steps. The purpose is to group every $\mathcal{F}$-branching vertices of high-diameter trees into clusters. By construction in Phase 1, vertices in this phase are low-degree.

(Step 1) Let $\mathcal{T}$ be a minimal subtree of $\mathcal{F}$ of effective diameter at least 2ℓ and at most 4ℓ that has a $\mathcal{T}$-branching vertex, say $\mathbf{x}$. We add $\mathcal{T}$ to Γ, remove it from $\mathcal{F}$ and repeat. After Step 1, every component of $\mathcal{F}$ is a tree of effective diameter most 2ℓ or is a path of effective diameter at least 4ℓ.

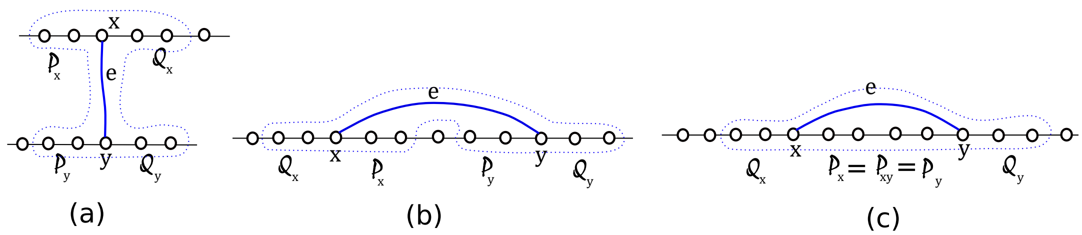

Figure 2.11: Subgraphs in Phase 3 are edges and vertices enclosed in dotted blue curves. There are three different forms that a subgraph in Phase 3 can take. The blue thick edge is Π-edge $\mathbf{e}$ with two endpoints $\mathbf{x}, \mathbf{y}$.

(Step 2) We call a path of $\mathcal{F}$ *high-diameter* if it has effective diameter at least 4ℓ. Let $\mathbf{y}$ be a vertex in a high-diameter path which is $\mathcal{F}$-branching before Step 1. That is, all but at most two neighbors of $\mathbf{y}$ in $\mathcal{F}$ are removed from $\mathcal{F}$ in Step 1. Let $\mathbf{z}$ be a removed neighbor of $\mathbf{y}$ and $\mathbf{e} \in \mathcal{E}$ be the ST edge between $\mathbf{y}$ and $\mathbf{z}$. Let $\mathcal{T}$ be the tree in Step 1 that contains $\mathbf{z}$. We augment $\mathcal{T}$ with $\mathbf{y}$ and $\mathbf{e}$. We then remove $\mathbf{y}$ from $\mathcal{F}$ and repeat. (See Figure 2.10(b).)

Phase 3: High-diameter paths of $\mathcal{F}$ We say a vertex $\mathbf{v}$ in a high-diameter path $\mathcal{P}$ *deep* if it is not an endpoint of $\mathcal{P}$ and the two subpaths of $\mathcal{P} - \{\mathbf{v}\}$ each has effective diameter at least 2ℓ. Let $\mathbf{e}$ be a Π-edge with two endpoints, say $\mathbf{x}, \mathbf{y}$, that are deep vertices. Let $\mathcal{X}, \mathcal{Y}$ be two cluster paths of $\mathcal{F}$ that contain $\mathbf{x}, \mathbf{y}$, respectively. It may be that $\mathcal{X} \equiv \mathcal{Y}$. Let $\mathcal{P}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{x}}$ be two minimal subpaths of $\mathcal{X} - \{\mathbf{x}\}$ incident to $\mathbf{x}$ that have effective diameter at least 2ℓ. $\mathcal{P}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{x}}$ exist since $\mathbf{x}$ is deep. We define two minimal subpaths $\mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{y}}$ of $\mathcal{Y}$ similarly. We then group $\mathbf{e}$, $\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{y}}$ into a new subgraph of Γ. We then remove $\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{y}}, \mathbf{x}, \mathbf{y}$ from $\mathcal{X} \cup \mathcal{Y}$ and update the set of high-diameter paths of $\mathcal{F}$. We repeat until this phase no longer applies.

See Figure 2.11 for an illustration of subgraphs formed in this step. Note that there could be two paths in $\{\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{y}}\}$, say $\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}$, that are overlap. In this case, we redefine $\mathcal{P}_{\mathbf{x}} = \mathcal{P}_{\mathbf{y}} = \mathcal{P}_{\mathbf{xy}}$ where $\mathcal{P}_{\mathbf{xy}} = \mathcal{P}[\mathbf{x}, \mathbf{y}]$.

Phase 4: Remaining high-diameter paths of $\mathcal{F}$ Let $\mathcal{P}$ be a high-diameter path of $\mathcal{F}$ after Phase 3. We break $\mathcal{P}$ into segments of effective diameter at least 2ℓ and at most 4ℓ. Let $\mathcal{X}$ be a segment of $\mathcal{P}$. If $\mathcal{X}$ has an ST edge to an existing subgraph in Γ (formed in previous phases), we defer the processing of $\mathcal{X}$ to Phase 5. (By construction in Step 2 of Phase 2, $\mathcal{X}$ must be an affix of $\mathcal{P}$.) Otherwise, we form a new subgraph of Γ from $\mathcal{X}$.

Phase 5: Remaining low-diameter trees of $\mathcal{F}$ Remaining components of $\mathcal{F}$ are trees (and paths) of effective diameter at most 4ℓ. Since $\mathcal{ST}(\mathcal{V},\mathcal{E})$ is a spanning tree of $\mathcal{G}(\mathcal{V},\mathcal{E}\cup\mathcal{E}')$, each tree in $\mathcal{F}$, say $\mathcal{T}$, must has at least one ST edge, say $\mathbf{e}$, to an existing subgraph in Γ, say $\mathcal{S}$, that is originated in the first three phases. We augment $\mathcal{F}$ with $\mathcal{T}$ and $\mathbf{e}$. We apply the augmentation to every tree of $\mathcal{F}$.

This completes the construction of Γ.

2.3.4.4 Maintaining cluster invariants and paying for spanner edges

In this section, we show how to maintain invariants (I1)-I(3) for level-i clusters constructed in the previous section. Recall that there are two types of Π-edges in $\mathcal{E}'$: (first type) a Π-edge $\mathbf{e}$ where $\kappa(\mathbf{e})$ is a Π-path incident to a low-degree ϵ-cluster and (second type) a Π-edge $\mathbf{e}$ where $\kappa(\mathbf{e})$ is a shortest path (in K) that approximates a Π-path incident to two high-degree ϵ-clusters. The only difference is that first type Π-edges have length at most ℓ while the second type Π-edges have length at most $(1+s\epsilon)\ell$. Since the second type of Π-path is only involved in Phase 1, we abuse notation by saying Π-edges to refer to first type Π-edges. When there is possible confusion, we would clearly indicate which type of Π-edge we are referring to.

The observation below allows us to work with subgraphs in Γ instead of level-i clusters. The reason is that subgraphs in Γ are vertex-disjoint, that makes the amortized argument easier. Recall weight of a path $\mathcal{P}$ in $\mathcal{G}(\mathcal{V},\mathcal{E}\cup\mathcal{E}')$ is the total weight of vertices and edges of $\mathcal{P}$.

Observation 2.3.21. *Let $\mathcal{S}$ be a subgraph in Γ and D be a diameter path of $\kappa(\mathcal{S})$. Let $\mathbf{x}$ and $\mathbf{y}$ be two vertices of $\mathcal{S}$ such that the corresponding ϵ-clusters $\kappa(\mathbf{x})$ and $\kappa(\mathbf{y})$ contain the two endpoints of D. Let $\mathcal{P}$ be a shortest $\mathbf{x}$-to-$\mathbf{y}$ path in $\mathcal{S}$. Then, $w(D) \leq \boldsymbol{\omega}(\mathcal{P})$.*

Proof. Let x, y be D's endpoints where $x \in \kappa(\mathbf{x})$ and $y \in \kappa(\mathbf{y})$. Write

$$\mathcal{P} = \{\mathbf{x}_1, \mathbf{e}_1, \mathbf{x}_2, \mathbf{e}_2, \ldots, \mathbf{e}_{q-1}, \mathbf{x}_q\}$$

where $\mathbf{x} = \mathbf{x}_1$, $\mathbf{y} = \mathbf{x}_q$ and $\mathbf{e}_j$ is the edge between $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ in $\mathcal{S}$, $1 \leq j \leq q-1$.

For each j, $2 \leq j \leq q-1$, we define P_j to be a shortest path in $\kappa(\mathbf{x}_j)$ between the endpoints of $\kappa(\mathbf{e}_{j-1})$ and $\kappa(\mathbf{e}_j)$ in $\kappa(\mathbf{x}_j)$. Let P_1 (P_q) be the shortest path in $\kappa(\mathbf{x})$ ($\kappa(\mathbf{y})$) between x (y) and the endpoint of $\kappa(\mathbf{e}_1)$ ($\kappa(\mathbf{e}_{q-1})$) in $\kappa(\mathbf{x})$ ($\kappa(\mathbf{y})$). Let W be the walk:

$$W \stackrel{\text{def}}{=} P_1 \circ \kappa(\mathbf{e}_1) \circ P_2 \circ \kappa(\mathbf{e}_2) \circ \ldots \circ \kappa(\mathbf{e}_{p-1}) \circ P_q \tag{2.26}$$

We call W an x-to-y walk *tracing* $\mathcal{P}$. By Equation (2.26), we have $w(W) \leq \boldsymbol{\omega}(\mathcal{P})$. Since $w(D) \leq w(W)$, $w(D) \leq \boldsymbol{\omega}(\mathcal{P})$. □

Claim 2.3.22. *Let* $\mathbf{e}$ *be a first type* Π*-edge between two vertices* $\mathbf{x}, \mathbf{y}$ *and* $\mathcal{P}$ *be the* $\mathbf{x}$*-to-*$\mathbf{y}$ *path in* $\mathcal{ST}(\mathcal{V}, \mathcal{E})$*. Then,* $\boldsymbol{\omega}(\mathcal{P}) \geq (1 + s \cdot \epsilon)\ell(\mathbf{e})$*.*

Proof. Suppose otherwise. Let x, y be $\kappa(\mathbf{e})$'s endpoints in S and W be an x-to-y walk tracing $\mathcal{P}$. Since $\kappa(\mathcal{ST})$ is a subgraph of S, W is a walk in S. Since $w(W) \leq \boldsymbol{\omega}(\mathcal{P})$, we have:

$$w(W) \leq (1 + s \cdot \epsilon)\, \boldsymbol{\omega}(\mathbf{e}) = (1 + s \cdot \epsilon) w(\kappa(\mathbf{e}))$$

However, that implies there is already an $(1+\epsilon)$-stretch path between the two endpoints of $\kappa(\mathbf{e})$ and hence Π-path $\kappa(\mathbf{e})$ is removed in the Π-path removal process. □

Invariant (I1)

Observation 2.3.21 allows us to derive an upper bound on the diameter of level-i clusters by upper-bounding the diameter of subgraphs in Γ with weight function $\boldsymbol{\omega}(.)$. Recall each ST edge has length at most w_0. we use $\mathcal{S}$ to denote a subgraph of Γ that is initiated before Phase 5 and $\mathcal{S}'$ to denote the augmentation of $\mathcal{S}$ after Phase 5. Since in Phase 5, we only augment existing subgraphs in Γ (originating in Phase 1,2 and 3) by attaching trees of diameter at most 4ℓ via ST edges, we have:

Observation 2.3.23. $\mathrm{diam}(\mathcal{S}') \leq \mathrm{diam}(\mathcal{S}) + 8\ell + 2w_0$.

Observation 2.3.24. *For every path* $\mathcal{P}$ *in* $\mathcal{ST}(\mathcal{V}, \mathcal{E})$*,* $\mathrm{diam}(\mathcal{P}) \leq 2\mathrm{ediam}(\mathcal{P})$*.*

Proof. $\mathcal{P}$ contains ST-edges only and each ST-edge has weight at most w_0 while each vertex in $\mathcal{P}$ has weight at least $\ell/2$. Recall $\ell/2 = \frac{2^{j-1}}{\epsilon^i} w_0 \geq w_0$ since $i, j \geq 1$. □

Lemma 2.3.25. *Level-i clusters have diameter at most* 125ℓ *when* ϵ *is sufficiently smaller than* $1/g$*.*

Proof. We consider four cases:

Case 1: $\mathcal{S}$ is formed in Phase 1 Observe from the construction of Phase 1 that $\mathcal{S}$ is a tree whose edges are Π-edge. In Step 1, $\mathcal{S}$ is a star centered at a vertex $\mathbf{x}$ and its incident edges are Π-edges. In Step 2, $\mathcal{S}$ is further augmented by attaching high-degree vertices and their unmarked Π-neighbors via Π-edges. In Step 3, if a low-degree vertex $\mathbf{y}$ and its incident Π-edge is added to $\mathcal{S}$ via its neighbor $\mathbf{z}$, $\mathbf{z}$ is also a neighbor of $\mathbf{x}$. Thus, any diameter path of $\mathcal{S}$ in Phase 1 has at most 7 vertices and 6 Π-edges. By invariant (I1) for level $i-1$, each vertex has weight at most $g\epsilon\ell$. Since all 6 Π-edges could be of second type, we have:

$$\mathrm{diam}(C) \leq 6(1+s\epsilon)\ell + 7g\epsilon\ell = 6(1+(16g+1)\epsilon)\ell + 7g\epsilon\ell \leq 12\ell + 103g\epsilon\ell$$

Case 2: $\mathcal{S}$ is formed in Phase 2 By Observation 2.3.24, subgraphs in Step 1 have diameter at most 8ℓ. By construction in Step 2, subgraphs in Step 1 are augmented by attaching vertices via ST edges. Thus, the augmentation in Step 2 blows up the diameter by at most $2w_0 + 2g\epsilon\ell$ which is at most $2\ell + 2g\epsilon\ell$. Hence, $\mathrm{diam}(S) \leq 10\ell + 2g\epsilon\ell$.

Case 3: $\mathcal{S}$ is formed in Phase 3 Let $\mathcal{P}'_{\mathbf{x}}$ ($\mathcal{P}'_{\mathbf{y}}$) be a minimal segment of $\mathcal{X}$ ($\mathcal{Y}$) that spans $\mathcal{P}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{x}}$ and $\mathbf{x}$ ($\mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{y}}$ and $\mathbf{y}$). By minimality, segments in $\{\mathcal{P}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{y}}\}$ have effective diameter at most $(2+g\epsilon)\ell$. Thus, the effective diameter of $\mathcal{P}'_{\mathbf{x}}$ and $\mathcal{P}'_{\mathbf{y}}$ are at most $(4+3g\epsilon)\ell$. Since $\mathrm{diam}(\mathcal{S}) \leq \mathrm{diam}(\mathcal{P}'_{\mathbf{x}}) + \mathrm{diam}(\mathcal{P}'_{\mathbf{y}}) + \ell$, by Observation 2.3.24, we have:

$$\mathrm{diam}(\mathcal{S}) \leq 2(4+3g\epsilon)\ell + \ell = 9\ell + 6\epsilon\ell.$$

Case 4: $\mathcal{S}$ is formed in Phase 4 By construction, $\mathcal{S}$ has effective diameter at most 4ℓ. By Observation 2.3.24, $\mathrm{diam}(\mathcal{S}) \leq 8\ell$.

Thus, in any case, $\mathrm{diam}(\mathcal{S}) \leq 12\ell + 103\epsilon g\ell$. By Observation 2.3.23, $\mathrm{diam}(\mathcal{S}') \leq 20\ell + 2w_0 + 103\epsilon g\ell$ which is at most 125ℓ since $\epsilon < 1/g$ and $w_0 \leq \ell$. By Observation 2.3.21, level-i cluster $\kappa(\mathcal{S}')$ has diameter at most 125ℓ. $\square$

Since $g = 125$, invariant (I1) is satisfied.

Invariant (I2) and Invariant (I3)

We would argue that credits of vertices and edges inside subgraphs of Γ are sufficient to both maintain invariant (I2) and pay for spanner edges added in this level. That is, we would not use credits of ST edges in $\mathcal{S}(\mathcal{V},\mathcal{E}) \setminus \Gamma$. Recall Γ is a collection of connected, vertex-disjoint subgraphs of $\mathcal{G}(\mathcal{V},\mathcal{E}\cup\mathcal{E}')$. Thus, by contracting each subgraph in Γ into a vertex, we obtain a multigraph $\mathcal{G}'$ from $\mathcal{G}(\mathcal{V},\mathcal{E}\cup\mathcal{E}')$. Since $\mathcal{ST}(\mathcal{V},\mathcal{E})$ is a spanning tree of $\mathcal{G}(\mathcal{V},\mathcal{E}\cup\mathcal{E}')$, there is a spanning tree, say $\mathcal{ST}_i(\mathcal{V}_i,\mathcal{E}_i)$, of $\mathcal{G}'$ that contains only ST edges. Since we never use credits of ST edges outside subgraphs in Γ to maintain invariant (I2), $\mathcal{ST}_i(\mathcal{V}_i,\mathcal{E}_i)$ satisfies invariant (I3).

Recall for each low-degree ϵ-cluster, we add all of its incident Π-paths to the spanner. There are at most $\frac{3g}{\epsilon}$ such paths incident to each low-degree ϵ-cluster. For high-degree ϵ-clusters, we construct a 3ℓ-close spanner K and argue that (Claim 2.3.19) for each Π-path incident to both high-degree ϵ-clusters, there is an $(1+s\cdot\epsilon)$-approximate shortest path for Q in K. The weight of K is given in Equation (2.21).

Let $\mathcal{S}$ be a subgraph of $\mathcal{G}(\mathcal{V},\mathcal{E})$ created before Phase 5 and $\mathcal{S}'$ be the augmentation of $\mathcal{S}$ after Phase 5. Let $\mathcal{D}'$ be the diameter path of $\mathcal{S}'$. By construction, $\mathcal{S}$ is augmented by attaching trees via ST edges. Thus, $\mathcal{D}'\cap\mathcal{S}$ has only one connected component which is a path. Let $\mathcal{D}=\mathcal{D}'\cap\mathcal{S}$.

Recall each ϵ-cluster, say X, has at least $c(\epsilon)\max(\mathrm{diam}(X),\epsilon\ell/2)$ by invariant (I2) for level $i-1$. We assign credits of X to its corresponding vertex $\kappa^{-1}(X)\in\mathcal{V}$. Since $\boldsymbol{\omega}(\kappa^{-1}(X))=\mathrm{diam}(X)$, each vertex in $\mathcal{V}$ has credit at least $c(\epsilon)$ times its weight. Recall the effective diameter of a path $\mathcal{P}$ is the total weight of its vertices. Thus, we have:

Observation 2.3.26. *If $\mathcal{P}$ is a path in $\mathcal{ST}(\mathcal{V},\mathcal{E})$, then total credit of its vertices is at least $c(\epsilon)\cdot\mathrm{ediam}(\mathcal{P})$.*

Recall each ST edge, say e, has credit at least $c(\epsilon)$ times its length. We assign credit of e to its corresponding edge $\kappa^{-1}(e)\in\mathcal{E}$. Thus, to maintain invariant (I2), by Observation 2.3.21, it suffices to guarantee that $\mathcal{S}'$ is assigned credits of value at least:

$$c(\epsilon)\max(\boldsymbol{\omega}(\mathcal{D}'),\ell/2)$$

Subgraphs originating in Phase 4 By construction in Phase 4, $\mathcal{S}$ is a path whose edges are ST edges. $\mathcal{S}$ would not be augmented in Phase 5. Thus, $\mathcal{S}=\mathcal{S}'$. We say $\mathcal{S}$

long if it has at least $\frac{2g}{\epsilon}+1$ vertices and *short* otherwise. We first consider the case when $\mathcal{S}$ is a long path.

Claim 2.3.27. *A long path can both maintain invariant (I2) and pay for its incident* Π*-edges when* $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$*.*

Proof. Recall that every vertex in $\mathcal{S}$ has at least $\Omega(c(\epsilon)\epsilon\ell/2)$ credits and is incident to at most $3g/\epsilon$ Π-edges, each of weight at most ℓ. Let $\mathcal{X}$ be a set of any $\frac{2g}{\epsilon}$ vertices of $\mathcal{S}$ and $\mathbf{v}$ be a vertex in $\mathcal{S} \setminus \mathcal{X}$. Total credit of vertices in $\mathcal{X}$ is at least $\frac{2g}{\epsilon}c(\epsilon)\epsilon\ell/2 = gc(\epsilon)\ell$ which is at least $c(\epsilon)\cdot \max(\mathrm{diam}(\mathcal{S}), \ell/2)$ since $\mathrm{diam}(\mathcal{S}) \leq g\ell$. Thus, we can use credits of $\mathcal{X}$ to maintain invariant (I2) for $\mathcal{S}$.

We use credit of $\mathbf{v}$ to pay for Π-edges incident to vertices in $\mathcal{X}\cup\{\mathbf{v}\}$. Since vertices involved in Phase 4 are low-degree, there are at most:

$$(\frac{2g}{\epsilon}+1)\frac{3g}{\epsilon} = O(\frac{g^2}{\epsilon^2})$$

such Π-edges. Credit of $\mathbf{v}$ (of value at least $c(\epsilon)\epsilon\ell/2$) is sufficient when $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$.

For every other vertex $\mathbf{x} \in \mathcal{S} \setminus (\mathcal{X}\cup\{\mathbf{v}\})$, we use its credit to pay for its incident Π-edges. Since $\mathbf{x}$ is incident to at most $\frac{3g}{\epsilon}$ Π-edges and has at least $c(\epsilon)\epsilon\ell/2$ credits, $\mathbf{x}$'s credit is sufficient when $c(\epsilon) = \Omega(\frac{g}{\epsilon^2})$. $\square$

We now consider the case when $\mathcal{S}$ is a short path.

Claim 2.3.28. *A short path can maintain invariant (I2) using credits of its vertices and* ST *edges.*

Proof. Since $\mathrm{ediam}(\mathcal{S}) \geq 2\ell$, by Observation 2.3.26, its total vertex credit is at least $c(\epsilon)2\ell$. Since $\mathrm{diam}(\mathcal{S}) \leq \sum_{\mathbf{v}\in\mathcal{S}} \boldsymbol{\omega}(\mathbf{v}) + \sum_{\mathbf{e}\in\mathcal{S}} \boldsymbol{\omega}(\mathbf{e})$ and each vertex or ST edge has more credits than its weight, the total vertex and edge credit of $\mathcal{S}$ is at least $c(\epsilon)\cdot \max(\mathrm{diam}(\mathcal{S}), \ell/2)$. $\square$

By Claim 2.3.28, credits of a short path are only sufficient for maintaining (I2). We say a short path $\mathcal{S}$ *internal* if it is not an affix of a high-diameter path $\mathcal{P}$ in Phase 4.

Observation 2.3.29. *There is no* Π*-edge that has both endpoints in internally short subpaths.*

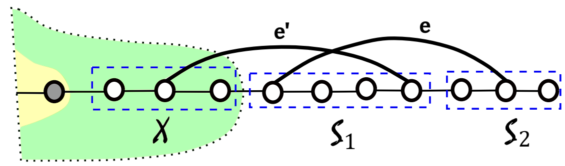

Figure 2.12: Hollow vertices are in a cluster path $\mathcal{P}$ that is broken into three subpaths (enclosed by dashed blue squares) in Phase 4. $\mathcal{X}$ and $\mathcal{S}_2$ are affices of $\mathcal{P}$ and $\mathcal{S}_1$ is an internally short subpath of $\mathcal{P}$. Two short subpaths S_1 and S_2 are grouped into two subgraphs of Γ while $\mathcal{X}$ is augmented to an existing subgraph of Γ in Phase 5. Two Π-edges $\mathbf{e}$ and $\mathbf{e}'$ are incident to $\mathcal{S}_1$ whose vertex and edge credits are only sufficient to maintain invariant (I2). To pay for $\mathbf{e}'$, we use credits of $\mathbf{e}'$'s endpoint in $\mathcal{X}$. However, to pay for $\mathbf{e}$, we need a different way since $\mathcal{S}_2$ is also short. $\mathcal{X}$ and $\mathcal{S}_2$ are siblings of each other.

Proof. If there is such an edge $\mathbf{e}$, both of its endpoints are deep and thus, would be in Phase-3 subgraphs. $\square$

Intuitively, Observation 2.3.29 said that for each Π-edge $\mathbf{e}$ that is incident to an internally short subpath, we can use credits of another endpoint of $\mathbf{e}$, say $\mathbf{v}$, to pay for $\mathbf{e}$. However, if $\mathbf{v}$ is in an affix short subpath, say $\mathcal{S}$, then we need a different way to pay for $\mathbf{e}$ (see Figure 2.12).

Observation 2.3.30. *Let $\mathcal{P}$ be a high-diameter path after Phase 3. Then at least one affix of $\mathcal{P}$ is deferred to Phase 5 for augmentation, except when $\Gamma = \emptyset$ after Phase 3.*

Proof. Assume that $\Gamma \neq \emptyset$ after Phase 3. Recall that $\mathcal{ST}(\mathcal{V}, \mathcal{E})$ is a spanning tree of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$ and by construction in Step 2 of Phase 2, every vertex of $\mathcal{P}$ has degree 2 in $\mathcal{ST}(\mathcal{V}, \mathcal{E})$. Thus, $\mathcal{P}$ must has an ST edge connecting one of its endpoint, say $\mathbf{v}$, to a subgraph formed in previous phases (there must be at least one such subgraph since $\Gamma \neq \emptyset$.). That implies the observation. $\square$

Assume that $\Gamma \neq \emptyset$ after Phase 3 (the case when $\Gamma = \emptyset$ would be handled at the end of this section.). If $\mathcal{S}$ is an affix of a long path $\mathcal{P}$, we would use credits of ϵ-clusters in another affix of $\mathcal{P}$, say $\mathcal{X}$, to pay for Π-edges incident to $\mathcal{S}$. This is possible because by Observation 2.3.30, $\mathcal{X}$ would be merged to other subgraphs in Γ during Phase 5. We say $\mathcal{X}$ is a *sibling affix* of $\mathcal{S}$, and vice versa (see Figure 2.12).

Subgraphs originating in Phase 1 By construction, each Phase-1 subgraph has at least $\frac{3g}{\epsilon}$ vertices. Let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be any two disjoint subsets of vertices of $\mathcal{S}$ such that $|\mathcal{Z}_1| = \frac{2g}{\epsilon}, |\mathcal{Z}_2| = \frac{g}{\epsilon}$. Recall each vertex has at least $c(\epsilon)\epsilon\ell/2$ credits. Thus, the total credit of vertices in $\mathcal{Z}_1$ is at least $\frac{2g}{\epsilon}(c(\epsilon)\epsilon\ell/2) = c(\epsilon)g\ell$ which is at least $c(\epsilon)\max(\mathrm{diam}(\mathcal{S}'), \ell/2)$ since $\mathrm{diam}(\mathcal{S}') \leq g\ell$ by invariant (I1). We can use vertex credit of $\mathcal{Z}_1$ to maintain invariant (I2) of $\mathcal{S}'$. We then redistribute vertex credit of $\mathcal{Z}_2$ to every vertex in $\mathcal{Z}_1 \cup \mathcal{Z}_2$. On average, each vertex has at least $(\frac{g}{\epsilon}c(\epsilon)\epsilon\ell/2)/(\frac{3g}{\epsilon}) = c(\epsilon)\epsilon\ell/6$ credits.

Note that other vertices in $\mathcal{S}' \setminus (\mathcal{Z}_1 \cup \mathcal{Z}_2)$ have $c(\epsilon)\epsilon\ell/2$ credits each. Thus, after maintaining invariant (I2), every vertex in Phase-1 subgraphs has at least $c(\epsilon)\epsilon\ell/6$ credits left. Let $\mathbf{x}$ be a low-degree vertex of $\mathcal{S}'$. We consider two cases:

1. If $\mathbf{x}$ is in an affix, say $\mathcal{X}$, of a long cluster path of Phase 4 (that is added to $\mathcal{S}'$ in Phase 5), we use $\mathbf{x}$'s remaining credits to pay for its incident Π-edges and Π-edges incident to vertices of $\mathcal{X}$'s sibling short affix, if any. $\mathcal{X}$ has at most $\frac{2g}{\epsilon}$ vertices since it is short and every vertex of $\mathcal{X}$ is low-degree. Thus, the total number of Π-edges paid by $\mathbf{x}$ is at most $\frac{3\mathrm{g}}{\epsilon} + (\frac{2g}{\epsilon})\frac{3\mathrm{g}}{\epsilon} = O(\frac{g^2}{\epsilon^2})$. Credit of $\mathbf{x}$ is sufficient if $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$.
2. Otherwise, we use $\mathbf{x}$'s remaining credits to pay for its incident Π-edges only. Since $\mathbf{x}$ is incident to at most $\frac{3\mathrm{g}}{\epsilon}$ Π-edges, its credit is sufficient when $c(\epsilon) = \Omega(\frac{g}{\epsilon^2})$.

