

MATCHINGS IN GEOMETRIC GRAPHS

AHMAD BINIAZ

A thesis submitted to the Faculty of Graduate and Post Doctoral Affairs
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science

Carleton University
Ottawa, Ontario, Canada, 2016

ABSTRACT

A *geometric graph* is a graph whose vertex set is a set of points in the plane and whose edge set contains straight-line segments. A *matching* in a graph is a subset of edges of the graph with no shared vertices. A matching is called *perfect* if it matches all the vertices of the underlying graph. A *geometric matching* is a matching in a geometric graph. In this thesis, we study matching problems in various geometric graphs. Among the family of geometric graphs we look at complete graphs, complete bipartite graphs, complete multipartite graphs, Delaunay graphs, Gabriel graphs, and Θ -graphs. The classical *matching problem* is to find a matching of maximum size in a given graph. We study this problem as well as some of its variants on geometric graphs. The *bottleneck matching problem* is to find a maximum matching that minimizes the length of the longest edge. The *plane matching problem* is to find a maximum matching so that the edges in the matching are pairwise non-crossing. A geometric matching is *strong* with respect to a given shape S if we can assign to each edge in the matching a scaled version of S such that the shapes representing the edges are pairwise disjoint. The *strong matching problem* is to find the maximum strong matching with respect to a given shape. The *matching packing problem* is to pack as many edge-disjoint perfect matchings as possible into a geometric graph. We study these problems and establish lower and upper bounds on the size of different kinds of matchings in various geometric graphs. We also present algorithms for computing such matchings. Some of the presented bounds are tight, while the others need to be sharpened.

ACKNOWLEDGMENTS

I would not have been able to write this thesis without the help and support of many people.

I wish to express my profound and sincere gratitude to Anil Maheshwari and Michiel Smid. They are all I could have wished for and more in my supervisors. Their continuous encouragement and belief in me were vital in the advancement of my graduate career, and made these past four years into a journey of exploration and excitement.

I also want to thank the other members and students of the Computational Geometry lab at Carleton University for making it such a nice place to work (and occasionally not work).

Finally, I am deeply thankful to my family for all the things they have done for my sake. I am grateful to my parents for their unconditional support, constant encouragement, and unflinching faith in me throughout my entire life. I am also deeply thankful to my wife, Mohadeseh, for her unfailing love and support, and to my lovely son, Mehrdad.

Thank you!

CONTENTS

| | |
|---|----|
| PREFACE | ix |
| 1 INTRODUCTION | 1 |
| 1.1 Geometric Graphs | 1 |
| 1.2 Matching Problems | 2 |
| 1.5 Summary of the thesis | 4 |
| 1.5.1 Maximum Matchings | 5 |
| 1.5.2 Strong Matchings | 6 |
| 1.5.3 Bottleneck Matchings | 7 |
| 1.5.4 Plane Matchings | 9 |
| 1.5.5 Matching Packing | 10 |
| 2 MATCHING IN GABRIEL GRAPHS | 15 |
| 2.1 Introduction | 15 |
| 2.2 Preliminaries | 17 |
| 2.3 Euclidean Bottleneck Matching | 17 |
| 2.4 Maximum Matching | 22 |
| 2.5 Blocking Higher-Order Gabriel Graphs | 30 |
| 2.6 Conclusions | 31 |
| 3 MATCHING IN TD-DELAUNAY GRAPHS | 33 |
| 3.1 Introduction | 33 |
| 3.2 Preliminaries | 36 |
| 3.3 Connectivity | 37 |
| 3.4 Hamiltonicity | 39 |
| 3.5 Perfect Matching Admissibility | 43 |
| 3.5.1 Bottleneck Perfect Matching | 43 |
| 3.5.2 Perfect Matching | 45 |
| 3.6 Blocking TD-Delaunay graphs | 50 |
| 3.7 Conclusions | 52 |
| 4 STRONG MATCHING OF POINTS WITH GEOMETRIC SHAPES | 55 |
| 4.1 Introduction | 55 |
| 4.2 Preliminaries | 58 |
| 4.3 Strong Matching in $G_S(P)$ | 60 |
| 4.4 Strong Matching in $G_\ominus(P)$ | 61 |
| 4.5 Strong Matching in $G_\nabla(P)$ | 64 |
| 4.6 Strong Matching in $G_\star(P)$ | 70 |
| 4.7 Strong Matching in $G_\square(P)$ | 75 |
| 4.8 A Conjecture on Strong Matching in $G_\ominus(P)$ | 77 |
| 4.9 Conclusions | 80 |
| 5 BOTTLENECK PLANE MATCHINGS IN A POINT SET | 83 |
| 5.1 Introduction | 83 |
| 5.2 Preliminaries | 84 |
| 5.3 Plane Matching in Unit Disk Graphs | 86 |
| 5.3.1 $\frac{1}{3}$ -approximation algorithm | 86 |

| | | |
|-------|---|-----|
| 5.3.2 | $\frac{2}{5}$ -approximation algorithm for connected unit disk graphs | 86 |
| 5.4 | Approximating Bottleneck Plane Perfect Matching | 88 |
| 5.4.1 | First Approximation Algorithm | 89 |
| 5.4.2 | Second Approximation Algorithm | 93 |
| 5.5 | Conclusions | 101 |
| 6 | BOTTLENECK PLANE MATCHINGS IN BIPARTITE GRAPHS | 103 |
| 6.1 | Introduction | 103 |
| 6.2 | Points in Convex Position | 103 |
| 6.2.1 | Points on Circle | 104 |
| 6.3 | Blue Points on Straight Line | 107 |
| 6.3.1 | First algorithm | 107 |
| 6.3.2 | Second algorithm | 108 |
| 7 | PLANE MATCHINGS IN COMPLETE MULTIPARTITE GRAPHS | 111 |
| 7.1 | Introduction | 111 |
| 7.2 | Balanced Cut Theorem | 115 |
| 7.3 | Plane Colored Matching Algorithm | 120 |
| 7.3.1 | Maximum Matching | 121 |
| 8 | PACKING MATCHINGS INTO A POINT SET | 125 |
| 8.1 | Introduction | 125 |
| 8.2 | Preliminaries | 128 |
| 8.3 | Packing Plane Matchings into Point Sets | 130 |
| 8.3.1 | Points in Convex Position | 130 |
| 8.3.2 | Points in Wheel Configurations | 131 |
| 8.3.3 | Points in General Position | 134 |
| 8.3.4 | Non-crossing Plane Matchings | 140 |
| 8.4 | Matching Removal Persistency | 141 |
| 8.5 | Conclusions | 144 |

PREFACE

This thesis is in “integrated article format” in which each chapter is based on published papers, conference proceedings, or papers awaiting publication.

- Chapter 2 considers the matching problems in Gabriel graphs. This chapter is a combination of results that have been published in the journal of Theoretical Computer Science [9] and results that have been presented in the 32nd European Workshop on Computational Geometry (EuroCG’16) [5].
- Chapter 3 considers matching problems in triangular-distance Delaunay graphs. This chapter presents the results that have been published in the journal of Computational Geometry: Theory and Applications [7]. A preliminary version of these results have been published in the proceedings of the First International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2015) [8].
- Chapter 4 considers the strong matching problem. This chapter is based on the results that have been accepted for publication in the journal of Computational Geometry: Theory and Applications, special issue in memoriam: Ferran Hurtado [4].
- Chapter 5 considers the non-crossing bottleneck matching problem in a point set. The results of this chapter have been published in the journal of Computational Geometry: Theory and Applications [1].
- Chapter 6 considers the non-crossing bottleneck matching problem in bipartite geometric graphs. The results of this chapter have appeared in the proceedings of the 26th Canadian Conference on Computational Geometry (CCCG 2014) [6].
- Chapter 7 considers the non-crossing maximum matching problem in complete multipartite geometric graphs. The results of this chapter have been published in the proceedings of the 14th International Symposium on Algorithms and Data Structures (WADS 2015) [3].
- Chapter 8 considers the problem of packing edge-disjoint perfect matchings in to a complete geometric graph. The results of this chapter have been published in the journal of Discrete Mathematics & Theoretical Computer Science [2].

BIBLIOGRAPHY

- [1] A. K. Abu-Affash, A. Biniáz, P. Carmi, A. Maheshwari, and M. H. M. Smid. Approximating the bottleneck plane perfect matching of a point set. *Computational Geometry: Theory and Applications*, 48(9):718–731, 2015.
- [2] A. Biniáz, P. Bose, A. Maheshwari, and M. H. M. Smid. Packing plane perfect matchings into a point set. *Discrete Mathematics & Theoretical Computer Science*, 17(2):119–142, 2015.
- [3] A. Biniáz, A. Maheshwari, S. C. Nandy, and M. H. M. Smid. An optimal algorithm for plane matchings in multipartite geometric graphs. In *Proceedings of the 14th International Symposium on Algorithms and Data Structures, WADS’15*, pages 66–78, 2015.

- [4] A. Biniáz, A. Maheshwari, and M. Smid. Strong matching of points with geometric shapes. *Accepted to Computational Geometry: Theory and Applications, special issue in memoriam: Ferran Hurtado*.
- [5] A. Biniáz, A. Maheshwari, and M. Smid. Bottleneck matchings and Hamiltonian cycles in higher-order Gabriel graphs. In *Proceedings of the 32nd European Workshop on Computational Geometry (EuroCG'16)*, pages 179–182, 2016.
- [6] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Bottleneck bichromatic plane matching of points. In *Proceedings of the 26th Canadian Conference on Computational Geometry, CCCG'14*, pages 431–435, 2014.
- [7] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. *Computational Geometry: Theory and Applications*, 48(9):646–660, 2015.
- [8] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. In *Proceedings of the First International Conference on Algorithms and Discrete Applied Mathematics, CALDAM'15*, pages 89–100, 2015.
- [9] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Matchings in higher-order Gabriel graphs. *Theoretical Computer Science*, 596:67–78, 2015.

INTRODUCTION

A matching in a graph is a subset of edges of the graph with no shared vertices. A geometric graph is a graph whose vertex set is a set of points in the plane and whose edge set contains straight-line segments between the points. In this thesis we consider matching problems in various geometric graphs.

In this section we provide a summary of the thesis. First we give a description of various geometric graphs that are considered in this thesis. We also provide the definition of different kinds of matching problems that we are looking at. Then we provide a brief review of the previous work on each problem, followed by a summary of the obtained results. Moreover, we present a brief description of the main ideas that are used to obtain these results.

1.1 GEOMETRIC GRAPHS

Let P be a set of n points in the plane. A *geometric graph*, $G = (P, E)$, is a graph whose vertex set is P and whose edge set E contains straight-line segments with endpoints in P . A *complete geometric graph*, $K(P)$, is a geometric graph on P that contains a straight-line edge between every pair of points in P . Let $\{P_1, \dots, P_k\}$, where $k \geq 2$, be a partition of P . A *complete multipartite geometric graph*, $K(P_1, \dots, P_k)$, is a geometric graph on P that contains a straight-line edge between every point in P_i and every point in P_j where $i \neq j$.

For two points p and q in P let $D(p, q)$ be the closed disk having \overline{pq} as diameter, and let $\nabla(p, q)$ (resp. $\triangle(p, q)$) be the smallest downward (resp. upward) equilateral triangle having p and q on its boundary. Moreover, let the *lune* $L(p, q)$ be the intersection of two disks of radius $|pq|$ that are centered at p and q , where $|pq|$ denote the Euclidean distance between p and q . See Figure 1.1 for illustration of the following geometric graphs. Assume P does not contain any four co-circular points. Then the *Delaunay triangulation* on P , denoted by $DT(P)$, has an edge between two points p and q if and only if there exists a circle that contains p and q on its boundary and whose interior does not contain any point of P . The *Gabriel graph* on P , denoted by $GG(P)$, is defined to have an edge between two points p and q if and only if $D(p, q)$ is empty. The *relative neighborhood graph* on P , denoted by $RNG(P)$, has an edge (p, q) if and only if $L(p, q)$ is empty. Similarly, the *triangular-distance Delaunay graph* on P , denoted by $TD(P)$, is defined to have an edge between two points p and q if and only if $\nabla(p, q)$ is empty. The *theta-six graph* on P , denoted by $\Theta_6(P)$, has an edge (p, q) if and only if $\nabla(p, q)$ or $\triangle(p, q)$ is empty. The *L_∞ -Delaunay graph* on P is defined to have an edge between two points p and q if and only if there exists an empty axis-parallel square that has p and q on its boundary.

The *order- k Gabriel graph* on P , denoted by k - GG , has an edge (p, q) if and only if $D(p, q)$ contains at most k points of $P \setminus \{p, q\}$. Note that 0 - GG is equal to $GG(P)$. The k - RNG graph is defined similarly. The *order- k triangular-distance Delaunay graph* on P , denoted by k - TD , has an edge (p, q) if and only if $\nabla(p, q)$ contains at most k points of $P \setminus \{p, q\}$. Note that 0 - TD is equal to $TD(P)$.

Given a radius $r \geq 0$, the *disk graph* on P , denoted by $DG(P, r)$, has an edge (p, q) if and only if the Euclidean distance between p and q is at most r . The *unit disk graph* on P , which is denoted by $UDG(P)$, is $DG(P, 1)$.

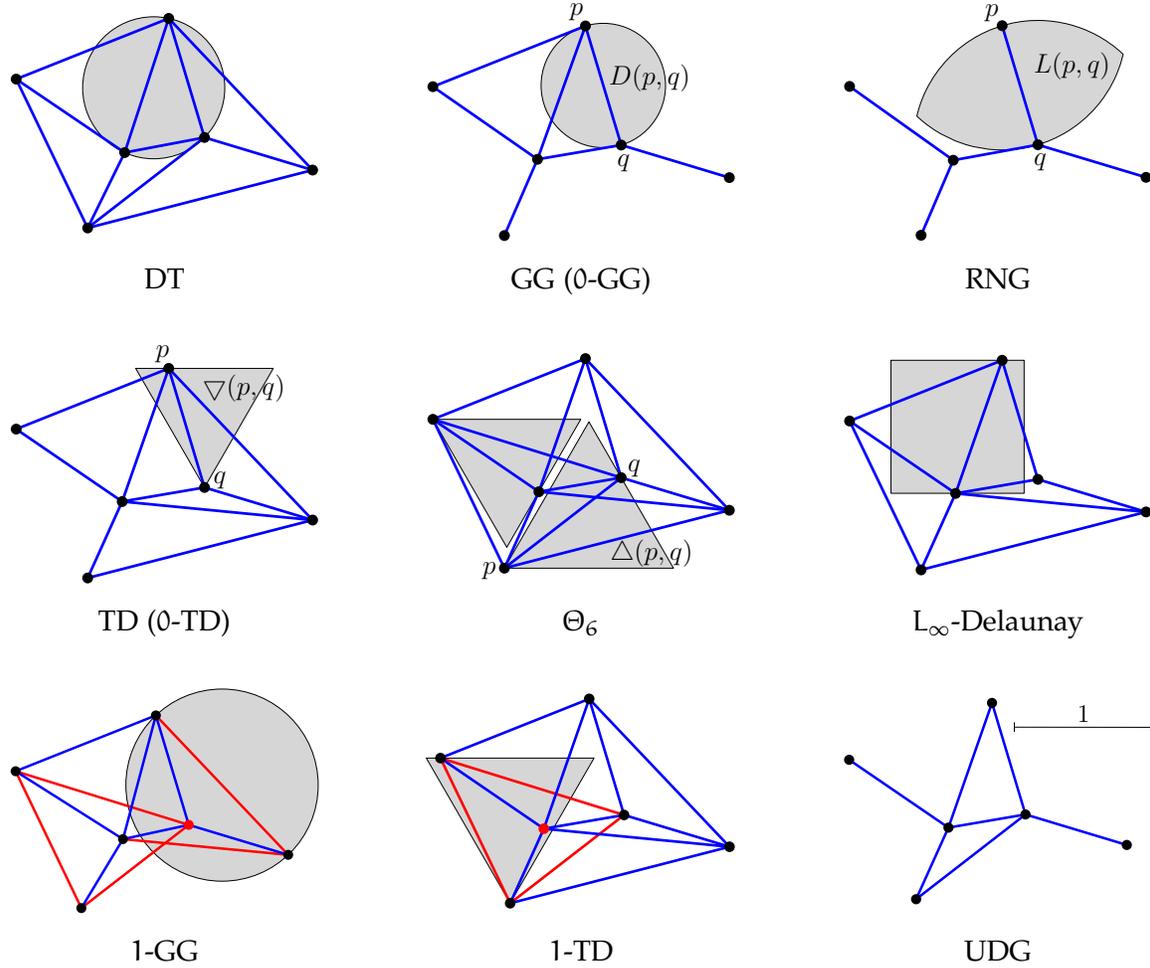


Figure 1.1: Some geometric graphs.