Let $\mathcal{C}$ be the set of high-degree vertices of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$. The total remaining vertex credit of $\mathcal{C}$ is at least $|\mathcal{C}|c(\epsilon)\ell/6$. Recall in the spanner construction step, we construct a 3ℓ-close spanner K to approximate Π-paths incident to both high-degree ϵ-clusters. By Equation (2.21), remaining vertex credit of $\mathcal{C}$ is sufficient to pay for K when $c(\epsilon) = \Omega(\mathrm{poly}(\frac{1}{\epsilon})\,\tau(\epsilon, k, n))$. Note that $|\mathcal{C}| = |\mathcal{C}_\epsilon|$. Thus, we have:

Claim 2.3.31. *If $c(\epsilon) = \Omega(\mathrm{poly}(\frac{1}{\epsilon})\,\tau(\epsilon, k, n))$, subgraphs originated in Phase 1 can maintain invariant (I2) and pay for (i) edges of 3ℓ-close spanner K, (ii) their incident Π-edges and (iii) Π-edges incident to short affix subpaths whose sibling affices are augmented to Phase-1 subgraphs in Phase 5.*

Subgraphs originating in Phase 2+3 We first define a canonical pair of $\mathcal{S}'$ that reflects how we use credits to maintain invariant (I2) and pay for spanner edges.

Definition 2.3.32 (Canonical pair)**.** *Let $\mathcal{Z}$ be a subset of vertices of $\mathcal{S} \cup \mathcal{D}'$ of size at most $\frac{2g}{\epsilon}$ and $\mathbf{z}$ is a vertex in $\mathcal{S} \setminus \mathcal{Z}$. If credits of vertices in $\mathcal{Z}$ and* ST *edges in $\mathcal{S}'$ are sufficient to maintain invariant (I2) for $\mathcal{S}'$, we call $(\mathcal{Z}, \mathbf{z})$ a* canonical pair *of $\mathcal{S}'$.*

We show that maintaining invariant (I2) and paying for spanner edges can be reduced to showing the existence of a canonical pair.

Claim 2.3.33. *If $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$ and $\mathcal{S}'$ has a canonical pair $(\mathcal{Z}, \mathbf{z})$, then vertex credit of $\mathcal{S}' \setminus \mathcal{Z}$ is sufficient to pay for all* Π*-edges incident to vertices of $\mathcal{S}'$ and* Π*-edges of short affices that have sibling affices in $\mathcal{S}'$.*

Proof. See Figure 2.13 for an illustration. Recall $\mathbf{z}$ has at least $c(\epsilon)\epsilon\ell/2$ credits. Since $\mathcal{S}$ is augmented in Phase 5 by attaching trees of $\mathcal{F}$ via ST edges, there are at most two short affices, say $\mathcal{Z}_1, \mathcal{Z}_2$, in Phase 4 whose sibling affices in $\mathcal{S}'$, say $\mathcal{Z}'_1, \mathcal{Z}'_2$, respectively, both have vertices in diameter path $\mathcal{D}'$. Let $\mathcal{R} = \mathcal{Z} \cup \mathcal{Z}_1 \cup \mathcal{Z}_2 \cup \{\mathbf{z}\}$. Since $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are short and $\mathcal{Z}$ has at most $\frac{2g}{\epsilon}$ vertices, there are at most:

$$(1 + 3\frac{2g}{\epsilon})\frac{3\text{g}}{\epsilon} = O(\frac{g^2}{\epsilon^2})$$

Π-edges incident to vertices of $\mathcal{R}$. Thus, credit of $\mathbf{z}$ is sufficient to pay for $\mathcal{R}$'s incident Π-edges when $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^3})$.

Each vertex in $\mathcal{S} \setminus (\mathcal{Z} \cup \{\mathbf{z}\})$ has at least $c(\epsilon)\epsilon\ell/2$ credits that can be used to pay for its incident Π-edges edges. However, we need to pay for Π-edges incident to other short affices in Phase 4 ($\mathcal{Z}_3$ in Figure 2.13 for example). To do this, we only use half the credit of each vertex in $\mathcal{S} \setminus (\mathcal{Z} \cup \{\mathbf{z}\})$ to pay for its Π-edges edges. This is sufficient when $c(\epsilon) = \Omega(\frac{g}{\epsilon^2})$.

Let $\mathcal{Z}_3$ be a short affix cluster in Phase 4, $\mathcal{Z}_3 \notin \{\mathcal{Z}_1, \mathcal{Z}_2\}$, where its sibling affix, say $\mathcal{Z}'_3$, is in $\mathcal{S}'$. There are at most $\frac{2g}{\epsilon} \cdot \frac{3\text{g}}{\epsilon} = O(\frac{g^2}{\epsilon^2})$ Π-edges incident to vertices of $\mathcal{Z}_3$. Since $\mathcal{Z}'_3$ has effective diameter at least 2ℓ, by Observation 2.3.26, half the total credit of its vertices is at least $c(\epsilon)\ell$. This credit is sufficient to pay for Π-edges incident to vertices of $\mathcal{Z}_3$ if $c(\epsilon) = \Omega(\frac{g^2}{\epsilon^2})$. □

By Claim 2.3.33, it remains to show that canonical pairs exist for subgraphs formed in Phase 2+3.

Claim 2.3.34. *If $\mathcal{S} \cup \mathcal{D}'$ has at least $\frac{2g}{\epsilon} + 1$ vertices, then $\mathcal{S}'$ has a canonical pair.*

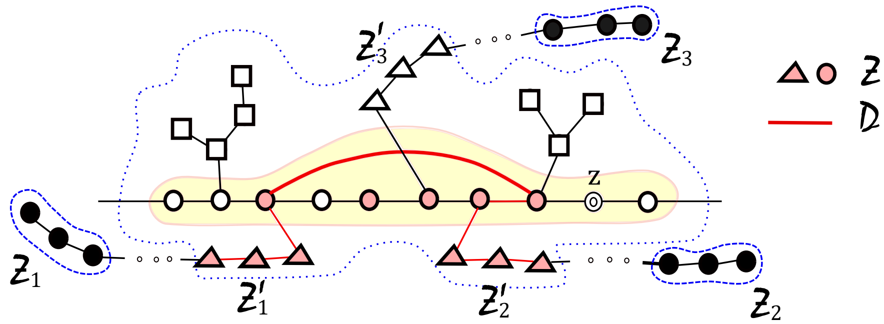

Figure 2.13: A subgraph $\mathcal{S}$ formed in Phase 3 (round vertices) augmented in Phase 5 (square and triangular vertices) is enclosed the dotted blue curve. Triangular vertices are in short affices in Phase 4 that are deferred to Phase 5. Dashed blue curves enclose short affix subgraphs of Γ in Phase 4 ($\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3$) whose sibling affices ($\mathcal{Z}'_1, \mathcal{Z}'_2, \mathcal{Z}'_3$) are augmented to $\mathcal{S}$ in Phase 5. $\mathcal{Z}$ contains red-filled vertices. Vertex $\mathbf{z}$ is marked by double cycles. The red path is the diameter path $\mathcal{D}'$ of $\mathcal{S}'$.

Proof. Let $\mathcal{Z}$ be any subset of $\frac{2g}{\epsilon}$ vertices of $\mathcal{S} \cup \mathcal{D}'$ and $\mathbf{z}$ be a vertex in $(\mathcal{S} \cup \mathcal{D}') \setminus \mathcal{Z}$. Since the total vertex credit of $\mathcal{Z}$ is at least $\frac{2g}{\epsilon}c(\epsilon)\epsilon\ell/2 = c(\epsilon)g\ell$ which is at least $c(\epsilon)\max(\mathrm{diam}(S'), \ell/2)$ by invariant (I1), $\mathcal{Z}$'s credit is sufficient to maintain invariant (I2). Hence, $(\mathcal{Z}, \mathbf{z})$ is a canonical pair of $\mathcal{S}'$. $\square$

By Claim 2.3.34, we can assume that $|\mathcal{S} \cup \mathcal{D}'| \leq \frac{2g}{\epsilon}$. We consider subgraphs formed in Phase 2 first.

Claim 2.3.35. *If $\mathcal{S}$ is constructed in Phase 2, then $\mathcal{S}'$ has a canonical pair.*

Proof. By construction in Phase 2, $\mathcal{S}$ is a tree and thus, $\mathcal{S}'$ is also a tree whose edges are ST edges only. Since $\mathcal{S}$ has a $\mathcal{S}$-branching vertex $\mathbf{x}$, there must be a neighbor of $\mathbf{x}$, say $\mathbf{z}$, that is not in $\mathcal{D}'$.

Since $\mathcal{D}'$ contains ST edges only, credit of its vertices and edges is at least $c(\epsilon)\,\boldsymbol{\omega}(\mathcal{D}')$. Since $|\mathcal{S} \cup \mathcal{D}'| \leq \frac{2g}{\epsilon}$, $\mathcal{D}'$ has at most $\frac{2g}{\epsilon}$ vertices. Thus, by letting $\mathcal{Z}$ to be the set of all vertices in $\mathcal{D}'$, we obtain a canonical pair for $\mathcal{S}'$. $\square$

Claim 2.3.36. *If $\mathcal{S}$ is constructed in Phase 3, then $\mathcal{S}'$ has a canonical pair.*

Proof. We reuse notation in Phase 3 here. As noted in Phase 3, there are three different forms that $\mathcal{S}$ can have (see Figure 2.11). We first consider the case when four subpaths

$\mathcal{P}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{y}}$ are pairwise disjoint. In this case, $\mathcal{S}$ is an acyclic subgraph of $\mathcal{G}(\mathcal{V}, \mathcal{E} \cup \mathcal{E}')$. That implies at most two subpaths in $\{\mathcal{P}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}, \mathcal{Q}_{\mathbf{y}}\}$ contain vertices of $\mathcal{D}'$. We denote such two paths by $\mathcal{P}_0, \mathcal{P}_1$ and two other paths that do not contain vertices of $\mathcal{D}'$ by $\mathcal{Q}_0, \mathcal{Q}_1$.

Let $\mathbf{z}$ be a vertex in $\mathcal{Q}_1$. Let $\mathcal{Z}$ be the set of vertices of $\mathcal{D}' \cup \mathcal{P}_0 \cup \mathcal{P}_1 \cup \mathcal{Q}_0$. Since $|\mathcal{S} \cup \mathcal{D}'| \leq \frac{2g}{\epsilon}$, $|\mathcal{Z}| \leq \frac{2g}{\epsilon}$. If $\mathcal{D}'$ does not contain $\mathbf{e}$, then credits of vertices and ST edges in $\mathcal{D}'$ are sufficient to maintain invariant (I2). Otherwise, we assign credits of vertices in $\mathcal{Q}_0$ to $\mathbf{e}$. Since $\mathrm{ediam}(\mathcal{Q}_0) \geq 2\ell$, by Observation 2.3.26, the total assigned credit is at least $2c(\epsilon)\ell$ which is at least $c(\epsilon)\,\boldsymbol{\omega}(\mathbf{e})$ (recall $\boldsymbol{\omega}(\mathbf{e}) \leq \ell$.). Thus, credits of vertices in $\mathcal{Z}$ and ST edges in $\mathcal{D}'$ suffice to maintain invariant (I2). Hence, $(\mathcal{Z}, \mathbf{z})$ is a canonical pair of $\mathcal{S}'$.

It remains to consider the case when two of four paths, say $\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}$, are not disjoint. In this case, $\mathcal{P}_{\mathbf{x}} = \mathcal{P}_{\mathbf{y}} = \mathcal{P}[\mathbf{x}, \mathbf{y}]$ (see Figure 2.11(c)). Observe that $\mathcal{P}_{\mathbf{xy}} \cup \{\mathbf{e}\}$ is the only cycle in $\mathcal{S}$. We consider two cases:

(i) If one of two paths $\mathcal{Q}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{y}}$, say $\mathcal{Q}_{\mathbf{y}}$, does not contain any vertex of $\mathcal{D}'$. Let $\mathbf{z}$ by any vertex of $\mathcal{Q}_{\mathbf{y}}$ and $\mathcal{Z}$ be the set of vertices of $(\mathcal{Q}_{\mathbf{x}} \cup \mathcal{Q}_{\mathbf{y}} \cup \mathcal{P}_{\mathbf{xy}} \cup \mathcal{D}') \setminus \{\mathbf{z}\}$. $|\mathcal{Z}| \leq \frac{2g}{\epsilon}$ since $\mathcal{Z} \subseteq \mathcal{S} \cup \mathcal{D}'$. If $\mathcal{D}'$ does not contain $\mathbf{e}$, then its vertex and edge credits suffice to maintain invariant (I2). Thus, $(\mathcal{Z}, \mathbf{z})$ is a canonical pair of $\mathcal{S}'$.

If $\mathcal{D}'$ contains $\mathbf{e}$, we assign credits of $\mathcal{Q}_{\mathbf{y}} \setminus \{\mathbf{z}\}$ to $\mathbf{e}$. Since $\mathrm{ediam}(\mathcal{Q}_{\mathbf{y}}) \geq 2\ell$, $\mathrm{ediam}(\mathcal{Q}_{\mathbf{y}} \setminus \{\mathbf{z}\}) \geq 2\ell - g\epsilon\ell$. By Observation 2.3.26, the total assigned credit is at least $c(\epsilon)2\ell - c(\epsilon)g\epsilon\ell$ which is at least $c(\epsilon)\ell$ since $\epsilon < \frac{1}{g}$. Thus, $\mathbf{e}$ receives at least $c(\epsilon)\,\boldsymbol{\omega}(\mathbf{e})$ credits. Hence, vertex credit $\mathcal{Z}$ and credit of ST edges in $\mathcal{D}'$ suffice to maintain invariant (I2). That implies $(\mathcal{Z}, \mathbf{z})$ is a canonical pair.

(ii) If both $\mathcal{Q}_{\mathbf{x}}, \mathcal{Q}_{\mathbf{y}}$ contain vertices of $\mathcal{D}'$. We first show that $\mathbf{e} \in \mathcal{D}'$. Assume otherwise. Then, we can shortcut $\mathcal{D}'$ through $\mathcal{P}_{\mathbf{xy}}$ at a cost of:

$$\begin{aligned}
&\boldsymbol{\omega}(\mathbf{e}) - \boldsymbol{\omega}(\mathcal{P}_{\mathbf{xy}}) + \boldsymbol{\omega}(\mathbf{x}) + \boldsymbol{\omega}(\mathbf{y}) \\
&\leq \boldsymbol{\omega}(\mathbf{e}) - (1 + s \cdot \epsilon)\,\boldsymbol{\omega}(\mathbf{e}) + \boldsymbol{\omega}(\mathbf{x}) + \boldsymbol{\omega}(\mathbf{y}) \qquad \text{(by Claim 2.3.22)} \\
&\leq -s\epsilon\ell/2 + 2\epsilon g\ell \qquad \text{(since } w(e) \geq \ell/2\text{)} \\
&= (4g - s)\epsilon\ell/2
\end{aligned}$$

which is negative since $s > 4g$. Thus, $\mathbf{e} \in \mathcal{D}'$ and hence, there is a vertex $\mathbf{z} \in \mathcal{P}_{\mathbf{xy}} \setminus \mathcal{D}$. We assign all vertex and edge credits of $\mathcal{P}_{\mathbf{xy}}$, excluding credits of $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$, to $\mathbf{e}$. Since $\boldsymbol{\omega}(\mathcal{P}_{\mathbf{xy}}) \geq (1 + s \cdot \epsilon)\,\boldsymbol{\omega}(\mathbf{e})$, the total credit assigned to $\mathbf{e}$ is at least:

$$\begin{aligned}
&c(\epsilon)(1 + s \cdot \epsilon)\,\boldsymbol{\omega}(\mathbf{e}) - 3c(\epsilon)g\epsilon\ell \\
&\geq c(\epsilon)\,\boldsymbol{\omega}(\mathbf{e}) + s \cdot \epsilon c(\epsilon)\ell/2 - 3c(\epsilon)g\epsilon\ell \qquad (\text{since } w(e) \geq \ell/2) \\
&\geq c(\epsilon)\,\boldsymbol{\omega}(\mathbf{e}) + c(\epsilon)(s - 6g)\epsilon\ell/2 \\
&> c(\epsilon)\,\boldsymbol{\omega}(\mathbf{e}) \qquad (\text{since } s > 6g)
\end{aligned}$$

Thus, credits of vertices in $\mathcal{Z}$ and ST edges in $\mathcal{S}'$ suffice to maintain invariant (I2). Hence, $(\mathcal{Z}, \mathbf{z})$ is a canonical pair. □

We now handling the exception in Observation 2.3.30. That is, $\Gamma = \emptyset$ after Phase 3.

No subgraphs in Phase 1+2+3 Since there are no subgraphs in Phase 1, every vertex is low-degree. Since there are no subgraphs in Phase 2, $\mathcal{ST}(\mathcal{V}, \mathcal{E})$ is a path and/or $\mathrm{ediam}(\mathcal{ST}(\mathcal{V}, \mathcal{E})) < 2\ell$. We consider two cases:

1. If $\mathrm{ediam}(\mathcal{ST}(\mathcal{V}, \mathcal{E})) < 2\ell$, by Observation 2.3.24, $\mathrm{diam}(\mathcal{ST}(\mathcal{V}, \mathcal{E})) < 4\ell$. Our cluster construction stops at this level. We then use credits of each vertex to pay for its incident Π-edges. Since each vertex is low-degree, its vertex credit is sufficient when $c(\epsilon) = \Omega(\frac{g}{\epsilon^2})$. Note that there is no path in level $i + 1$ or higher in $\mathcal{H}$ since such a path, say Q, would have length at least $\frac{\ell}{2\epsilon}$ which is at least $\mathrm{diam}(\mathcal{ST}(\mathcal{V}, \mathcal{E}))$ when $\epsilon < \frac{1}{8}$; contradicting that Q is a shortest path.
2. If $\mathrm{ediam}(\mathcal{ST}(\mathcal{V}, \mathcal{E}))$ is a path of effective diameter at least 2ℓ. Since there are no subgraphs in Phase 3, every Π-path must be incident to a vertex in an affix of $\mathcal{ST}$ of effective diameter at most 2ℓ. Since only short affix subpaths cannot pay for its incident Π-edges (Claim 2.3.27), we put all the Π-paths corresponding to Π-edges incident to short affix subpaths to the holding bag $\mathcal{B}_j$ (in the statement of Theorem 2.3.17). There are at most $2\frac{2g}{\epsilon}\frac{3\mathrm{g}}{\epsilon} = O(\frac{g^2}{\epsilon^2})$ such Π-paths. Thus, over all levels, the total weight of $\mathcal{B}$ is at most:

$$\begin{aligned}
&O(\frac{g^2}{\epsilon^2})\sum_i \ell_i \leq O(\frac{g^2}{\epsilon^2})\ell_{\max}\sum_i \epsilon^i \leq O(\frac{g^2}{\epsilon^2})w(\mathrm{ST})\sum_i \epsilon^i \\
&\leq O(\frac{g^2}{\epsilon^2})w(\mathrm{ST})\frac{1}{1-\epsilon} = O(\epsilon^{-2}) \cdot w(\mathrm{ST})
\end{aligned} \tag{2.27}$$

where $\ell_{\max}$ is the maximum length of shortest paths in $\mathcal{H}$, which is at most $w(\mathrm{ST})$.

Proof of Theorem 2.3.17. By Equation (2.27), the holding bag $\mathcal{B}_j$ has weight at most $O(\epsilon^{-2}) \cdot w(\mathrm{ST})$. By Claim 2.3.27, 2.3.31 and 2.3.33, we have:

$$w(S_j) \leq \max(O(\mathrm{poly}(\frac{1}{\epsilon})\tau(\epsilon,k,n)), O(\frac{g}{\epsilon^3})) = O(\mathrm{poly}(\frac{1}{\epsilon})\tau(\epsilon,k,n))$$

The stretch guarantee in (3) of Theorem 2.3.17 follows directly from Claim 2.3.19. $\square$

2.3.5 A singly exponential time algorithm for the subset TSP problem in bounded treewidth graphs

In this section, we give a dynamic program that can solve subset TSP in $2^{O(\mathrm{tw})}n^{O(1)}$ as stated in Theorem 2.3.3. Our algorithm is based on a method introduced by Bodlaender, Cygan, Kratsch, Nederlof [16] to design deterministic singly exponential time algorithms for connectivity problems in bounded treewidth graphs.

2.3.5.1 Representing partitions

Let $U = [n]$ be a ground set of n elements and $\Pi(U)$ be the set of all partitions of U. We abuse notation by denoting U to be the partition $\{U\} \in \Pi(U)$, i.e, the partition that has U as the only set. For each partition $\pi \in \Pi(U)$, define a *partition graph* G_π where $V(G_\pi) = U$ and there is an edge between u and v in G_π if they are both in the same set of π. Thus, there is a bijection between sets in π and cliques in G_π. For two elements $u, v \in U$, we denote by $U[ab]$ the partition of U that has $\{u, v\}$ as a set and other sets are singletons. By $\pi \setminus \{v\}$, we denote the partition of $U \setminus \{v\}$ obtained from π by removing v from π.

Let α, β be two partitions of $\Pi(U)$ and G_α, G_β be two corresponding partition graphs. We define a join operation $\sqcup$ as follows: $\alpha \sqcup \beta$ is a partition $\Pi(U)$ where each set of $\alpha \sqcup \beta$ is a connected component of the graph with vertex set U and edge set $E(G_\alpha) \cup E(G_\beta)$.

We say partition β is an *extension* of partition α if $\alpha \sqcup \beta = U$. Note that a partition can have many different extensions.

Let $\Gamma \subseteq \Pi(U)$ be a set of partitions of U. We say $\widehat{\Gamma}$ is a *representative set* of Γ if (i) $\widehat{\Gamma} \subseteq \Gamma$ and (ii) for any partition $\alpha \in \Gamma$ and any extension, say β, of α, then there is a partition $\hat{\alpha} \in \widehat{\Gamma}$ such that β is also an extension of $\hat{\alpha}$ ($\hat{\alpha} \sqcup \beta = U$). We say $\hat{\alpha}$ is a β-representation of α in $\widehat{\Gamma}$.

Suppose every partition $\alpha \in \Gamma$ has a weight $w(\alpha)$. We say $\widehat{\Gamma}$ is a *min representative set* of Γ, denoted by $\widehat{\Gamma} \subseteq_{\min} \Gamma$ if (i) $\widehat{\Gamma}$ is a representative set of Γ and (ii) for every $\alpha \in \Gamma$ and any extension β of α, there is a β-representation $\hat{\alpha}$ of α in $\widehat{\Gamma}$ such that $w(\hat{\alpha}) \leq w(\alpha)$.

The key idea in speeding up dynamic programs [16] is the following *representation theorem*.

Theorem 2.3.37 (Theorem 3.7 [16])**.** *Any set of weighted partitions Γ of U has a min representative set $\widehat{\Gamma}$ of size at most 2^{n-1} that can be found in time $|\Gamma| 2^{(\omega-1)n} n^{O(1)}$ where $|U| = n$ and ω is the matrix multiplication exponent.*

We note that size of Γ can be up to $2^{\Omega(n \log n)}$ but the representation theorem said that it has a min representative set of size at most 2^{n-1}.

2.3.5.2 Tree decompositions

Let $(\mathcal{T}, \mathcal{X})$ be a tree decomposition of G. For each node $t \in \mathcal{T}$, we denote its corresponding bag by X_t. Traditionally, each bag X_t is a set of vertices of G. However, for simplifying presentation of the dynamic program, we think of X_t as a bag of vertices and edges of G. That is, X_t is a subgraph of G. A tree decomposition $(\mathcal{T}, \mathcal{X})$ is *nice* if it is rooted at a node r where $|X_r| = \emptyset$ and other nodes are one of five following types:

Leaf node A leaf node t of $\mathcal{T}$ has $|X_t| = \emptyset$.

Introduce vertex node An introduce vertex node $t \in \mathcal{T}$ has only one child t' such that $X_{t'}$ is a subgraph of X_t, $|V(X_t)| = |V(X_{t'})| + 1$ and $E(X_t) = E(X_{t'})$.

Introduce edge node An introduce edge node $t \in \mathcal{T}$ has only one child t' such that $X_{t'}$ is a subgraph of X_t. $V(X_{t'}) = V(X_t)$ and $|E(X_t)| = |E(X_{t'})| + 1$.

Forget node A forget node $t \in \mathcal{T}$ has only one child t' such that X_t is an induced subgraph of $X_{t'}$ and $|V(X_t)| = |V(X_{t'})| - 1$.

Join node A join node t has two children t_1, t_2 such that $V(X_t) = V(X_{t_1}) = V(X_{t_2})$, $E(X_{t_1}) \cap E(X_{t_2}) = \emptyset$ and $E(X_t) = E(X_{t_1}) \cup E(X_{t_2})$.

A nice tree decomposition has $O(n)$ nodes and can be obtained from any tree decomposition of the same width of G in $O(n)$ time (see Proposition 2.2 [16]).

2.3.5.3 A dynamic programming algorithm for subset TSP

Recall T is a set of terminals in a treewidth-tw graph G. Let $k = |T|$. We modify G by adding $k-1$ parallel edges to each edge $e \in G$ and subdividing each new edge by a single vertex. Weight of each edge is splitted equally in the two new edges. The resulting graph is simple and has treewidth $\max(\mathrm{tw}, 2)$. This modification of G would guarantee that there is an optimal tour W that visits every edge at most once. (W is an Eulerian subgraph of G.)

For simplicity of presentation, we assume that the optimal solution W is unique. This assumption can also be technically enforced by imposing a lexicographic order on optimal solutions or by perturbation using Isolation Lemma [89].

For two edge sets E_1, E_2 of E. We use $E_1 \uplus E_2$ to be the *multiset addition* of E_1 and E_2. That is, we keep two copies of an edge in $E_1 \uplus E_2$ if it appears in both E_1 and E_2.

Let t be a node in $\mathcal{T}$. If t' is a descendant of t, we write $t' \preceq t$. Note that t is a descendant of itself. Let $G_t = \cup_{t' \preceq t} X_{t'}$. We regard the optimal solution W as a graph of G with vertex set spans by edges of W. Let $\mathrm{W}_t = G_t \cap W$. Note that there could be connected components of W_t that are isolated vertices. We call W_t a *partial solution*. It is straightforward to see that W_t satisfies one of the following two conditions for every node t:

1. W_t is a feasible solution. That is, W_t is an Eulerian subgraph of G and spans T.
2. Every vertex of T in $G_t \setminus X_t$ is in W_t, every vertex of $(\mathrm{W}_t \cap G_t) \setminus X_t$ has even degree and every connected component of W_t contains at least one vertex of X_t.

For each vertex $v \in X_t$, we assign a label $c_t(v) \in \{0, 1, 2\}$, where $c_t(v) = 0$ if v is not in W_t, $c_t(v) = 1$ if v has odd degree in W_t and $c_t(v) = 2$ if v has even degree in W_t. We denote the labeling restricted to a subset Y of X_t by $c_t(Y)$.

Let $Y_t = V(\mathrm{W}_t \cap X_t)$. Let α_t be the partition of Y_t *induced by* W_t. That is, vertices in the same connected component of W_t are in the same set of α_t. Let $R_t = E(\mathrm{W}) \setminus E(\mathrm{W}_t)$ be a subset edges of W not in G_t. Let β_t be the partition of Y_t induced by R_t. Since W is connected, $\alpha_t \sqcup \beta_t = Y_t$. We define the weight of α_t to be $w_t(\alpha_t) = w(\mathrm{W}_t)$.

We call tuple $(c_t(Y_t), \alpha_t, Y_t)$ the *encoding* of W_t, denoted by $\mathrm{Enc}(W_t)$. By definition of Y_t, any vertex $v \in X_t \setminus Y_t$ is not in W, hence, $c_t(v) = 0$. Thus, the labeling of vertices X_t is implicitly defined by labeling of vertices in Y_t. A encoding is *valid* if it encodes at least one partial solution. We only keep track of valid encodings during dynamic programming. There could be many partial solutions that have the same encoding. However, we only keep track of one partial solution, denoted by $\mathrm{Dec}(c_t(Y), \alpha_t, Y_t)$ for each each encoding $(c_t(Y_t), \alpha_t, Y_t)$, that has smallest $w(\alpha_t)$. The correctness follows from the following observation.

Observation 2.3.38. *Let W_t and W_t' be two partial solutions that have the same encoding $(c_t(Y_t), \alpha_t, Y_t)$ such that $w(W_t) < w(W_t)'$. If R_t is the set of edges such that $R_t \uplus W_t$ is a feasible solution, then $R_t \uplus W_t'$ is also a feasible solution with smaller weight.*

Claim 2.3.39. *If $(c_t(Y_t), \alpha_t, Y_t)$ is the encoding of the partial solution W_t of the optimal solution W in G_t, then $\mathrm{Dec}(c_t(Y_t), \alpha_t, Y_t) = W_t$.*

Proof. Let $\widehat{W_t} = \mathrm{Dec}(c_t, \alpha_t, Y_t)$ and $\widehat{W} = E(\widehat{W_t}) \uplus R_t$. By definition of decoding, $w(\widehat{W_t}) \leq w(W_t)$. Since $\widehat{W_t} \cap X_t = W_t \cap X_t$ (both are equal to Y_t) and labels of vertices in Y_t are the same in both $\widehat{W_t}$ and W_t, every vertex in $\widehat{W}$ has even degree. Since $\widehat{W_t}$ is a partial solution, $\widehat{W}$ spans all terminals. Since R_t has no edge in G_t, there are no parallel edges in $\widehat{W_t}$. Thus, $\widehat{W}$ is a feasible solution of subset TSP problem.

However, $w(\widehat{W}) = w(R_t) + w(\widehat{W_t}) \leq w(R_t) + w(W_t) = w(W)$. By the uniqueness assumption, $W_t = \widehat{W_t}$; the claim follows. □

For each node $t \in \mathcal{T}$, we would inductively maintain a set of encodings $\hat{\eta}_t$ that satisfies the following correctness invariant:

> **Correctness invariant:** $\hat{\eta}_t$ contains the encoding of the partial solution W_t of W.

By Claim 2.3.39, the correctness invariant implies that we are keeping track of W via encodings and their decodings. The key idea of an efficient dynamic program is to guarantee that $|\hat{\eta}_t| \leq 2^{O(\mathrm{tw})}$ for every node t. We do that by applying size reduction based on the representation theorem (Theorem 2.3.37).

> **Size reduction:** We guarantee that $|\hat{\eta}_t| \leq 12^{\mathrm{tw}}$ for every node t as follows. For a fixed labeling c_t of X_t and a fixed susbet $Y \subseteq X_t$, let $\eta_t(c_t, Y) = \{(c_t(Y'), \alpha, Y') : (c_t(Y'), \alpha, Y') \in \hat{\eta}_t, Y' = Y\}$ be the set of all encodings in $\hat{\eta}_t$ with the same set Y and vertex labeling c_t but different partitions of Y. Let Γ be the set of partitions of Y_t associated with encodings in $\eta_t(c_t, Y)$. Let

$\widehat{\Gamma} \subseteq_{\min} \Gamma$. By Theorem 2.3.37, $|\widehat{\Gamma}| \leq 2^{\mathrm{tw}-1}$. We now construct a new set of encodings $\hat{\eta}_t(c_t, Y)$ from $\eta_t(c_t, Y)$ as follows: for each partition $\hat{\alpha} \in \widehat{\Gamma}$, we add the encoding $(c_t(Y), \hat{\alpha}, Y)$ to $\hat{\eta}_t(c_t, Y)$. Then, we set $\hat{\eta}_t \leftarrow (\hat{\eta}_t \setminus \eta_t(c_t, Y)) \cup \hat{\eta}_t(c_t, Y)$. We repeat the reduction for every fixed Y and c_t. Since there are at most 2^{tw} different subsets Y and 3^{tw} different labelings c_t, $\hat{\eta}_t \leq 12^{\mathrm{tw}}$. We denote by $\mathrm{SR}(\hat{\eta}_t)$ the set of encodings obtained by applying size reduction to $\hat{\eta}_t$.

To see the correctness invariant of $\hat{\eta}_t$ after size reduction, consider encoding $(c_t(Y_t), \alpha_t, Y_t)$ of W_t. Before reduction, $(c_t(Y_t), \alpha_t, Y_t) \in \hat{\eta}_t$. Recall β_t is the partition of Y_t induced by R_t. By Theorem 2.3.37, there is an encoding $(c_t(Y_t), \hat{\alpha}_t, Y_t) \in \hat{\eta}_t$ after reduction such that $\hat{\alpha}_t \sqcup \beta_t = Y_t$ and $w(\hat{\alpha}_t) \leq w(\alpha_t)$. Let $\hat{W}_t = \mathrm{Dec}(c_t(Y_t), \hat{\alpha}_t, Y_t)$. Since labels of vertces in Y_t are the same for $\hat{W}_t$ and W_t, every vertex of $R_t \uplus \hat{W}_t$ has even degree. Recall R_t has no edge in G_t, thus, $R_t \uplus \hat{W}_t$ is an Eulerian subgraph of G that spans T. However, $w(R_t \uplus \hat{W}_t) \leq w(R_t \uplus W_t)$ since $w(\hat{W}_t) \leq w(W_t)$. By the uniqueness of W, $\hat{W}_t = W_t$. Hence, $\mathrm{Dec}(c_t(Y_t), \hat{\alpha}_t, Y_t) \in \hat{\eta}_t$. Thus, $\hat{\eta}_t$ satisfies correctness invariant.

We denote the empty encoding $(\emptyset, \{\emptyset\}, \emptyset)$ by $\emptyset$. If G_t has a feasible solution, then $\mathrm{Dec}(\emptyset)$ is the smallest weight feasible solution, say S_t, in G_t and the weight of the corresponding empty partition is $w(S_t)$. Otherwise, $\mathrm{Dec}(\emptyset) = \emptyset$ and the weight of the corresponding empty partition is $+\infty$.

Since the root node r has $X_r = \emptyset$, $G_r = G$. Thus, the feasible solution $\mathrm{Dec}(\emptyset)$ is the optimal solution W.

Leaf node For each leaf node t, $\hat{\eta}_t$ only contains the empty encoding $\emptyset$.

Introduce vertex node Let t be an introduce vertex node and t' be a child of t. Let $v = X_t \setminus X_{t'}$. By the definition of introduce vertex nodes, v is an isolated vertex in G_t. For each encoding $(c_{t'}(Y'), \alpha', Y')$ of $\hat{\eta}_{t'}$, we construct a new encoding $(c_t(Y), \alpha, Y)$ where:

(i) $Y = Y' \cup \{v\}$.

(ii) $c_t(v) = 0$ and $c_t(u) = c_{t'}(u)$ for every $u \in Y'$.

(iii) $\alpha = \alpha' \cup \{\{v\}\}$ (add v as a singleton to α').

Let $\mathrm{Dec}(c_t(Y), \alpha, Y) = \mathrm{Dec}(c_{t'}(Y'), \alpha', Y') \cup \{v\}$. Let η_t^{new} be the set of new encodings. Let $\eta_t = \eta_t^{new} \cup \hat{\eta}_{t'}$. We now show the correctness invariant for η_t.

Recall W_t and $W_{t'}$ are the partial solutions of W in G_t and $G_{t'}$, respectively. Since $V(G_t) = V(G_{t'}) \cup \{v\}$ and $E(G_t) = E(G_{t'})$, either (a) $W_t = W_{t'}$ or (b) $W_t = W_{t'} \cup \{v\}$ (v is added to $W_{t'}$ as an isolated vertex). In case (a), $(c_t(Y_t), \alpha_t, Y_t) = (c_{t'}(Y_{t'}), \alpha_{t'}, Y_{t'})$. Thus, encoding of W_t is in $\eta_{t'}$. In case (b), v is an isolated vertex of W_t, thus has $c_t(v) = 0$. Since we add $(c_t(Y_t), \alpha_t, Y_t)$ to η_t^{new} where $Y_t = Y_{t'} \cup \{v\}$ and $\alpha_t = \alpha_{t'} \cup \{\{v\}\}$, η_t contains the encoding of W_t.

Let $\hat{\eta}_t = \mathrm{SR}(\eta_t)$. Since $|\eta_t^{new}| \leq |\hat{\eta}_{t'}| \leq 12^{\mathrm{tw}}$, $|\eta_t| \leq |\eta_t^{new}| + |\hat{\eta}_{t'}| \leq 2 \cdot 12^{\mathrm{tw}} = 2^{O(\mathrm{tw})}$. Thus, by Theorem 2.3.37, the running time of size reduction is at most $2^{O(\mathrm{tw})} tw^{O(1)}$.

Introduce edge node Let t be an introduce edge node where an edge uv is introduced. Let t' be the only child of t. By the definition of introduce edge nodes, $V(H_t) = V(H_{t'})$ and $E(H_t) = E(Ht') \cup \{uv\}$.

Let $g(x) = ((x+1) \mod 2) + 1$. Function $g(x)$ has following properties: $g(x+1) = 1$ when $x = 0$ or $x = 2$ and $g(x+1) = 2$ when $x = 1$.

For each encoding $(c_{t'}(Y'), \alpha', Y')$ of $\hat{\eta}_{t'}$, we construct a new encoding $(c_t(Y), \alpha, Y)$ where:

(i) $Y = Y'$.

(ii) $c_t(u) = g(c_{t'}(u)+1)$, $c_t(v) = g(c_{t'}(v)+1)$ and $c_t(w) = c_{t'}(w)$ for every $w \in Y' \setminus \{u, v\}$.

(iii) $\alpha_t = \alpha_{t'} \sqcup Y'[uv]$. We the assign $w(\alpha_t) = w(\alpha_{t'}) + w(uv)$.

Let $\mathrm{Dec}(c_t(Y), \alpha, Y) = \mathrm{Dec}(c_{t'}(Y'), \alpha', Y') \cup \{uv\}$. Let η_t^{new} be the set of new encodings. We then remove duplicates from η_t^{new}: if there are two encodings $(c_t(Y), \alpha, Y)$, $(c_t(Y), \beta, Y)$ in η_t^{new} where $\alpha = \beta$ but $w(\alpha) < w(\beta)$ or $(c_t(Y), \beta, Y)$ is just another version of the same encoding $(c_t(Y), \alpha, Y), c_t(Y))$ (two versions are constructed from different encodings in $\hat{\eta}_t$.), we remove $\mathrm{Dec}(c_t(Y), \beta, Y)$ from η_t^{new}. Let $\eta_t = \eta_t^{new} \cup \eta_{t'}$. We now show the correctness invariant for η_t.

Since $V(G_t) = V(G_{t'})$ and $E(G_t) = E(G_{t'}) \cup \{uv\}$, either (a) $W_t = W_{t'}$ or (b) $W_t = W_{t'} \cup \{uv\}$. In case (a), $(c_t(Y_t), \alpha_t, Y_t) = (c_{t'}(Y_{t'}), \alpha_{t'}, Y_{t'})$. Thus, the encoding of W_t is in $\eta_{t'}$. In case (b), adding edge uv change the label of u and v in $W_{t'}$ to $g(c_{t'}(u)+1)$ and $g(c_{t'}(v)+1)$, respectively. If u, v are in two different components of $W_{t'}$, say C'_u, C'_v, respectively, adding uv merges C'_u and C'_v into one connected component. Thus, $\alpha_t = \alpha_t \cup Y_{t'}[uv]$. That implies the encoding $(c_t(Y_t), \alpha_t, Y_t)$ of W_t is in η_t^{new}. By Observation 2.3.38, $(c_t(Y_t), \alpha_t, Y_t)$ is not removed in η_t^{new} during the duplicate removal; the correctness invariant of η_t follows.

Let $\hat{\eta}_t = \mathrm{SR}(\eta_t)$. Since $|\eta_t^{new}| \leq |\hat{\eta}_{t'}| \leq 12^{\mathrm{tw}}$, $|\eta_t| \leq 2 \cdot 12^{\mathrm{tw}} = 2^{O(\mathrm{tw})}$. Thus, the running time of size reduction is at most $2^{O(\mathrm{tw})} tw^{O(1)}$.

Forget node Let t be a forget node and t' be the only child of t. Let $v = X_{t'} \setminus X_t$. We first discard any encoding $(c_{t'}(Y'), \alpha', Y)$ in $\hat{\eta}_{t'}$ that satisfies one of three following conditions:

1. $c_{t'}(v) = 1$.
2. $c_{t'}(v) = 0$ and $v \in T$.
3. $c_{t'}(v) = 2$, v is a singleton in the partition α' and $\mathrm{Dec}(c_{t'}(Y'), \alpha', Y')$ is not a feasible solution.

For each remaining encoding, say $(c_{t'}(Y'), \alpha', Y')$, of $\hat{\eta}_{t'}$, we construct a new encoding $(c_t(Y), \alpha, Y)$ where:

(i) $Y = Y' \setminus \{v\}$.
(ii) $c_t(u) = c_{t'}(u)$ for every $u \in Y$
(iii) $\alpha = \alpha' \setminus \{v\}$ and $w(\alpha) = w(\alpha')$.

Let $\mathrm{Dec}(c_t(Y), \alpha, Y) = \mathrm{Dec}(c_{t'}(Y'), \alpha', Y')$. Let η_t^{new} be the set of new encodings. Let $\eta_t = \eta_t^{new} \cup \hat{\eta}_{t'}$. We then remove duplicates from η_t. We now show the correctness invariant for η_t.

Observe that if $v \in W_{t'}$, it must have label 2 in the encoding of $W_{t'}$ since $V(G_t) = V(G_{t'} \setminus \{v\})$. Furthermore, if v is a singleton in $\alpha_{t'}$, $W_{t'} = W$. Thus, $\mathrm{Dec}(c_{t'}(Y_{t'}), \alpha_{t'}, Y_{t'})$ is a feasible solution. That implies $\mathrm{Dec}(c_{t'}(Y_{t'}), \alpha_{t'}, Y_{t'})$ is not discarded at the beginning (the new encoding constructed from $\mathrm{Dec}(c_{t'}(Y_{t'}), \alpha_{t'}, Y_{t'})$ is empty.).

We consider two cases: (a) $W_{t'}$ does not contain v and (b) $W_{t'}$ contains v. In case (a), $(c_t(Y_t), \alpha_t, Y_t) = (c_{t'}(Y_{t'}), \alpha_{t'}, Y_{t'})$. Thus, the encoding of W_t is in $\hat{\eta}_{t'}$. In case (b), $Y_t = Y_{t'} \setminus \{v\}$, $c_t(Y_t) = c_{t'}(Y_{t'} \setminus \{v\})$ and $\alpha_t = \alpha_{t'} \setminus \{v\}$. Thus, $(c_t(Y_t), \alpha_t, Y_t)$ is in η_t^{new}; the correctness invariant of η_t follows.

Let $\hat{\eta}_t = \mathrm{SR}(\eta_t)$. Since $|\eta_t^{new}| \leq |\hat{\eta}_{t'}| \leq 12^{\mathrm{tw}}$, $|\eta_t| \leq 2 \cdot 12^{\mathrm{tw}} = 2^{O(\mathrm{tw})}$. Thus, the running time of size reduction is at most $2^{O(\mathrm{tw})} tw^{O(1)}$.

Join node Let t be a join node with two children t_1, t_2. Note that $X_t = X_{t_1} = X_{t_2}$. Let $h(x, y)$ be a function where:

$$h(x,y) = \begin{cases} 0, & \text{if } x = y = 0 \\ 1, & \text{if } x + y \text{ is odd} \\ 2, & \text{otherwise} \end{cases}$$

For each encoding $(c_{t_1}(Y_1), \alpha_1), Y_1$ of $\hat{\eta}_{t_1}$ and $(c_{t_2}(Y_2), \alpha_2, Y_2)$ of $\hat{\eta}_{t_1}$ such that $Y_1 = Y_2$, we construct a new encoding $(c_t(Y), \alpha, Y)$ where:

(i) $Y = Y_1 = Y_2$.

(ii) $c_t(u) = h(c_{t_1}(u), c_{t_2}(u))$ for every $u \in Y$

(iii) $\alpha = \alpha_1 \sqcup \alpha_2$ and $w(\alpha) = w(\alpha_1) + w(\alpha_2)$.

Since $E(X_{t_1}) \cap E(X_{t_2}) = \emptyset$, $E(\mathrm{Dec}(c_{t_1}(Y_1), \alpha_1, Y_1)) \cap E(\mathrm{Dec}(c_{t_2}(Y_2), \alpha_2, Y_2)) = \emptyset$. Let $\mathrm{Dec}(c_t(Y), \alpha, Y) = \mathrm{Dec}(c_{t_1}(Y_1), \alpha_1, Y_1) \cup \mathrm{Dec}(c_{t_2}(Y_2), \alpha_2, Y_2)$. Let η_t be the set of new encodings. We then remove duplicates from η_t. We now show the correctness invariant for η_t.

Recall W_{t_1}, W_{t_2} are the partial solutions of W in G_{t_1} and G_{t_2}, respectively. We consider the relationship between the encoding $(c_t(Y_t), \alpha_t, Y_t)$ of W_t and the encodings of its two children $(c_{t_1}(Y_{t_1}), \alpha_{t_1}, Y_{t_1})$ and $(c_{t_2}(Y_{t_2}), \alpha_{t_2}, Y_{t_2})$.

Since $E(G_{t_1}) \cap E(G_{t_2}) = \emptyset$, $E(W_{t_1}) \cap E(W_{t_2}) = \emptyset$. Since $X_t = X_{t_1} = X_{t_2}$, we have $Y_t = Y_{t_1} = Y_{t_2}$. Since degree in W_t of a vertex $v \in Y_t$ is the sum of its degrees in Y_{t_1} and Y_{t_2}, $c_t(v) = h(c_{t_1}(v), c_{t_2}(v))$. Since $W_t = W_{t_1} \cup W_{t_2}$, we have $\alpha_t = \alpha_{t_1} \sqcup \alpha_{t_2}$. That implies $(c_t(Y_t), \alpha_t, Y_t)$ is in η_t.

Let $\hat{\eta}_t = \mathrm{SR}(\eta_t)$. Since $|\eta_t| \leq |\hat{\eta}_{t_1}||\hat{\eta}_{t_2}| \leq 12^{2\mathrm{tw}} = 2^{O(\mathrm{tw})}$, size reduction can be done in $2^{O(\mathrm{tw})} tw^{O(1)}$ time.

Claim 2.3.40. *The dynamic programming table of each node can be constructed in time* $2^{O(\mathrm{tw})} tw^{O(1)} n$.

The n factor in Claim 2.3.40 is for maintaining decodings in each step. This factor can be removed, but it is not the purpose of our paper. Thus, the total running time of the dynamic programming algorithm is $2^{O(\mathrm{tw})} tw^{O(1)} n^2$, thereby implying Theorem 2.3.3.

Chapter 3: Local Search PTASes

Local search is a widely used heuristic for solving optimization problems. It has been successfully applied to the traveling salesperson problem [33], k-median [8], facility location [8] problems. A striking property of local search is simplicity: repeatedly changing a constant number of vertices or edges to improve the current solution until no such changes possible. In this chapter, we show that local search yields PTASes for two problems:

r-dominating set Given a graph G, find a subset of vertices D of minimum size such that every vertex in $V \setminus D$ is in distance at most r from at least one vertex in D.

Feedback vertex set Given a graph G, find a subset of vertices S of minimum size such that $G \setminus S$ is acyclic.

3.1 Local search PTASes

The first PTASes by local search were discovered independently for geometric problems by Chan and Har-Peled [27] and Mustafa and Ray [78]. The algorithm, shown in Figure 3.1, is conceptually simple and can be applied to any problem. Intuitively, the algorithm checks whether a better solution can be obtained from the current solution by exchanging a constant number of vertices. We show below that for most problems, the constant c in Figure 3.1 is $O(\frac{1}{\epsilon^2})$. Thus, one can search for a set of vertices to exchange in $n^{O(\frac{1}{\epsilon^2})}$ time. Since the algorithm terminates after at most n exchanges, the total running time is $n^{O(\frac{1}{\epsilon^2})}$.

The main technical challenge is the analysis: showing that local search gives a $(1+\epsilon)$-approximation upon termination. There have been several results on the analysis of local search for problems in H-minor-free graphs. Cabello and Gajser [26] showed that local search gives PTASes for maximum independent set, vertex cover and dominating set problems. Cohen-Addad, Klein and Mathieu [32] showed that local search yields PTASes for k-means, k-median and uniform facility location problems. Chaplick, De,

```
LocalSearch(G, ϵ)
    S ← an arbitrary feasible solution of G.
    c ← a constant depending on ϵ.
    while there is a solution S′ such that |S \ S′| ≤ c, |S′ \ S| ≤ c and |S′| < |S|
        S ← S′
    return S
```

Figure 3.1: The local search algorithm.

Ravsky and Spoerhase [28] showed the same result for maximum k-coverage problems. A key ingredient in the analysis are δ-divisions of H-minor-free graphs. We say a vertex v is a boundary vertex of a subgraph K of G if v is incident to an edge in $E(G) \setminus E(K)$.

Definition 3.1.1. *For an integer δ, an δ-division of a graph G is a collection of edge-disjoint subgraphs of G, called* regions*, with the following properties.*

1. *Each region contains at most δ vertices.*
2. *The number of regions is at most $c_d \frac{n}{\delta}$.*
3. *The number of boundary vertices, summed over all regions, is at most $c_d \frac{n}{\sqrt{\delta}}$.*

where c_d is a constant.

Frederickson [49] introduced δ-divisions of planar graphs to speed up planar shortest path computation. A similar division was obtained earlier by Lipton and Tarjan [73] to approximate independent set in planar graphs. Since δ-divisions in planar graphs only rely on the separator theorem by Lipton and Tarjan [73], we can extend δ-division naturally to H-minor-free graphs using the separator theorem by Alon, Seymour and Thomas [6] or a similar theorem by Kawarabayashi and Reed [58].

3.2 Domination problems

When r is a fixed parameter for the r-domination problem, the bidimensionality framework by Demain and Hajiaghayi [36] gives an EPTAS where the running time depends exponentially on r. The essential part of the bidimensionality framework is to reduce the original problem to the same problem in a bounded treewidth graph where dynamic programming can be used to solve the problem optimally.

In joint work with Glencora Borradaile [22], we show that unless the Exponential Time Hypothesis fails, there is no subexponential algorithm for r-dominating set problem in bounded treewidth graphs. In that sense, the exponential dependency of r is unavoidable, and that means it is hard obtain a PTAS for r-dominating set when r is beyond $O(\log n)$. When r is a part of the input, Eisenstat, Klein and Mathieu [44] designed a *bicritera* PTAS that approximates both the size of the optimal solution and the domination distance r. We show that local search gives a PTAS for this problem for all possible values of r, even on edge-weighted graphs. Our analysis is simple and similar to the analysis of Cabello and Gajser [26].

Theorem 3.2.1. *Local search gives a PTAS for r-dominating set problem in H-minor-free graphs with non-negative edge-weights.*

Let O be the set of vertices in an optimal solution. Let L be the local search solution (Figure 3.1). Let $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$ be a graph, called the *exchange graph*, with vertex set $V_{\mathrm{Ex}} = L \cup O$ obtained by contracting every vertex of G not in V_{Ex} to its nearest vertex in V_{Ex} (breaking ties by lexicographic order). We make $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$ simple by removing self-loops and parallel edges. We say a vertex v is *r-dominated* by a vertex $u \in V_{\mathrm{Ex}}$ if $d_G(u, v) \leq r$ (a vertex r-dominates itself).

Lemma 3.2.2 (Exchange property)**.** *For every vertex v of G, either v is r-dominated by a vertex in $L \cap O$ or there is an edge $uw \in E_{\mathrm{Ex}}$ where $u \in L \setminus O$ and $w \in O \setminus L$ and v is r-dominated by both u and w in G.*

Proof. For each vertex $v \in G$, we use $V_{\mathrm{Ex}}(v)$ to denote the vertex in V_{Ex} that v is contracted to. If v is in V_{Ex}, we let $V_{\mathrm{Ex}}(v) = v$.

Consider any vertex v of g. If v is contracted to a vertex, say $x \in L \cap O$, then v is r-dominated by x. Thus, the lemma holds. Hence, we can assume that v is not contracted to a vertex $L \cap O$. We suppose w.l.o.g. that $V_{\mathrm{Ex}}(v) \in L$. Let v' be a vertex in O that r-dominates v. Let $v = v_0, v_1, \ldots, v_k = v'$ be vertices on a shortest v-to-v' path, say P, in G. Let $P_S = \{s_1, s_2, \ldots, s_\ell\}$ be the corresponding path in Ex such that: $s_1 = V_{\mathrm{Ex}}(v), s_\ell = v'$ (recall $V_{\mathrm{Ex}}(v') = v'$) and for any j, $s_j = V_{\mathrm{Ex}}(v_i)$ for some i.

Claim 3.2.3. *Vertex v is r-dominated by every vertex in P_S.*

Proof. Let s_j be a vertex in P_S and v_i be a vertex in P such that $V_{\mathrm{Ex}}(v_i) = s_j$. Since v_i is contracted to s_j, $d_G(v_i, s_j) \leq d_G(v_i, s_\ell)$. Thus, we have:

$$\begin{aligned} d_G(v, s_j) &\leq d_G(v, v_i) + d_G(v_i, s_j) \\ &\leq d_G(v, v_i) + d_G(v_i, s_\ell) \\ &= d(v, s_\ell) \quad \text{(since } v_i \text{ is on the shortest path } P\text{)} \\ &= d(v, v') \leq r \end{aligned}$$

□

If any vertex of P_S is in $L \cap O$, then by Claim 3.2.3, v is r-dominated by a vertex in $L \cap O$; the lemma holds. Thus, we can assume that $V(P_S) \cap L \cap O = \emptyset$. Since path P_S starts with a vertex in L and ends with a vertex in O, there must be two adjacent vertices u and w such that $u \in L \setminus O$ and $w \in O \setminus L$. By Claim 3.2.3, u and w both r-dominate v; the lemma follows. □

Proof of Theorem 3.2.1. By construction, $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$ is H-minor-free. Thus, there exists an δ-division of $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$. Let $c = \delta = \frac{1}{\epsilon^2}$ (c is the constant in the algorithm in Figure 3.1). Let $R_1, R_2, \ldots, R_k$ be the set of regions in the δ-division of G. Let B_i be the set of vertices on the boundary of R_i and $int(R_i) = R_i \setminus B_i$, $1 \leq i \leq k$. By (ii) of Definition 3.1.1, we have:

$$\sum_{i=1}^{k} |B_i| = c_d \left(\frac{|O \cup L|}{\sqrt{\delta}} \right) \leq c_d \epsilon (|O| + |L|) \tag{3.1}$$

Let $O_i = O \cap int(R_i)$ and $L_i = L \cap int(R_i)$. Let $M_i = (L \setminus R_i) \cup B_i \cup O_i$.

Claim 3.2.4. *M_i is an r-dominating set.*

Proof. Suppose otherwise; there is a vertex $v \in G$ that is not r-dominated by M_i. Since L is an r-dominating set, v must be r-dominated by a vertex u in $L_i \setminus O$. By Lemma 3.2.2, there exists a vertex $w \in O$ that r-dominates v and $uw \in \mathrm{Ex}$. That implies $w \in B_i \cup O_i$; contradicting that v is not r-dominated by M_i. □

Since $|M_i \setminus L| < |R_i|$ and $|L \setminus M_i| < |R_i|$ and $|R_i| \leq \delta = c$, it must be that $|M_i| > L$. That implies:

$$|L_i| < |B_i| + |O_i| \tag{3.2}$$

Thus, we have:

$$\begin{aligned}
|L| = \sum_{i=1}^{k} |L \cap R_i| &\le \sum_{i=1}^{k} |L_i| + |B_i| \\
&< \sum_{i=1}^{k} |O_i| + 2\sum_{i=1}^{k} |B_i| \qquad \text{(by Equation 3.2)} \\
&\le |O| + 2\sum_{i=1}^{k} |B_i| \\
&\le |O| + 2c_d\epsilon(|O| + |L|) \qquad \text{(by Equation 3.1)}
\end{aligned}$$

Thus, $|L| < \frac{1+2c_d\epsilon}{1-2c_d\epsilon}|O| = (1 + O(\epsilon))|O|$ when ϵ is sufficiently small. □

3.3 The feedback vertex set problem

FVS problem is in Karp's list of 21 NP-complete problems [56] and has many real-world applications: deadlock recover in operating systems [86], VLSI design [47], wireless networks [91]. In general graphs, the first approximation algorithm guaranteed $O(\log n)$-approximate solution by Bar-Yehuda, Geiger, Naor and Roth [12] and the best known approximation algorithm guarantes a 2-approximate solution [10, 14]. FVS is also one of the central problems in parameterized complexity [42] and there was a PACE challenge in 2016 [35] on this problem.

In planar graphs, the FVS problem has played an important role as motivation for new PTASes. Baker's technique [11] is a powerful technique that gives PTASes for various optimization problems. However, as said by Demaine and Hajiaghayi [36],

> "[...] all applications of Baker's approach so far are to optimization problems arising from "local" properties (such as those definable in first-order logic). Intuitively, such local properties can be decided by locally checking every constant-size neighborhood. In particular, this restriction has limited attempts at characterizing the complexity class of problems admitting PTASs."

Prominent representatives of problems that have "local" properties are vertex cover and dominating set. However, the FVS problem does not have such local properties and

Baker's technique [11] does not apply to this problem. In fact, the FVS problem is one of two motivating problems (the other being connected dominating set) for Demaine and Hajiaghayi [36] to develop the bidimensionality PTAS framework. We note that the first PTAS for the FVS problem in planar graphs was obtained earlier by Kleignberg and Kumar [62], but their algorithm is quite complicated.