1.2 MATCHING PROBLEMS

Consider a geometric graph G . We say that two edges of G *cross* if they have a point in common that is interior to both edges. Two edges are *disjoint* if they have no point in common. A *matching* in G is a set of edges that do not share vertices. A *maximum matching* is a matching with maximum cardinality. A *perfect matching* is a matching that matches all the vertices of G . When we talk about perfect matchings we assume that G has an even number of points. A *plane matching* is a matching whose edges do not cross. A *bottleneck matching* is a maximum matching in G in which the length of the longest edge is minimized. Given a matching M and geometric shape S , we say that M is a *strong matching* if we can assign to each edge (p, q) in M a scaled copy of the shape S such that p and q are on the boundary of S and all the shapes are pairwise disjoint (do not overlap). A *bottleneck biconnected subgraph* of G is a 2-connected spanning subgraph of G in which the length of the longest edge is minimized. A *bottleneck Hamiltonian cycle* is defined similarly.

Let $\{P_1, \dots, P_k\}$, where $k \geq 2$, be a partition of P . Assume the points in P_i are colored C_i . P is called *color-balanced* if no color is in strict majority, i.e., $|P_i| \leq |P|/2$ for all $i \in \{1, \dots, k\}$. A matching in $K(P_1, \dots, P_k)$ is called a *colored matching*. A *balanced cut* is a line ℓ that partitions a color-balanced point set P into two color-balanced point sets Q_1 and Q_2 such that $\max\{|Q_1|, |Q_2|\} \leq \frac{2}{3}|P|$. See Figure 1.2(a). The ham-sandwich cut is a balanced cut; given a set of red and blue points,

there exists a line that simultaneously bisects the red points and the blue points. A ham-sandwich cut can be computed in $O(n)$ time [30].

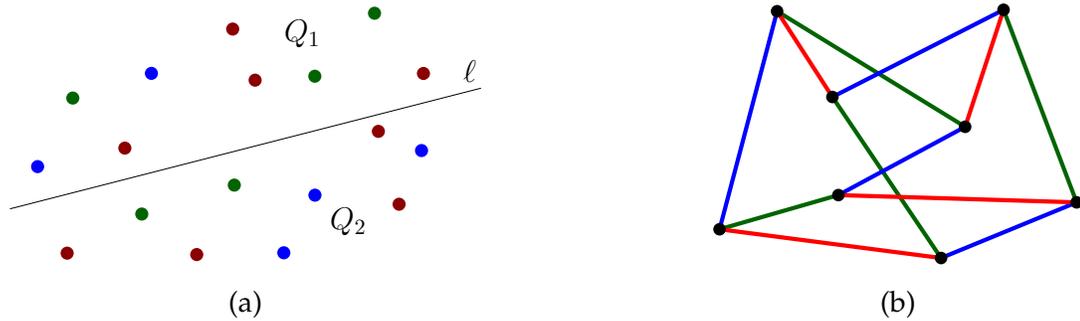


Figure 1.2: (a) A balanced cut. (b) A packing example of three edge-disjoint plane perfect matchings.

We say that a set of subgraphs is *packed into* $K(P)$, if the subgraphs in the set are pairwise edge-disjoint (do not share any edge). In a packing problem, we ask for the largest number of subgraphs of a given type that can be packed into $K(P)$. Among all subgraphs of $K(P)$, plane perfect matchings, plane spanning trees, and plane spanning paths are of interest. See Figure 1.2(b). We say that a graph G is *matching persistent* if by removing any perfect matching M from G , the resulting graph, $G - M$, has a perfect matching. We define the *matching persistency* of G as the size of the smallest set \mathcal{M} of edge-disjoint perfect matchings that can be removed from G such that $G - \mathcal{M}$ does not have any perfect matching. The *plane matching persistency* of G is defined similarly while each of the matchings in \mathcal{M} is plane.

1.3 PREVIOUS WORK

In a (strong) matching problem, usually we are looking for a (strong) matching of maximum cardinality in a specific geometric graph. Dillencourt [26] proved that the Delaunay triangulation of P , denoted by $DT(P)$, contains a perfect matching. Ábrego et al. [2] proved that $DT(P)$ has a strong circle matching of size at least $\lceil (n-1)/8 \rceil$. Ábrego et al. [2, 3] proved that L_∞ - $DT(P)$ has a perfect matching and a strong matching of size at least $\lceil n/5 \rceil$.

A problem related to the bottleneck matching problem is the *containment problem* in which we want to determine the smallest value k^* for which k^* -GG contains a bottleneck perfect matching of every point set P (or of $K(P)$). Chang et al. [25] proved that a bottleneck perfect matching of P is contained in 16-RNG. Since for a given point set k -RNG is a subgraph of k -GG, this implies that 16-GG contains a bottleneck matching of P . Chang et al. [24] showed that a bottleneck biconnected subgraph of P is contained in 1-RNG, and hence in 1-GG. In [1] it is shown that a bottleneck Hamiltonian cycle for P is contained in 15-GG. Recently, Kaiser et al. [28] improved the bound by showing that 10-GG contains a bottleneck Hamiltonian cycle of P . This implies that 10-GG has a perfect matching. They also showed that 5-GG may not contain any bottleneck Hamiltonian cycle of P .

The bottleneck plane matching problem is to compute a plane perfect matching in a geometric graph in which the length of the longest edge is minimized. Let $\{R, B\}$ be a partition of P such that $|R| = |B|$. Carlsson et al. [23] showed that the bottleneck plane perfect matching problem in $K(R, B)$ is NP-hard. When $R \cup B$ is in general position, they presented an $O(n^4 \log n)$ -time algorithm. Abu-Affash et al. [5] showed that the bottleneck plane perfect matching problem in

$K(P)$ is NP-hard and does not admit a PTAS (Polynomial Time Approximation Scheme), unless $P=NP$. Let λ^* be the length of the longest edge in an optimal bottleneck plane perfect matching in $K(P)$. Abu-Affash et al. [5] presented a polynomial-time algorithm that computes a plane perfect matching whose edges have length at most $2\sqrt{10}\lambda^*$.

A plane perfect matching in $K(R, B)$, with $|R| = |B|$, can be computed optimally in $\Theta(n \log n)$ time by recursively applying the ham-sandwich theorem [30]. Let $\{P_1, \dots, P_k\}$ with $k \geq 2$, be a partition of P . Aichholzer et al. [6] and Kano et al. [29] showed that $K(P_1, \dots, P_k)$ has a plane colored matching if and only if P is color-balanced. Kano et al. [29] presented an algorithm that computes such a matching in $O(n^2 \log n)$ time.

In a packing problem we usually ask for the maximum number of edge-disjoint structures that can be packed into a complete graph. Aichholzer et al. [7] considered the problem of packing plane spanning trees and plane Hamiltonian paths into $K(P)$. They showed that at least \sqrt{n} plane spanning trees can be packed into $K(P)$. They also showed how to pack 2 plane Hamiltonian paths into $K(P)$.

1.4 OBTAINED RESULTS

We looked at the maximum matching problem in Gabriel graphs and TD-Delaunay graphs. We showed that o-GG, 1-GG, and 2-GG has a matching of size at least $\frac{n-1}{4}$, $\frac{2(n-1)}{5}$, and $\frac{n-1}{2}$, respectively [22]. We also showed that o-TD, 1-TD, 2-TD contains a matching of size at least $\frac{n-1}{3}$, $\frac{2(n-1)}{5}$, and $\frac{n-1}{2}$, respectively [20, 21]. As for Euclidean bottleneck matching, we showed that a bottleneck perfect matching of P is contained in 9-GG, but 8-GG may not have any bottleneck perfect matching [22, 18]. As for triangular-distance bottleneck matching, we showed that a bottleneck perfect matching of P is contained in 6-TD, but 5-TD may not have any bottleneck perfect matching [20, 21]. In [8, 9] we considered the maximum matching problem in o-TD.

In [17] we considered the strong matching problem. We showed that if shape S is a diametral-disk (a disk whose diameter is a line segment between two input points), a downward equilateral triangle, and an axis-aligned square then P admits a strong matching of size at least $\frac{n-1}{17}$, $\frac{n-1}{9}$, and $\frac{n-1}{4}$, respectively. If both downward and upward equilateral triangles are allowed, we compute a strong matching of size at least $\frac{n-1}{4}$.

For the plane matching problem in complete multipartite geometric graphs, we show how to compute a plane maximum matching in $K(P_1, \dots, P_k)$ in $\Theta(n \log n)$ time [16]. We also extended this results to the case where the points are in the interior of a simple polygon [15, 13].

In [4] we considered the plane bottleneck matching problem in $K(P)$, which is NP-hard. We show how to compute a plane bottleneck matching of size at least $\frac{n}{5}$ in $K(P)$ with edges of length at most λ^* in $O(n \log^2 n)$ time, where λ^* is the length of the longest edge in an optimal bottleneck plane perfect matching. We also presented an $O(n \log n)$ -time approximation algorithm that computes a plane matching of size at least $\frac{2n}{5}$ in $K(P)$ whose edges have length at most $(\sqrt{2} + \sqrt{3})\lambda^*$. In [19] we considered the bottleneck matching problem in complete bipartite geometric graphs.

In [14] we considered the problem of packing plane edge-disjoint matchings into $K(P)$. We proved that at least $\lfloor \log_2 n \rfloor - 1$ edge-disjoint plane perfect matchings can be packed into $K(P)$.

1.5 SUMMARY OF THE THESIS

In this section we describe the obtained results in more detail. Moreover we give a brief description of the approaches used to obtain these results.

1.5.1 Maximum Matchings

In [22, 18, Chapter 2] and [21, 20, Chapter 3] we consider the maximum matching and bottleneck matching containment problems in k -GG and k -TD graphs. In [22] we showed that 0 -GG has a matching of size at least $\frac{n-1}{4}$ and this bound is tight. We also proved that 1 -GG has a matching of size at least $\frac{2(n-1)}{5}$ and 2 -GG has a perfect matching. As for Euclidean bottleneck matching, we showed that a bottleneck matching of P is contained in 9 -GG, but 8 -GG may not have any bottleneck matching; see [22, 18]. In [18] we also showed that 7 -GG may not contain any bottleneck Hamiltonian cycle of P . In [9, 8] we have shown that 0 -TD contains a matching of size at least $\frac{n-1}{3}$ and this bound is tight (this result is not included in the thesis). In [21, 20] we proved that 1 -TD has a matching of size at least $\frac{2(n-1)}{5}$ and 2 -TD has a perfect matching. As for triangular-distance bottleneck matching, we showed that a bottleneck matching of P is contained in 6 -TD, but 5 -TD may not have any bottleneck matching. We also showed that a bottleneck biconnected subgraph of P is contained in 1 -TD. In addition, we showed that 7 -TD contains a bottleneck Hamiltonian cycle of P while 5 -TD may not contain any. Tables 1.1 and 1.2 summarize the results.

Table 1.1: Lower bounds on the size of maximum (strong) matchings.

| Graph | Max matching | Reference | Strong matching | Reference |
|----------------|--------------------------------|---------------------|--|-----------------------------|
| DT | $\lfloor \frac{n}{2} \rfloor$ | [26] | $\lceil \frac{n-1}{8} \rceil$ | [2] |
| L_∞ -DT | $\lfloor \frac{n}{2} \rfloor$ | [2, 3] | $\lceil \frac{n}{5} \rceil$ $\lceil \frac{n-1}{4} \rceil$ | [2, 3] [17, Section 4.7] |
| 0 -GG | $\lceil \frac{n-1}{4} \rceil$ | [22, Section 2.4] | $\lceil \frac{n-1}{17} \rceil$ | [17, Section 4.4] |
| 1 -GG | $\lceil \frac{2n-2}{5} \rceil$ | [22, Section 2.4] | - | - |
| 2 -GG | $\lfloor \frac{n}{2} \rfloor$ | [22, Section 2.4] | - | - |
| 0 -TD | $\lceil \frac{n-1}{3} \rceil$ | [9] | $\lceil \frac{n-1}{9} \rceil$ | [17, Section 4.5] |
| 1 -TD | $\lceil \frac{2n-2}{5} \rceil$ | [21, Section 3.5.2] | - | - |
| 2 -TD | $\lfloor \frac{n}{2} \rfloor$ | [21, Section 3.5.2] | - | - |
| Θ_6 | $\lceil \frac{n-1}{3} \rceil$ | [9] | $\lceil \frac{n-1}{4} \rceil$ | [17, Section 4.6] |

OUR APPROACH: In order to provide a lower bound on the size of a maximum matching in k -GG and k -TD, we first give a lower bound on the number of components that result after removing a set S of vertices from k -GG and k -TD. Then we use the following theorem of Tutte and Berge. For a graph $G = (V, E)$ and $S \subseteq V$, let $G - S$ be the subgraph obtained from G by removing all vertices in S , and let $o(G - S)$ be the number of odd components in $G - S$, i.e., connected components with an odd number of vertices. In a graph G , the *deficiency*, $\text{def}_G(S)$, is $o(G - S) - |S|$. Let $\text{def}(G) = \max_{S \subseteq V} \text{def}_G(S)$.

THEOREM 1.1 (Tutte-Berge formula [12]). *The size of a maximum matching in G is*

$$\frac{1}{2}(n - \text{def}(G)).$$

In order to show that 9 -GG and 6 -TD contain bottleneck matchings, we do the following. For a matching M we define the *length sequence* of M , $LS(M)$, as the sequence containing the lengths

Table 1.2: Bottleneck structures containment in k-GG and k-TD.

| Distance | Bott. struct. | \notin | Reference | \in | Reference |
|------------|----------------------|----------|---------------------|-------|---------------------|
| Euclidean | biconnected subgraph | 0-GG | [24] | 1-GG | [24] |
| | matching | 8-GG | [22, Section 2.3.2] | 9-GG | [18, Section 2.3.1] |
| | cycle | 7-GG | [18, Section 2.3.3] | 10-GG | [28] |
| Triangular | biconnected subgraph | 0-TD | [21, Section 3.3.2] | 1-TD | [21, Section 3.3.2] |
| | matching | 5-TD | [21, Section 3.5.1] | 6-TD | [21, Section 3.5.1] |
| | cycle | 5-TD | [21, Section 3.4] | 7-TD | [21, Section 3.4] |

of the edges of M in non-increasing order. A matching M_1 is said to be less than a matching M_2 if $LS(M_1)$ is lexicographically smaller than $LS(M_2)$. Then, we show that the matching with the minimum length sequence is contained in 10-GG and 6-TD.

1.5.2 Strong Matchings

A geometric matching is *strong* with respect to a given shape S if we can assign to each edge in the matching a scaled version of S with the endpoints of the edge on the boundary, such that the shapes representing the edges are pairwise disjoint.

In [17, Chapter 4] we considered the strong matching problem with respect to a given geometric object S . We proved that if S is a diametral-disk (a disk whose diameter is a line segment between two input points), then P (and hence $GG(P)$) has a strong matching of size at least $\lceil \frac{n-1}{17} \rceil$, and if S is a downward equilateral triangle, then P (and hence $TD(P)$) has a strong matching of size at least $\lceil \frac{n-1}{9} \rceil$. In case both downward and upward equilateral triangles are allowed, we compute a strong matching of size at least $\lceil \frac{n-1}{4} \rceil$ in P (and hence in $\Theta_6(P)$). If S is an axis-aligned square, then we compute a strong matching of size at least $\lceil \frac{n-1}{4} \rceil$ in P (and hence in L_∞ -DT(P)); this improves the previous bound of $\lceil \frac{n}{5} \rceil$. The results are summarized in Table 1.1.

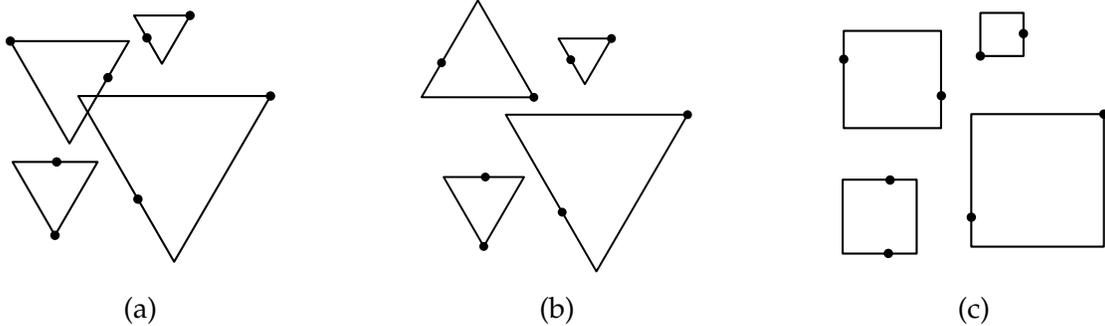


Figure 1.3: Point set P and (a) a non-strong perfect matching in $G_{\nabla}(P)$ (two shapes overlap), (b) a perfect strong matching in $G_{\blacktriangledown}(P)$, and (c) a perfect strong matching in $G_{\square}(P)$.

OUR APPROACH: To compute a strong matching with diametral-disks and equilateral triangles we present the following algorithm that is depicted in Algorithm 1. For a given point set P and a geometric shape $S \in \{\ominus, \nabla\}$, the edge weighted geometric graph $G_S(P)$ on P is defined to have an edge between two points of P if and only if there exists an empty shape S having the two points

on its boundary. The weight of each edge e is equal to the area of $S(e)$, where $S(e)$ is a smallest scaled copy of S containing e . Let T be a minimum spanning tree of $G_S(P)$. For each edge $e \in T$ we denote by $T(e^+)$ the set of all edges in T whose weight is at least $w(e)$. Moreover, we define the *influence set* of e , as the set of all edges in $T(e^+)$ whose representing shapes overlap with $S(e)$, i.e.,

$$\text{Inf}(e) = \{e' : e' \in T(e^+), S(e') \cap S(e) \neq \emptyset\}.$$

Note that $\text{Inf}(e)$ is not empty, as $e \in \text{Inf}(e)$. Consequently, we define the *influence number* of T to be the maximum size of a set among the influence sets of edges in T , i.e.,

$$\text{Inf}(T) = \max\{|\text{Inf}(e)| : e \in T\}.$$

Algorithm 1 receives $G_S(P)$ as input and computes a strong matching in $G_S(P)$ as follows. The algorithm starts by computing a minimum spanning tree T of $G_S(P)$, where the weight of each edge is equal to the area of its representing shape. Then it initializes a forest F by T , and a matching \mathcal{M} by an empty set. Afterwards, as long as F is not empty, the algorithm adds to \mathcal{M} , the smallest edge e in F , and removes the influence set of e from F . Finally, it returns \mathcal{M} .

Algorithm 1 *Strong-matching*($G_S(P)$)

```

1:  $T \leftarrow \text{MST}(G_S(P))$ 
2:  $F \leftarrow T$ 
3:  $\mathcal{M} \leftarrow \emptyset$ 
4: while  $F \neq \emptyset$  do
5:    $e \leftarrow$  smallest edge in  $F$ 
6:    $\mathcal{M} \leftarrow \mathcal{M} \cup \{e\}$ 
7:    $F \leftarrow F - \text{Inf}(e)$ 
8: return  $\mathcal{M}$ 

```

THEOREM 1.2. *Given a set P of n points in the plane and a shape $S \in \{\ominus, \nabla\}$, Algorithm 1 computes a strong matching of size at least $\lceil \frac{n-1}{\text{Inf}(T)} \rceil$ in $G_S(P)$, where T is a minimum spanning tree of $G_S(P)$.*

In order to compute a strong matching with diametral-disks and equilateral triangles, we show that if S is \ominus then $\text{Inf}(T) \leq 17$, and if S is ∇ , then $\text{Inf}(T) \leq 9$.

The bounds for the strong matching when S is an square is proved by induction on the area of the smallest axis-aligned square containing P . The bounds for the strong matching when S is allowed to be upward or downward equilateral triangle is proved similarly.

1.5.3 Bottleneck Matchings

A *bottleneck matching* in a graph G is a maximum matching in which the length of the longest edge is minimized. A *plane bottleneck matching* is a bottleneck matching that is non-crossing.

Table 1.3: Approximating plane bottleneck matching.

| time complexity | plane bottleneck | size of matching | Reference |
|----------------------------|----------------------------------|------------------|--------------------|
| $O(n^{1.5} \sqrt{\log n})$ | $2\sqrt{10}\lambda^*$ | $n/2$ | [5] |
| $O(n \log^2 n)$ | λ^* | $n/5$ | [4, Section 5.4.1] |
| $O(n \log n)$ | $(\sqrt{2} + \sqrt{3})\lambda^*$ | $2n/5$ | [4, Section 5.4.2] |

Problems related to computing bottleneck plane matchings in geometric graphs are considered in [4, Chapter 5] and [19, Chapter 6]. Computing a bottleneck plane matching in $K(P)$ is NP-hard. In [4], we present an $O(n \log n)$ -time algorithm that computes a plane matching of size at least $\frac{n-1}{5}$ in a connected disk graph. Using this algorithm we obtain a bottleneck plane matching of size at least $\frac{n}{5}$ in $K(P)$ with edges of length at most λ^* in $O(n \log^2 n)$ time, where λ^* is the length of the longest edge in an optimal bottleneck plane perfect matching in $K(P)$. We also presented an $O(n \log n)$ -time approximation algorithm that computes a plane matching of size at least $\frac{2n}{5}$ in $K(P)$ whose edges have length at most $(\sqrt{2} + \sqrt{3})\lambda^*$. Table 1.3 summarizes the results. In [19] we considered the bottleneck plane matching problem in $K(R, B)$ with $|R| = |B|$. This problem is NP-hard; we provided polynomial-time algorithms that compute exact solutions for some special cases of the problem. When $R \cup B$ is in convex position we solve this problem in $O(n^3)$ time that improves upon the previous algorithm of [23] by a factor of $n \log n$. If $R \cup B$ is on the boundary of a circle we solve this problem in $O(n \log n)$ time. If the points in R are on a line and the points in B are on one side of the line, we solve this problem in $O(n^4)$ time. Table 1.4 summarizes the results.

OUR APPROACH: In order to compute a bottleneck plane matching in $K(P)$ we do the following. First we present an $O(n \log n)$ -time algorithm that computes a plane matching of size at least $\frac{n-1}{5}$ in any connected disk graph. Using this algorithm we obtain a bottleneck plane matching of size at least $\frac{n}{5}$ in $K(P)$ with edges of length at most λ^* in $O(n \log^2 n)$ time.

We show that every minimum spanning tree on $K(P)$ is a minimum spanning tree of $UDG(P)$. Monma et al. [31] proved that every set of points in the plane admits a minimum spanning tree of degree at most five that can be computed in $O(n \log n)$ time. Now we present an algorithm that extracts a plane matching M from a minimum spanning tree T of $UDG(P)$ with vertices of degree at most five. We define the *skeleton tree*, T' , as the tree obtained from T by removing all its leaves; see Figure 1.4. Clearly $T' \subseteq T \subseteq UDG(P)$. For clarity we use u and v to refer to the leaves of T and T' respectively. In addition, let v and v' , respectively, refer to the copies of a vertex v in T and T' . In each step, pick an arbitrary leaf $v' \in T'$. By the definition of T' , it is clear that the copy of v' in T , i.e. v , is connected to vertices u_1, \dots, u_k , for some $1 \leq k \leq 4$, that are leaves of T (if T' has one vertex then $k \leq 5$). Pick an arbitrary leaf u_i and add (v, u_i) as a matched pair to M . For the next step we update T by removing v and all its adjacent leaves. We also compute the new skeleton tree and repeat this process. In the last iteration, T' is empty and we may be left with a tree T consisting of one single vertex or one single edge. If T consists of one single vertex, we disregard it, otherwise we add its only edge to M . M has size at least $\frac{n-1}{5}$ and can be computed in $O(n \log n)$ time.

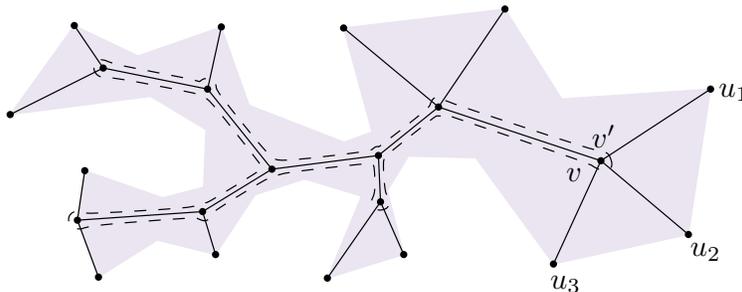


Figure 1.4: Minimum spanning tree T with union of empty convex hulls. The skeleton tree T' is surrounded by dashed line, and v' is a leaf in T' .

Then we compute a (possibly crossing) bottleneck perfect matching M_\times of $K(P)$ using the algorithm in [25]. Let λ_{M_\times} denote the length of the bottleneck edge in M_\times . It is obvious that the bottleneck length of any plane perfect matching is not less than λ_{M_\times} . Therefore, $\lambda^* \geq \lambda_{M_\times}$. We consider a “unit” disk graph $DG(\lambda_{M_\times}, P)$ over P , in which there is an edge between two vertices p and q if $|pq| \leq \lambda_{M_\times}$. Note that $DG(\lambda_{M_\times}, P)$ is not necessarily connected. Let G_1, \dots, G_k be the connected components of $DG(\lambda_{M_\times}, P)$. For each component G_i , consider a minimum spanning tree T_i of degree at most five. We show how to extract from T_i a plane matching M_i of proper size $\frac{n}{5}$ and length λ^* . We show how to compute M in $O(n \log^2 n)$ time.

To compute a plane matching of size at least $\frac{2n}{5}$ in $K(P)$ with edges of length at most $(\sqrt{2} + \sqrt{3})\lambda^*$, we do the following. Let $DT(P)$ denote the Delaunay triangulation of P . Let the edges of $DT(P)$ be, in sorted order of their lengths, e_1, e_2, \dots . Initialize a forest F consisting of n trees, each one being a single node for one point of P . Run Kruskal’s algorithm on the edges of $DT(P)$ and terminate as soon as every tree in F has an even number of nodes. Let e_l be the last edge that is added by Kruskal’s algorithm. Observe that e_l is the longest edge in F . Denote the trees in F by T_1, \dots, T_k and for $1 \leq i \leq k$, let P_i be the vertex set of T_i and let $n_i = |P_i|$. Then we prove the following lemma.

LEMMA 1.3. $\lambda^* \geq |e_l|$.

By Lemma 1.3 the length of the longest edge in F is at most λ^* . For each $T_i \in F$, where $1 \leq i \leq k$, we compute a plane matching M_i of P_i of size at least $\frac{2n_i}{5}$ with edges of length at most $(\sqrt{2} + \sqrt{3})\lambda^*$ and return $\bigcup_{i=1}^k M_i$. This gives a plane matching of P of size at least $\frac{2}{5}n$ with bottleneck at most $(\sqrt{2} + \sqrt{3})\lambda^*$. This matching can be computed in $O(n \log n)$ time.

The plane bottleneck matchings for the special cases of $R \cup B$ in $K(R, B)$ are computed by dynamic programming.

1.5.4 Plane Matchings

There has been much research to extend the well-known ham-sandwich theorem—that partitions a two colored point set—to more colors; see [10, 11, 29]. For a color-balanced point set P in the plane we showed how to compute a balanced cut in linear time [16, Chapter 7]. Moreover, by applying balanced cuts recursively, we computed a plane matching in $K(P_1, \dots, P_k)$ in $\Theta(n \log n)$ time. We have also extended this notion for points that are in the interior of a simple polygon [15] (this extension is not included in the thesis).

OUR APPROACH: Let $\{R, B\}$ be a partition of P such that $|R| = |B|$. Assume the points in R are colored red and the points in B are colored blue. A plane perfect matching in $K_n(R, B)$ can be computed in $\Theta(n \log n)$ time by recursively applying the following Ham Sandwich Theorem.

THEOREM 1.4 (Ham Sandwich Theorem). *For a point set P in general position in the plane that is partitioned into sets R and B , there exists a line that simultaneously bisects R and B .*

Let $\{P_1, \dots, P_k\}$, with $k > 2$, be a partition of P . A necessary and sufficient for the existence of a plane perfect matching in $K_n(P_1, \dots, P_k)$ is obtained by the following theorem.