Though a PTAS for FVS in H-minor-free graphs is known by the bidimentionality framework of Demaine and Hajiaghayi, it was not known whether local search admits a PTAS for FVS. Known local search PTASes in H-minor-free graphs are for problems that as for Baker's technique, have "local" properties. Such locality is also reflected in the analysis where exchange graphs are built by taking a subgraph of the input graph on $O \cup L$, such as in the analysis of vertex cover and independent set [26], or by contracting vertices outside V_{Ex} to nearest vertices in $O \cup L$, such as in the analysis of dominating set [26], k-means, k-median, uniform facility location [32] and the maximum coverage problem [28]. However, FVS doesn't have such locality and simply taking a subgraph or taking a contraction of the input graph is not enough to guarantee the *exchange property* of the resulting graph, that is, for any cycle of G, there must be a vertex in $L \cap O$ or an edge between a vertex of O and a vertex of L.

Thus, constructing an exchange graph, say $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$, for analysis is the major technical hurdle. Instead of limiting $V_{\mathrm{Ex}} = L \cup O$ as previous work, we add to $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$ vertices in $V \setminus (O \cup L)$, called *Steiner vertices*. However, for the analysis to work, we need to guarantee that the number of Steiner vertices is $O(|L| + |O|)$.

Theorem 3.3.1. *There is an H-minor-free graph $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$ such that:*

(1) $L \cup O \subseteq V(\mathrm{Ex}) \subseteq V(G)$.

(2) $|V(\mathrm{Ex})| \leq c_e(|L| + |O|)$ *for* $c_e = \mathrm{poly}(|V(H)|)$.

(3) For every cycle C of G, there is (3a) a vertex of C in $O \cap L$ or (3b) an edge $uv \in E_{\mathrm{Ex}}$ between a vertex $u \in L$ and a vertex $v \in O$ in C or (3c) a cycle C' of Ex *such that $V(C') \subseteq V(C)$ and $C \cap (O \cup L) = C' \cap (O \cup L)$.*

Given Theorem 3.3.1, we can show the following theorem.

Theorem 3.3.2. *For any fixed $\epsilon > 0$, local search algorithm finds a $(1+\epsilon)$-approximate solution for the FVS problem in H-minor-free graphs with running time $O(n^c)$ where $c = \frac{\mathrm{poly}(|V(H)|)}{\epsilon^2}$.*

Proof. The proof is similar to the proof of Theorem 3.2.1. The only difference is that we have Steiner vertices in the exchange graph. We set the constant c in the algorithm in Figure 3.1 as $1/\tau^2$ where $\tau = \frac{\epsilon}{2c_d c_e(2+\epsilon)} = O(\frac{\epsilon}{c_d c_e})$. Note that c_d is the constant in Definition 3.1.1. Since both c_d and c_e are polynomial in $|V(H)|$, c is also polynomial in $|V(H)|$ which implies the claimed running time.

Let $\mathrm{Ex}(V_{\mathrm{Ex}}, E_{\mathrm{Ex}})$ be the exchange graph for O and L as in Theorem 3.3.1. Since Ex is H-minor-free, we can find a δ-division of Ex for $\delta = c = \lceil 1/\tau^2 \rceil$. Let B be the multi-set containing all the boundary vertices in the r-division. By the third property of δ-division, $|B|$ is at most $c_d \frac{|V(\mathrm{Ex})|}{\sqrt{\sigma}}$. By the second property of exchange graph, $|V(\mathrm{Ex})| \leq c_e(|O| + |L|)$. Thus, we have:

$$|B| \leq c_d c_e \tau(|O| + |L|) \tag{3.3}$$

Below, we will show that:

$$|L| \leq |O| + 2|B| \tag{3.4}$$

Then by Equation 3.3, we have:

$$|L| \leq |O| + 2c_d c_e \tau(|O| + |L|) = |O| + \frac{\epsilon}{2+\epsilon}(|O| + |L|)$$

which implies $|L| \leq (1+\epsilon)|O|$.

To prove Equation (3.4), we need to study some properties of Ex. For any region R_i of the δ-division, let B_i be the boundary of R_i and M_i be the union of $L \setminus R_i$, $O \cap R_i$ and B_i.

Claim 3.3.3. *M_i is a feedback vertex set of G.*

Proof. For a contradiction, assume that there is a cycle C of G that is not covered by M_i. Then C does not contain any vertex of $L \setminus R_i$, $O \cap R_i$ and B_i. So C can only be covered by vertices of $(L \setminus O) \cap int(R_i)$ and vertices of $O \setminus (L \cup R_i)$. This implies that C does not contain any vertex of $O \cap L$ and there is no edge in Ex between $C \cap O$ and $C \cap L$. By the third property of the exchange graph, there must be a cycle C' in Ex such that $V(C') \subseteq V(C)$ and $C \cap (O \cup L) = C' \cap (O \cup L)$. Let u be the vertex of $(L \setminus O) \cap int(R_i)$ in C and v be the vertex of $O \setminus (L \cup R_i)$ in C. Then cycle C' contains both u and v, which implies C' crosses the boundary of R_i, that is $C' \cap B_i \neq \emptyset$. Let w

be a vertex in $C' \cap B_i$, then w also belongs to C in G. This implies M_i contains a vertex of C, a contradiction. $\square$

By the construction of M_i, we know the difference between L and M_i is bounded by the size of the region R_i, which is δ. Recall that $c = \delta = 1/\tau^2$. Since L is the output of the local search algorithm, we know L cannot be improved by changing at most δ vertices. So we have $|L| \leq |M_i|$. By the construction of M_i, this implies

$$|L \cap R_i| \leq |M_i \cap R_i| \leq |O \cap int(R_i)| + |B_i|.$$

which in turn implies

$$|L \cap int(R_i)| \leq |L \cap R_i| \leq |O \cap int(R_i)| + |B_i|.$$

Since $int(R_i)$ and $int(R_j)$ are vertex-disjoint for any two distinct i and j, by summing over all regions in the δ-division, we can have

$$|L| - |B| \leq \sum_i |L \cap int(R_i)| \leq \sum_i (|O \cap int(R_i)| + |B_i|) \leq |O| + |B|.$$

This proves Equation (3.4) and so, Theorem 3.3.2. $\square$

3.3.1 Exchange Graph Construction

Recall that $\sigma_H = |V(H)|\sqrt{\log |V(H)|}$ is the sparsity of H-minor-free graphs. We will show that the constant c_e in Theorem 3.3.1 is $O(\sigma_H)$. We construct the exchange graph in three steps:

Step 1 We delete all edges in G that are incident to vertices of $O \cap L$. We then remove all components that do not contain any vertex $O \cup L$. Note that the removed components are acyclic.

Step 2 We contract edges that have an endpoint that is not a solution vertex and has degree at most two until there is no such edge left. Since L and O are both feedback vertex sets of G, every cycle after the contraction must contain a vertex from L and a vertex from O. Since edges incident to vertices of $O \cap L$ are removed, there is no self-loop after this step.

Step 3 We keep the graph simple by removing parallel edges.

Let K be the resulting graph. We now show that K satisfies three properties in Theorem 3.3.1. Property (1) is obvious because we never delete a vertex in $L \cup O$ from K. To show property (3), let C be a cycle of G. If any edge of C is removed in Step 1, C must contain a vertex in $O \cap L$; implying (3a). Thus, we can assume that no edge of C is deleted after Step 1. Since contraction does not destroy cycles, after the contraction in Step 2, there is a cycle C' such that $V(C') \subseteq V(C)$. If $|V(C')| = 2$ (C' is a cycle of two parallel edges), then (3b) holds. Thus, we can assume that every edge of C' remains intact after removing parallel edges. But that implies (3c) since we never remove solution vertices from G. Thus, K satisfies property (3).

It remains to show K satisfies property (2) in Theorem 3.3.1, that is, $|V(K)| \leq O(\sigma_H)(|L| + |O|)$. By Step 2 we have the following observation.

Observation 3.3.4. *Every Steiner vertex of K has degree at least 3.*

Since $O \cup L$ is a feedback vertex set of K, $K \setminus (O \cup L)$ is a forest F containing only Steiner vertices. For each tree T in F, we define the *degree* of T, denoted by $\deg_K(T)$, as the number of edges in K between T and $O \cup L$. Let $\ell(T)$ be the number of leaves of T. By Observation 3.3.4, every internal vertex of T has degree at least 3. Thus, $|V(T)| \leq 2\ell(T)$. That implies:

$$|V(T)| \leq 2\deg_K(T). \tag{3.5}$$

We contract each tree T of F into a single Steiner vertex s_T. Let K' be the resulting graph. Then we have the following observation.

Observation 3.3.5. *Graph K' is simple.*

Proof. Since every cycle of K must contain a vertex from L and a vertex from O, there cannot be any solution vertex in K that is incident to more than one vertex of a tree T of F. So there cannot be parallel edges in K'. □

To bound the size of K', we need the following structural lemma. We remark that this lemma holds for general graphs.

Lemma 3.3.6. *For a graph G and two disjoint nonempty vertex subsets A and B, let $D = V(G) \setminus (A \cup B)$. If (i) D is an independent set, (ii) every vertex in $V(G) \setminus (A \cup B)$ has degree at least 3 in G and (iii) for every cycle C in G, we have $C \cap A \neq \emptyset$ and $C \cap B \neq \emptyset$, then we have $|V(G)| \leq 2(|A| + |B|)$.*

Proof. We remove every edge that only has endpoints in $A \cup B$ and let the resulting graph be G'. Then G' is a bipartite graph with $A \cup B$ in one side and D in the other side since D is an independent set. Let D_A (resp. D_B) be the subset of D only containing the vertices that have at least two neighbors in A (resp. B). Since every vertex of D has degree at least 3, we have $D_A \cup D_B = D$.

Let H_A be the subgraph of G' induced by $A \cup D_A$. Then H_A is acyclic since otherwise every cycle of H_A would correspond to a cycle in G that does not contain any vertex in B. We now construct a graph H^*_A on vertex set A. For each vertex $v \in D_A$, we arbitrarily choose its two neighbors x and y in A and add an edge between x and y in H^*_A. By construction, there is a one-to-one mapping between edges of H^*_A and vertices of D_A.

Since H_A is acyclic, H^*_A is also acyclic. Thus, $|E(H^*_A)| \leq V(H^*_A) = |A|$. This implies $|D_A| \leq |A|$. By a similar argument, we can show that $|D_B| \leq |B|$. Thus, $|D| = |D_A \cup D_B| \leq |A| + |B|$, and the lemma follows. $\square$

Let Z be an arbitrary component of K' that contains at least one Steiner vertex. Then the two sets $V(Z) \cap O$ and $V(Z) \cap L$ must be disjoint since any vertex in $O \cap L$ is isolated in K'. And each of the two sets cannot be empty since there must be a cycle in Z through the Steiner vertex which also contains a vertex of O and a vertex of L respectively. Let X be the set of Steiner vertices in Z. By the construction of K', vertex set X is an independent set of Z. By Observation 3.3.4, every vertex of X has degree at least 3. So we can apply Lemma 3.3.6 for Z, $V(Z) \cap O$ and $V(Z) \cap L$, and obtain $|V(Z)| \leq 2(|V(Z) \cap O| + |V(Z) \cap L|) = 2(|V(Z) \cap O| + |V(Z) \cap (L \setminus O)|)$. Note that this bound holds trivially if Z does not contain any Steiner vertex. Thus, summing over all components of K', we have $|V(K')| \leq 2(|V(K') \cap O| + |V(K') \cap (L \setminus O)|) \leq 2(|O| + |L|)$. Since K' is a minor of G , it is also H-minor-free. By Lemma 1.1.3, we have

$$\begin{aligned} |E(K')| &= O(\sigma_H |V(K')|) \\ &= O(\sigma_H)(|O| + |L|) \end{aligned} \tag{3.6}$$

We now ready to bound the size of $V(K)$. Recall each tree T of F is contracted into a single Steiner vertex s_T in K'. We have:

$$\begin{aligned}
|V(K) \setminus (O \cup L)| &= \textstyle\sum_{T \in F} |V(T)| \\
&\leq 2 \textstyle\sum_{T \in F} \deg_K(T) && \text{(Equation (3.5))} \\
&= 2 \textstyle\sum_{T \in F} \deg_{K'}(s_T) \\
&\leq 2|E(K')| && (\{s_T | T \in F\} \text{ is an independent set}) \\
&= O(\sigma_H)(|O| + |L|) && \text{(Equation (3.6))}
\end{aligned}$$

This implies $V(K) \leq O(\sigma_H)(|O| + |L|)$. Thus K satisfies property (2) in Theorem 3.3.1.

3.3.2 Negative Results

In this section, we show some negative results for the FVS problem and its variants. A graph is *1-planar* if it can be drawn in the Euclidean plane such that every edge has at most one crossing, where it crosses a single additional edge. We first show that FVS is APX-hard in 1-planar graphs. Then for the two variants, odd cycle transversal and subset feedback vertex set, we construct examples where local search with constant exchanges cannot give a constant approximation in planar graphs. The *odd cycle transversal* problem (also called *bipartization*) asks for a minimum set of vertices in an undirected graph whose removal results in a bipartite graph. Given an undirected graph and a subset U of vertices, the *subset feedback vertex set* problem asks for a minimum set S of vertices such that after removing S the resulting graph contains no cycle that passes through any vertex of U.

Theorem 3.3.7. *Given a general graph G, we can construct a 1-planar graph H in polynomial time, such that G has a feedback vertex set of size at most k if and only if H has a feedback vertex set of size at most k.*

Proof. Consider a drawing of G on the plane where each pair of edges can cross at most once. For each crossed edge e in G, we subdivide e into edges so that there is exactly one crossing per new edge. Let H be the resulting graph. By construction, graph H is 1-planar.

Let n be the size of G. Since there are at most $O(n^2)$ crossings per edge in the drawing, the size of H is at most $O(n^4)$. Sine we only subdivide edges, there is a one-to-one mapping between cycles of G and cycles of H. It is straightforward to see that any feedback vertex set of G is also a feedback vertex set of H.

Let S be a feedback vertex set of H. If $S \subseteq V(H) \cap V(G)$, then it is also a feedback vertex set for G. Otherwise, let $v \in V(H) \setminus V(G)$ be a vertex in S. Then v must be added to subdivide an edge, say e, in G. We remove v from S and add an arbitrary endpoint of e in G to S. Then S is still a feedback vertex set for H. We repeat this process until S is a subset of $V(H) \cap V(G)$. Observe that S is a feedback vertex set of size at most k for G. Thus, the lemma holds. $\square$

Since the FVS problem is APX-hard in general graphs (by an approximation preserving reduction [56] from vertex cover problem, which is APX-hard [40]), Theorem 3.3.7 implies that FVS is APX-hard in 1-planar graphs.

To show that simple local search cannot give a constant approximation for the odd cycle transversal problem and the subset feedback vertex set problem, we construct a counter-example from a $k \times k$ grid as shown in Figure 3.2.

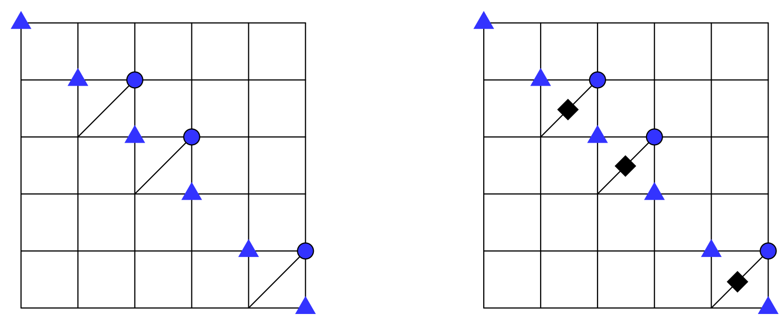

Figure 3.2: Counterexamples for local search on odd cycle transversal and subset feedback vertex set. Circle vertices represent vertices of the optimal solution, and triangle vertices represent vertices of the local search solution. The grid could be arbitrarily large. We add one edge in some diagonal cells of the grid. Left: counterexample for odd cycle transversal. Since any grid is bipartite and does not contain any odd cycle, any odd cycle in the example must contain an edge in the diagonal cell. All the vertices in the diagonal, represented by triangles, give a solution that is locally optimal, that is, we cannot improve this solution by changing a small number of vertices. This is because each triangle vertex and each new edge, together with some other edges, can form at least one odd cycle in the graph. For a constant c that is smaller than the size of optimal solution, if we remove c triangle vertices, say V', in the locally optimal solution, there will be c vertex-disjoint odd cycles in the resulting graph, each of which contains one removed triangle. Thus, there is no subset of size less than c that can replace V'. Then the ratio between the two solutions could be arbitrarily big if the gird is arbitrarily big and the number of added diagonal edges is super-constant and sublinear to the size of the diagonal. Right: counterexample for subset feedback vertex set. The diamonds represent the vertices in the given set U. Similarly, any cycle through a given vertex must contain the two edges in the diagonal cell. By the same reason, the local search solution cannot be improved.

Chapter 4: Large Induced Forests in Planar Graphs

A subset of verties S of a graph G induces *a forest* if the graph obtained from G by removing every vertex in $V(G) \setminus V(S)$ has no cycle. In this chapter, we explore two related questions concerning the existence of large induced forests in planar graphs. The first question is the conjecture proposed by Albertson and Berman [4].

Conjecture 4.0.1 (Albertson and Berman [4]). *Any planar graph of n vertices has an induced forest of size at least $\frac{n}{2}$.*

We show in Section 4.2 that the Albertson-Berman Conjecture holds for 2-outerplanar graphs. The second question is the conjecture proposed by Akiyama and Watanabe [3].

Conjecture 4.0.2 (Akiyama and Watanabe [3]). *Any bipartite planar graph of n vertices has an induced forest of size at least $\frac{5n}{8}$.*

The best result toward the Akiyama-Watanabe Conjecture is by Wang, Xie and Yu [90], who show the existence of an induced forest of size $\lceil (4n+3)/7 \rceil$. However, a stronger version of Akiyama-Watanabe Conjecture that has attracted attention recently is studied in this thesis. Its proof would immediately imply the Akiyama-Watanabe Conjecture.

Conjecture 4.0.3 (Strong Akiyama-Watanabe Conjecture). *Any triangle-free planar graph of n vertices has an induced forest of size at least $\frac{5n}{8}$.*

In Section 4.3, we show that triangle-free planar graph of n vertices has an induced forest of size at least $\frac{5n}{9}$, coming very close to proving the Strong Akiyama-Watanabe Conjecture

4.1 Definitions

We will assume that we are given a fixed embedding of a connected planar graph. A *face* of a planar graph is a connected region of the complement of the image of the drawing.

There is one *infinite* face, which we denote by f_∞. We denote the boundary of f_∞, which is the boundary of G, by ∂G. We say that a vertex v is enclosed by a cycle C if every curve from the image of v to an infinite point must cross the image of C. Every planar graph G has a corresponding dual planar graph G^*: the vertices of G^* correspond to the faces of G and the faces of G^* correspond to the vertices of G; an edge of G^* connects two vertices of G^* if the corresponding faces of G share an edge. (In this way the edges of the two graphs are in bijection.) We use $d_H(v)$ to denote the degree of vertex v in graph H and $|H|$ to denote the number of vertices of graph H.

Block-Cut Tree A *block* of a graph G is a maximal two-connected component of G. A *block-cut tree* $\mathcal{T}$ of a connected graph G is a tree where each vertex of $\mathcal{T}$ corresponds to a block and there is an edge between two vertices X, Y of $\mathcal{T}$ if two blocks X and Y share a common vertex or are incident to a common edge.

Outerplanarity A non-empty planar graph G with a given embedding is *outerplanar* (or 1-*outerplanar*) if all vertices are in ∂G. A planar graph is k-*outerplanar* for $k > 1$ if deleting the vertices in ∂G results in a $(k-1)$-outerplanar graph. A k-outerplanar graph has a natural partition of the vertices into k *layers*: L_1 is the set of vertices in ∂G; L_i is the set of vertices in the boundary of $G \backslash \cup_{j<i} L_j$. We denote $G(V, E)$ by $G(L_1, \ldots, L_k; E)$ if G is k-outerplanar. For a 2-outerplanar graph, we define the *between degree* of a vertex $v \in L_i$ to be the number of adjacent vertices in $L_j, j \neq i$.

Facial Block Let $\mathcal{C}$ be the set of facial cycles bounding finite faces of $G[L_1]$. For each $C \in \mathcal{C}$, let S_C be the set of vertices enclosed by C in G. Then we call the graph $G[C \cup S_C]$ a facial block of G.

4.2 Albertson-Berman Conjecture

Albertson and Berman conjectured that every planar graph has an induced forest on at least half of its vertices [4]; K_4 illustrates that this would be the best possible lower bound. A proof of the Albertson-Berman Conjecture would, among other things, provide an alternative proof, avoiding the 4-Color Theorem, that every planar graph has an independent set with at least one-quarter of the vertices.

The best-known lower bound toward the Albertson-Berman Conjecture has stood for 40 years: Borodin showed that planar graphs are *acyclically 5-colorable* (i.e. have a 5-coloring, every two classes of which induce a forest), thus showing that every planar graph has an induced forest on at least two-fifths of its vertices [17]. This is the best lower bound achievable toward the Albertson-Berman Conjecture via acyclic colorings as there are planar graphs which do not have an acyclic 4-coloring (for example $K_{2,2,2}$ or the octahedron).

The Albertson-Berman Conjecture has been proven for certain subclasses of planar graphs. Shi and Xu [85] showed that the Albertson-Berman Conjecture holds when $m < \lfloor 7n/4 \rfloor$ where m and n are the number of edges and vertices of the graphs, respectively. Hosono showed that outerplanar graphs have induced forests on at least two-thirds of the vertices [54].

One direction toward proving the Albertson-Berman Conjecture is to partition the vertices of graph G into sets such that each set induces a forest; the minimum number, $a(G)$, of such sets is the *vertex arboricity* of G. This implies that G has an induced forest with at least $1/a(G)$ of its vertices. Chartran and Kronk first proved that all planar graphs have vertex arboricity at most 3 [29]. Raspaud and Wang proved that $a(G) \leq 2$ if G is planar and either G has no 4-cycles, any two triangles of G are at distance at least 3, or G has at most 20 vertices; they also illustrated a 3-outerplanar graph on 21 vertices with vertex arboricity 3 [80]. Yang and Yuan [2] proved that $a(G) \leq 2$ if G is planar and has diameter at most 2. In this section, we show that:

Theorem 4.2.1. *If G is a 2-outerplanar graph, then the vertex arboricity of G is at most 2: $a(G) \leq 2$.*

4.2.1 Proof of Theorem 4.2.1

We call a set of vertex-disjoint induced forests of G *induced p-forests* if their vertices partition the vertex set of G. We consider a counterexample graph G of minimal order. By studying the structure of this minimal counterexample, we will derive a contradiction. Let e be an edge that is not in G. We observe:

Observation 4.2.2. *If $a(G \cup \{e\}) \leq 2$, then $a(G) \leq 2$.*

Observation 4.2.2 allows us to assume w.l.o.g. that G is connected (by adding edges between components while maintaining 2-outerplanarity) and that G is a disk triangulation, i.e., that every face except the outer face of G is a triangle (by adding edges inside non-triangular faces while maintaining 2-outerplanarity). Let L_1, L_2 be the bipartition of the vertices of G into layers.

Observation 4.2.3. *$G[L_1]$ is two-connected.*

Proof. Suppose otherwise. Let v be a cut vertex of $G[L_1]$. Then v is also a cut vertex of G since L_1 is the outermost layer. Let B_1, B_2 be two induced subgraphs of G that share the cut vertex v and $V(B_1) \cup V(B_2) = V$. Since G is minimal, we can partition each B_i into two induced forests F_{1i} and F_{2i}, $1 \leq i \leq 2$. W.l.o.g, we assume that $V(F_{11}) \cap V(F_{12}) = \{v\}$. Then, $F_{11} \cup F_{12}$ and $F_{21} \cup F_{22}$ are two induced p-forests of G, contradicting that G is a counter-example. □

Claim 4.2.4. *Every vertex in G has degree at least 4.*

Proof. Suppose G has a vertex v of degree at most 3. Since G is a minimal order counterexample and $G - v$ is a 2-outerplanar graph, $a(G - v) \leq 2$. Let F_0 and F_1 be two induced p-forests of $G - v$. Since v has at most 3 neighbors in G, one of F_0 or F_1, w.l.o.g. say F_0, contains at most one of these neighbors. Therefore $F_0 \cup \{v\}$ is a forest of G and $F_0 \cup \{v\}, F_1$ are two induced p-forests of G, contradicting that G is a counterexample. □

By Observation 4.2.3, $\partial G[L_1]$ is a simple cycle. Thus, the graph, say H_1^*, of $G[L_1]$ obtained from the dual graph of $G[L_1]$ by removing the dual vertex corresponding to the infinite face of $G[L_1]$ is a tree. Let B be a facial block of G that has the boundary cycle corresponding to a leaf of H_1^*. Then, either ∂B has exactly one edge not in ∂G or $B \equiv G$. In the former case, let e_B be the shared edge; in the later case, let e_B be any edge of B. Denote $L_2^B = L_2 \cap V(B)$. We have:

Claim 4.2.5. *$|L_2^B| \geq 2$.*

Proof. If $|L_2^B| = 0$, then B is a triangle since G is a disk-triangulation and vertices have degree at least 4. Then, the vertex of B that is not an endpoint of e_B has degree 2 in G, contradicting Claim 4.2.4. If $L_2^B = \{v\}$, by Claim 4.2.4, v has at least four neighbors in L_1 and thus, at least one neighbor u of v in L_1 is not an endpoint of e_B. Then the degree of u in G is 3, contradicting Claim 4.2.4. □

Claim 4.2.6. *Let v be a vertex in L_2^B that has between degree at least 3. Then, either v is a cut vertex of $G[L_2^B]$ or v is adjacent to both endpoints of e_B.*

Proof. Let v_1, v_2, v_3 be neighbors of v in ∂B in clockwise order around v. Let $\partial B[v_i, v_j]$ be the clockwise segment of ∂B from v_i to v_j, $i \neq j$. We define $C_{ij} = \partial B[v_i, v_j] \cup \{vv_i, vv_j\}$, which is a cycle of B. Assume v is not a cut vertex, at most one cycle of $\{C_{12}, C_{23}, C_{31}\}$ encloses a vertex of L_2^B, say C_{31}. Thus, v_2 is only adjacent to v and two other neighbors, say v_1', v_3', of ∂B. Since C_{12} and C_{23} enclose no vertex of L_2^B, $vv_1'v_2$ and vv_2v_3' are faces of G. If neither $v_1'v_2 = e_B$ nor $v_2v_3' = e_B$, then $d_G(v_2) = 3$, contradicting Claim 4.2.4. □

Suppose $v \in L_2^B$ is such that $d_{G[L_2^B]}(v) = 1$. By Claim 4.2.4, $d_G(v) \geq 4$ so v has between degree at least 3. Thus, by Claim 4.2.6, we have:

Observation 4.2.7. *If there exists $v \in L_2^B$ such that $d_{G[L_2^B]}(v) = 1$, then v must be adjacent to both endpoints of e_B.*

Let x_B, y_B be the endpoints of e_B. Since G is a triangulation, there is a vertex $v \in L_2^B$ such that vx_By_B is a face of G. We call v the *separating vertex* of B.

Claim 4.2.8. *If $v' \neq v$ is a vertex in L_2^B that is adjacent to both endpoints of e_B, then, v' is a cut vertex of L_2^B.*

Proof. We will prove that v' has at least one neighbor in L_2^B inside the triangle $v'x_By_B$ and at least one neighbor in L_2^B outside the triangle $v'x_By_B$; thus v' is a cut vertex of L_2^B.

By planarity, the triangle $v'x_By_B$ encloses v. Let $C_{vv'} = \{v, x_B, v', y_B\}$ which is a cycle of G. Since G is a disk triangulation and the edge x_B, y_B is embedded outside $C_{vv'}$, there must be an edge or a path inside $C_{vv'}$ connecting v and v'. Thus, v' has at least one neighbor in L_2^B inside the triangle $v'x_By_B$.

Suppose that the cycle $C_{v'} = \{\partial B \setminus e_B\} \cup \{v'x_B, v'y_B\}$ does not enclose any vertex of L_2^B. Since B is a facial block that only has e_B as a possible edge not in ∂G, every vertex in $C_{v'} \setminus \{v'\}$ must have v' as a neighbor and has degree 3, contradicting Claim 4.2.4. Thus, $C_{v'}$ must enclose at least one vertex of L_2^B. That implies v' has at least one neighbor in L_B^2 outside the triangle $v'x_By_B$ as desired. □

Since every cut vertex of L_2^B has degree at least 2 in $G[L_2^B]$, by Claim 4.2.8 and Observation 4.2.7, we have:

Observation 4.2.9. *Only the separating vertex v of B can have $d_{G[L_2]}(v) = 1$.*

If the block-cut tree of $G[L_2^B]$ has at least two vertices, let K be a leaf block of $G[L_2^B]$ that does not contain the separating vertex of B. In this case, by Observation 4.2.9, $|K| \geq 3$. Otherwise, let $K = G[L_2^B]$. We refer to the cut vertex of K in the former case and the separating vertex of B in the latter case as the *separating vertex* of K. By Claim 4.2.6, we have:

Observation 4.2.10. *Non-separating vertices of K have between degree at most 2.*

We call a triangle abc of K a *critical triangle* with top c if $d_K(c) = 2$ and c is non-separating. By Observation 4.2.10 and Claim 4.2.4, c has exactly two neighbors in L_1, that we denote by d, e (see Figure 4.1). Since G is a disk triangulation, two edges da and eb are edges of G.

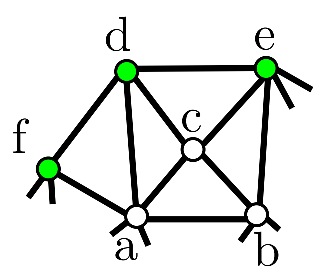

Figure 4.1: The critical triangle abc and two neighbors d, e of c in L_1. Hollow vertices are in L_2.

Claim 4.2.11. *Vertices d and e have degree at least 5.*

Proof. Neither d nor e has degree less than 4 by Claim 4.2.4. For contradiction, w.l.o.g, we assume that $d_G(d) = 4$. Recall a, c, e are three neighbors of d. Let f be the only other neighbor of d. Since $a, c \in L_2$ and ∂G is a simple cycle (Observation 4.2.3), f must be in L_1 (see Figure 4.1). Since G is a disk triangulation, $af \in E(G)$. Let G' be the graph obtained from G by contracting fd and dc and removing parallel edges. Then G' is a minor of G (and so is 2-outerplanar) with fewer vertices. Let F_0, F_1 be two induced p-forests of G' that exist by the minimality of G. Without loss of generality, we assume that $f \in F_0$. We have two cases:

1. If $b \in F_0$, then $a, e \in F_1$. If $bf \notin G$, adding c, d to F_0 does not destroy the acyclicity of F_0 in G. Thus, $F_0 \cup \{c, d\}, F_1$ are two induced p-forests of G. If $bf \in G$, the cycle $\{b, f, d, c\}$ separates a from e so a and e are in different trees in F_1. Thus, $F_0 \cup \{c\}, F_1 \cup \{d\}$ are two induced p-forests of G.

2. Otherwise, $b \in F_1$. We have three subcases:

   (a) If a, e are both in F_0, then $F_0, F_1 \cup \{c, d\}$ are two induced p-forests of G.

   (b) If $a, e \in F_1$, then, $F_0 \cup \{c, d\}, F_1$ are two induced p-forests of G.

   (c) If a, e are in different induced p-forests of G', then, $F_0 \cup \{c\}, F_1 \cup \{d\}$ are two induced p-forests of G.

In each case, the resulting p-forests contradict that G is a minimal order counter example.

□

Claim 4.2.12. $|K| \geq 4$.

Proof. If $K \neq G[L_2^B]$, as noted in the definition of K, $|K| \geq 3$. If $K = G[L_2^B]$, then by Claim 4.2.5, $|K| \geq 2$ and by Observation 4.2.9, $|K| \geq 3$. Suppose that $|K| = 3$. Then, K is a triangle. Let u, w be two neighbors of the separating vertex v in K. Then, wuv is a critical triangle with top u (or w). By Claim 4.2.4 and Observation 4.2.10, u and w both have between degree 2. Thus, u and w have a common neighbor on L_1 which therefore has degree 4, contradicting Claim 4.2.11. □

Suppose that a and b of a critical triangle abc with top c of K have a common neighbor f in L_2. We have:

Claim 4.2.13. *If fa (resp. fb) is in $\partial G[L_2^B]$, then a (resp. b) must be the separating vertex.*

Proof. For a contradiction (and w.l.o.g), we assume that $fa \in \partial G[L_2^B]$ and a is non-separating. See Figure 4.2. Let G' be the graph obtained from G by contracting ac and ce and removing parallel edges. Then, G' is a minor of G (and so is 2-outerplanar) with fewer vertices. Let F_0, F_1 be two induced p-forests of G', which are guaranteed to exist by the minimality of G. Without loss of generality, we assume that $f \in F_0$. We consider two cases:

1. If $e \in F_0$, then d, b are in F_1. If edge $fe \notin G$, then, $F_0 \cup \{a, c\}, F_1$ are two induced p-forests of G. If $fe \in G$, cycle $\{f, a, c, e\}$ separates d from b so d and b are in different trees of F_1. Thus, $F_1 \cup \{c\}, F_0 \cup \{a\}$ are two induced p-forests of G.

2. Otherwise, $e \in F_1$. We have three subcases:

   (a) If b, d are both in F_0, then $F_0, F_1 \cup \{a, c\}$ are two induced p-forests of G.

   (b) If b, d are both in F_1, then $F_0 \cup \{a, c\}, F_1$ are two induced p-forests of G.

   (c) If b, d are in different forests of G', then, $F_0 \cup \{c\}, F_1 \cup \{a\}$ are two induced p-forests of G.

In each case, the resulting p-forests contradicts that G is a minimal order counter example. □

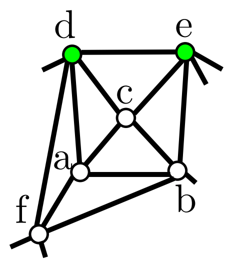

Figure 4.2: The critical triangle abc with edge $fa \in \partial G[L_2]$. Hollow vertices are in L_2.

If the edge fb is shared with another critical triangle fbg with top g, then we call $\{abc, bfg\}$ a *pair of critical triangles*. See Figure 4.3. Note that we are assuming that f is a common neighbor of a and b in L_2.

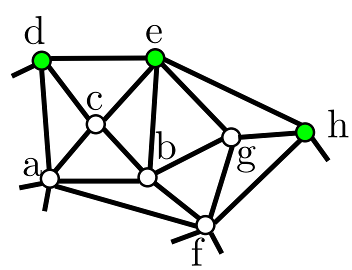

Figure 4.3: A pair of critical triangles abc and bfg. Hollow vertices are in L_2

Claim 4.2.14. *If there exists a pair of critical triangles abc and bfg in K, then b must be the separating vertex of K.*

Proof. Note that neither c nor g can be the separating vertex by definition of critical triangles. Suppose for contradiction that b is non-separating. Let d, e be two neighbors of c as defined above and i and h be the neighbors of g in L_1. We first argue that $i \equiv e$. Suppose otherwise. Since G is a disk triangulation, ec, eb, ig, ih, ib are edges of G. Let P be the subpath of ∂G between e and i that does not contain d and h. Note that P could simply be edge ei. Since B is a facial block that shares at most one edge with other facial blocks and b is non-separating, e has exactly one neighbor on P. That implies e would have degree 4, contradicting Claim 4.2.11.

We also note that $h \neq d$ (for otherwise, e would not be in L_1) and $hf \in E(G)$. See Figure 4.3. Let G' be the graph obtained from G by contracting ec, eb, eg and eh and removing parallel edges. Thus, G' is a minor of G with fewer vertices. By minimality, G' has two induced p-forests F_0, F_1. Without loss of generality, we assume that $a \in F_0$. We will reconstruct two induced p-forests of G by considering two cases:

1. If $h \in F_0$, then $d, f \in F_1$. If edge $ah \in G$, then, by planarity, d and f are in different trees of F_1. Thus, $F_1 \cup \{e, b\}$ has no cycle which implies $F_1 \cup \{e, b\}, F_0 \cup \{c, g\}$ are two induced p-forests in G. Otherwise, $F_0 \cup \{b, e\}$ has no cycle. Thus, $F_0 \cup \{b, e\}, F_1 \cup \{c, g\}$ are two induced p-forests in G.

2. Otherwise, $h \in F_1$. We have four subcases:

   (a) If d, f are both in F_1, then $F_0 \cup \{c, e, g\}, F_1 \cup \{b\}$ are two induced p-forests of G.

   (b) If d, f are both in F_0, then $F_0 \cup \{e\}, F_1 \cup \{b, c, g\}$ are two induced p-forests of G.

   (c) If $d \in F_0, f \in F_1$, then $F_0 \cup \{e, g\}, F_1 \cup \{b, c\}$ are two induced p-forests of G.

   (d) If $d \in F_1, f \in F_0$, then $F_0 \cup \{c, g\}, F_1 \cup \{b, e\}$ are two induced p-forests of G.

Thus, in all cases, the resulting induced p-forests contradict that G is a counterexample.

□

We are now ready to complete the proof of Theorem 4.2.1 by considering a triangle of K of $G[L_2^B]$, say uvw, containing the separating vertex v of K and has the most edges in common with ∂K. Since v is separating, uvw contains at least one edge in ∂K. We note that $K^* \setminus (\partial K)^*$ where $(\partial K)^*$ is the dual vertex of the infinite face of K, is a tree that we denote by T_K^*. Recall that K is a block of $G[L_2^B]$. We root T_K^* at the vertex corresponding to the triangle uvw. Consider the deepest leaf $x^* \in T_K^*$ and its parent y^*. Let abc be the triangle corresponding to x^* such that the dual edge of ab is x^*y^*. Then $d_K(c) = 2$. Since $K \geq 4$, $abc \not\equiv uvw$ and thus, it is a critical triangle with top c. Let abf be the triangle that corresponds to y^*. Note here it may be that $abf \equiv uvw$. We have three cases:

1. If $d_{T_K^*}(y^*) = 1$, then $abf \equiv uvw$. Thus, two edges fa, fb are both in ∂K but only one of the two vertices a, b can be the separating vertex of K. This contradicts Claim 4.2.13.

2. If $d_{T_K^*}(y^*) = 2$, then exactly one of two edges $af, bf \in \partial K$; w.l.o.g, we assume that $bf \in \partial K$. Then, by Claim 4.2.13, b must be the separating vertex of K. Thus, only two triangles abc and abf contain the separating vertex. Since uvw is the triangle containing the separating vertex with most edges in ∂K, $uvw \equiv abc$, contradicting our choice of triangle abc.

3. Otherwise, we have $d_{T_K^*}(y^*) = 3$. Then, none of $\{ab, bf, af\}$ is in ∂K, so $abf \not\equiv uvw$. Let z^* and t^* be the other two neighbors of y^* in T_K^* with t^* as the parent of y^*. Then, x^* and z^* have the same depth. By our choice of x^*, z^* must also be a leaf. Thus, the triangle, say bfg, corresponding to z^* is critical. Thus $\{abc, bfg\}$ is a pair of critical triangles. Since t^* is the parent of y^*, b cannot be the separating vertex of K, contradicting Claim 4.2.14.

This completes the proof of Theorem 4.2.1.

4.3 Strong Akiyama-Watanabe Conjecture

Salavatipour [84] was the first to make significant progress toward the strong Akiyama-Watanabe Conjecture. He showed that a triangle-free planar graph of order n has an induced forest of order at least $\frac{17n+24}{32}$, which is approximately $\frac{5n}{9.41}$ (we ignore the additive constant factor as it is insignificant when n is big). Dross, Montassier and Pinlou [43] improved this bound to $\frac{6n+7}{11}$ which is approximately $\frac{5n}{9.17}$. In this section, we further improve this bound to $\frac{5n}{9}$ (Theorem 4.3.1). We note that Kowalik, Lužar and Škrekovski [64] claimed a $\frac{5n}{9.01}$ bound, but there is a serious flaw in their proof, as pointed out by Dross, Montassier and Pinlou [43]. We also note that the example by Akiyama and Watanabe [3] for bipartite planar graphs implies that there exists triangle-free planar graphs of order n that have induced forests of order at most $\lceil \frac{5n}{8} \rceil$. We believe the bound of $\frac{5n}{8}$ would be a tight bound, as evidenced by the work of Alon, Mubayi and Thomas [5], who showed that if a triangle-free graph planar graph is cubic, its largest induced forest has order at least $\frac{5n}{8}$.

Theorem 4.3.1. *Every triangle-free planar graph of n vertices contains an induced forest of order at least $\frac{5n}{9}$.*

4.3.1 Previous techniques

Here in, we assume that our graph in question, denoted by G, is triangle-free. Let $n(G)$ and $m(G)$ be the number of vertices and edges of G, respectively. Let $\varphi(G)$ be the order of the largest induced forest in G. Previous techniques use discharging to prove:

$$\varphi(G) \geq an(G) - bm(G) \quad \text{for some appropriate constants } a \text{ and } b \tag{4.1}$$

Since $m(G) \leq 2n(G) - 4$ when G is triangle-free planar and $n(G) \geq 3$, Inequality (4.1) implies the existence of an induced forest of order at least $(a - 2b)n(G) + 4b$. (When $n(G) \leq 2$, the Akiyama-Watanabe conjecture becomes trivial.) Salavatipour [84] proved that Inequality (4.1) holds when (a, b) is $(\frac{29}{32}, \frac{6}{32})$, thereby, obtained the bound $\frac{17n(G)+24}{32}$. Dross, Montassier and Pinlou [43] proved that Inequality (4.1) holds when (a, b) is $(\frac{38}{44}, \frac{7}{44})$ and obtained the bound $\frac{6n(G)+7}{11}$. Kowalik, Lužar and Škrekovski [64] tried to modify Inequality (4.1) by adding an additive constant to the right-hand side, but that makes their proof erroneous as noted by Dross, Montassier and Pinlou [43].

To get a good bound on the order of the largest induced forest, one should choose a and b that maximize $(a - 2b)$. However, a and b are constrained by how many vertices one can add to the final induced forest after deleting a subset of vertices and edges of the graph. Roughly speaking, if we delete a set of α vertices, β edges from G to obtain a subgraph G' and we can add γ vertices from α deleted vertices to the largest induced forest of G' to get an induced forest in G, we should choose a and b such that:

$$a\alpha - b\beta \leq \gamma \tag{4.2}$$

If so, we can apply the inductive proof to show that Inequality (4.1) is satisfied as follows:

$$\begin{aligned} \varphi(G) &\geq \varphi(G') + \gamma \geq a(n(G) - \alpha) - b(m(G) - \beta) + \gamma \\ &\geq an(G) + bm(G) \end{aligned} \tag{4.3}$$

This process is repeated until we get down to base cases. As a result, we get a linear program. We then solve the linear program for a and b that maximize $a - 2b$. For example, Linear Program (4.4) is from the work of Dross, Montassier and Pinlou [43].

$$b \geq 0 \tag{4.4a}$$

$$0 \leq a \leq 1 \tag{4.4b}$$

$$8a - 12b \leq 5 \tag{4.4c}$$

$$a - 6b \leq 0 \tag{4.4d}$$

$$3a - 10b \leq 1 \tag{4.4e}$$

We will not try to go into details of Linear Program (4.4), but we would like to make a few points that motivate our technique. To get a better bound, one could manage to relax one or more constraints in the linear program. For technical reasons, the first two constraints and the last constraint seems unavoidable. The fourth constraint allows us to only consider graphs of maximum degree at most 5. Thus, one can relax the fourth constraint by considering graphs of higher maximum degree, say 6. But this makes the number of configurations unmanageable. The third constraint, called the *planar cube constraint*, is due to the planar cube. Specifically, by deleting a planar cube component from G, we remove 8 vertices, 12 edges and we can only add 5 vertices back to the forest since the largest induced forest of the planar cube contains 5 vertices. It turns out that we can relax the planar cube constraint in a different way by introducing two other terms to the right-hand side of Inequality (4.1). Our idea is inspired by the ideas of Lukot'ka, Mazák and Zhu [74].

4.3.2 Our technique

We use $V(G)$ and $E(G)$ to denote the set of vertices and set of edges, respectively, of G. Let H be an induced subgraph of G. The degree of H, denoted by $\deg_G(H)$, is the number of edges of G with exactly one endpoint in $V(H)$. We use H^d, H^{d+} and H^{d-} to denote an induced subgraph H of degree exactly d, at least d and at most d, respectively, of G. Two special graphs of interest in this paper are the planar cube, denoted by Q_3, and $K_{3,3}$ minus an edge, denoted by T_6 (see Figure 4.4(b)). The planar cube is a 3-regular planar graph that has 8 vertices and 12 edges (see Figure 4.4(a)).

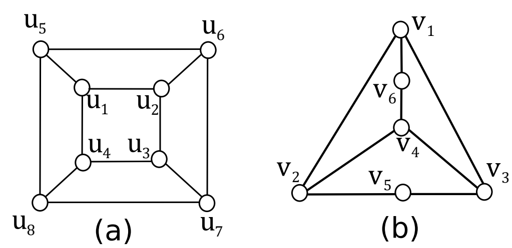

Figure 4.4: Two special graphs (a) Q_3 and (b) T_6.

Let $p(G)$ and $q(G)$ be the maximum number of Q_3^{1-} vertex-disjoint subgraphs and T_6 components of G, respectively. We will use discharging technique to prove:

$$\varphi(G) \geq an(G) - bm(G) - cp(G) - dq(G) \tag{4.5}$$

for appropriate constants a, b, c, d. Essentially, we add two terms depending on $p(G)$ and $q(G)$ to the right-hand side of Inequality (4.1). That would give us more room to find a and b that maximize $a - 2b$. Since $m(G) \leq 2n(G)$ for every triangle-free planar graphs, Inequality (4.5) gives us:

$$\varphi(G) \geq (a - 2b)n(G) - cp(G) - dq(G) \tag{4.6}$$

However, we need a bound that is independent of $p(G), q(G)$. This forces us to introduce another technical layer. In the ideal case, both $p(G)$ and $q(G)$ are 0, Inequality (4.6) gives us a good bound on $\varphi(G)$. When $p(G) + q(G)$ is at least 1, Lemma 4.3.4 and Lemma 4.3.5 allow us to reduce to the ideal case by adding a large portion of vertices from Q_3^{1-} subgraphs and T_6 components to the large induced forest.

While the introduction of Q_3 is natural, it is not clear why there is a term involving T_6 in Inequality (4.5). Indeed, our original attempt only introduces Q_3. But the bound we get is $\frac{6n(G)}{11}$ which is the same as the bound by obtained by Dross, Montassier and Pinlou [43]. However, when looking at the linear program carefully, we realize that T_6 is another obstruction to obtain a better bound. Thus, we further introduce T_6 into the linear program.

The main tool in our proof is Theorem 4.3.2 whose proof is deferred to Section 4.3.4.

Theorem 4.3.2. *If a, b, c, d are constants that satisfy all constraints in the Linear Program (4.7), then every triangle-free planar graph G has an induced forest of order at least $an(G) - bm(G) - cp(G) - dq(G)$.*

$$a \geq 0 \tag{4.7a}$$
$$1 - a \geq 0 \tag{4.7b}$$
$$b \geq 0 \tag{4.7c}$$
$$c \geq 0 \tag{4.7d}$$
$$d \geq 0 \tag{4.7e}$$
$$1 - a + b - c \geq 0 \tag{4.7f}$$
$$1 - a + b - d \geq 0 \tag{4.7g}$$
$$5b - a \geq 0 \tag{4.7h}$$
$$5 - 8a + 12b + c \geq 0 \tag{4.7i}$$
$$4 - 6a + 8b + d \geq 0 \tag{4.7j}$$
$$5 - 8a + 13b \geq 0 \tag{4.7k}$$
$$5 + 13b - 8a + c - d \geq 0 \tag{4.7l}$$
$$4 + 9b - 6a - d \geq 0 \tag{4.7m}$$
$$5 - 8a + 14b - d \geq 0 \tag{4.7n}$$
$$5 - 8a + 14b - c \geq 0 \tag{4.7o}$$
$$5 - 8a + 15b - c - d \geq 0 \tag{4.7p}$$
$$3 - 5a + 10b - c \geq 0 \tag{4.7q}$$
$$3 - 5a + 10b - d \geq 0 \tag{4.7r}$$
$$3 - 4a + 4b \geq 0 \tag{4.7s}$$

Corollary 4.3.3. *If a, b, c, d are constants that satisfy all constraints in the Linear Program (4.7), then every triangle-free planar graph G that contains no Q_3^{1-} subgraph and T_6 component has an induced forest of order at least $(a - 2b)n(G)$.*

Proof. Since $p(G)$ and $q(G)$ are both 0, Theorem 4.3.2 implies that G has an induced forest of order at least $an(G) - bm(G)$. Thus, the corollary follows from the fact that $m(G) \leq 2n(G)$. □

4.3.3 Proof of Theorem 4.3.1

Lemma 4.3.4. *If H is a Q_3^{3-} subgraph of a planar graph G, then any induced forest F in $G \setminus H$ can be extended to an induced forest of G of order $|F| + 5$.*

Proof. Let $a, b, c \in \{u_1, u_2, \ldots, u_8\}$ be three highest-degree vertices of H in G. If a, b and c are pairwise non-adjacent. By symmetry of Q_3, we can assume w.l.o.g that a, b, c are u_1, u_3, u_6, respectively. Then, $F \cup \{u_2, u_4, u_5, u_7, u_8\}$ is an induced forest in G. Thus, we can suppose that two vertices, say a, b, are adjacent. We consider two cases:

Case 1 Three vertices a, b, c induce a connected subgraph of H. Then, there is a face in any planar embedding of Q_3 that contains all a, b and c. By symmetry of Q_3, we can assume that a, b, c are u_1, u_2, u_3, respectively. Since $\deg_G(H) \leq 3$, at least one vertex in $\{u_1, u_3\}$ is a 4^--vertex of G. Let x be a 4^--vertex in $\{u_1, u_3\}$. Then, $F \cup \{x, u_4, u_5, u_6, u_7\}$ is an induced forest in G.

Case 2 Three vertices a, b, c induce a dis-connected subgraph of H. By symmetry of Q_3, we can assume that a, b, c are u_1, u_2, u_7, respectively. Then, $F \cup \{u_3, u_4, u_5, u_6, u_8\}$ is an induced forest in G.

□

Lemma 4.3.5. *If K is a T_6^{3-} subgraph of G, then any forest F in $G \setminus K$ can be extended to an induced forest of G of order $|F| + 4$.*

Proof. By symmetry of T_6, we can assume w.l.o.g that cycle $C = v_1v_2v_5v_3$ has the highest degree among cycles inducing faces of K. Let $X = \{v_1, v_6, v_4, v_5\}$. Suppose K has a between vertex, say v, that has at least two non-K edges in G. By the degree assumption of C, v must be a vertex in C. If $v \notin X$, then $F \cup X$ is an induced forest of G of order $|F| + 4$. If $v \in X$, then $F \cup \{v_2, v_4, v_6, v_3\}$ is an induced forest of G.

Thus, we can assume that every vertex of K has at most one non-K edge. If at most one vertex in X is a between vertex of K, then $F \cup X$ is an induced forest of G. Thus, we can assume that at least two vertices in X are between. Since $\deg_G(K) \leq 3$, at most one of two vertices v_2 and v_3 is a between vertex. Let x be the non-between vertex in $\{v_2, v_3\}$. By the degree assumption of C, at most one vertex among $\{v_4, v_6\}$ is between. We have two cases:

Case 1 No vertex in $\{v_4, v_6\}$ is between. Then, $F \cup \{v_2, v_4, v_6, v_3\}$ is an induced forest of G.

Case 2 Exactly one vertex in $\{v_4, v_6\}$ is between. Let y be the non-between vertex in $\{v_4, v_6\}$. If both v_1 and v_5 are between, then v_2 and v_3 have no non-K edge since $\deg_G(K) \leq 3$. Thus, $F \cup \{v_2, v_4, v_6, v_3\}$ is an induced forest of G. If v_1 is between and v_5 is non-between, then $F \cup \{v_5, v_4, v_6, x\}$ is an induced forest of G. Otherwise, v_5 is between and v_1 is non-between. Then, $F \cup \{v_1, x, y, v_5\}$ is an induced forest of G.

□

Observation 4.3.6. *Any two Q_3^{2-} subgraphs of G must be vertex-disjoint.*

Proof. We observe that any non-trivial edge cut of Q_3 has at least 3 edges. Let H and K be two Q_3^{2-} subgraphs of G that share a subset of vertices X. Then, the cut $(V(H) \setminus X, X)$ has at least 3 edges. Thus, $\deg_G(K) \geq 3$, contradicting that K is a Q_3^{2-} subgraph. □

Proof of Theorem 4.3.1. Let $\rho(G) = p(G) + q(G) + n(G)$. We prove Theorem 4.3.1 by induction on $\rho(G)$. The base case is when $\rho(G) = 0$, Theorem 4.3.1 trivially holds. We consider three cases:

Case 1 G has no Q_3^{1-} subgraph or T_6 component. Then, $p(G) + q(G) = 0$. Using a linear programming solver [1] to solve Linear Program 4.7, we found that $a - 2b$ is

[1] We use lp_solve package. The full implementation can be found at `http://web.engr.oregonstate.edu/~lehu/res/lp_final.lp`

maximized when $a = \frac{25}{27}, b = c = \frac{5}{27}, d = \frac{2}{27}$. Corollary 4.3.3 implies that if G has an induced forest F of order at least $\frac{5n(G)}{9}$.

Case 2 G contains a T_6 component, then $p(G \setminus T_6) \leq p(G)$ and $q(G \setminus T_6) < q(G)$. Thus, $\rho(G \setminus T_6) < \rho(G)$. By induction, $\varphi(G \setminus T_6) \geq \frac{5n(G \setminus T_6)}{9} = \frac{5(n(G)-6)}{9}$. By Lemma 4.3.5, we can collect 4 vertices from T_6. That implies:

$$\varphi(G) \geq \varphi(G \setminus T_6) + 4 \geq \frac{5(n(G)-6)}{9} + 4 > \frac{5n(G)}{9}$$

Case 3 G contains a Q_3^{1-} subgraph, say H. Since H has degree at most 1 in G, removing H from G can create at most one T_6 component and at most one new Q_3^{1-} subgraph. Thus, $p(G \setminus H) \leq p(G)$ and $q(G \setminus H) \leq q(G) + 1$. Since $n(G \setminus H) \leq n(G) - 8$, we have $\rho(G \setminus H) < \rho(G)$. By induction, we have $\varphi(G \setminus H) \geq \frac{5(n(G)-8)}{9}$. By Lemma 4.3.4, we can collect 5 vertices from H. That implies:

$$\varphi(G) \geq \varphi(G \setminus H) + 5 \geq \frac{5(n(G)-8)}{9} + 5 > \frac{5n(G)}{9}$$

4.3.4 Proof of Theorem 4.3.2

Let G be a counter-example of minimal order. We begin our proof with Observation 4.3.7, that we will frequently make use of in deriving a contradiction.

Observation 4.3.7. *Let L be a subgraph of G. Let $\alpha, \beta, \gamma, \eta \geq 0$ be such that:*

$$\begin{aligned} \alpha &= n(G) - n(G \setminus L) \\ \beta &\leq m(G) - m(G \setminus L) \\ \gamma &\leq p(G) - p(G \setminus L) \\ \eta &\leq q(G) - q(G \setminus L) \end{aligned} \tag{4.8}$$

If we can collect λ vertices from L, then, $\lambda - \alpha a + \beta b + c\gamma + d\eta$ must be negative.

Proof. Suppose that $\lambda - \alpha a + \beta b + c\gamma + d\eta$ is non-negative. Since G is a minimal counter-example, $G \setminus L$ has an induced forest of order at least $an(G \setminus L) - bm(G \setminus L) - cp(G \setminus L) - dq(G \setminus L)$ which is at least:

$$an(G) - bm(G) - cp(G) - dq(G) + \beta b + c\gamma + d\eta - \alpha a.$$

By collecting λ vertices from L, we get an induced forest in G of order at least:

$$an(G) - bm(G) - cp(G) - dq(G) + \lambda + \beta b + c\gamma + d\eta - \alpha a$$

Since $\lambda - \alpha a + \beta b + c\gamma + d\eta$ is non-negative, $\varphi(G) \geq an(G) - bm(G) - cp(G) - dq(G)$, contradicting that G is a counter-example. $\square$

Overview of the proof. Our proof of Theorem 4.3.2 relies on the following structural theorem that was proved by Salavatipour [84].

Theorem 4.3.8. *If G is a two-edge connected triangle-free planar graph, then, G contains (1) a 2^--vertex, or (2) a 4-face with at least one 3-vertex, or (3) a 5-face with at least four 3-vertices.*

At high level, we build a linear program, called $\mathcal{LP}$, that initially contains trivial constraints (4.7a), (4.7b), (4.7c), (4.7d) and (4.7e). We then consider a finite set of subgraphs, say $\mathcal{L}$, that a triangle-free planar graph can have. For each subgraph, say H, in $\mathcal{L}$, by removing H from G, we reduce the number of vertices, edges, Q_3^{1-} subgraphs and T_6 components of G by at least, say, α, β, γ and η, respectively. Then, we show that we can add λ vertices from H to a large induced forest of $G \setminus H$ to get an induced forest of G. Observation 4.3.7 tells us that if we choose a, b, c and d such that $\lambda - \alpha a + \beta b + c\gamma + d\eta \geq 0$, then G cannot be a counter-example. Thus, a counter-example graph G cannot contain the subgraph H. In other words, by adding the constraint $\lambda - \alpha a + \beta b + c\gamma + d\eta \geq 0$ to $\mathcal{LP}$, we *exclude* H from G. We repeat this argument for every subgraph in $\mathcal{L}$ and keep adding linear constraints along the way to $\mathcal{LP}$. Finally, we get a linear program represented by $\mathcal{LP}$ and we show that $\mathcal{LP}$ is equivalent to Linear Program (4.7) after removing redundant constraints. Thus, by choosing a, b, c and d that satisfy Linear Program (4.7), the counter-example G does not exist, thereby, proving Theorem 4.3.2.