THEOREM 1.5 (Aichholzer et al. [6], and Kano et al. [29]). *Let $k \geq 2$ and consider a partition $\{P_1, \dots, P_k\}$ of a point set P , where $|P|$ is even. Then, $K_n(P_1, \dots, P_k)$ has a plane colored perfect matching if and only if P is color-balanced.*

Table 1.4: Plane colored matchings.

| Problem | Point set | Time complexity | Reference |
|--|---------------------|--------------------|---------------------|
| bottleneck 2-colored plane perfect matching | general position | NP-hard | [23] |
| | convex position | $O(n^4 \log n)$ | [23] |
| | convex position | $O(n^3)$ | [19, Section 6.2] |
| | on circle | $O(n \log n)$ | [19, Section 6.2.1] |
| | one color on a line | $O(n^4)$ | [19, Section 6.3] |
| plane maximum matching | 2-colored | $\Theta(n \log n)$ | [27] |
| | k-colored | $\Theta(n \log n)$ | [16, Section 7.3] |

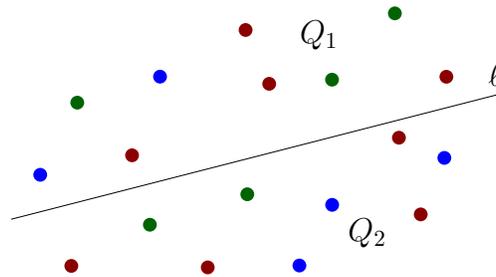


Figure 1.5: Illustrating the balanced cut theorem.

Based on that, we first prove the existence of a balanced cut in a color-balanced point set.

THEOREM 1.6 (Balanced Cut Theorem). *Let P be a color-balanced point set of $n \geq 4$ points in general position in the plane. In $O(n)$ time we can compute a line ℓ such that*

1. ℓ does not contain any point of P .
2. ℓ partitions P into two point sets Q_1 and Q_2 , where
 - a) both Q_1 and Q_2 are color-balanced,
 - b) both Q_1 and Q_2 contains at most $\frac{2}{3}n + 1$ points.
 - c) if $|P|$ is even, then both $|Q_1|$ and $|Q_2|$ are even.

Then, by applying the Balanced Cut Theorem recursively, we can compute a plane perfect matching in matching in $K_n(P_1, \dots, P_k)$ in $\Theta(n \log n)$ time.

1.5.5 Matching Packing

Recall that in a matching packing problem, we ask for the largest number of matchings that can be packed into $K(P)$. A plane matching packing problem is to pack non-crossing matchings to $K(P)$.

In [14, Chapter 8] we consider the problem of packing perfect matchings into $K(P)$. Let n be the number of points in P , and assume that n is an even number. We proved that if P is in general position, then at least $\lfloor \log_2 n \rfloor - 1$ plane perfect matchings can be packed into $K(P)$. Moreover, we show that for some point set P in general position, no more than $\lceil \frac{n}{3} \rceil$ can be packed into $K(P)$.

If P is in convex position we show that the maximum number of plane perfect matchings that can be packed to $K(P)$ is $\frac{n}{2}$, and $\frac{n}{2} - 1$ when P is in regular wheel configuration. As for matching persistency, we showed that the matching persistency of $K(P)$ is $\frac{n}{2}$ if $n \equiv 2 \pmod{4}$, and $\frac{n}{2} + 1$ if $n \equiv 0 \pmod{4}$. As for plane matchings we showed that if P is in convex position then the plane matching persistency of $K(P)$ is 2. We also show the existence of a set of points in general position with plane matching persistence of at least 3.

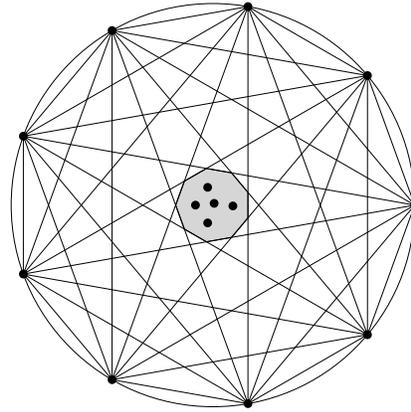


Figure 1.6: About 1/3 of the points are in the middle and the boundary has an odd number of points.

OUR APPROACH: In order to show the upper bound, we provide an example, as depicted in Figure 1.6, that does not contain more than $\lceil \frac{n}{3} \rceil$ edge-disjoint plane perfect matchings.

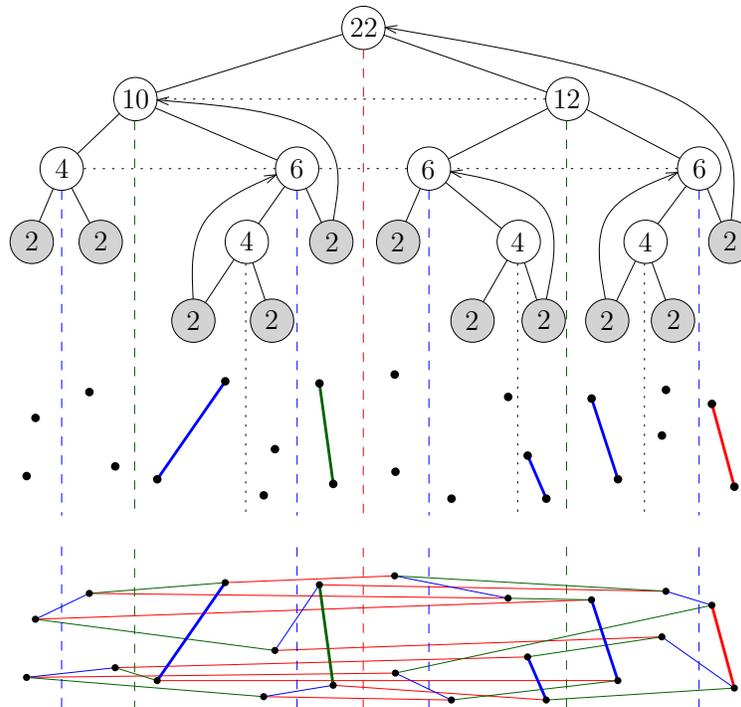


Figure 1.7: The points in P are assigned, in pairs, to the leaves of T , from left to right. Internal nodes store the number of points in their subtree. The three edge disjoint plane perfect matchings are shown in red, green, and blue.

To prove the lower bound of $\lfloor \log_2 n \rfloor - 1$, we first build a binary tree T on the point set P , then we assign the points of P to the leaves of T , and then we extract a plane matching from the internal nodes in each level of T . See Figure 1.7.

BIBLIOGRAPHY

- [1] M. Abellanas, P. Bose, J. García-López, F. Hurtado, C. M. Nicolás, and P. Ramos. On structural and graph theoretic properties of higher order Delaunay graphs. *International Journal of Computational Geometry & Applications*, 19(6):595–615, 2009.
- [2] B. M. Ábrego, E. M. Arkin, S. Fernández-Merchant, F. Hurtado, M. Kano, J. S. B. Mitchell, and J. Urrutia. Matching points with circles and squares. In *Proceedings of the Japanese Conference Discrete and Computational Geometry, JCDCG*, pages 1–15, 2004.
- [3] B. M. Ábrego, E. M. Arkin, S. Fernández-Merchant, F. Hurtado, M. Kano, J. S. B. Mitchell, and J. Urrutia. Matching points with squares. *Discrete & Computational Geometry*, 41(1):77–95, 2009.
- [4] A. K. Abu-Affash, A. Biniáz, P. Carmi, A. Maheshwari, and M. H. M. Smid. Approximating the bottleneck plane perfect matching of a point set. *Computational Geometry: Theory and Applications*, 48(9):718–731, 2015.
- [5] A. K. Abu-Affash, P. Carmi, M. J. Katz, and Y. Trabelsi. Bottleneck non-crossing matching in the plane. *Computational Geometry: Theory and Applications*, 47(3):447–457, 2014.
- [6] O. Aichholzer, S. Cabello, R. F. Monroy, D. Flores-Peñaloza, T. Hackl, C. Huemer, F. Hurtado, and D. R. Wood. Edge-removal and non-crossing configurations in geometric graphs. *Disc. Math. & Theo. Comp. Sci.*, 12(1):75–86, 2010.
- [7] O. Aichholzer, T. Hackl, M. Korman, M. van Kreveld, M. Löffler, A. Pilz, B. Speckmann, and E. Welzl. Packing plane spanning trees and paths in complete geometric graphs. In *Proceedings of the 26th Annual Canadian Conference on Computational Geometry, CCCG'14*, pages 233–238, 2014.
- [8] J. Babu, A. Biniáz, A. Maheshwari, and M. H. M. Smid. Fixed-orientation equilateral triangle matching of point sets. In *Proceedings of the 7th International Workshop on Algorithms and Computation, WALCOM'13*, pages 17–28, 2013.
- [9] J. Babu, A. Biniáz, A. Maheshwari, and M. H. M. Smid. Fixed-orientation equilateral triangle matching of point sets. *Theoretical Computer Science*, 555:55–70, 2014.
- [10] S. Bereg, F. Hurtado, M. Kano, M. Korman, D. Lara, C. Seara, R. I. Silveira, J. Urrutia, and K. Verbeek. Balanced partitions of 3-colored geometric sets in the plane. *Discrete Applied Mathematics*, 181:21–32, 2015.
- [11] S. Bereg and M. Kano. Balanced line for a 3-colored point set in the plane. *The Electronic Journal of Combinatorics*, 19(1):P33, 2012.
- [12] C. Berge. Sur le couplage maximum d'un graphe. *C. R. Acad. Sci. Paris*, 247:258–259, 1958.

- [13] A. Biniiaz, P. Bose, A. Maheshwari, and M. Smid. Plane geodesic spanning trees, Hamiltonian cycles, and perfect matchings in a simple polygon. *Computational Geometry: Theory and Applications*, 2016.
- [14] A. Biniiaz, P. Bose, A. Maheshwari, and M. H. M. Smid. Packing plane perfect matchings into a point set. *Discrete Mathematics & Theoretical Computer Science*, 17(2):119–142, 2015.
- [15] A. Biniiaz, P. Bose, A. Maheshwari, and M. H. M. Smid. Plane geodesic spanning trees, Hamiltonian cycles, and perfect matchings in a simple polygon. In *Proceedings of the first international conference on Topics in Theoretical Computer Science, TTCS'15*, pages 56–71, 2015.
- [16] A. Biniiaz, A. Maheshwari, S. C. Nandy, and M. H. M. Smid. An optimal algorithm for plane matchings in multipartite geometric graphs. In *Proceedings of the 14th International Symposium on Algorithms and Data Structures, WADS'15*, pages 66–78, 2015.
- [17] A. Biniiaz, A. Maheshwari, and M. Smid. Strong matching of points with geometric shapes. *Accepted to Computational Geometry: Theory and Applications, special issue in memoriam: Ferran Hurtado*.
- [18] A. Biniiaz, A. Maheshwari, and M. Smid. Bottleneck matchings and Hamiltonian cycles in higher-order Gabriel graphs. In *Proceedings of the 32nd European Workshop on Computational Geometry (EuroCG'16)*, pages 179–182, 2016.
- [19] A. Biniiaz, A. Maheshwari, and M. H. M. Smid. Bottleneck bichromatic plane matching of points. In *Proceedings of the 26th Canadian Conference on Computational Geometry, CCCG'14*, pages 431–435, 2014.
- [20] A. Biniiaz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. In *Proceedings of the First International Conference on Algorithms and Discrete Applied Mathematics, CALDAM'15*, pages 89–100, 2015.
- [21] A. Biniiaz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. *Computational Geometry: Theory and Applications*, 48(9):646–660, 2015.
- [22] A. Biniiaz, A. Maheshwari, and M. H. M. Smid. Matchings in higher-order Gabriel graphs. *Theoretical Computer Science*, 596:67–78, 2015.
- [23] J. Carlsson and B. Armbruster. A bottleneck matching problem with edge-crossing constraints. Manuscript, see <http://users.iems.northwestern.edu/~armbruster/2010matching.pdf>, 2010.
- [24] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. Solving the Euclidean bottleneck biconnected edge subgraph problem by 2-relative neighborhood graphs. *Discrete Applied Mathematics*, 39(1):1–12, 1992.
- [25] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. Solving the Euclidean bottleneck matching problem by k-relative neighborhood graphs. *Algorithmica*, 8(3):177–194, 1992.
- [26] M. B. Dillencourt. Toughness and Delaunay triangulations. *Discrete & Computational Geometry*, 5:575–601, 1990.

- [27] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32(2):249–267, 1992.
- [28] T. Kaiser, M. Saumell, and N. V. Cleemput. 10-Gabriel graphs are hamiltonian. *Information Processing Letters*, 115(11):877–881, 2015.
- [29] M. Kano, K. Suzuki, and M. Uno. Properly colored geometric matchings and 3-trees without crossings on multicolored points in the plane. In *16th Japanese Conference on Disc. Comp. Geom. and Graphs, JCDCGG*, pages 96–111, 2013.
- [30] C. Lo, J. Matousek, and W. L. Steiger. Algorithms for ham-sandwich cuts. *Discrete & Computational Geometry*, 11:433–452, 1994.
- [31] C. L. Monma and S. Suri. Transitions in geometric minimum spanning trees. *Discrete & Computational Geometry*, 8:265–293, 1992.

Given a set P of n points in the plane, the order- k Gabriel graph on P , denoted by k -GG, has an edge between two points p and q if and only if the closed disk with diameter pq contains at most k points of P , excluding p and q . We study matching problems in k -GG graphs. We show that a Euclidean bottleneck matching of P is contained in 9-GG, but 8-GG may not have any Euclidean bottleneck matching. In addition we show that 0-GG has a matching of size at least $\frac{n-1}{4}$ and this bound is tight. We also prove that 1-GG has a matching of size at least $\frac{2(n-1)}{5}$ and 2-GG has a perfect matching. Finally we consider the problem of blocking the edges of k -GG.

This chapter is a combination of results that have been published in the journal of Theoretical Computer Science [6] and results that have been presented in the 32nd European Workshop on Computational Geometry (EuroCG'16) [5].

2.1 INTRODUCTION

Let P be a set of n points in the plane. For any two points $p, q \in P$, let $D[p, q]$ denote the closed disk which has the line segment \overline{pq} as diameter. Let $|pq|$ be the Euclidean distance between p and q . The *Gabriel graph* on P , denoted by $GG(P)$, is defined to have an edge between two points p and q if $D[p, q]$ is empty of points in $P \setminus \{p, q\}$. Let $C(p, q)$ denote the circle which has \overline{pq} as diameter. Note that if there is a point of $P \setminus \{p, q\}$ on $C(p, q)$, then $(p, q) \notin GG(P)$. That is, (p, q) is an edge of $GG(P)$ if and only if

$$|pq|^2 < |pr|^2 + |rq|^2 \quad \forall r \in P, \quad r \neq p, q.$$

Gabriel graphs were introduced by Gabriel and Sokal [12] and can be computed in $O(n \log n)$ time [14]. Every Gabriel graph has at most $3n - 8$ edges, for $n \geq 5$, and this bound is tight [14].

A *matching* in a graph G is a set of edges without common vertices. A *maximum matching* in G is a matching of maximum cardinality, i.e., maximum number of edges. A *perfect matching* is a matching which matches all the vertices of G . In the case that G is an edge-weighted graph, a *bottleneck matching* is defined to be a perfect matching in G in which the weight of the maximum-weight edge is minimized. For a perfect matching M , we denote the *bottleneck* of M , i.e., the length of the longest edge in M , by $\lambda(M)$. For a point set P , a *Euclidean bottleneck matching* is a perfect matching which minimizes the length of the longest edge.