In Subsection 4.3.4.2, we prove that G is two-edge connected and $\delta(G) \geq 3$. In Subsection 4.3.4.6, we prove that G has no 4-face with at least one 3-vertex. In Subsection 4.3.4.7, we prove that G has no 5-face with at least four 3-vertices. This is a contradiction by Theorem 4.3.8.

4.3.4.1 Excluding Q_3^d and T_6^d subgraphs

In this section, by adding more constraints to $\mathcal{LP}$, we will prove that the minimal counter example G cannot contain any Q_3^d or T_6^d subgraph for $d \leq 5$ if $\mathcal{LP}$ is satisfied.

Claim 4.3.9. *G has no Q_3 component.*

Proof. Let H be a Q_3 component of G. By Lemma 4.3.4, we can collect 5 vertices from H. Since Q_3 has 8 vertices, 12 edges, by Observation 4.3.7 with $L = Q_3$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 12, 1, 0, 5)$, $5 - 8a + 12b + c$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.9) to $\mathcal{LP}$.

$$5 - 8a + 12b + c \geq 0 \tag{4.9}$$

□

Claim 4.3.10. *G has no T_6 component.*

Proof. Let H be a T_6 component of G. By Lemma 4.3.5, we can collect 4 vertices from H. By Observation 4.3.7 with $L = T_6$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (6, 8, 0, 1, 4)$, $4 - 6a + 8b + d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.10) to $\mathcal{LP}$.

$$4 - 6a + 8b + d \geq 0 \tag{4.10}$$

□

Claim 4.3.10 implies that if $\mathcal{LP}$ is satisfied, the counter-example G has no T_6 component.

Claim 4.3.11. *G excludes Q_3^{1-} as a subgraph.*

Proof. By Claim 4.3.9, we only need to exclude Q_3^1 from G. Let H be a Q_3^1 subgraph of G. Let $G' = G \setminus H$. If H is adjacent to a Q_3^2 subgraph of G, then $p(G') = p(G)$ and $q(G') = q(G) = 0$. By Lemma 4.3.4, we can collect 5 vertices from H. By applying Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 13, 0, 0, 5)$, $5 - 8a + 13b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.11) to $\mathcal{LP}$.

$$5 - 8a + 13b \geq 0 \tag{4.11}$$

If H is not adjacent to a Q_3^2 subgraph, then $p(G') = p(G) - 1$. Note that G' can has a T_6 component if H is adjacent to a T_6^1 subgraph in G. By Observation 4.3.7 with $L = H$

and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 13, 1, -1, 5)$, $5 - 8a + 13b + c - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.12) to $\mathcal{LP}$.

$$5 - 8a + 13b + c - d \geq 0 \tag{4.12}$$

$\square$

Claim 4.3.9 and 4.3.11 imply that if $\mathcal{LP}$ is satisfied, G has no Q_3^{1-} subgraph. Herein, we can assume that the counter-example graph G has $p(G) = q(G) = 0$.

Claim 4.3.12. *G excludes T_6^{1-} as a subgraph.*

Proof. By Claim 4.3.10, we only need to exclude T_6^1 from G. Let K be a T_6^1 subgraph of G. Let $H_1, \ldots, H_t$ be the subgraphs of G such that H_j is a Q_3^1 subgraph of $G \setminus \{K \cup \{H_1, \ldots, H_{j-1}\}\}$ and H_j is adjacent to H_{j-1} in G. Let t be the maximum index such that $G \setminus \{K \cup H_1 \cup \ldots \cup H_t\}\}$ contains no Q_3^1 subgraph. It may be that none of H_j exists and we define $t = 0$ in this case. Let $KH = K \cup \{H_1, \ldots, H_t\}$. We have $\deg_G(KH) = 1$. Thus, $G \setminus KH$ cannot contain any Q_3 component, since otherwise, it would be Q_3^1 in G, contradicting Claim 4.3.11. Since $\deg_G(KH) = 1$, $G \setminus KH$ contains at most one T_6 component. By Lemma 4.3.4 and Lemma 4.3.5, we can collect $5t + 4$ vertices from KH. By Observation 4.3.7 with $L = KH$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8t + 6, 13t + 9, 0, -1, 5t + 4)$, $(5t + 4) - (8t + 6)a + (13t + 9)b - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.13) to $\mathcal{LP}$.

$$(5t + 4) - (8t + 6)a + (13t + 9)b - d \geq 0 \tag{4.13}$$

$\square$

Claim 4.3.13. *G excludes Q_3^{2-} as a subgraph.*

Proof. By Claim 4.3.11, we only need to exclude Q_3^2 from G. Let H be a Q_3^2 subgraph of G. Suppose that $G \backslash H$ contains a T_6 component, say K. By Claim 4.3.12, K is the only T_6 component of $G \backslash H$. Since G excludes T_6^{1-} by Claim 4.3.12, K must be T_6^2 in G and hence, two edges incident to H are between H and K. Thus, by Claim 4.3.11, $p(G \setminus H) = 0$. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 14, 0, -1, 5)$, $5 - 8a + 14b - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.14) to $\mathcal{LP}$.

$$5 - 8a + 14b - d \geq 0 \tag{4.14}$$

Thus, we may assume that $G \setminus Q_3^2$ has no T_6 component for any Q_3^2 subgraph of G. Without loss of generality, we choose H to be a Q_3^2 subgraph such that $G \setminus H$ has the least number of Q_3^{1-} subgraphs. By Claim 4.3.11, $G \setminus H$ has at most two Q_3^{1-} subgraphs. If $G \setminus H$ has exactly one Q_3^{1-} subgraph, say M, then M must be adjacent to H. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 14, -1, 0, 5)$, $5 - 8a + 14b - c$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.15) to $\mathcal{LP}$.

$$5 - 8a + 14b - c \geq 0 \tag{4.15}$$

If $G \setminus H$ has two Q_3^{1-} subgraphs. By Claim 4.3.11, two Q_3^{1-} subgraphs must be Q_3^1 subgraphs in $G \setminus H$. By our choice of H, we conclude that, for any Q_3^2 subgraph of G, $G \setminus Q_3^2$ must have exactly two Q_3^1 subgraphs. Since G excludes Q_3^1 by Claim 4.3.11, any Q_3^2 subgraph of G must adjacent to two other Q_3^2 subgraphs. Let $\mathcal{H}$ be a graph such that each vertex of $\mathcal{H}$ corresponds to a Q_3^2 subgraph of G and each edge of $\mathcal{H}$ connects two corresponding adjacent Q_3^2 subgraphs of G. Then, $\mathcal{H}$ is a 2-regular graph. In other words, $\mathcal{H}$ is a collection of cycles. By Lemma 4.3.4, we can collect $5|V(\mathcal{H})|$ vertices from Q_3^2 subgraphs of G. Let L be the subgraph of G induced by the vertices in all Q_3^2 subgraphs. By Observation 4.3.7 with L and $(\alpha, \beta, \gamma, \eta, \lambda) = (8|V(\mathcal{H})|, 13|V(\mathcal{H})|, 0, 0, 5|V(\mathcal{H})|)$, $|V(\mathcal{H})|(5 - 8a + 13b)$ must be negative, this contradicts Inequality (4.11). □

Claim 4.3.14. *G excludes Q_3^{3-} as a subgraph.*

Proof. By Claim 4.3.13, we only need to exclude Q_3^3 from G. Let H be a Q_3^3 subgraph in G. Observe that $G \setminus H$ contains at most one Q_3^{1-} subgraph since otherwise, G has a Q_3^{2-} subgraph, contradicting Claim 4.3.13. Similarly, by Claim 4.3.12, $G \setminus H$ contains at most one T_6 component. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 15, -1, -1, 5)$, $5 - 8a + 15b - c - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.16) to $\mathcal{LP}$.

$$5 - 8a + 15b - c - d \geq 0 \tag{4.16}$$

□

We obtain the following corollary of Claim 4.3.14.

Corollary 4.3.15. *If H is a subgraph of degree 2 of G and $\mathcal{LP}$ is satisfied, then $G \setminus H$ has no Q_3^{1-} subgraph.*

Claim 4.3.16. *G excludes T_6^{2-} as a subgraph.*

Proof. By Claim 4.3.12, we only need to exclude T_6^2 from G. Let H be a T_6^2 subgraph of G. By Corollary 4.3.15, $G \setminus H$ has no Q_3^{1-} subgraph. By Claim 4.3.12, $G \setminus H$ has at most one T_6 component. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (6, 10, 0, -1, 4)$, $4 - 6a + 10b - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.17) to $\mathcal{LP}$.

$$4 - 6a + 10b - d \geq 0 \tag{4.17}$$

□

Claim 4.3.17. *G has no 5^+-vertex.*

Proof. Let v be a 5^+ vertex in G and $G' = G - \{v\}$. Suppose that G' has a Q_3^{1-} subgraph H. By planarity, v must be embedded in one face of H. Since faces of H has length 4 and G is triangle-free, v has at most two neighbors in H. That implies H is a Q_3^{3-} subgraph of G, contradicting Claim 4.3.14. Thus $p(G') = 0$. Suppose that G' has a T_6 component K. By planarity, v must be embedded in one face of K. Since G is triangle-free, v has at most two neighbors in K. That implies K is T_6^{2-}, contradicting Claim 4.3.16. Thus $q(G') = 0$. By Observation 4.3.7 with $L = v$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (1, 5, 0, 0, 0)$, $5b - a$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.18) to $\mathcal{LP}$.

$$5b - a \geq 0 \tag{4.18}$$

□

Lemma 4.3.18. *If H is a Q_3^{5-} subgraph of G and every vertex of H has degree at most 4 in G, then any induced forest F of $G \setminus H$ can be extended to an induced forest of G with order $|F| + 5$.*

Proof. By Lemma 4.3.4, we can assume that H is Q_3^4 or Q_3^5. By symmetry of Q_3, we can choose an embedding of G such that the inner-most face $u_1u_2u_3u_4$, denoted by f, of H has the most number of 3-vertices. We have three cases:

Case 1 f has at least three 3-vertices, say u_1, u_2, u_3, then $F \cup \{u_1, u_2, u_3, u_6, u_8\}$ is an induced forest of G.

Case 2 f has only one 3-vertex, say u_1, then every face that contains u_1 on the boundary must has at least three 4-vertices by the choice of the inner-most face of H. That implies $\deg_G(H) \geq 6$, contradicting that H is a Q_3^{5-} subgraph of G.

Case 3 f has exactly two 3-vertices. By the choice of f, every face of H has at most two 3-vertices. Suppose that H has two adjacent 3-vertices. By symmetry of Q_3, we can choose an embedding of G such that two 3-vertices of f are adjacent. We can assume w.l.o.g they are u_1 and u_2. Thus, u_5 and u_6 must be 4-vertices. Since $\deg_G(H) \leq 5$, at most one vertex in $\{u_7, u_8\}$ is a 4-vertex. Let u^* be a 3-vertex in $\{u_7, u_8\}$. Since non-H edges of u_6 and u_4 are embedded in different faces of G, $F \cup \{u_6, u_1, u_2, u_4, u^*\}$ is an induced forest of G. If H has no two adjacent 3-vertices, we can assume w.l.o.g that u_1 and u_3 are two 3-vertices of H. Thus, u_2, u_4, u_5, u_7 are 4-vertices. Since $\deg_G(H) \leq 5$, at least one vertex in $\{u_6, u_8\}$ is a 3-vertex. We define $u^* = u_2$ if u_8 is a 4-vertex and $u^* = u_4$ if u_6 is a 4-vertex. Then, $F \cup \{u_1, u_3, u_6, u_8, u^*\}$ is an induced forest of G.

□

Claim 4.3.19. *G excludes Q_3^{4-} as a subgraph.*

Proof. By Claim 4.3.14, we only need to exclude Q_3^4 from G. Let H be a Q_3^4 subgraph of G. By Claim 4.3.17, between vertices of H are 4-vertices. Observe that $G \setminus H$ contains at most one Q_3^{1-} subgraph since otherwise, G has a Q_3^{3-} subgraph, contradicting Claim 4.3.14. Similarly, by Claim 4.3.16, $G \setminus H$ has at most one T_6 component. By Lemma 4.3.18, we can collect 5 vertices from H. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 16, -1, -1, 5)$, $5 - 8a + 16b - c - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.19) to $\mathcal{LP}$.

$$5 - 8a + 16b - c - d \geq 0 \tag{4.19}$$

□

Claim 4.3.20. *G excludes T_6^{3-} as a subgraph.*

Proof. By Claim 4.3.16, we only need to exclude T_6^3 from G. Let H be a T_6^3 subgraph of G. Observe that $G \setminus H$ has no Q_3^{1-} subgraph since such a subgraph would have degree at most 4 in G, contradicting Claim 4.3.19. By Claim 4.3.16, $G \setminus H$ has at most one T_6 component. By Lemma 4.3.5, we can collect 4 vertices from H. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (6, 11, 0, -1, 4)$, $4-6a+11b-d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.20) to $\mathcal{LP}$.

$$4 - 6a + 11b - d \geq 0 \tag{4.20}$$

□

Claim 4.3.21. *G excludes any T_6^{5-} subgraph that has all between vertices on the same face.*

Proof. Suppose that G contains a T_6^{5-} subgraph H as in the claim. By Claim 4.3.20, $\deg_G(H) \geq 4$. By symmetry of H, we can assume w.l.o.g that the outer face $v_1v_2v_5v_3$ of H contains all between vertices. Let K be the subgraph of G induced by $\{v_1, v_2, v_3, v_4, v_6\}$. By Claim 4.3.17, v_1 has at most one non-H incident edge. Thus, we can collect $\{v_1, v_4, v_6\}$ from K. Since $\deg_G(K) \leq 5$, by Claim 4.3.19, $G \setminus K$ has at most one Q_3^{1-} subgraph. If $G \setminus K$ has exactly one Q_3^{1-} subgraph, the Q_3^{1-} subgraph in $G \setminus K$ must has at least three edges to K in G. That implies $G \setminus K$ has no T_6 component since such a component would have degree at most 2 in G, contradicting Claim 4.3.20. Since $\deg_G(H)$ is at least 4 and v_5 has degree at most 4, $m(G) - m(G \setminus K) \geq 10$. By Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 10, 0, -1, 3)$, $3 - 5a + 10b - c$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.21) to $\mathcal{LP}$.

$$3 - 5a + 10b - c \geq 0 \tag{4.21}$$

If $G \setminus K$ has no Q_3^{1-} subgraph, by Claim 4.3.20, $G \setminus K$ has at most one T_6 component. By Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 10, -1, 0, 3)$, $3 - 5a + 10b - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.22) to $\mathcal{LP}$.

$$3 - 5a + 10b - d \geq 0 \tag{4.22}$$

□

Claim 4.3.22. *G excludes Q_3^{5-} as a subgraph.*

Proof. By Claim 4.3.19, we only need to exclude Q_3^5 from G. Let H be a Q_3^5 subgraph of G. By Claim 4.3.17, between vertices of H has degree exactly 4. $G \backslash H$ must have at most one Q_3^{1-} subgraph since otherwise, there would be a Q_3^{4-} subgraph in G, contradicting Claim 4.3.19. Similarly, by Claim 4.3.20, $G \setminus H$ has at most one T_6 component. By Lemma 4.3.18, we can collect 5 vertices from H. By Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 17, -1, -1, 5)$, $5 - 8a + 17b - c - d$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.23) to $\mathcal{LP}$.

$$5 - 8a + 17b - c - d \geq 0 \tag{4.23}$$

□

Claim 4.3.23. *If H is a connected subgraph of G, then $G \setminus H$ has no Q_3^{1-} subgraph and T_6 component.*

Proof. Suppose that $G \backslash H$ contains a Q_3^{1-} subgraph K. Since H is connected, its vertices are embedded in on face of K, say the infinite face. Thus, by Claim 4.3.17, G has at most 4 edges connecting vertices of H and vertices of K. Since K has degree at most one in $G \setminus H$, K has degree at most 5 in G, contradicting Claim 4.3.22. Suppose that $G \setminus H$ contains a T_6 component M. Since H is connected, their vertices are embedded inside one face of M. Thus, there exists one face of M contains all between vertices. Since G only has 4^--vertices, $\deg_G(M) \leq 5$, contradicting Claim 4.3.21. □

4.3.4.2 Excluding low degree vertices

As shown in Section 4.3.4.1, if $\mathcal{LP}$ is satisfied, G has $p(G) = 0$ and $q(G) = 0$. Thus, we only need to prove $\varphi(G) \geq an(G) - bm(G)$ to obtain a contradiction.

Claim 4.3.24. *G is two-edge connected.*

Proof. Suppose that the claim fails, then either G is disconnected or G is connected and has a bridge e. If G is disconnected, let G_1 be any connected component of G and $G_2 = G \setminus G_1$. If G is connected and has a bridge e, let G_1, G_2 be the two components of $G \setminus \{e\}$. Since $\deg_G(G_1) \leq 1$, by Claim 4.3.22, $p(G_1) = p(G_2) = 0$. By Claim 4.3.20, $q(G_1) = q(G_2) = 0$. Since G_1, G_2 has strictly smaller order than G, they have two induced forests F_1, F_2 of order at least $an(G_1) - bm(G_1), an(G_2) - bm(G_2)$, respectively. Thus, $F_1 \cup F_1$ is an induced forest of G of order at least:

$$a(n(G_1) + n(G_2)) - b(m(G_1) + m(G_2)) \geq an(G) - bm(G)$$

This contradicts that G is a counter-example. □

A direct corollary of Claim 4.3.24 is that $\delta(G) \geq 2$.

Claim 4.3.25. *If v is a 2-vertex, then its two neighbors must have another common neighbor.*

Proof. Let G' be the graph obtained from G by contracting an incident edge of v. Suppose that v is the only common neighbor of its neighbors, then, G' is triangle-free. Let u be the neighbor that v is contracted to. Any Q_3^{1-} subgraph and T_6 component of G' must contain u. Thus, $p(G') + q(G') \leq 1$. Since G' has strictly smaller order than G, G' has a forest F' of order at least $an(G') - bm(G') - cp(G') - dq(G')$. We note that $n'(G) = n(G) - 1$ and $m(G') = m(G) - 1$. If $p(G') = 1$, $F' \cup \{v\}$ is an induced forest in G of order at least:

$$1 + a(n(G) - 1) - b(m(G) - 1) - c = an(G) - bm(G) + 1 - a + b - c$$

Thus, by adding Inequality (4.24) to $\mathcal{LP}$, we deduce that $\varphi(G) \geq an(G) - bm(G)$, contradicting that G is a counter-example.

$$1 - a + b - c \geq 0 \tag{4.24}$$

If $q(G') = 1$, $F' \cup \{v\}$ is an induced forest in G of order at least:

$$1 + a(n(G) - 1) - b(m(G) - 1) - d = an(G) - bm(G) + 1 - a + b - d$$

Thus, by adding Inequality (4.25) to $\mathcal{LP}$, we obtain a contradiction.

$$1 - a + b - d \geq 0 \tag{4.25}$$

□

Claim 4.3.26. *None neighbor of a 2-vertex is a 4-vertex.*

Proof. Suppose that a neighbor u of a 2-vertex v is a 4-vertex. Let $G' = G - \{u, v\}$. By Claim 4.3.23, $p(G') = q(G') = 0$. Observe that we can add v to any induced forest of G' to get an induced forest of G. By Observation 4.3.7 with $L = uv$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (2, 5, 0, 0, 1)$, $1-2a+5b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.26) to $\mathcal{LP}$.

$$1 - 2a + 5b \geq 0 \tag{4.26}$$

□

Claim 4.3.27. *None neighbor of a 2-vertex is a 2-vertex.*

Proof. Let u and v be two adjacent 2-vertices. Let w and w' be other neighbors of u and v, respectively. By Claim 4.3.26, w and w' are 3^--vertices. By Claim 4.3.25, w and w' must be adjacent. If both w and $w'$are 2-vertices, then G is a cycle of 4 vertices. Since G has a forest of order 3, by Observation 4.3.7 with $L = G$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (4, 4, 0, 0, 3)$, $3-4a+4b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.27) to $\mathcal{LP}$.

$$3 - 4a + 4b \geq 0 \tag{4.27}$$

Thus, we may assume w has degree exactly 3. By Claim 4.3.23, $G - \{u, v, w\}$ has no Q_3^{1-} subgraph or T_6 component. Since we can collect $\{u, v\}$, by Observation 4.3.7 with $L = G[\{u, v, w\}]$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (3, 5, 0, 0, 2)$, $2 - 3a + 5b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.28) to $\mathcal{LP}$.

$$2 - 3a + 5b \geq 0 \tag{4.28}$$

□

Claim 4.3.28. *Any 3-vertex in G is adjacent to at most one 2-vertex.*

Proof. Suppose otherwise. Let w be a 3-vertex that is adjacent to two 2-vertices u and v. Let $G' = G - \{u, v, w\}$. By Claim 4.3.23, $p(G') = q(G') = 0$. Since we can collect $\{u, v\}$, by Observation 4.3.7 with $L = \{u, v, w\}$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (3, 5, 0, 0, 2)$, $2 - 3a + 5b$ must be negative, contradicting Inequality (4.28). □

Lemma 4.3.29. $\delta(G) \geq 3$.

Proof. Let w_1 be a 2-vertex of G with two neigbors w_2, w_4. By Claim 4.3.25, w_2 and w_4 must have another common neighbor, say w_3. Let C be the cycle $w_1w_2w_3w_4$. By Claim 4.3.27 and 4.3.26, w_2 and w_4 are 3-vertices. Let u be the non-C neighbor of w_2. By Claim 4.3.28, u and w_3 are a 3^+-vertices. Since G is triangle free, u cannot be a neighbor of w_3. Let H be the induced subgraph of G induced by $\{w_1, w_2, w_3, w_4, u\}$. We can collect 3 vertices w_4, w_1, w_2 from H. By Claim 4.3.23, $p(G \setminus H) = q(G \setminus H) = 0$. If $m(G) - m(G \setminus H)$ is at least 9, by Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 9, 0, 0, 3)$, $3 - 5a + 9b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.29) to $\mathcal{LP}$.

$$3 - 5a + 9b \geq 0 \tag{4.29}$$

Thus, we can assume $m(G)-m(G\setminus H) \leq 8$. That implies u must be a neighbor of w_4 and both u and w_3 are a 3-vertices (see Figure 4.5(a)). Since G is two connected, the non-H neighbor of u must be embedded in the same side with the non-H neighbor of w_3 with respect to the cycle $uw_4w_3w_2$. Let v be the non-H neighbor of w_3. Let K be the subgraph of G induced by $\{w_1, w_2, w_3, w_4, u, v\}$. If u and v are adjacent, K is T_6^{3-}, contradicting Claim 4.3.16. Thus, u and v are not adjacent. Hence, $m(G \setminus K) \geq 9$. Observe that we can collect $\{u, w_1, w_2, w_3\}$ from K. By Claim 4.3.23, $p(G \setminus K) = q(G \setminus K) = 0$. By Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (6, 9, 0, 0, 4)$, $4 - 6a + 9b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.30) to $\mathcal{LP}$.

$$4 - 6a + 9b \geq 0 \tag{4.30}$$

□

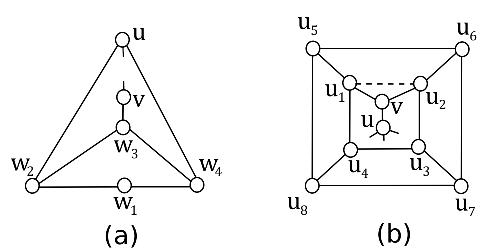

Figure 4.5: (a) A configuration in the proof of Lemma 4.3.29 (b) A configuration in the proof of Claim 4.3.30

4.3.4.3 Avoiding small vertex cut

A separating cycle is a cycle that separates the plane into two regions, each contains at least one vertex of G that is not on its boundary.

Claim 4.3.30. *Let v be a 3-vertex that is adjacent to a 4-vertex u. Then two neighbors of v other than u must share a neighbor other than v.*

Proof. Let x, y be neighbors of v such that $x, y \neq u$. Let G' be the graph obtained from G by deleting u, v and adding an edge between x and y. Suppose that the claim fails. Then G is triangle-free.

We first show that $q(G') = 0$. By Claim 4.3.23, $G - \{u, v\}$ contains no T_6 component. If G' contains a T_6 component K, then K must contain edge xy. Since $\delta(G) \geq 3$, x and y must be 3-vertices in K. By symmetry of T_6, we can assume w.l.o.g that $x \equiv v_4$ and $y \equiv v_2$. If u is embedded inside the cycle $v_1v_2v_4v_3$, then v_5 must be a 2-vertex in G. Otherwise, v_6 must be a 2-vertex in G. Both cases contradict that $\delta(G) \geq 3$. Thus, $q(G) = 0$.

Suppose that G' contains a Q_3^{1-} component H. Then, edge xy must belong to H. By symmetry of Q_3, We assume w.l.o.g that $x \equiv u_1$ and $y \equiv u_2$. Let M be the subgraph of G induced by $\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, v\}$. By Claim 4.3.23, $p(G \backslash M) = q(G \backslash M) = 0$. By the symmetry of H, we can assume that u is embedded inside the cycle $u_1vu_2u_3u_4$ (see Figure 4.5(b)). Since $\deg_{G'}(H) \leq 1$, at most one vertex in $\{u_1, u_2\}$ is a 4-vertex. Let z be a 3-vertex in $\{u_1, u_2\}$. Since G is triangle-free, u can have at most one neighbor in $\{u_3, u_4\}$. If u_3 is a 3-vertex, then we can collect $\{v, z, u_3, u_5, u_7, u_8\}$ from M. If u_4 is a 3-vertex, then we can collect $\{v, z, u_4, u_6, u_7, u_8\}$ from M. Thus, in any case, we can collect 6 vertices from M. By Observation 4.3.7 with $L = M$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (9, 14, 0, 0, 6)$, $6 - 9a + 14b$ must be negative. We obtain a contradiction by adding Inequality (4.31) to $\mathcal{LP}$.

$$6 - 9a + 14b \geq 0 \tag{4.31}$$

Thus, we can assume $p(G') = 0$. Hence, G' has an induced forest F' of order at least $an(G') - bm(G')$. Since xy is not an edge of G, $V(F') \cup \{v\}$ induces a forest in G. Since $n(G') = n(G) - 2$ and $m(G') = m(G) - 5$, G has an induced forest of order at least:

$$a(n(G) - 2) - b(m(G) - 5) + 1 = an(G) - bm(G) + 1 + 5b - 2a$$

Thus, we obtain a contradiction by adding Inequality (4.32) to $\mathcal{LP}$.

$$1 + 5b - 2a \geq 0 \tag{4.32}$$

$\square$

Claim 4.3.31. *Let C be a 4-cycle of G that has at least one 3-vertex and at most two 3-vertices. Then, (i) any two 3-vertices of C must be adjacent and two non-C edges adjacent to two 3-vertices must be embedded in the same side of C and (ii) two non-C edges of a* 4*-vertex, say v, must be embedded in the same side of C if C has a* 3*-vertex that is not adjacent to v.*

Proof. Let $\{w_1, w_2, w_3, w_4\}$ be clockwise ordered vertices of C. Without loss of generality, we assume w_1 is a 3-vertex of C and its non-C edge is embedded outside C. Suppose that the claim fails. We show that we can collect 2 vertices from C. If (i) fails, the other 3-vertex of C, denoted by x, is w_3 or a neighbor of w_1 such that its non-C edge is embedded inside C. Then, we can collect $\{x, w_1\}$ from C. If (ii) fails, let w_i and w_j be two non-adjacent vertices of C such that w_i is a 3-vertex and w_j is a 4 vertex that has two non-C edges that are embedded in different sides of C. Then, we can collect $\{w_i, w_j\}$ from C. By Claim 4.3.23, $p(G \setminus C) = q(G \setminus C) = 0$. Since C has at most two 3-vertices and $\delta(G) \geq 3$, $m(G \setminus C) \leq m(C) - 10$. By Observation 4.3.7 with $L = C$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (4, 10, 0, 0, 2)$, $2 - 4a + 10b$ must be negative, contradicting Inequality (4.32). $\square$

Claim 4.3.32. *G excludes any separating* 4*-cycle that has four 3-vertices.*

Proof. Let w_1, w_2, w_3, w_4 be 3-vertices of a separating 4-cycle C. Since C is separating and G is two-edge connected, two non-C edges of C must be embedded inside C and two other non-C edges must be embedded outside C. We assume w.l.o.g that the non-C edge of w_1 is embedded outside C. Let u be the non-C neighbor of w_1. Let w_i, $i \neq 1$, be a vertex of C that has its non-C edge embedded outside C and w_j be a vertex of C that has its non-C edge embedded inside C. Let H be the subgraph of G induced by $\{w_1, w_2, w_3, w_4, u\}$. We can collect $\{w_1, w_i, w_j\}$ from H. We now argue that $m(G \setminus H) \leq m(G) - 10$. Since G is triangle-free, u has at most two neighbors in C. If u has only one neighbor in G which is w_1, then $m(G \setminus H) \leq m(G) - 10$ since $\delta(G) \geq 3$. If u has exactly two neighbors in G, they must be w_1 and w_3. That means the non-C

edge of w_3 is embedded outside C. Since C is separating, two non-C edges incident to w_2 and w_4 must be embedded inside C. Thus, u must have two non-H incident edges since G is two-edge connected and $\delta(G) \geq 3$. That implies $m(G \setminus H) \leq m(G) - 10$.

By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 10, 0, 0, 3)$, $3 - 5a + 10b$ must be negative, contradicting Inequality (4.29) since b is non-negative. □

Claim 4.3.33. *Any separating 4-cycle of G must have at most two 3-vertices.*

Proof. Let C be a separating 4-cycle of G that has at least three 3-vertices. By Claim 4.3.32, C must have exactly three 3-vertices. Let w_1, w_2, w_3, w_4 be vertices in the clock-wise order of C such that w_1, w_2, w_3 are three 3-vertices. By Claim 4.3.17, w_4 is a 4-vertex. Let x, y, z be the non-C neighbors of w_1, w_2, w_3, respectively. Note that x and z may be the same vertex. We assume that x is embedded outside C. By Claim 4.3.30, two vertices w_2 and x must have a non-C common neighbor and two vertices w_2 and z must also have a non-C common neighbor. That implies xy and yz are edges of G. By planarity, y and z must also be embedded outside C. Since C is separating and G is two-edge connected, two edges of w_4 must be embedded inside C. If x and z are the same vertex (see Figure 4.6(a)), then we can collect $\{w_1, w_2, w_3\}$ from the subgraph H that is induced by $\{w_1, w_2, w_3, w_4, x\}$. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 10, 0, 0, 3)$, $3 - 5a + 10b$ must be negative, contradicting Inequality (4.29) since $b \geq 0$.

Thus, we can assume that x and z are two different vertices (see Figure 4.6(b)). If x, y, z are 3-vetices, then we can collect $\{w_2, w_3, x, y\}$ from the subgraph K of G induced by $\{w_1, w_2, w_3, x, y, z\}$. Since $m(G \setminus K) = m(G) - 11$, by Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (6, 11, 0, 0, 4)$, $4 - 6a + 11b$ must be negative, contradicting Inequality 4.30. Thus, at least one vertex in $\{x, y, z\}$ is a 4-vertex. Let M be the subgraph of G induced by $\{w_1, w_2, w_3, w_4, x, y, z\}$. Observe that we can collect $\{y, w_1, w_2, w_3\}$ from M. Since $m(G \setminus M) \leq m(G) - 14$, by Observation 4.3.7 with $L = M$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (7, 14, 0, 0, 4)$, $4 - 7a + 14b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.33) to $\mathcal{LP}$.

$$4 - 7a + 14b \geq 0 \tag{4.33}$$

□

Claim 4.3.34. *Any separating 4-cycle of G must have at most one 3-vertex.*

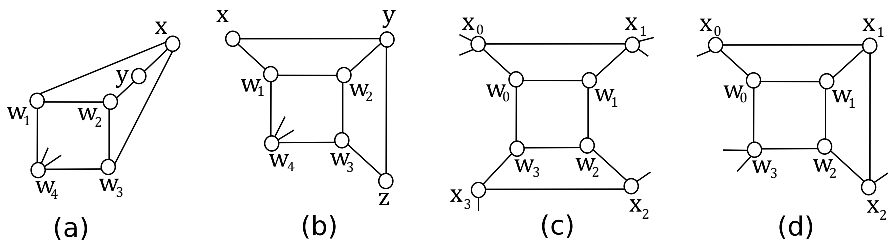

Figure 4.6: (a) A configuration in the proof of Claim 4.3.33 when $x = z$ (b) A configuration in the proof of Claim 4.3.33 when $x \neq z$ (c) A configuration in the proof of Lemma 4.3.38 (d) A configuration in the proof of Lemma 4.3.40

Proof. Let $w_1w_2w_3w_4$ be a separating 4-cycle, denoted by C, of G that has at least two 3-vertices. By Claim 4.3.33, C has exactly two 3-vertices. By (i) of Claim 4.3.31, we assume that w_1, w_2 are two 3-vertices of C and their non-C edges are embedded outside C. Since C is separating and G is two-edge connected, at least two non-C edges of C must be embedded inside C. By (ii) of Claim 4.3.31, two non-C edges of any 4-vertex of C must be embedded in the same side of C. We assume w.l.o.g that two non-C edges of w_3 are embedded inside C. Let u and v be non-C neighbors of w_1 and w_2, respectively. By Claim 4.3.30, v must be a common neighbor of u and w_2. If v is a 4-vertex, let H be the subgraph of G induced by $\{w_1, w_2, v, w_4\}$. Observe that we can collect $\{w_1, w_2\}$ from H. Since v and w_4 can share at most one incident edge, $m(G \setminus H) \leq m(G) - 10$. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (4, 10, 0, 0, 2)$, $2 - 4a + 10b$ must be negative, contradicting Inequality (4.26). Thus, we can assume that v is a 3-vertex. Let K be a subgraph of G induced by $\{u, v, w_1, w_2, w_4\}$. We can collect $\{v, w_2, w_1\}$ from K. Since v and w_4 can share at most one incident edge, $m(G \setminus K) \leq m(G) - 10$. By Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 10, 0, 0, 3)$, $3 - 5a + 10b$ must be negative, contradicting Inequality (4.29). □

4.3.4.4 Excluding a 4-face with exactly four 3-vertices

In this subsection, we denote $C = w_0w_1w_2w_3$ to be a 4-face of G such that each w_i is a 3-vertex, $0 \leq i \leq 3$. Let $X = \{x_0, x_1, x_2, x_3\}$ where each x_i is the non-C neighbor of w_i. All indices in this subsection are mod 4 and to simplify the presentation, we write w_j (x_j) instead of writing $w_{j \bmod 4}$ ($x_{j \bmod 4}$).

Claim 4.3.35. *Vertices in X are pairwise distinct and x_j is not adjacent to x_{j+2} for any $j \in \{0,1\}$.*

Proof. To prove that vertices in X are pairwise distinct, we only need to prove that $x_j \neq x_{j+2}$ since G is triangle-free. If $x_0 = x_2$, then $w_0w_1w_2x_0$ is a separating 4-cycle with at least three 3-vertices. If $x_1 = x_3$, then $w_0w_1x_1w_3$ is a separating 4-cycle with at least three 3-vertices. Both cases contradict Claim 4.3.34.

We now show that x_j and x_{j+2} are non-adjacent. By symmetry, it suffices to show the non-adjacency of x_0 and x_2. Suppose otherwise. By planarity, x_1 and x_3 cannot be adjacent and if they have a common neighbor, it must be x_0 and/or x_2. Since G is triangle-free, at most one of x_0 and x_2 can be a common neighbor of x_1 and x_3. We assume w.l.o.g that x_2 is a non-common neighbor of x_1 and x_3. We consider two cases:

Case 1 x_0 is a 3-vertex. Then removing x_2, w_2, w_0 disconnects x_3 and x_1. Let H be the subgraph induced by $\{w_0, w_1, w_2, w_3, x_2\}$. We can collect $\{w_1, w_2, w_3\}$ from H. By Claim 4.3.23, $p(G \setminus K) = q(G \setminus K) = 0$. Since $m(G \setminus K) \leq m(G) - 10$, by Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (5, 10, 0, 0, 3)$, $3 - 5a + 10b$ must be negative, contradicting Inequality (4.29).

Case 2 x_0 is a 4-vertex. Let G' be the graph obtained by removing $\{x_0, w_0, w_1, w_2, w_3\}$ from G and adding edge x_1x_3. G' is triangle-free since common neighbors of x_1 and x_3 are all removed. By Claim 4.3.23, $G \setminus C$ contains no Q_3^{1-} subgraph and T_6 component. Thus, $p(G') + q(G') \leq 1$. Let F' be the largest induced forest in G'. Observe that we can add $\{w_0, w_1, w_3\}$ to F' to get an induced forest in G. Since G' has strictly smaller order than G, F' has order at least $an(G') - bm(G') - cp(G') - dq(G')$. Since $n(G') = n(G) - 5$ and $m(G') \leq m(G) - 10$, by adding $\{w_0, w_1, w_3\}$ to F', we get an induced forest in G of order at least:

$$\begin{aligned} an(G') - bm(G') - cp(G') - dq(G') \geq & an(G) - bm(G) + 3 \\ & - 5a + 10b - cp(G') - dq(G') \end{aligned}$$

By Inequality (4.21) and Inequality (4.22), $3 - 5a + 10b - c$ and $3 - 5a + 10b - d$ are both non-negative. Since $p(G') + q(G') \leq 1$, $3 - 5a + 10b - cp(G') - dq(G')$ is non-negative. Thus, G has an induced forest of order at least $an(G) - bm(G)$, contradicting that G is a counter-example.

□

Claim 4.3.36. *At least one of two edges $w_j w_{j+1}$ and $w_{j+1} w_{j+2}$, for any j in $\{0, 1, 2, 3\}$, is not on the boundary of a 5^+-face.*

Proof. Suppose that there exists $j \in \{0, 1, 2, 3\}$ such that $w_j w_{j+1}$ and $w_{j+1} w_{j+2}$ are on the boundaries of 5^+-faces. We assume w.l.o.g that $j = 0$. Let G' be the graph obtained from G by removing $\{w_0, w_2, w_3\}$ and adding two edges $x_0 w_1, w_1 x_2$. By Claim 4.3.35, x_0 and x_2 are not adjacent. Thus, G' is triangle-free. Since $\delta(G) \geq 3$, x_3 is the only possible 2-vertex of G'. Thus, G' contains no T_6 component. We consider two cases:

Case 1 G' contains no Q_3^{1-} subgraph. Then G' has an induced forest F' of order at least $an(G') - bm(G')$. Since $n(G') = n(G) - 3$ and $m(G') = m(G) - 5$, by adding $\{w_0, w_2\}$ to F', wet get an induced forest of G of order at least:

$$a(n(G) - 3) - b(m(G) - 5) + 2 = an(G) - bm(G) + 2 - 3a + 5b$$

Since $2 - 3a + 5b \geq 0$ by Inequality (4.28), G has an induced forest of order at least $an(G) - bm(G)$, contradicting that G is a counter-example.

Case 2 G' contains at least one Q_3^{1-} subgraph. By Claim 4.3.23, $G - \{w_0, w_2, w_3\}$ contains no Q_3^{1-} subgraph. Thus, any Q_3^{1-} subgraph of G must contain w_1. By Observation 4.3.6, G' has exactly one Q_3^{1-} subgraph. If G' contains a Q_3 component, then the subgraph of G induced by $V(Q_3) \cup \{w_0, w_2, w_3\}$ has degree 1 in G, contradicting that G is two-edge connected. Thus, we can assume that G' contains a Q_3^1 subgraph K. Let $G'' = G' \setminus K$. We observe that G'' can also be obtained from G by removing $V(K) \cup \{w_0, w_2, w_3\}$. Since $V(K) \cup \{w_0, w_2, w_3\}$ induces a connected subgraph of G, $p(G'') = q(G'') = 0$ by Claim 4.3.23. Thus, G'' has a forest F'' of order at least $an(G'') - bm(G'')$. By Lemma 4.3.4, we can collect 5 vertices from K to obtain an induced forest F' of G' of order at least $an(G'') - bm(G'') + 5$. By adding $\{w_0, w_2\}$ to F', we get an induced forest F of G of order at least $an(G'') - bm(G'') + 7$. Since $n(G'') = n(G) - 11$ and $m(G'') = m(G) - 18$, F has order at least:

$$an(G) - bm(G) + 7 - 11a + 18b$$

We obtain a contradiction by adding Inequality (4.34) to $\mathcal{LP}$.

$$7 - 11a + 18b \geq 0 \tag{4.34}$$

□

Claim 4.3.37. *At least one of two vertices x_j, x_{j+2} is a 3-vertex, for any j in $\{0,1\}$.*

Proof. Suppose that x_j and x_{j+2} are two 4-vertices for some $j \in \{0,1\}$. Let H be the graph induced by $V(C) \cup \{x_j, x_{j+2}\}$. Observe that we can collect $\{w_j, w_{j+1}, w_{j+2}\}$ from H. By Claim 4.3.35, x_j and x_{j+1} are non-adjacent. Thus, $m(G \setminus H) \leq m(G) - 14$. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (6, 14, 0, 0, 3)$, $3 - 6a + 14b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.35) to $\mathcal{LP}$.

$$3 - 6a + 14b \geq 0 \tag{4.35}$$

□

Lemma 4.3.38. *G has no 4-face with four 3-vertices.*

Proof. Let H be the subgraph of G induced by $V(C) \cup X$. If no edge of C is on the boundary of a 5^+-face, then H is a Q_3^{4-} subgraph of G, contradicting Claim 4.3.19. Thus, we can assume at least one edge of C is on the boundary of a 5^+-face. By Claim 4.3.36, C has at most two edges on the boundaries of 5^+-faces and they cannot be incident to the same vertex of C. Thus, there exists $j \in \{0,1\}$ such that two edges $w_j w_{j+1}$ and $w_{j+2} w_{j+3}$ are on the boundary of 4-faces. Without loss of generality, we assume that $j = 0$. Thus, $x_0 x_1$ and $x_2 x_3$ are edges of G (see Figure 4.6(c)). By symmetry, we can assume that x_0 is the highest degree vertex of X. By Claim 4.3.37, x_2 is a 3-vertex.

We claim that (i) x_1 is a 4-vertex and (ii) $x_1 x_2$ and $x_3 x_0$ are non-edges of G. Suppose that at least one of two claims fails, we show that we can collect 5 vertices from H. If $x_1 x_2$ is an edge of G, then we can collect $\{x_1, w_1, w_0, w_3, x_2\}$ from H. If $x_0 x_3$ is an edge of G, then we can collect $\{x_0, w_0, w_1, w_3, x_2\}$ from H. If x_1 is a 3-vertex, then we can collect $\{w_0, w_2, w_3, x_2, x_1\}$ from H. Since $m(G \setminus H) \leq m(G) - 13$, by Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 13, 0, 0, 5)$, $5 - 8a + 13b$ must be negative, contradicting Inequality (4.11). Thus, both claims hold.

Let K be the subgraph induced by $V(C) \cup \{x_0, x_2, x_3\}$. By Claim 4.3.37, x_3 is a 3-vertex. Thus, we can collect $\{x_3, w_0, w_2, w_3\}$ from K. Since x_0 is the highest degree vertex of C, x_0 is a 4-vertex. Thus, $m(G \setminus K) \leq m(G) - 14$. By Observation 4.3.7 with $L = K$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (7, 14, 0, 0, 4)$, $4 - 7a + 14b$ must be negative, contradicting Inequality (4.33). □

By combining Lemma 4.3.38 and Claim 4.3.34, we get:

Corollary 4.3.39. *G has no 4-cycle with four 3-vertices.*

4.3.4.5 Excluding a 4-face with at least two 3-vertices

Lemma 4.3.40. *G has no 4-face with three 3-vertices.*

Proof. Let $C = w_0w_1w_2w_3$ be a 4-face of G that has three 3-vertices, say w_0, w_1, w_2. Suppose that w_i and w_{i+2} share a neighbor, say x, for some i in $\{0, 1\}$. Then, the cycle $xw_iw_{i+1}w_{i+2}$ is a separating 4-cycle that has at least two 3-vertices, contradicting Claim 4.3.34. Thus, w_i and w_{i+2} have no common neighbor for any i in $\{0, 1\}$. Let x_0, x_1, x_2 be the neighbors of w_0, w_1, w_2, respectively. By Claim 4.3.30, x_0x_1 and x_1x_2 are edges of G (see Figure 4.6(d)). Let H be the subgraph of G induced by $V(C)\cup\{x_0, x_1, x_2\}$. By Corollary 4.3.39, at least one vertex in $\{x_0, x_1, x_2\}$ is a 4-vertex. Since G is triangle-free, x_0 and x_2 are non-adjacent. Since w_1 and w_3 do not have a common neighbor, w_3 and x_1 are non-adjacent. Thus, $m(G \setminus H) \leq m(G) - 14$. Observe that we can collect $\{x_1, w_0, w_1, w_2\}$ from H. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (7, 14, 0, 0, 4)$, $4 - 7a + 14b$ must be negative, contradicting Inequality (4.33). $\square$

By Lemma 4.3.40 and Claim 4.3.34, we have:

Corollary 4.3.41. *G has no 4-cycle with at least three 3-vertices.*

Lemma 4.3.42. *G has no 4-face with exactly two 3-vertices.*

Proof. Let w_0, w_1, w_2, w_3 be vertices in clock-wise order of a 4-face C of G that has exactly two 3-vertices. By Claim 4.3.31 two 3-vertices of C must be adjacent. Without loss of generality, we assume that two 3-vertices are w_0 and w_1. Let x_0, x_1 be the neighbors of w_0, w_1, respectively. By Claim 4.3.30, x_0x_1 is an edge of G. By Corollary 4.3.41, x_0 and x_1 are 4-vertices. We now show that x_j and w_{j+2} are non-adjacent for any j in $\{0, 1\}$. If x_0 and w_2 are adjacent, then the cycle $x_0w_0w_1w_2$ is a separating 4-cycle that has at least two 3-vertices. If x_1 and w_3 are adjacent, then the cycle $x_1w_1w_0w_3$ is a separating 4-cycle that has at least two 3-vertices. Both cases contradict Claim 4.3.34. Thus, x_j and w_{j+2} are non-adjacent.

If x_0 and w_2 share a common neighbor and x_1 and w_3 also share a common neighbor, by planarity, they all share a common neighbor, contradicting that G is triangle-free. Thus, by symmetry (see Figure 4.7(a)), we can assume that x_0 and w_2 share no common neighbor. Let G' be the graph obtained by removing $\{x_1, w_0, w_1, w_3\}$ from G and adding edge x_0w_2. Then, G' is a triangle-free planar graph. By Claim 4.3.23, the graph obtained by removing $\{x_1, w_0, w_1, w_3\}$ from G has no T_6 component and Q_3^{1-} subgraph. Thus, any T_6 component or Q_3^{1-} subgraph of G' must contain edge x_0w_2. That implies $p(G') + q(G') = 1$. We consider three cases:

Case 1 $p(G') = q(G') = 0$. Then G' has an induced forest F' of order at least $an(G') - bm(G')$. By adding $\{w_0, w_1\}$ to F', we obtain an induced forest of order at least $an(G') - bm(G') + 2$. Since $n(G') = n(G) - 4$ and $m(G') = m(G) - 10$, we have:

$$an(G') - bm(G') + 2 = an(G) - bm(G) + 2 - 4a + 10b$$

Since $2 - 4a + 10b$ is non-negative by Inequality (4.26), G has an induced forest of order at least $an(G) - bm(G)$, contradicting that G is a counter-example.

Case 2 $p(G') = 1$ and $q(G') = 0$. Let G'' be the graph obtained from G' by removing the T_6 component of G'. Since G'' can also be obtained from G by removing $V(T_6) \cup \{x_1, w_0, w_1, w_3\}$ which induces a connected subgraph of G, by Claim 4.3.23, $p(G'') = q(G'') = 0$. Thus, G'' has a forest F'' of order at least $an(G'') - bm(G'')$. By Lemma 4.3.5, we can add 4 vertices from the T_6 component to F'' to get an induced forest $\hat{F}$ of G' of order at least $an(G'') - bm(G'') + 4$. By adding w_0 and w_1 to $\hat{F}$, we get an induced forest of order at least $an(G'') - bm(G'') + 6$ in G. Since $n(G'') = n(G) - 10$ and $m(G'') = m(G) - 18$, we have:

$$an(G'') - bm(G'') + 6 = an(G) - bm(G) + 6 - 10a + 18b$$

Since $6 - 10a + 18b$ is non-negative by Inequality (4.29), G has an induced forest of order at least $an(G) - bm(G)$, contradicting that G is a counter-example.

Case 3 $p(G') = 0$ and $q(G') = 1$. Let M be the Q_3^{1-} subgraph of G'. We consider two subcases:

Subcase 1 M is Q_3^1 in G'. Let $G''' = G' \setminus M$. Then, G''' can also be obtained from G by removing $V(M) \cup \{x_1, w_0, w_1, w_3\}$ which induces a connected subgraph of G. Thus, by Claim 4.3.23, $p(G''') = q(G''') = 0$. Let F''' be a forest of G''' of order at least $an(G''') - bm(G''')$. By Lemma 4.3.4, we can add 5 vertices of M to F''' to get an induced forest $\overline{F}$ in G' of order at least $an(G''') - bm(G''') + 5$. By adding w_0 and w_1 to $\overline{F}$, we get an induced forest of order at least $an(G''') - bm(G''') + 7$ in G. Since $n(G''') = n(G) - 12$ and $m(G'') = m(G) - 23$, we have:

$$an(G''') - bm(G''') + 6 = an(G) - bm(G) + 7 - 12a + 23b$$

Thus, we obtain a contradiction by adding Inequality (4.36) to $\mathcal{LP}$.

$$7 - 12a + 23b \geq 0 \tag{4.36}$$

Subcase 2 M is Q_3 in G'. Recall that x_0w_2 must be an edge of M. By symmetry of Q_3, we can assume w.l.o.g that $x_0 = u_1$ and $w_2 = u_2$ (see Figure 4.7(b)). We will argue that 4-cycle $u_5u_6u_7u_8$ has at least 3-vertices in G. Consider the cycle $\hat{C} = x_0u_5u_6w_2u_3u_4$ of G. Since edge x_0w_2 is embedded inside $\hat{C}$, x_1, w_0, w_1, w_3 is embedded inside $\hat{C}$. That implies u_7 and u_8 are 3-vertices in G. Observe that the path $x_0w_0w_1w_2$ separate the internal part of $\hat{C}$ into two parts, one contains x_1 and another contains w_3. Let $C' = x_0w_0w_1w_2u_6u_5$ and $y \in \{x_1, w_3\}$ be the vertex inside C'. If u_5 is adjacent to y, then u_5x_0y is a triangle in G, contradicting that G is triangle-free. Thus, u_5 is also a 3-vertex in G. Hence, we can conclude that 4-cycle $u_5u_6u_7u_8$ has at least three 3-vertices in G, contradicting Corollary 4.3.41.

□

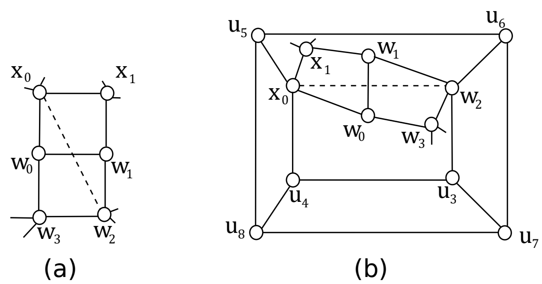

Figure 4.7: (a) A configuration in the proof of Lemma 4.3.42 (b) A configuration in the proof of Subcase 2 in Lemma 4.3.42.

4.3.4.6 Excluding a 4-face with at least one 3-vertex

In this subsection, we denote $C = w_0w_1w_2w_3$ to be a 4-face of G such that w_0 is a 3-vertex and w_1, w_2, w_3 are 4-vertices. From Lemma 4.3.38, 4.3.40 and 4.3.42 and Claim 4.3.34, we have:

Corollary 4.3.43. *Any* 4*-cycle of* G *has at most one 3-vertex.*

Claim 4.3.44. *G has no 3-vertex that has a 3-vertex and a 4-vertex as neighbors.*

Proof. Suppose that G has a 3-vertex u that has a 3-vertex v and a 4-vertex w as neighbors. Let x be a neighbor of u such that $x \notin \{v, w\}$. By Claim 4.3.30, x and v has a neighbor $y \neq u$. Thus, 4-cycle $uxyv$ has two 3-vertices, contradicting Corollary 4.3.43. □

Let x_0 be the non-C neighbor of w_0. By Claim 4.3.30, x_0 and w_1 have a common neighbor, say x_1, and x_0 and w_3 have a common neighbor, say x_3. By Corollary 4.3.43, x_0, x_1, x_3 are 4-vertices. Note that x_1 and x_3 could be the same vertex, or even $x_1 = x_3 = w_2$.

Claim 4.3.45. *x_0 and w_2 are non-adjacent.*

Proof. Suppose otherwise. Let x_2 be the non-C neighbor of w_2 such that $x_2 \neq x_0$. Let $C_1 = x_0w_0w_3w_2$ and $C_2 = x_0w_0w_1w_2$ be two 4-cycles of G. By applying (ii) of Claim 4.3.31 to C_1, two edges w_2x_2 and w_2w_1 must be embedded in the same side of C_1. That implies two edges w_2x_2 and w_2w_3 are embedded in different sides of C_2, contradicting (ii) of Claim 4.3.31 for C_2. □

Claim 4.3.46. *w_1 and w_3 have no non-C common neighbors.*

Proof. We note that w_1 and w_3 can have up to 4 common neighbors. Let x be a non-C common neighbor of w_1 and w_3. Since w_0 is a 3-vertex in 4-cycle $w_1w_0w_3x$, by Corollary 4.3.43, x must be a 4-vertex. We consider two cases:

Case 1 Three vertices x_0, w_1, w_3 share a common neighbor, that we, w.l.o.g, assume to be x (see Figure 4.8(a)). Let $C_3 = x_0w_0w_1x$, $C_4 = x_0w_0w_3x$ and $C_5 = xw_3w_0w_1$. By applying (ii) of Claim 4.3.31 to C_3 and to C_4, two non-C_3 edges incident to x must be embedded in the same side of C_3 and two non-C_4 edges incident to x must be embedded in the same side of C_4. That implies two non-C_5 edges incident to x are embedded in different side of C_5, contradicting (ii) of Claim 4.3.31 for C_5.

Case 2 Three vertices x_0, w_1, w_3 do not share a common neighbor. Then, x, x_1 and x_3 are pairwise distinct(see Figure 4.8(b)). Let H be the subgraph of G induced by $V(C) \cup \{x, x_0, x_1, x_3\}$. Observe that we can collect $\{x_0, w_0, w_1, w_3\}$ from H. Since G is triangle-free, xw_2, xx_3, xx_1, x_1x_3 are non-edges of G. Thus, $n(H) = n(G) - 8$ and $m(H) = m(G) - 20$. By Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 20, 0, 0, 4)$, $4 - 8a + 20b$ must be negative, contradicting Inequality (4.26).

□

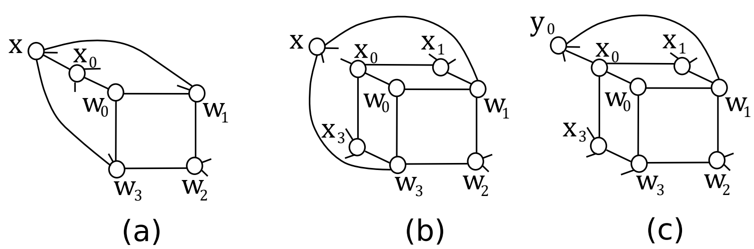

Figure 4.8: (a) A configuration in the proof of Case 1 of Claim 4.3.46 (b) A configuration in the proof of Case 2 Claim 4.3.46 (c) A configuration in the proof of Claim 4.3.48

Let y_1 and y_3 be the non-C neighbors of w_1 and w_3, respectively, such that $y_1 \neq x_1$ and $y_3 \neq x_3$. Let y_0 be the remaining neighbor of x_0 that is not in $\{x_1, w_0, x_3\}$. Let $Z = V(C) \cup \{x_0, x_1, x_3, y_0, y_1, y_3\}$.