In this chapter we consider perfect matching and bottleneck matching admissibility of higher order Gabriel Graphs. The *order- k Gabriel graph* on P , denoted by k -GG, is the geometric graph which has an edge between two points p and q iff $D[p, q]$ contains at most k points of $P \setminus \{p, q\}$. The standard Gabriel graph, $GG(P)$, corresponds to 0-GG. It is obvious that 0-GG is plane, but k -GG may not be plane for $k \geq 1$. Su and Chang [17] showed that k -GG can be constructed in $O(k^2 n \log n)$ time and contains $O(k(n - k))$ edges. In [7], the authors proved that k -GG is $(k + 1)$ -connected.

2.1.1 Previous Work

It is well-known that a maximum matching in a graph with n vertices and m edges can be computed in $O(m\sqrt{n})$ time, e.g., by Edmonds algorithm (see [10, 15]). Any Gabriel graph is planar, and thus, has $O(n)$ edges. Therefore a maximum matching in a Gabriel graph can be computed in $O(n^{1.5})$ time. Mucha and Sankowski [16] showed that a maximum matching in a planar graph can be found in $O(n^{\omega/2})$ time, where ω is the exponent of matrix multiplication. Since $\omega < 2.38$ (see [19]) a maximum matching in a Gabriel graph can be computed in $O(n^{1.18})$ time.

For any two points p and q in P , the *lune* of p and q , denoted by $L(p, q)$, is defined as the intersection of the open disks of radius $|pq|$ centred at p and q . The *order- k Relative Neighborhood Graph* on P , denoted by k -RNG, is the geometric graph which has an edge (p, q) iff $L(p, q)$ contains at most k points of P . The *order- k Delaunay Graph* on P , denoted by k -DG, is the geometric graph which has an edge (p, q) iff there exists a circle through p and q which contains at most k points of P in its interior. It is obvious that

$$k\text{-RNG} \subseteq k\text{-GG} \subseteq k\text{-DG}.$$

The problem of determining whether a geometric graph has a (bottleneck) perfect matching is quite of interest. Dillencourt showed that the Delaunay triangulation (0-DG) admits a perfect matching [9]. Chang et al. [8] proved that a Euclidean bottleneck perfect matching of P is contained in 16-RNG.¹ This implies that 16-GG and 16-DG contain a (bottleneck) perfect matching of P . In [1] the authors showed that 15-GG is Hamiltonian. Recently, Kaiser et al. [13] improved the bound by showing that 10-GG is Hamiltonian. This implies that 10-GG has a perfect matching.

Given a geometric graph $G(P)$ on a set P of n points, we say that a set K of points *blocks* $G(P)$ if in $G(P \cup K)$ there is no edge connecting two points in P , in other words, P is an independent set in $G(P \cup K)$. Aichholzer et al. [2] considered the problem of blocking the Delaunay triangulation (i.e. 0-DG) for P in general position. They show that $\frac{3n}{2}$ points are sufficient to block $DT(P)$ and at least $n - 1$ points are necessary. To block a Gabriel graph, $n - 1$ points are sufficient, and $\frac{3}{4}n - o(n)$ points are sometimes necessary [3].

2.1.2 Our Results

In this chapter we consider the following three problems: (a) for which values of k does every k -GG have a Euclidean bottleneck matching of P ? (b) for a given value k , what is the size of a maximum matching in k -GG? (c) how many points are sufficient/necessary to block a k -GG? In Section 2.2 we review and prove some graph-theoretic notions. In Section 2.3 we consider the problem (a) and prove that a Euclidean bottleneck matching of P is contained in 9-GG. In addition, we show that for some point sets, 8-GG does not have any Euclidean bottleneck matching. Moreover, we show that for some point sets, 7-GG does not have any Euclidean Hamiltonian cycle; this improves the previous bound of 5-GG that is obtained in [13]. In Section 2.4 we consider the problem (b) and give some lower bounds on the size of a maximum matching in k -GG. We prove that 0-GG has a matching of size at least $\frac{n-1}{4}$, and this bound is tight. In addition we prove that 1-GG has a matching of size at least $\frac{2(n-1)}{5}$ and 2-GG has a perfect matching. In Section 2.5 we consider the problem (c). We show that at least $\lceil \frac{n-1}{3} \rceil$ points are necessary to block a Gabriel

¹ They defined k -RNG in such a way that $L(p, q)$ contains at most $k - 1$ points of P .

graph and this bound is tight. We also show that at least $\lceil \frac{(k+1)(n-1)}{3} \rceil$ points are necessary and $(k+1)(n-1)$ points are sufficient to block a k -GG. The open problems and concluding remarks are presented in Section 2.6.

2.2 PRELIMINARIES

Let G be an edge-weighted graph with vertex set V and weight function $w : E \rightarrow \mathbb{R}^+$. Let T be a minimum spanning tree of G , and let $w(T)$ be the total weight of T .

LEMMA 2.1. *Let $\delta(e)$ be a cycle in G that contains an edge $e \in T$. Let δ' be the set of edges in $\delta(e)$ which do not belong to T and let e'_{\max} be the largest edge in δ' . Then, $w(e) \leq w(e'_{\max})$.*

Proof. Let $e = (u, v)$ and let T_u and T_v be the two trees obtained by removing e from T . Let $e' = (x, y)$ be an edge in δ' such that one of x and y belongs to T_u and the other one belongs to T_v . By definition of e'_{\max} , we have $w(e') \leq w(e'_{\max})$. Let $T' = T_u \cup T_v \cup \{(x, y)\}$. Clearly, T' is a spanning tree of G . If $w(e') < w(e)$ then $w(T') < w(T)$; contradicting the minimality of T . Thus, $w(e) \leq w(e')$, which completes the proof of the lemma. \square

For a graph $G = (V, E)$ and $S \subseteq V$, let $G - S$ be the subgraph obtained from G by removing all vertices in S , and let $o(G - S)$ be the number of odd components in $G - S$, i.e., connected components with an odd number of vertices. The following theorem by Tutte [18] gives a characterization of the graphs which have perfect matching:

THEOREM 2.2 (Tutte [18]). *G has a perfect matching if and only if $o(G - S) \leq |S|$ for all $S \subseteq V$.*

Berge [4] extended Tutte's theorem to a formula (known as the Tutte-Berge formula) for the maximum size of a matching in a graph. In a graph G , the *deficiency*, $\text{def}_G(S)$, is $o(G - S) - |S|$. Let $\text{def}(G) = \max_{S \subseteq V} \text{def}_G(S)$.

THEOREM 2.3 (Tutte-Berge formula; Berge [4]). *The size of a maximum matching in G is*

$$\frac{1}{2}(n - \text{def}(G)).$$

For an edge-weighted graph G we define the *weight sequence* of G , $WS(G)$, as the sequence containing the weights of the edges of G in non-increasing order. A graph G_1 is said to be less than a graph G_2 if $WS(G_1)$ is lexicographically smaller than $WS(G_2)$.

2.3 EUCLIDEAN BOTTLENECK MATCHING

In Subsection 2.3.1 we prove the following theorem.

THEOREM 2.4. *For every point set P in the plane, 9-GG contains a Euclidean bottleneck matching of P .*

In Subsection 2.3.2 we prove the following proposition.

PROPOSITION 1. *There exist point sets P in the plane such that 8-GG does not contain any Euclidean bottleneck matching of P .*

In Subsection 2.3.3 we prove the following proposition.

PROPOSITION 2. *There exist point sets P in the plane such that 7-GG does not contain any Euclidean bottleneck Hamiltonian cycle of P .*

Proposition 2 improves the previous bound of 5-GG that is obtained by Kaiser *et al.* [13]

2.3.1 Proof of Theorem 2.4

In this section we prove Theorem 2.4. The proofs for Lemmas 2.5 and 2.6 are similar to the proofs in [8] which are adjusted for Gabriel graphs. The proof of Lemma 2.8 is based on a similar technique that is used in [13] for the Hamiltonicity of Gabriel graphs.

Let \mathcal{M} be the set of all perfect matchings of the complete graph with vertex set P . For a matching $M \in \mathcal{M}$ we define the *weight sequence* of M , $WS(M)$, as the sequence containing the weights of the edges of M in non-increasing order. A matching M_1 is said to be less than a matching M_2 if $WS(M_1)$ is lexicographically smaller than $WS(M_2)$. We define a total order on the elements of \mathcal{M} by their weight sequence. If two elements have exactly the same weight sequence, break ties arbitrarily to get a total order.

Let $M^* = \{(a_1, b_1), \dots, (a_{\frac{p}{2}}, b_{\frac{p}{2}})\}$ be a matching in \mathcal{M} with minimum weight sequence. Observe that M^* is a Euclidean bottleneck matching for P . In order to prove Theorem 2.4, we will show that all edges of M^* are in 9-GG. Consider any edge (a, b) in M^* . If $D[a, b]$ contains no point of $P \setminus \{a, b\}$, then (a, b) is an edge of 9-GG. Suppose that $D[a, b]$ contains k points of $P \setminus \{a, b\}$. We are going to prove that $k \leq 9$. Let $R = \{r_1, r_2, \dots, r_k\}$ be the set of points of $P \setminus \{a, b\}$ that are in $D[a, b]$. Let $S = \{s_1, s_2, \dots, s_k\}$ represent the points for which $(r_i, s_i) \in M^*$.

Without loss of generality, we assume that $D[a, b]$ has diameter 1 and is centered at the origin $o = (0, 0)$, and $a = (-0.5, 0)$ and $b = (0.5, 0)$. For any point p in the plane, let $\|p\|$ denote the distance of p from o . Note that $|ab| = 1$, and for any point $x \in D[a, b] \setminus \{a, b\}$ we have $\max\{|xa|, |xb|\} < 1$.

LEMMA 2.5. For each point $s_i \in S$, $\min\{|s_i a|, |s_i b|\} \geq 1$.

Proof. The proof is by contradiction; suppose that $|s_i a| < 1$. Let M be the perfect matching obtained from M^* by deleting $\{(a, b), (r_i, s_i)\}$ and adding $\{(s_i, a), (r_i, b)\}$. The lengths of the two new edges are smaller than 1, and hence both (s_i, a) and (r_i, b) are shorter than (a, b) . Thus, $WS(M) <_{\text{lex}} WS(M^*)$, which contradicts the minimality of M^* . \square

As a corollary of Lemma 2.5, R and S are disjoint.

LEMMA 2.6. For each pair of points $s_i, s_j \in S$, $|s_i s_j| \geq \max\{|r_i s_i|, |r_j s_j|, 1\}$.

Proof. The proof is by contradiction; suppose that $|s_i s_j| < \max\{|r_i s_i|, |r_j s_j|, 1\}$. Let M be the perfect matching obtained from M^* by deleting $\{(a, b), (r_i, s_i), (r_j, s_j)\}$ and adding $\{(a, r_i), (b, r_j), (s_i, s_j)\}$. Note that $\max\{|ar_i|, |br_j|, |s_i s_j|\} < \max\{|r_i s_i|, |r_j s_j|, |ab|\}$. Thus, we get $WS(M) <_{\text{lex}} WS(M^*)$, which contradicts the minimality of M^* . \square

Let $C(x, r)$ (resp. $D(x, r)$) be the circle (resp. closed disk) of radius r that is centered at a point x in the plane. For $i \in \{1, \dots, k\}$, let s'_i be the intersection point between $C(o, 1.5)$ and the ray with origin at o passing through s_i . Let the point p_i be s_i , if $\|s_i\| < 1.5$, and s'_i , otherwise. See Figure 2.1. Let $S' = \{a, b, p_1, \dots, p_k\}$.

OBSERVATION 2.7. Let s_j be a point in S , where $\|s_j\| \geq 1.5$. Then, the disk $D(s_j, \|s_j\| - 0.5)$ is contained in the disk $D(s_j, |s_j r_j|)$. Moreover, the disk $D(p_j, 1)$ is contained in the disk $D(s_j, \|s_j\| - 0.5)$. See Figure 2.1.

LEMMA 2.8. The distance between any pair of points in S' is at least 1.

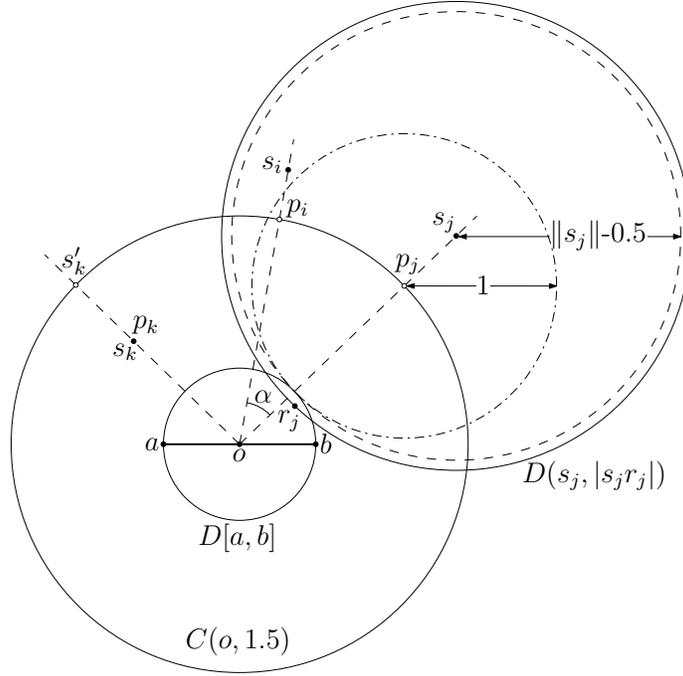


Figure 2.1: Proof of Lemma 2.8; $p_i = s'_i$, $p_j = s'_j$, and $p_k = s_k$.

Proof. Let x and y be two points in S' . We are going to prove that $|xy| \geq 1$. We distinguish between the following three cases.

- $\{x, y\} = \{a, b\}$. In this case the claim is trivial.
- $x \in \{a, b\}, y \in \{p_1, \dots, p_k\}$. If $\|y\| = 1.5$, then y is on $C(o, 1.5)$, and hence $|xy| \geq 1$. If $\|y\| < 1.5$, then y is a point in S . Therefore, by Lemma 2.5, $|xy| \geq 1$.
- $x, y \in \{p_1, \dots, p_k\}$. Without loss of generality assume $x = p_i$ and $y = p_j$, where $1 \leq i < j \leq k$. We differentiate between three cases:

Case (i): $\|p_i\| < 1.5$ and $\|p_j\| < 1.5$. In this case p_i and p_j are two points in S . Therefore, by Lemma 2.6, $|p_i p_j| \geq 1$.

Case (ii): $\|p_i\| < 1.5$ and $\|p_j\| = 1.5$. In this case p_i is a point in S . By Observation 2.7, the disk $D(p_j, 1)$ is contained in the disk $D(s_j, |s_j r_j|)$, and by Lemma 2.6, p_i is not in the interior of $D(s_j, |s_j r_j|)$. Therefore, p_i is not in the interior of $D(p_j, 1)$, which implies that $|p_i p_j| \geq 1$.

Case (iii): $\|p_i\| = 1.5$ and $\|p_j\| = 1.5$. In this case $\|s_i\| \geq 1.5$ and $\|s_j\| \geq 1.5$. Without loss of generality assume $\|s_i\| \leq \|s_j\|$. For the sake of contradiction assume that $|p_i p_j| < 1$; see Figure 2.1. Then, for the angle $\alpha = \angle s_i o s_j$ we have $\sin(\alpha/2) < \frac{1}{3}$. Then, $\cos(\alpha) = 1 - 2 \sin^2(\alpha/2) > \frac{7}{9}$. By the law of cosines in the triangle $\triangle s_i o s_j$, we have

$$|s_i s_j|^2 < \|s_i\|^2 + \|s_j\|^2 - \frac{14}{9} \|s_i\| \|s_j\|. \quad (2.1)$$

By Observation 2.7, the disk $D(s_j, \|s_j\| - 0.5)$ is contained in the disk $D(s_j, |s_j r_j|)$, and by Lemma 2.6, s_i is not in the interior of $D(s_j, |s_j r_j|)$. Therefore, s_i is not in the interior of $D(s_j, \|s_j\| - 0.5)$. Thus, $|s_i s_j| \geq \|s_j\| - 0.5$. In combination with Inequality (2.1), this implies

$$\|s_j\| \left(\frac{14}{9} \|s_i\| - 1 \right) < \|s_i\|^2 - \frac{1}{4}. \quad (2.2)$$

In combination with the assumption $\|s_i\| \leq \|s_j\|$, Inequality (2.2) implies

$$\frac{5}{9} \|s_i\|^2 - \|s_i\| + \frac{1}{4} < 0,$$

i.e.,

$$\frac{5}{9} \left(\|s_i\| - \frac{3}{10} \right) \left(\|s_i\| - \frac{3}{2} \right) < 0.$$

This is a contradiction, because, since $\|s_i\| \geq 1.5$, the left-hand side is non-negative. Thus $|p_i p_j| \geq 1$, which completes the proof of the lemma. \square

By Lemma 2.8, the points in S' have mutual distance at least 1. Moreover, the points in S' lie in $D(o, 1.5)$. Fodor [11] proved that the smallest circle which contains 12 points with mutual distances at least 1 has radius 1.5148. Therefore, S' contains at most 11 points. Since $a, b \in S'$, this implies that $k \leq 9$. Therefore, S , and consequently R , contains at most 9 points. Thus, (a, b) is an edge in 9-GG. This completes the proof of Theorem 2.4.

2.3.2 Proof of Proposition 1

In this section we prove Proposition 1. We show that for some point sets P , 8-GG does not contain any Euclidean bottleneck matching of P .

Consider Figure 2.2 which shows a configuration of a set P of 20 points. The closed disk $D[a, b]$ is centred at c and has diameter one, i.e., $|ab| = 1$. $D[a, b]$ contains 9 points $U = \{u_1, \dots, u_9\}$ which lie on a circle with radius $\frac{1}{2} - \epsilon$ which is centred at c . Nine points in $U' = \{r_1, \dots, r_9\}$ are placed on a circle with radius 1.5 which is centred at c in such a way that $|r_j u_j| = 1 + \epsilon$, $|r_j a| > 1 + \epsilon$, $|r_j b| > 1 + \epsilon$, and $|r_j r_k| > 1 + \epsilon$ for $1 \leq j, k \leq 9$ and $j \neq k$. Consider a perfect matching $M = \{(a, b)\} \cup \{(r_i, u_i) : i = 1, \dots, 9\}$ where each point $r_i \in U'$ is matched to its closest point u_i . It is obvious that $\lambda(M) = 1 + \epsilon$, and hence the bottleneck of any bottleneck perfect matching is at most $1 + \epsilon$. We will show that any Euclidean bottleneck matching of P contains (a, b) . By contradiction, let M^* be a Euclidean bottleneck matching which does not contain (a, b) . In M^* , a is matched to a point $x \in U \cup U'$. If $x \in U'$, then $|ax| > 1 + \epsilon$. If $x \in U$, w.l.o.g. assume that $x = u_1$. Thus, in M^* the point r_1 is matched to a point y where $y \neq u_1$. Since u_1 is the closest point to r_1 and $|r_1 u_1| = 1 + \epsilon$, $|r_1 y| > 1 + \epsilon$. In both cases $\lambda(M^*) > 1 + \epsilon$, which is a contradiction. Therefore, M^* contains (a, b) . Since $D[a, b]$ contains 9 points of $P \setminus \{a, b\}$, $(a, b) \notin 8\text{-GG}$. Therefore 8-GG does not contain any Euclidean bottleneck matching of P .

2.3.3 Proof of Proposition 2

In this section we prove Proposition 2. We show that for some point sets P , 7-GG does not contain any Euclidean bottleneck Hamiltonian cycle of P .

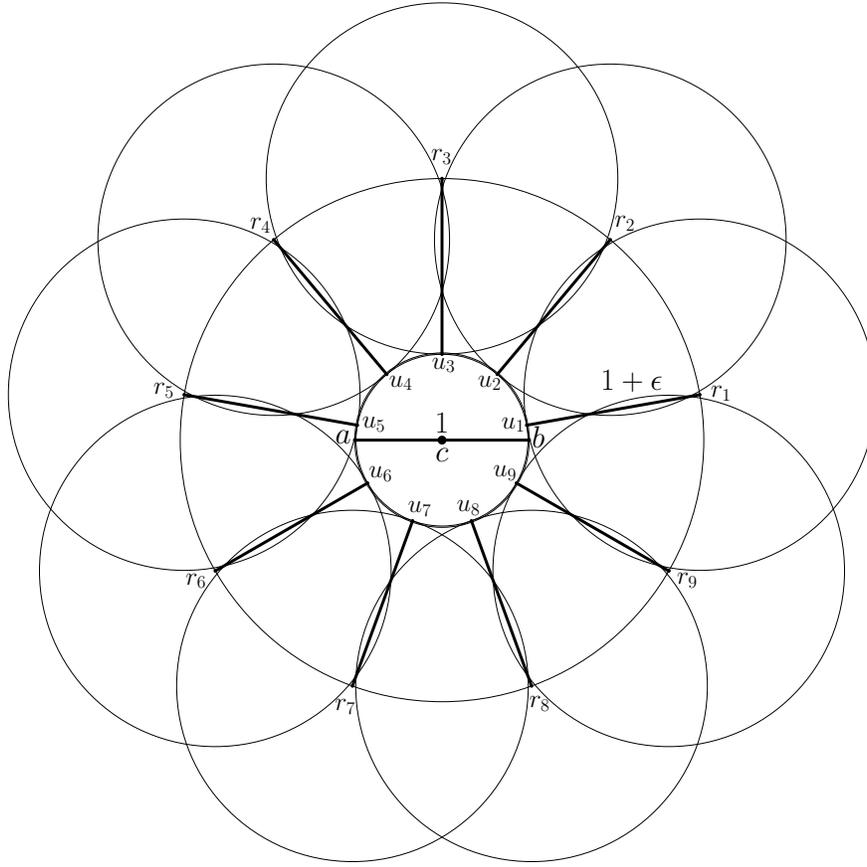


Figure 2.2: A set of 20 points such that 8-GG does not contain any Euclidean bottleneck matching.

Figure 2.3 shows a configuration of a multiset $P = \{a, b, x, r_1, \dots, r_8, s_1, \dots, s_7\}$ of 26 points, where s_5 is repeated nine times. The closed disk $D[a, b]$ is centered at o and has diameter one, i.e., $|ab| = 1$. $D[a, b]$ contains all 8 points of the set $R = \{r_1, \dots, r_8\}$; these points lie on the circle with radius $\frac{1}{2} - \epsilon$ that is centered at o ; all points of R are in the interior of $D[a, b]$. Let $S = \{s_1, \dots, s_7\}$ be the multiset of 15 points, where s_5 is repeated nine times. The red circles have radius 1 and are centered at points in S . Each point in S is connected to its first and second closest point (the black edges in Figure 2.3). Let B the chain formed by these edges. Note that r_1 and r_8 are the endpoints of B . Specifically, $|r_1 s_1| = |r_8 s_7| = 1$, and for each point r_i , where $2 \leq i \leq 7$, $|s_i a| > 1$, $|s_i b| > 1$, $|s_i x| > 1$, and $|r_i s_{i-1}| = |r_i s_i| = 1$ (here by s_5 we mean the first and last endpoints of the chain defined by points labeled s_5). Consider the Hamiltonian cycle $H = B \cup \{(r_1, a), (a, b), (b, x), (x, r_8)\}$. The longest edge in H has length 1. Therefore, the length of the longest edge in any bottleneck Hamiltonian cycle for P is at most 1. In the rest we will show—by contradiction—that any bottleneck Hamiltonian cycle of P contains (a, b) . Since in B each point of S is connected to its first and second closest point, every bottleneck Hamiltonian cycle of P contains B , because otherwise, one of the points in S should be connected to a point that is farther than its second closest point, and hence that edge is longer than 1. Now we consider possible ways to construct a bottleneck Hamiltonian cycle, say H^* , using the edges in B and the points a, b, x . Assume $(a, b) \notin H^*$. Then, in H^* , a is connected to two points in $\{r_1, r_8, x\}$. We differentiate between two cases:

- $(a, x) \in H^*$. In this case $|ax| > 1$, and hence the longest edge in H^* is longer than 1, which is a contradiction.
- $(a, x) \notin H^*$. In this case $(a, r_1) \in H^*$ and $(a, r_8) \in H^*$. This means that H^* does not contain x and b , which is a contradiction.

Therefore, we conclude that H^* , and consequently any bottleneck Hamiltonian cycle of P , contains (a, b) . Since $D[a, b]$ contains 8 points of $P \setminus \{a, b\}$, $(a, b) \notin 7\text{-GG}$. Therefore 7-GG does not contain any Euclidean bottleneck Hamiltonian cycle of P .

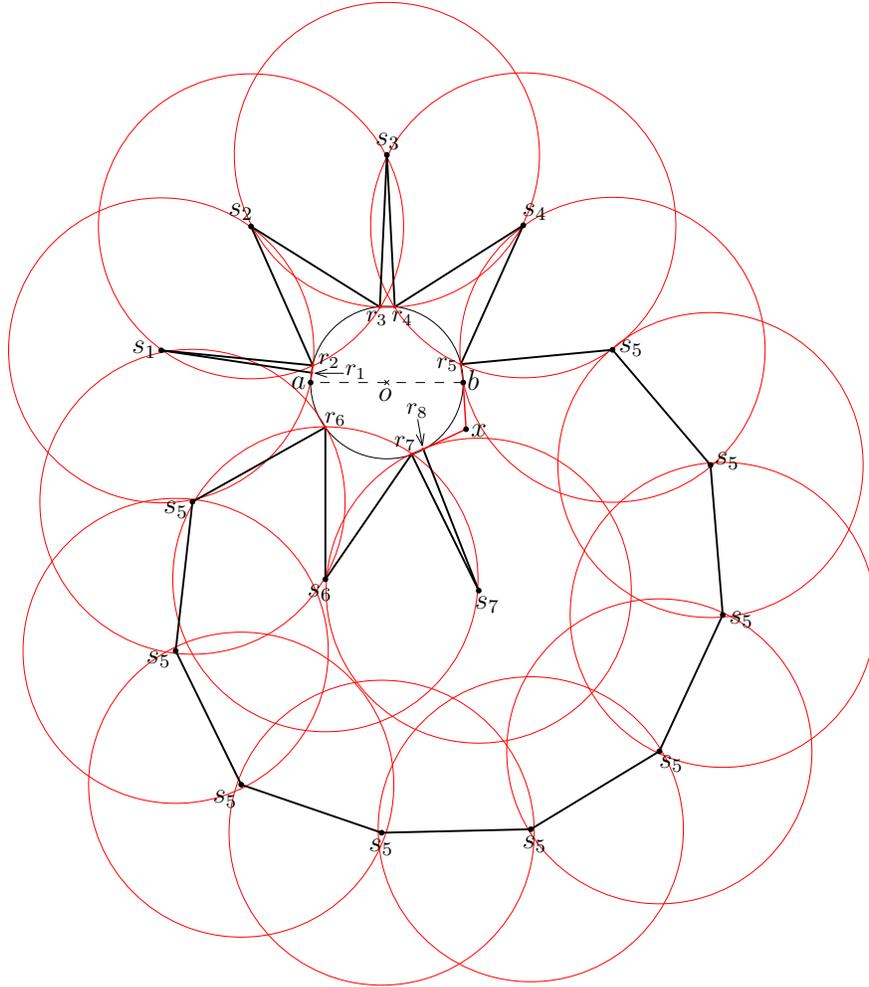


Figure 2.3: Proof of Proposition 2. The bold-black edges belong to B . $D[a, b]$ contains 8 points.

2.4 MAXIMUM MATCHING

Let P be a set of n points in the plane. In this section we will prove that 0-GG has a matching of size at least $\frac{n-1}{4}$; this bound is tight. We also prove that 1-GG has a matching of size at least $\frac{2(n-1)}{5}$ and 2-GG has a perfect matching (when n is even).

First we give a lower bound on the number of components that result after removing a set S of vertices from k -GG. Then we use Theorem 2.2 and Theorem 2.3, respectively presented by Tutte [18] and Berge [4], to prove a lower bound on the size of a maximum matching in k -GG.

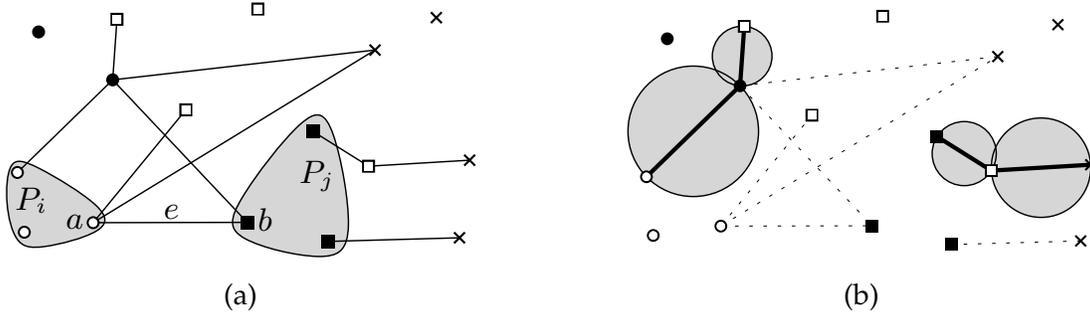


Figure 2.4: The point set P of 16 points is partitioned into white/black disks, white/black squares, and crosses. (a) The graph $G(\mathcal{P})$, (b) The set \mathcal{T} of straight-line edges corresponding to $\text{MST}(G(\mathcal{P}))$ is in bold, and the set \mathcal{D} of their corresponding disks.

Let $\mathcal{P} = \{P_1, P_2, \dots\}$ be a partition of the points in P . For two sets P_i and P_j in \mathcal{P} define the distance $d(P_i, P_j)$ as the smallest Euclidean distance between a point in P_i and a point in P_j , i.e., $d(P_i, P_j) = \min\{|ab| : a \in P_i, b \in P_j\}$. Let $G(\mathcal{P})$ be the complete edge-weighted graph with vertex set \mathcal{P} . For each edge $e = (P_i, P_j)$ in $G(\mathcal{P})$, let $w(e) = d(P_i, P_j)$. This edge e is defined by two points a and b , where $a \in P_i$ and $b \in P_j$. Therefore, an edge $e \in G(\mathcal{P})$ corresponds to a straight line edge between two points $a, b \in P$; see Figure 2.4(a). Let $\text{MST}(G(\mathcal{P}))$ be a minimum spanning tree of $G(\mathcal{P})$. It is obvious that each edge e in $\text{MST}(G(\mathcal{P}))$ corresponds to a straight line edge between $a, b \in P$. Let \mathcal{T} be the set of all these straight line edges. Let \mathcal{D} be the set of disks which have the edges of \mathcal{T} as diameter, i.e., $\mathcal{D} = \{D[a, b] : (a, b) \in \mathcal{T}\}$. See Figure 2.4(b).