Claim 4.3.47. *Vertices in Z are pairwise distinct.*

Proof. By Claim 4.3.46, two vertices x_1 and x_3 are distinct and two vertices y_1 and y_3 are distinct. By Claim 4.3.45, $y_0 \neq w_2$. It remains to prove that $y_0 \neq y_1$ and $y_0 \neq y_3$. By symmetry, it suffices to prove $y_0 \neq y_1$. Suppose otherwise. Let H be the subgraph of G induced by $V(C) \cup \{x_0, x_1, x_3, y_0\}$ (see Figure 4.8(c)). Since G is triangle-free, y_0 and x_3 are non-adjacent and y_0 and w_2 are non-adjacent. By Claim 4.3.46, y_0 and w_3 are non-adjacent. Thus, $m(G \setminus H) = m(G) - 20$. Since we can collect $\{x_0, w_0, w_1, w_3\}$ from H, by Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 20, 0, 0, 4)$, $4 - 8a + 20b$ must be negative, contradicting Inequality (4.26). □

Claim 4.3.48. *At least one of y_0y_1, y_1y_3, y_0y_3 is an edge of G.*

Proof. Suppose that y_0y_1, y_1y_3, y_0y_3 are non-edges of G. Let $N = \{x_0, x_1, x_3, w_1, w_2, w_3\}$. Let G' be the graph obtained from G by removing vertices in N and adding edges $\{w_0y_0, w_0y_1, w_0y_3\}$ (see Figure 4.9(a)). By Claim 4.3.23, the graph obtained from G by removing vertices in N has no T_6 component and Q_3^{1-} subgraph. Thus, any T_6 component and Q_3^{1-} subgraph of G' must contain w_0. That implies $p(G')+q(G') \leq 1$. If $q(G') = 1$, let H be a T_6 component of G. Then w_0 must be a 3-vertex of H. Since any 3-vertex of a T_6 component is adjacent to a 2-vertex, at least one neighbor of w_0 must be a 2-vertex in G'. However, w_0's neighbors all are 3^+-vertices in G'. Thus, $q(G') = 0$. Since G is a counter-example of minimal order, G' has an induced forest F' of order at least $an(G') - bm(G') - cp(G')$. By adding x_0, w_1, w_3 to F', we obtain an induced forest F of order at least $an(G') - bm(G') - cp(G') + 3$ in G. Since $n(G') = n(G) - 6$ and $m(G') = m(G) - 15$, we have:

$$an(G') - bm(G') - cp(G') + 3 \geq an(G) - bm(G) + 3 - 6a + 15b - c$$

Thus, we obtain a contradiction by adding Inequality (4.37) to $\mathcal{LP}$.

$$3 - 6a + 15b - c \geq 0 \tag{4.37}$$

□

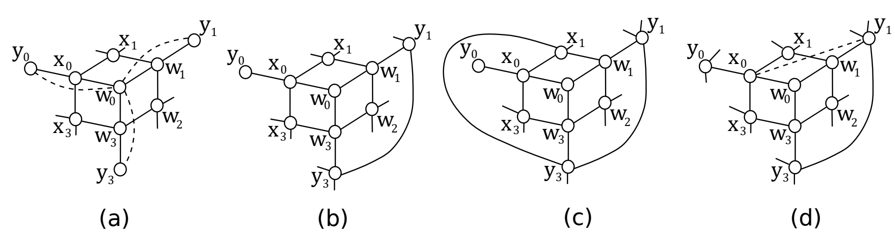

Figure 4.9: (a) A configuration in the proof of Claim 4.3.48 (b) A configuration in the proof of Claim 4.3.49 (c) A configuration in the proof of Claim 4.3.50 (d) A configuration in the proof of Lemma 4.3.51

Herein, we regard 4-face C as a 4-cycle so that we can speak of the symmetry of neighbors of w_0 in G (see Figure 4.9(a)). By symmetry, we can assume w.l.o.g that y_1y_3 is an edge of G.

Claim 4.3.49. *y_1 and y_3 are 4-vertices.*

Proof. Suppose otherwise. We can assume w.l.o.g that y_1 is a 3-vertex. By Claim 4.3.44, y_3 must be a 4-vertex (see Figure 4.9(b)). Let H be the subgraph of G induced by $\{w_0, w_1, w_3, x_0, x_1, x_3, y_1, y_3\}$. By planarity, if y_1x_3 is an edge of G, then x_1y_3 is non-edge of G and vice versa. Thus, $m(G \setminus H) \leq m(G) - 19$. Since we can collect $\{y_1, x_0, w_0, w_3\}$ from H, by Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 19, 0, 0, 4)$, $4 - 8a + 19b$ must be negative. Thus, we obtain a contradiction by adding Inequality (4.38) to $\mathcal{LP}$.

$$4 - 8a + 19b \geq 0 \tag{4.38}$$

□

Claim 4.3.50. *Two vertices y_3, x_1 are non-adjacent and two vertices x_3, y_1 are non-adjacent.*

Proof. By symmetry, we only need to prove y_3 and x_1 are non-adjacent. Suppose that y_3x_1 is an edge of G(see Figure 4.9(c)). Let C_6 be cycle $x_1w_1w_2w_3y_3$. Let H be the subgraph induced by $\{w_0, w_1, w_2, w_3, x_0, x_1, x_3, y_3\}$. Since y_1 and y_0 are embedded in different sides of C_6, we can collect x_0, w_0, w_1, w_3 from H. Since $m(G \setminus H) \leq m(G) - 20$, by Observation 4.3.7 with $L = H$ and $(\alpha, \beta, \gamma, \eta, \lambda) = (8, 20, 0, 0, 4)$, $4 - 8a + 20b$ must be negative, contradicting Inequality (4.26). □

Lemma 4.3.51. *G has no 4-face with exactly one 3-vertex.*

Proof. Since G is triangle-free, at most one of y_0y_1, y_0y_3 is an edge of G. By symmetry, we can assume w.l.o.g that y_0 and y_1 are non-adjacent. Let $J = \{w_0, w_1, w_2, w_3, x_1, x_3, y_3\}$. Let G' be the graph obtained from G by removing vertices in J and adding edge x_0y_1 (see Figure 4.9(d)). Since two vertices y_0, y_1 are non-adjacent, G' is triangle-free. By Claim 4.3.23, the graph obtained by removing J from G has no T_6 component and Q_3^{1-} subgraph. Thus, any T_6 and Q_3^{1-} subgraph of G' must contain both x_0 and y_1. Since x_0 is a 2-vertex of G' and every vertex in a Q_3^{1-} subgraph has degree at least 3, G' has no Q_3^{1-} subgraph. We now argue that G' contains no T_6 component.

Suppose that G' contains a T_6 component. Then, x_0 must be one of two 2-vertices of T_6. By symmetry of T_6, we can assume w.l.o.g that x_0 is v_5. Thus, edge x_0y_1 is v_2v_5 or v_5v_3. Let C_7 be cycle $v_1v_2v_4v_3$. C_7 separates v_6 and v_5 in both G' and G. Thus, C_7

also separates v_6 from every vertex reachable from v_5 in $G \setminus C_7$. That implies v_6 is also a 2-vertex in G, contradicting Lemma 4.3.29. Thus, G contains no T_6 component.

Since G is a minimal counter-example, G' has an induced forest F' of order at least $an(G') - bm(G')$. Let $F = F' \cup \{w_0, w_1, w_3\}$. F is an induced forest of G of order at least $an(G') - bm(G') + 3$. Since x_1 and y_3 are not adjacent by Claim 4.3.50, $m(G') = m(G) - 19$. Since $n(G') = n(G) - 7$, we have:

$$|F| \geq an(G) - bm(G) + 3 - 7a + 19b$$

Thus, we obtain a contradiction by adding Inequality (4.39) to $\mathcal{LP}$.

$$3 - 7a + 19b \geq 0 \tag{4.39}$$

□

4.3.4.7 Excluding a 5-face with at least four 3-vertices

Let w_0, w_1, w_2, w_3, w_4 be vertices in clock-wise order of a 5-face C of G such that C has at most one 4-vertex. Let $X = \{x_0, x_1, x_2, x_3, x_4\}$ be a set of vertices such that x_i is a non-C neighbor of w_i for all $0 \leq i \leq 4$.

Claim 4.3.52. *C has no 4-vertex.*

Proof. Suppose that w_i is a 4-vertex in G. Recall that C has at least four 3-vertices. Thus, w_{i+1} is a 3-vertex that has a 3-vertex and a 4-vertex as neighbors, contradicting Claim 4.3.44. □

Observation 4.3.53. *Vertices in X are and pairwise distinct and have degree* 3.

Proof. Suppose that $x_i = x_{i+2}$ for some $i \in \{0, 1, 2, 3, 4\}$ (indices are mod 5). Then, $w_i w_{i+1} w_{i+2} x_i$ is a 4-cycle that has at least three 3-vertices, contradicting Corollary 4.3.43. If x_i is a 4-vertex, then w_i is a 3-vertex that has a 3-vertex and a 4-vertex as neighbors, contradicting Claim 4.3.44. □

Lemma 4.3.54. *Any 5-face of G has at least two 4-vertices.*

Proof. By Claim 4.3.52, a 5-face C that has at most one 4-vertex actually has no 4-vertex. By Corollary 4.3.43, x_i and x_{i+1} are non-adjacent, for any i such that $0 \leq i \leq 4$. Let G' be the graph obtained from G by removing $\{w_0, w_3, w_4\}$ and adding two edges x_0w_1, x_3w_2. We first show that $p(G') = q(G') = 0$.

G' has no T_6 component since x_4 is the only 2-vertex in G'. Suppose that G' contains a Q_3^{1-} subgraph, say H, of G. Since the graph obtained from G by removing $\{w_0, w_3, w_4\}$ has no Q_3^{1-} subgraph, H must contain at least one of two new edges x_0w_1, x_3w_2. Since H has six 4-faces, there is at least one 4-face, say C_0, of H that contains no new edge. Thus, C_0 is also a 4-cycle in G. Except x_4, all vertices in G' has the same degree as in G. Thus, C_0 has at least two 3-vertices, contradicting Corollary 4.3.43.

Hence, G' has an induced forest F' of order at least $an(G') - bm(G')$. Let $F = F' \cup \{w_0, w_3\}$. F is an induced forest of G of order at least $an(G') - bm(G') + 2$. Since $n(G') = n(G) - 3$ and $m(G') = m(G) - 5$, we have:

$$|F| \geq an(G) - bm(G) + 2 - 3a + 5b$$

Since $2-3a+5b$ is non-negative by Inequality (4.28), $|F| \geq an(G)-bm(G)$, contradicting that G is a counter-example. $\square$

Proof of Theorem 4.3.2 We have shown that if a, b, c, d satisfy all constraints in $\mathcal{LP}$, a counter-example graph G must be two-connected, have $\delta(G) \geq 3$, have no 4-face with at least one 3-vertex and have no 5-face with at least four 3-vertices, contradicting Theorem 4.3.8. To finish the proof of Theorem 4.3.2, we only need to show that Linear Program $\mathcal{LP}$ that consists of constraints from (4.9) to (4.39) is equivalent to Linear Program (4.7). We observe that the set of constraints in Linear Program (4.7) is a subset of constraints in $\mathcal{LP}$ since:

$$\begin{gathered}(4.7\mathrm{f}) = (4.24), (4.7\mathrm{g}) = (4.25), (4.7\mathrm{h}) = (4.18), (4.7\mathrm{i}) = (4.9), (4.7\mathrm{j}) = (4.10)\\ (4.7\mathrm{k}) = (4.11), (4.7\mathrm{l}) = (4.12)(4.7\mathrm{m}) = (4.13), (4.7\mathrm{n}) = (4.14), (4.7\mathrm{o}) = (4.15)\\ (4.7\mathrm{p}) = (4.16), (4.7\mathrm{q}) = (4.21), (4.7\mathrm{r}) = (4.22), (4.7\mathrm{s}) = (4.27)\end{gathered}$$

Here we note that Inequality (4.7m) is equivalent to Inequality (4.13) when $t = 0$. We express remaining constraints of $\mathcal{LP}$ as linear combinations of constraints in Linear Program (4.7) as follows:

$$
\begin{aligned}
(5t+4)-(8t+6)a+(13t+9)b-d\ (4.13) &= (4.7\mathrm{m})+t(4.7\mathrm{k}) \\
4-6a+10b-d\ (4.17) &= (4.7\mathrm{m})+(4.7\mathrm{c}) \\
5-6a+16b-c-d\ (4.19) &= (4.7\mathrm{p})+(4.7\mathrm{c}) \\
4-6a+11b-d\ (4.20) &= (4.7\mathrm{m})+2(4.7\mathrm{c}) \\
5-8a+17b-c-d\ (4.23) &= (4.7\mathrm{p})+2(4.7\mathrm{c}) \\
1-2a+5b\ (4.26) &= (4.7\mathrm{h})+(4.7\mathrm{b}) \\
2-3a+5b\ (4.28) &= 2(4.7\mathrm{b})+(4.7\mathrm{h}) \\
3-5a+9b\ (4.29) &= (4.7\mathrm{s})+(4.7\mathrm{h}) \\
4-6a+9b\ (4.30) &= (4.7\mathrm{m})+(4.7\mathrm{e}) \\
6-9a+14b\ (4.31) &= (4.7\mathrm{k})+(4.7\mathrm{c})+(4.7\mathrm{b}) \\
1+5b-2a\ (4.32) &= (4.7\mathrm{h})+(4.7\mathrm{b}) \\
4-7a+14b\ (4.33) &= (4.7\mathrm{b})+2(4.7\mathrm{h})+(4.7\mathrm{s}) \\
7-11a+18b\ (4.34) &= (4.7\mathrm{b})+2(4.7\mathrm{h})+2(4.7\mathrm{s}) \\
3-6a+14b\ (4.35) &= (4.7\mathrm{s})+2(4.7\mathrm{h}) \\
7-12a+23b\ (4.36) &= 2(4.7\mathrm{s})+3(4.7\mathrm{h})+(4.7\mathrm{b}) \\
3-6a+15b-c\ (4.37) &= (4.7\mathrm{h})+(4.7\mathrm{q}) \\
4-8a+19b\ (4.38) &= 3(4.7\mathrm{h})+(4.7\mathrm{s})+(4.7\mathrm{b}) \\
3-7a+19b\ (4.39) &= (4.7\mathrm{s})+3(4.7\mathrm{h})
\end{aligned}
$$

Chapter 5: Conclusion

Our work in Chapter 2 gives the first EPTAS for TSP and the first PTAS for subset TSP in H-minor-free graphs. This is the first step toward designing PTASes for other connectivity problems. Steiner tree [21], Steiner forest [13], surivivable network design [20] are connectivity problems that have EPTASes in planar and bounded genus graphs. Their planar PTASes also share two ingredients with TSP: *the contraction decomposition framework* and *light spanners*. The main difficulty is that, like TSP and subset TSP, the spanners for other connectivity problems in surface-embedded graphs heavily use non-crossing embedding. We have shown that embedding is not necessary to obtain light spanners for TSP and subset TSP and that could be true for other connectivity problems as well. The most natural next problem that we believe to have such spanners is the Steiner tree problem.

In Section 2.3 of Chapter 2, we presented an algorithm that constructs a subset spanner of weight at most $O_H(\log k \operatorname{poly}(\frac{1}{\epsilon}))$ times the weight of a minimum Steiner tree ST. We conjecture that it is possible to remove the $\log k$ factor. That is, there exists a subset spanner of weight at most $O_H(\operatorname{poly}(\frac{1}{\epsilon}))w(\mathrm{ST})$. Such a spanner would imply an *efficient* PTAS for subset TSP in minor-free graphs. By Theorem 2.3.8, it suffices to construct an ℓ-close spanner of weight at most $O_H(\operatorname{poly}(\frac{1}{\epsilon})\, k\ell)$. A possible direction to construct such a light ℓ-close spanner is to extend Lemma 2.3.5 to H-minor-free graphs. In Section 2.3.2, we point out that a terminal preserving minor that preserves pairwise distances *exactly* is not possible due to the lower bound by Krauthgamer, Nguyễn, and Zondiner [65]. However, it is open to obtain an approximate version of Lemma 2.3.5 for H-minor-free graphs. The lower bound for approximate terminal preserving minors in planar graphs by Krauthgamer, Nguyễn, and Zondiner [65] (Theorem 3.3 in [65]) can be translated to a lower bound of $\Omega(\frac{n}{\epsilon^2})$ on the number of Steiner vertices. It does not rule out the existence of an approximate terminal preserving minor of size $O(\frac{n}{\epsilon^c})$ for some constant $c \geq 3$.

In Chapter 3, we have seen that the PTAS from local search is not efficient and the bottleneck lies in the local exchange that has running time $n^{c(\epsilon)}$ where $c(\epsilon)$ is a function depending on ϵ only. A natural research question is: can we do the local exchange in $2^{c(\epsilon)}n^{O(1)}$ time? This would make the local search PTAS *efficient*. Fellows, Fomin, Lokshtanov, Rosamond, Saurabh and Villanger [46] showed that it is possible to make local exchange efficient for several problems in apex-minor-free graphs, including: vertex cover, dominating set and odd cycle transversal. However, it is still open whether the same result can be archived for the FVS problem. It also would be interesting to have a meta theorem that characterizes the set of problems admitting efficient local search PTAS in H-minor-free graphs.

Bidimensionality is a powerful framework for obtaining a PTAS for many problems in H-minor-free graphs. One prominent property of a bidimensional problem is that the size of the solution in a $t \times t$ planar grid must be $\Omega(t^2)$. Subset feedback vertex set and the r-dominating set problem when r is a part of the input do not have such properties. Thus, no PTASes based on the bidimentionality framework are known for the two problems. We have seen that local search can give a PTAS for the r-dominating set problem. Chaplick, De, Ravsky, Spoerhase [28] also presented local search PTAS for several other non-bidimensional problems. In a certain sense, local search does offer advantages over the bidimensional framework. However, a wide range of bidimensional problems, especially contraction-bidimensional problems, are not known to have local search PTASes. We note that FVS problem is a contraction-bidimensional problem. Thus, our result puts the first contraction-bidimensional problem in the class of problems having local search PTAS. A natural question is: are there problems that have bidimensional PTAS but do not have a local search PTAS?

In Section 4.3 of Chapter 4, we have introduced a new approach that can handle special graphs of small order separately to find an induced forest of order at least $\frac{5n}{9}$ in triangle-free planar graphs of order n. It would be very interesting to see whether our method can be employed to give a better bound on the order of the largest induced forest in girth-5 planar graphs [43] and the order of the induced forest in subcubic (non-planar) graphs of girth at least four and five [59]. Another direction is to improve our analysis to obtained a bound of $\frac{4n}{7}$. This would match the bound obtained by Wang, Xie and Yu [90] for bipartite planar graphs and would possibly give a simpler and more general proof than the proof by Wang, Xie and Yu. The ultimate goal is to resolve the conjecture of Akiyama and Watanabe and we would like to see if our method can be extended to resolve this conjecture as well.

Bibliography

[1] I. Abraham and C. Gavoille. Object location using path separators. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 188–197, 2006.

[2] Y. Aifeng and Y. Jinjiang. On the vertex arboricity of planar graphs of diameter two. *Discrete Mathematics*, 307(19-20):2438–2447, 2007.

[3] J. Akiyama and M. Watanabe. Maximum iduced forests of planar graphs. *Graphs and Combinatorics*, 1(3):201–202, 1987.

[4] M. O. Albertson and D. M. Berman. A conjecture on planar graphs. *Graph Theory and Related Topics*, page 357, 1979.

[5] N. Alon, D. Mubayi, and R. Thomas. Large induced forests in sparse graphs. *Journal of Graph Theory*, 38(3):113–123, 2001.

[6] N. Alon, P. D. Seymour, and R. Thomas. A separator theorem for nonplanar graphs. *Journal of the American Mathematical Society*, 3(4):801–808, 1990.

[7] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Computational Geometry*, 9(1):81–100, 1993.

[8] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004.

[9] B. Awerbuch, M. Luby, A. V. Goldberg, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, FOCS '89, pages 364–369, 1989.

[10] V. Bafna, P. Berman, and T. Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 12(3):289–297, 1999.

[11] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.

[12] R. Bar-Yehuda, D. Geiger, J. Naor, and R. M. Roth. Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and bayesian inference. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 344–354, 1994.

[13] M. Bateni, M. Hajiaghayi, and D. Marx. Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth. *Journal of the ACM (JACM)*, 58(5):21:1–21:37, 2011.

[14] A. Becker. Approximation algorithms for the loop cutset problem. In *Proceedings of the 10th International Conference on Uncertainty in Artificial Intelligence*, UAI '94, pages 60–68, 1994.

[15] H. L Bodlaender. Dynamic programming on graphs with bounded treewidth. In *The 15th International Colloquium on Automata, Languages and Programming*, ICALP '88, pages 105–118, 1988.

[16] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243(Supplement C):86–111, 2015.

[17] O. V. Borodin. On acyclic colorings of planar graphs. *Discrete Mathematics*, 25(3):211–236, 1979.

[18] G. Borradaile, E. D. Demaine, and S. Tazari. Polynomial-time approximation schemes for subset-connectivity problems in bounded-genus graphs. *Algorithmica*, 68(2):287–311, 2014.

[19] G. Borradaile and P. Klein. An $O(n \log n)$-time algorithm for maximum st-flow in a directed planar graph. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 524–533, 2006.

[20] G. Borradaile and P. Klein. The two-edge connectivity survivable-network design problem in planar graphs. *ACM Transactions on Algorithms (TALG)*, 12(3):30:1–30:29, 2016.

[21] G. Borradaile, P. Klein, and C. Mathieu. An $O(n \log n)$ approximation scheme for steiner tree in planar graphs. *ACM Transactions on Algorithms (TALG)*, 5(3):31:1–31:31, 2009.

[22] G. Borradaile and H. Le. Optimal dynamic program for r-domination problems over tree decompositions. In *Proceedings of the 11th International Symposium on Parameterized and Exact Computation*, IPEC' 16, pages 8:1–8:23, 2016.

[23] G. Borradaile and H. Le. Light spanners for bounded treewidth graphs imply light spanners for h-minor-free graphs. *CoRR*, abs/1703.10633, 2017.

[24] G. Borradaile, H. Le, and C. Wulff-Nilsen. Minor-free graphs have light spanners. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, FOCS '17, pages 767–778, 2017.

[25] S. Cabello, E. W. Chambers, and J. Erickson. Multiple-source shortest paths in embedded graphs. *SIAM Journal on Computing*, 42(4):1542–1571, 2013.

[26] S. Cabello and D. Gajser. Simple PTAS's for families of graphs excluding a minor. *Discrete Applied Mathematics*, 189(C):41–48, 2015.

[27] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*, SocG'09, pages 333–340, 2009.

[28] S. Chaplick, M. De, A. Ravsky, and J. Spoerhase. Approximation schemes for geometric coverage problems. *CoRR*, abs/1607.06665, 2016.

[29] G. Chartrand and H.V. Kronk. The point-arboricity of planar graphs. *Journal of the London Mathematical Society*, 44:612–616, 1969.

[30] S. Chechik and C. Wulff-Nilsen. Near-optimal light spanners. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'16, pages 883–892, 2016.

[31] Y. K. Cheung, G. Goranci, and M. Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. In *43rd International Colloquium*

on Automata, Languages, and Programming (ICALP 2016), volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 131:1–131:14, 2016.

[32] V. Cohen-Addad, P. N. Klein, and C. Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, FOCS' 16, 2016.

[33] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

[34] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 150–159. IEEE, 2011.

[35] H. Dell, T. Husfeldt, Bart M. P. Jansen, P. Kaski, C. Komusiewicz, and F. A. Rosamond. The first parameterized algorithms and computational experiments challenge. In *11th International Symposium on Parameterized and Exact Computation*, volume 63 of *IPEC' 2016*, pages 30:1–30:9, 2016.

[36] E. D. Demaine and M. Hajiaghayi. Bidimensionality: New connections between FPT algorithms and PTASs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'05, pages 590–601, 2005.

[37] E. D. Demaine, M. Hajiaghayi, and K. Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 637–646, 2005.

[38] E. D. Demaine, M. Hajiaghayi, and K. Kawarabayashi. Contraction decomposition in H-minor-free graphs and algorithmic applications. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, STOC ' 11, pages 441–450, 2011.

[39] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010.

[40] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005.

[41] Frederic Dorn, Fedor V. Fomin, and Dimitrios M. Thilikos. Catalan structures and dynamic programming in H-minor-free graphs. *Journal of Computer and System Sciences*, 78(5):1606–1622, 2012.

[42] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.

[43] F. Dross, M. Montassier, and A. Pinlou. Large induced forests in planar graphs with girth 4 or 5. *arXiv preprint arXiv:1409.1348*, 2014.

[44] D. Eisenstat, P. N. Klein, and C. Mathieu. Approximating k-center in planar graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'14, pages 617–627, 2014.

[45] M. Elkin and D. Peleg. $(1+\epsilon,\beta)$-spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004.

[46] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, and Y. Villanger. Local search: Is brute-force avoidable? *Journal of Computer and System Sciences*, 78(3):707 – 719, 2012.

[47] P. Festa, P. M. Pardalos, and M .GC. Resende. Feedback set problems. In *Handbook of combinatorial optimization*, pages 209–258. Springer US, 1999.

[48] F. V. Fomin, D. Lokshtanov, F. Panolan, and S. Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of the ACM (JACM)*, 63(4):29:1–29:60, 2016.

[49] G. Frederickson. Fast algorithms for shortest paths in planar graphs with applications. *SIAM Journal on Computing*, 16:1004–1022, 1987.

[50] M. R. Garey, D. S. Johnson, and R. E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976.

[51] M. Grigni. Approximate TSP in graphs with forbidden minors. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming*, ICALP '00, pages 869–877, 2000.

[52] M. Grigni and H. Hung. Light spanners in bounded pathwidth graphs. In *Proceedings of the 37th International Conference on Mathematical Foundations of Computer Science*, MFCS'12, pages 467–477, 2012.

[53] M. Grigni and P. Sissokho. Light spanners and approximate TSP in weighted graphs with forbidden minors. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '02, pages 852–857, 2002.

[54] K. Hosono. Induced forests in trees and outerplanar graphs. *Proceedings of the Faculty of Science of Tokai University*, 25:27–29, 1990.

[55] D. S. Johnson. The NP-completeness column: An ongoing guide (column 19). *Journal of Algorithms*, 8(3):438–448, 1987.

[56] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.

[57] K. Kawarabayashi, P. N. Klein, and C. Sommer. Linear-space approximate distance oracles for planar, bounded-genus and minor-free graphs. In *Proceedings of the 38th International Colloquim Conference on Automata, Languages and Programming*, ICALP '11, pages 135–146, 2011.

[58] K. Kawarabayashi and B. Reed. A separator theorem in minor-closed classes. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 153–162, 2010.

[59] T. Kelly and C. Liu. Size of the largest induced forest in subcubic graphs of girth at least four and five. *arXiv:1603.03855*, 2016.

[60] P. N. Klein. A linear-time approximation scheme for planar weighted TSP. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 647–657, 2005.

[61] P. N. Klein. Subset spanner for planar graphs, with application to subset TSP. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, STOC '06, pages 749–756, 2006.

[62] J. Kleinberg and A. Kumar. Wavelength conversion in optical networks. *Journal of Algorithms*, 38:25–50, 2001.

[63] T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.

[64] L. Kowalik, B. Lužar, and R. Škrekovski. An improved bound on the largest induced forests for triangle-free planar graphs. *Discrete Mathematics and Theoretical Computer Science*, 12(1):87–100, 2010.

[65] R. Krauthgamer, H. L. Nguyễn, and T. Zondiner. Preserving terminal distances using minors. *SIAM Journal on Discrete Mathematics*, 28(1):127–141, 2014.

[66] C. Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930.

[67] G. Borradaile H. Le and M. Sherman-Bennett. Large induced acyclic and outerplanar subgraphs of 2-outerplanar graph. *Graphs and Combinatorics*, 33(6):1621–1634, 2017.

[68] H. Le. A better bound on the largest induced forests in triangle-free planar graphs. *ArXiv e-prints*, 2016.

[69] H. Le. A PTAS for subset TSP in minor-free graphs. *ArXiv e-prints*, 2018.

[70] H. Le and B. Zheng. Local search is a ptas for feedback vertex set in minor-free graphs. *ArXiv e-prints*, 2018.

[71] R. J. Lipton and K. W. Regan. *People, Problems, and Proofs: Essays from Gödel's Lost Letter: 2010*, chapter David Johnson: Galactic Algorithms, pages 109–112. Springer Berlin Heidelberg, 2013.

[72] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, FOCS'77, pages 162–170, 1977.

[73] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

[74] R. Lukot'ka, J. Mazák, and X. Zhu. Maximum 4-degenerate subgraph of a planar graph. *The Electronic Journal of Combinatorics*, 22(1):P1–11, 2015.

[75] W. Mader. Homomorphiesätze für graphen. *Mathematische Annalen*, 178(2):154–168, 1968.

[76] R. Matai, S. Singh, and M. Lal. Traveling salesman problem: An overview of applications, formulations, and solution approaches. In *Traveling Salesman Problem, Theory and Applications*. InTech, 2010.

[77] M. Mucha and P. Sankowski. Maximum matchings in planar graphs via gaussian elimination. In *Proceedings of the 12th Annual European Symposium on Algorithms*, ESA '04, pages 532–543, 2004.

[78] N. H. Mustafa and S. Ray. Improved results on geometric hitting set problems. *Discrete & Computational Geometry*, 44(4):883–895, 2010.

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

[80] A. Raspaud and W. Wang. On the vertex-arboricity of planar graphs. *European Journal of Combinatorics*, 29:1064–1075, 2008.

[81] G. Ringel. Das geschlecht des vollständigen paaren graphen. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 28, pages 139–150, 1965.

[82] N. Robertson and P. D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatoral Theory Series B*, 89(1):43–76, 2003.

[83] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory Series B*, 92(2):325–357, 2004.

[84] M. R. Salavatipour. Large induced forests in triangle-free planar graphs. *Graphs and Combinatorics*, 22:113–126, 2006.

[85] L Shi and H Xu. Large induced forests in graphs. *Journal of Graph Theory*, 2016.

[86] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating System Concepts*. Wiley Publishing, 2008.

[87] S. Tazari and M. Müller-Hannemann. Shortest paths in linear time on minor-closed graph classes, with an application to steiner tree approximation. *Discrete Applied Mathematics*, 157(4):673–684, 2009.

[88] A. Thomason. The extremal function for complete minors. *Journal of Combinatorial Theory Series B*, 82(2):318–338, 2001.

[89] L.G. Valiant and V.V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(0):85 – 93, 1986.

[90] Y. Wang, Q. Xie, and X. Yu. Induced forests in bipartite planar graphs. *arXiv:1605.00047*, 2016.

[91] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 7–14, 1999.