OBSERVATION 2.9. \mathcal{T} is a subgraph of a minimum spanning tree of P , and hence \mathcal{T} is plane.

LEMMA 2.10. A disk $D[a, b] \in \mathcal{D}$ does not contain any point of $P \setminus \{a, b\}$.

Proof. By Observation 2.9, \mathcal{T} is a subgraph of a minimum spanning tree of P . It is well known that any minimum spanning tree of P is a subgraph of $0\text{-GG}(P)$. Thus, \mathcal{T} is a subgraph of $0\text{-GG}(P)$, and hence, any disk $D[a, b] \in \mathcal{D}$ —representing an edge of \mathcal{T} —does not contain any point of $P \setminus \{a, b\}$. \square

LEMMA 2.11. For each pair D_i and D_j of disks in \mathcal{D} , D_i (resp. D_j) does not contain the center of D_j (resp. D_i).

Proof. Let (a_i, b_i) and (a_j, b_j) respectively be the edges of \mathcal{T} which correspond to D_i and D_j . Let C_i and C_j be the circles representing the boundary of D_i and D_j . W.l.o.g. assume that C_j is the bigger circle, i.e., $|a_i b_i| < |a_j b_j|$. By contradiction, suppose that C_j contains the center c_i of C_i . Let x and y denote the intersections of C_i and C_j . Let x_i (resp. x_j) be the intersection of C_i (resp. C_j) with the line through y and c_i (resp. c_j). Similarly, let y_i (resp. y_j) be the intersection of C_i (resp. C_j) with the line through x and c_i (resp. c_j).

As illustrated in Figure 2.5, the arcs $\widehat{x_i x}$, $\widehat{y_i y}$, $\widehat{x_j x}$, and $\widehat{y_j y}$ are the potential positions for the points a_i , b_i , a_j , and b_j , respectively. First we will show that the line segment $x_i x_j$ passes through x and $|a_i a_j| \leq |x_i x_j|$. The angles $\angle x_i x y$ and $\angle x_j x y$ are right angles, thus the line segment $x_i x_j$ goes through x . Since $\widehat{x_i x} < \pi$ (resp. $\widehat{x_j x} < \pi$), for any point $a_i \in \widehat{x_i x}$, $|a_i x| \leq |x_i x|$ (resp. $a_j \in \widehat{x_j x}$, $|a_j x| \leq |x_j x|$). Therefore,

$$|a_i a_j| \leq |a_i x| + |x a_j| \leq |x_i x| + |x x_j| = |x_i x_j|.$$

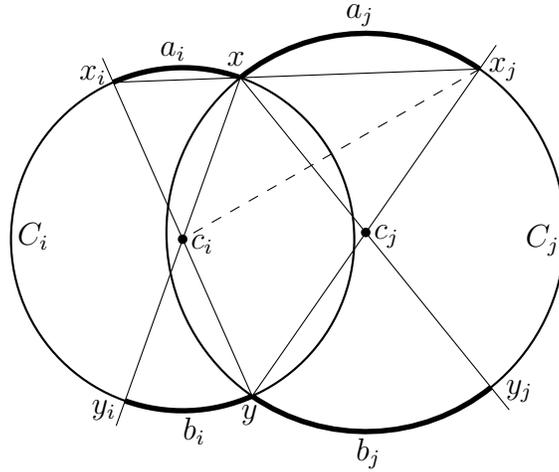


Figure 2.5: Illustration of Lemma 2.11: C_i and C_j intersect, and C_j contains the center of C_i .

Consider triangle $\triangle x_i x_j y$ which is partitioned by segment $c_i x_j$ into $t_1 = \triangle x_i x_j c_i$ and $t_2 = \triangle c_i x_j y$. Since $x_i y$ is a diameter of C_i which passes through the center c_i , the length of the segment $x_i c_i$ of t_1 is equal to the length of the segment $c_i y$ of t_2 . The segment $c_i x_j$ is shared by t_1 and t_2 . Since c_i is inside C_j and $\widehat{y x_j} = \pi$, the angle $\angle y c_i x_j > \frac{\pi}{2}$. Thus, $\angle x_i c_i x_j$ in t_1 is smaller than $\frac{\pi}{2}$ (and hence smaller than $\angle y c_i x_j$ in t_2). Therefore, the segment $x_i x_j$ of t_1 is smaller than the segment $x_j y$ of t_2 . Thus,

$$|a_i a_j| \leq |x_i x_j| < |x_j y| = |a_j b_j|.$$

By symmetry $|b_i b_j| < |a_j b_j|$. Therefore $\max\{|a_i a_j|, |b_i b_j|\} < \max\{|a_i b_i|, |a_j b_j|\}$. In addition $\delta = (a_i, a_j, b_j, b_i, a_i)$ is a cycle and at least one of (a_i, a_j) and (b_i, b_j) does not belong to \mathcal{T} . This contradicts Lemma 2.1 (Note that by Observation 2.9, \mathcal{T} is a subgraph of a minimum spanning tree of P). \square

Now we show that the intersection of every four disks in \mathcal{D} is empty. In other words, every point in the plane cannot lie in more than three disks in \mathcal{D} . In Section 2.4.1 we prove the following theorem, and in Section 2.4.2 we present the lower bounds on the size of a maximum matching in k -GG.

THEOREM 2.12. *For every four disks $D_1, D_2, D_3, D_4 \in \mathcal{D}$, $D_1 \cap D_2 \cap D_3 \cap D_4 = \emptyset$.*

2.4.1 Proof of Theorem 2.12

The proof is by contradiction. Let D_1, D_2, D_3 , and D_4 be four disks in \mathcal{D} . Let $\mathcal{X} = D_1 \cap D_2 \cap D_3 \cap D_4$ and let x be a point in \mathcal{X} . For $i = 1, 2, 3, 4$, let c_i be the center of D_i , let C_i be the boundary of D_i , and let (a_i, b_i) be the edge in \mathcal{T} which corresponds to D_i . Denote the angle $\angle a_i x b_i$ by α_i , for $i = 1, 2, 3, 4$. Since (a_i, b_i) is a diameter of D_i and x lies in D_i , $\alpha_i \geq \frac{\pi}{2}$. First we prove the following observation.

OBSERVATION 2.13. *For $i, j \in \{1, 2, 3, 4\}$, where $i \neq j$, the angles α_i and α_j are either disjoint or one is completely contained in the other.*

Proof. The proof is by contradiction. Suppose that α_i and α_j are not disjoint and none of them is completely contained in the other. Thus α_i and α_j share some part and w.l.o.g. assume that b_i is in the cone which is obtained by extending the edges of α_j , and b_j is in the cone which is obtained by extending the edges of α_i . Three cases arise:

- $b_i \in \Delta x a_j b_j$. In this case b_i is inside D_j which contradicts Lemma 2.10.
- $b_j \in \Delta x a_i b_i$. In this case b_j is inside D_i which contradicts Lemma 2.10.
- $b_i \notin \Delta x a_j b_j$ and $b_j \notin \Delta x a_i b_i$. In this case (a_i, b_i) intersects (a_j, b_j) which contradicts Observation 2.9.

□

We call α_i a *blocked angle* if α_i is contained in an angle α_j for some $j \in \{1, 2, 3, 4\}$, where $j \neq i$. Otherwise, we call α_i a *free angle*.

LEMMA 2.14. *At least one α_i , for $i \in \{1, 2, 3, 4\}$, is blocked.*

Proof. Suppose that all angles α_i , where $i \in \{1, 2, 3, 4\}$, are free. This implies that the α_i s are pairwise disjoint and $\alpha = \sum_{i=1}^4 \alpha_i \geq 2\pi$. If $\alpha > 2\pi$, we obtain a contradiction to the fact that the sum of the disjoint angles around x is at most 2π . If $\alpha = 2\pi$, then the four edges (a_i, b_i) where $i \in \{1, 2, 3, 4\}$, form a cycle which contradicts the fact that \mathcal{T} is a subgraph of a minimum spanning tree of P . □

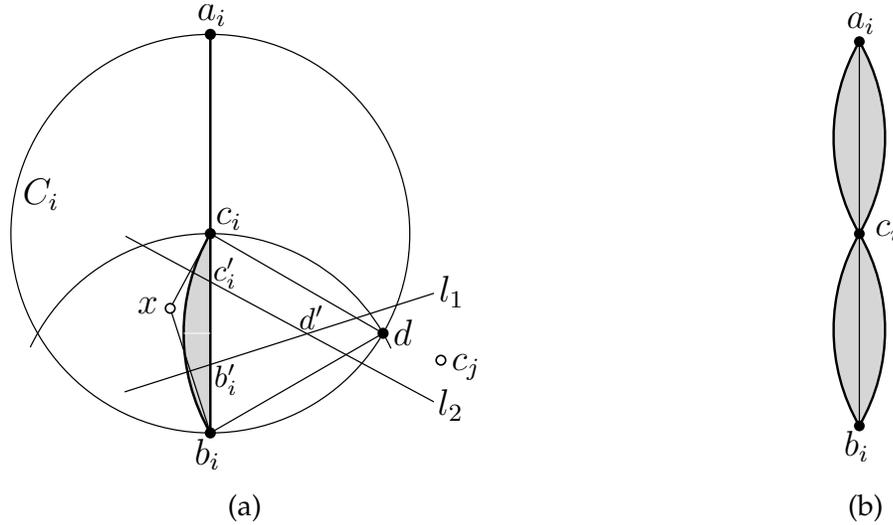


Figure 2.6: (a) The point x should be inside the circle segment $sc(c_i, b_i)$. (b) The trap (a_i, b_i) which consists of two lenses $\text{lens}(a_i, c_i)$ and $\text{lens}(b_i, c_i)$.

By Lemma 2.14 at least one of the angles is blocked. Hereafter, assume that α_j is blocked by α_i where $1 \leq i, j \leq 4$ and $i \neq j$. W.l.o.g. assume that $a_i b_i$ is a vertical line segment and the point x (which belongs to \mathcal{X}) is to the left of $a_i b_i$. Thus, $a_j b_j$ and c_j are to the right of $a_i b_i$. This implies that $a_i b_i \cap D_j \neq \emptyset$. See Figure 2.6(a). By Lemma 2.11, c_i cannot be inside D_j , thus either $a_i c_i \cap D_j \neq \emptyset$ or $c_i b_i \cap D_j \neq \emptyset$, but not both. W.l.o.g. assume that $c_i b_i \cap D_j \neq \emptyset$. Let C' be the circle with radius $|c_i b_i|$ which is centered at b_i . Let d denote the intersection of C' with C_i which is to the right of $c_i b_i$. Consider the circle C'' with radius $|db_i|$ centered at d . Note that C'' goes

through b_i and c_i . Let $sc(c_i, b_i)$ be the segment of the circle C'' which is between the chord $c_i b_i$ and the arc $\widehat{c_i b_i}$ as shown in Figure 2.6(a).

We show that x cannot be outside $sc(c_i, b_i)$. By contradiction suppose that x is outside $sc(c_i, b_i)$ (and to the left of $c_i b_i$). Let l_1 and l_2 respectively be the perpendicular bisectors of $x b_i$ and $x c_i$. Let b'_i and c'_i respectively be the intersection of l_1 and l_2 with $c_i b_i$ and let d' be the intersection point of l_1 and l_2 . Since x is outside $sc(c_i, b_i)$, the intersection point d' is to the left of (the vertical line through) d and inside triangle $\triangle b_i c_i d$. If c_j is below l_1 then $|c_j b_i| < |c_j x|$ and D_j contains b_i which contradicts Lemma 2.11. If c_j is above l_2 then $|c_j b_i| < |c_j x|$ and D_j contains c_i which contradicts Lemma 2.11. Thus, c_j is above l_1 and below l_2 , and (by the initial assumption) to the right of $c_i b_i$. That is, c_j is in triangle $\triangle b'_i c'_i d'$. Since $\triangle b'_i c'_i d' \subseteq \triangle b_i c_i d \subseteq D_i$, c_j lies inside D_i which contradicts Lemma 2.11. Therefore, x is contained in $sc(c_i, b_i)$.

By symmetry D_j can intersect $a_i c_i$ and/or c_j can be to the left of $a_i b_i$ as well. Therefore, if α_i blocks α_j , the point x can be in $sc(c_i, b_i)$ or any of the symmetric segments of the circles. For an edge $a_i b_i$ we denote the union of these segments by $trap(a_i, b_i)$ which is shown in Figure 2.6(b). For each disk D_i , let $trap(D_i) = trap(a_i, b_i)$ where (a_i, b_i) is the edge in \mathcal{T} corresponding to D_i . Therefore x is contained in $trap(D_i)$ which implies that

$$\mathcal{X} \subseteq trap(D_i).$$

Note that $trap(D_i)$ consists of two symmetric lenses $lens(a_i, c_i)$ and $lens(b_i, c_i)$, i.e., $trap(D_i) = lens(a_i, c_i) \cup lens(b_i, c_i)$.

LEMMA 2.15. For any point $x \in trap(a_i, b_i)$, $\angle a_i x b_i \geq 150^\circ$.

Proof. See Figure 2.6(a). The angle $\angle b_i d c_i = 60^\circ$, which implies that $\widehat{c_i b_i} = 60^\circ$. Thus, for any point x' on the arc $\widehat{c_i b_i}$, $\angle x' c_i b_i + \angle x' b_i c_i = 30^\circ$, and hence for any point x in the segment $sc(c_i, b_i)$, $\angle x c_i b_i + \angle x b_i c_i \leq 30^\circ$. This implies that in $\triangle x b_i c_i$, $\angle b_i x c_i \geq 150^\circ$. On the other hand $\angle b_i x c_i \leq \angle b_i x a_i$, which proves the lemma. \square

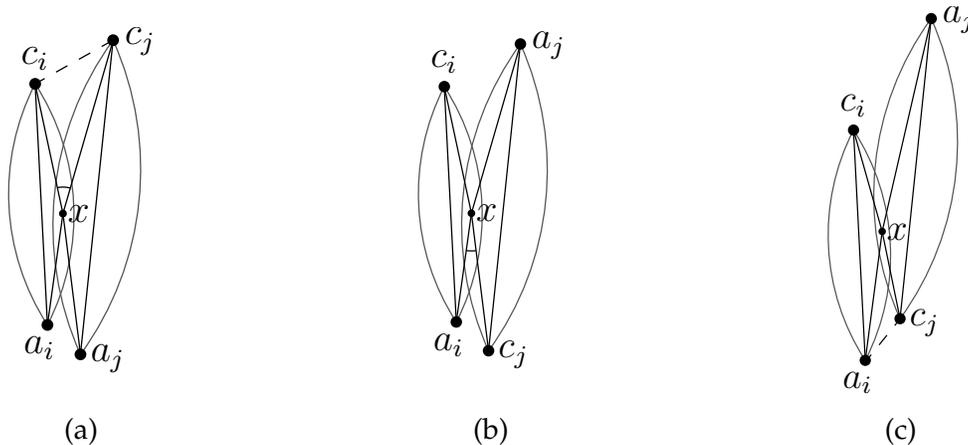


Figure 2.7: Illustration of Lemma 2.16.

LEMMA 2.16. For any two disks D_i and D_j in \mathcal{D} , $trap(D_i) \cap trap(D_j) = \emptyset$.

Proof. We prove this lemma by contradiction. Suppose $x \in trap(D_i) \cap trap(D_j)$ and w.l.o.g. assume that $x \in lens(a_i, c_i) \cap lens(a_j, c_j)$ as shown in Figure 2.7. Connect x to $a_i, c_i, a_j,$ and c_j (a_i may be identified with a_j). As shown in the proof of Lemma 2.15, $\min\{\angle a_i x c_i, \angle a_j x c_j\} > 150^\circ$. Two configurations may arise:

- $\angle c_i x c_j \leq 60^\circ$. In this case $|c_i c_j| \leq \max\{|x c_i|, |x c_j|\}$. W.l.o.g. assume that $|x c_i| \leq |x c_j|$ which implies that $|c_i c_j| \leq |x c_j|$; see Figure 2.7(a). Clearly $|x c_j| < |c_j a_j|$, and hence $|c_i c_j| < |c_j a_j|$. Thus, D_j contains c_i which contradicts Lemma 2.11.
- $\angle c_i x c_j > 60^\circ$. In this case $\angle a_i x c_j \leq 60^\circ$ and $\angle a_j x c_i \leq 60^\circ$, hence $|a_i c_j| \leq \max\{|a_i x|, |c_j x|\}$ and $|a_j c_i| \leq \max\{|a_j x|, |c_i x|\}$. Three configurations arise:
 - $|a_i x| < |c_j x|$, in this case $|a_i c_j| < |c_j x| < |c_j a_j|$ and hence D_j contains a_i . See Figure 2.7(b).
 - $|a_j x| < |c_i x|$, in this case $|a_j c_i| < |c_i x| < |c_i a_i|$ and hence D_i contains a_j .
 - $|a_i x| \geq |c_j x|$ and $|a_j x| \geq |c_i x|$, in this case w.l.o.g. assume that $|a_i x| \leq |a_j x|$. Thus $|a_i c_j| \leq |a_i x| \leq |a_j x| < |a_j c_j|$ which implies that D_j contains a_i . See Figure 2.7(b).

All cases contradict Lemma 2.10. □

Recall that each blocking angle represents a trap. Thus, by Lemma 2.14 and Lemma 2.16, we have the following corollary:

COROLLARY 2.17. *Exactly one α_i , where $1 \leq i \leq 4$, is blocked.*

Recall that α_j is blocked by α_i , $a_i b_i$ is vertical line segment, c_j is to the right of $a_i b_i$, and $x \in \text{sc}(c_i, b_i)$. As a direct consequence of Corollary 2.17, α_i , α_k , and α_l are free angles, where $1 \leq i, j, k, l \leq 4$ and $i \neq j \neq k \neq l$. In addition, c_k and c_l are to the left of $a_i b_i$. It is obvious that

$$x \subseteq \text{trap}(D_i) \cap D_k \cap D_l.$$

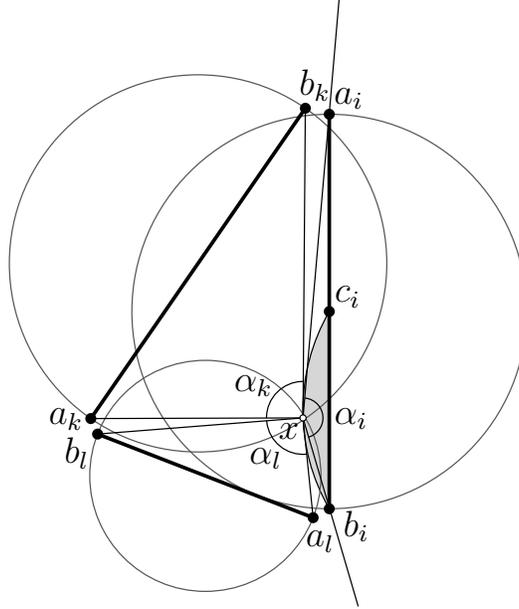


Figure 2.8: Illustration of Lemma 2.18.

LEMMA 2.18. *For a blocking angle α_i and free angles α_k and α_l , $\text{trap}(D_i) \cap D_k \cap D_l = \emptyset$.*

Proof. Since α_i is a blocking angle and α_k, α_l are free angles, c_k and c_l are on the same side of $a_i b_i$. By contradiction, suppose that $x \in \text{trap}(D_i) \cap D_j \cap D_k$. See Figure 2.8. It is obvious that $\max\{|xa_i|, |xb_i|\} < |a_i b_i|$, $\max\{|xa_k|, |xb_k|\} < |a_k b_k|$, and $\max\{|xa_l|, |xb_l|\} < |a_l b_l|$. By Lemma 2.15, $\alpha_i \geq 150^\circ$. In addition $\alpha_k, \alpha_l \geq 90^\circ$. Thus, $\max\{\angle a_i x b_k, \angle a_k x b_l, \angle a_l x b_i\} \leq 30^\circ$. Hence, $|a_i b_k| < \max\{|xa_i|, |xb_k|\}$, $|a_k b_l| < \max\{|xa_k|, |xb_l|\}$, and $|a_l b_i| < \max\{|xa_l|, |xb_i|\}$. Therefore, $\max\{|a_i b_k|, |a_k b_l|, |a_l b_i|\} < \max\{|a_i b_i|, |a_k b_k|, |a_l b_l|\}$. In addition $\delta = (a_i, b_i, a_l, b_l, a_k, b_k, a_i)$ is a cycle and at least one of (a_i, b_k) , (a_k, b_l) and (a_l, b_i) does not belong to \mathcal{T} . This contradicts Lemma 2.1. \square

Thus, $\mathcal{X} = \emptyset$; which contradicts the fact that $x \in \mathcal{X}$. This completes the proof of Theorem 2.12.

2.4.2 Lower Bounds

In this section we present some lower bounds on the size of a maximum matching in 2-GG, 1-GG, and 0-GG.

THEOREM 2.19. *For a set P of an even number of points, 2-GG has a perfect matching.*

Proof. First we show that by removing a set S of points from 2-GG, at most $|S| + 1$ components are generated. Then we show that at least one of these components must be even. Using Theorem 2.2, we conclude that 2-GG has a perfect matching.

Let S be a set vertices removed from 2-GG, and let $\mathcal{C} = \{C_1, \dots, C_{m(S)}\}$ be the resulting $m(S)$ components. Then, $\mathcal{P} = \{V(C_1), \dots, V(C_{m(S)})\}$ is a partition of the vertices in $P \setminus S$.

CLAIM 1. $m(S) \leq |S| + 1$.

Let $G(\mathcal{P})$ be the complete graph with vertex set \mathcal{P} which is constructed as described above. Let \mathcal{T} be the set of all edges between points in P corresponding to the edges of $\text{MST}(G(\mathcal{P}))$ and let \mathcal{D} be the set of disks corresponding to the edges of \mathcal{T} . It is obvious that \mathcal{T} contains $m(S) - 1$ edges and hence $|\mathcal{D}| = m(S) - 1$. Let $F = \{(p, D) : p \in S, D \in \mathcal{D}, p \in D\}$ be the set of all (point, disk) pairs where $p \in S$, $D \in \mathcal{D}$, and p is inside D . By Theorem 2.12 each point in S can be inside at most three disks in \mathcal{D} . Thus, $|F| \leq 3 \cdot |S|$. Now we show that each disk in \mathcal{D} contains at least three points of S in its interior. Consider any disk $D \in \mathcal{D}$ and let $e = (a, b)$ be the edge of \mathcal{T} corresponding to D . By Lemma 2.10, D does not contain any point of $P \setminus S$. Therefore, D contains at least three points of S , because otherwise (a, b) is an edge in 2-GG which contradicts the fact that a and b belong to different components in \mathcal{C} . Thus, each disk in \mathcal{D} has at least three points of S . That is, $3 \cdot |\mathcal{D}| \leq |F|$. Therefore, $3(m(S) - 1) \leq |F| \leq 3|S|$, and hence $m(S) \leq |S| + 1$.

CLAIM 2. $o(\mathcal{C}) \leq |S|$.

By Claim 1, $|\mathcal{C}| = m(S) \leq |S| + 1$. If $|\mathcal{C}| \leq |S|$, then $o(\mathcal{C}) \leq |S|$. Assume that $|\mathcal{C}| = |S| + 1$. Since $P = S \cup \{\bigcup_{i=1}^{|S|+1} V(C_i)\}$, the total number of vertices of P is equal to $n = |S| + \sum_{i=1}^{|S|+1} |V(C_i)|$. Consider two cases where (i) $|S|$ is odd, (ii) $|S|$ is even. In both cases if all the components in \mathcal{C} are odd, then n is odd; contradicting our assumption that P has an even number of vertices. Thus, \mathcal{C} contains at least one even component, which implies that $o(\mathcal{C}) \leq |S|$.

Finally, by Claim 2 and Theorem 2.2, we conclude that 2-GG has a perfect matching. \square

THEOREM 2.20. *For every set P of n points, 1-GG has a matching of size at least $\frac{2(n-1)}{5}$.*

Proof. Let S be a set of vertices removed from 1-GG, and let $\mathcal{C} = \{C_1, \dots, C_{m(S)}\}$ be the resulting $m(S)$ components. Then, $\mathcal{P} = \{V(C_1), \dots, V(C_{m(S)})\}$ is a partition of the vertices in $P \setminus S$. Note that $o(\mathcal{C}) \leq m(S)$. Let M^* be a maximum matching in 1-GG. By Theorem 2.3,

$$|M^*| = \frac{1}{2}(n - \text{def}(1\text{-GG})), \quad (2.3)$$

where

$$\text{def}(1\text{-GG}) = \max_{S \subseteq P} (o(\mathcal{C}) - |S|) \leq \max_{S \subseteq P} (|\mathcal{C}| - |S|) = \max_{0 \leq |S| \leq n} (m(S) - |S|). \quad (2.4)$$

Define $G(\mathcal{P})$, \mathcal{J} , \mathcal{D} , and F as in the proof of Theorem 2.19. By Theorem 2.12, $|F| \leq 3 \cdot |S|$. By the same reasoning as in the proof of Theorem 2.19, each disk in \mathcal{D} has at least two points of S in its interior. Thus, $2|\mathcal{D}| \leq |F|$. Therefore, $2(m(S) - 1) \leq |F| \leq 3|S|$, and hence

$$m(S) \leq \frac{3|S|}{2} + 1. \quad (2.5)$$

In addition, $|S| + m(S) = |S| + |\mathcal{C}| \leq |P| = n$, and hence

$$m(S) \leq n - |S|. \quad (2.6)$$

By Inequalities (2.5) and (2.6),

$$m(S) \leq \min \left\{ \frac{3|S|}{2} + 1, n - |S| \right\}. \quad (2.7)$$

Thus, by (2.4) and (2.7)

$$\begin{aligned} \text{def}(1\text{-GG}) &\leq \max_{0 \leq |S| \leq n} (m(S) - |S|) \leq \max_{0 \leq |S| \leq n} \left\{ \min \left\{ \frac{3|S|}{2} + 1, n - |S| \right\} - |S| \right\} \\ &= \max_{0 \leq |S| \leq n} \left\{ \min \left\{ \frac{|S|}{2} + 1, n - 2|S| \right\} \right\} = \frac{n+4}{5}, \end{aligned} \quad (2.8)$$

where the last equation is achieved by setting $\frac{|S|}{2} + 1$ equal to $n - 2|S|$, which implies $|S| = \frac{2(n-1)}{5}$. Finally by substituting (2.8) in Equation (2.3) we have

$$|M^*| \geq \frac{2(n-1)}{5}. \quad \square$$

By similar reasoning as in the proof of Theorem 2.20 we have the following Theorem.

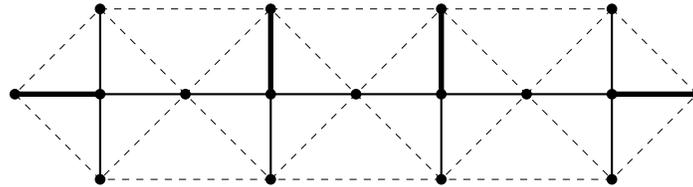


Figure 2.9: A 0-GG of $n = 17$ points with a maximum matching of size $\frac{n-1}{4} = 4$ (bold edges). The dashed edges do not belong to the graph because any of their corresponding closed disks has a point on its boundary.

THEOREM 2.21. *For every set P of n points, 0-GG has a matching of size at least $\frac{n-1}{4}$.*

The bound in Theorem 2.21 is tight, as can be seen from the graph in Figure 2.9, for which the maximum matching has size $\frac{n-1}{4}$. Actually this is a Gabriel graph of maximum degree four which is a tree. The dashed edges do not belong to 0-GG because any closed disk which has one of these edges as diameter has a point on its boundary. Observe that each edge in any matching is adjacent to one of the vertices of degree four.

NOTE: For a point set P , let $\nu_k(P)$ and $\alpha_k(P)$ respectively denote the size of a maximum matching and a maximum independent set in k -GG. For every edge in the maximum matching, at most one of its endpoints can be in the maximum independent set. Thus,

$$\alpha_k(P) \leq |P| - \nu_k(P).$$

By combining this formula with the results of Theorems 2.21, 2.20, 2.19, respectively, we have $\alpha_0(P) \leq \frac{3n+1}{4}$, $\alpha_1(P) \leq \frac{3n+2}{5}$, and $\alpha_2(P) \leq \lceil \frac{n}{2} \rceil$. The 0-GG graph in Figure 2.9 has an independent set of size $\frac{3n+1}{4} = 13$, which shows that this bound is tight for 0-GG. On the other hand, 0-GG is planar and every planar graph is 4-colorable; which implies that $\alpha_0(P) \geq \lceil \frac{n}{4} \rceil$. There are some examples of 0-GG in [14] such that $\alpha_0(P) = \lceil \frac{n}{4} \rceil$, which means that this bound is tight as well.

2.5 BLOCKING HIGHER-ORDER GABRIEL GRAPHS

In this section we consider the problem of blocking higher-order Gabriel graphs. Recall that a point set K blocks k -GG(P) if in k -GG($P \cup K$) there is no edge connecting two points in P .

THEOREM 2.22. *For every set P of n points, at least $\lceil \frac{n-1}{3} \rceil$ points are necessary to block 0-GG(P).*

Proof. Let K be a set of m points which blocks 0-GG(P). Let $G(\mathcal{P})$ be the complete graph with vertex set $\mathcal{P} = P$. Let \mathcal{T} be a minimum spanning tree of $G(\mathcal{P})$ and let \mathcal{D} be the set of closed disks corresponding to the edges of \mathcal{T} . Since $G(\mathcal{P})$ has n vertices, \mathcal{T} has $n - 1$ edges. Thus, $|\mathcal{D}| = n - 1$. By Lemma 2.10 each disk $D[a, b] \in \mathcal{D}$ does not contain any point of $P \setminus \{a, b\}$, thus, $\mathcal{T} \subseteq 0\text{-GG}(P)$. To block each edge of \mathcal{T} , corresponding to a disk in \mathcal{D} , at least one point is necessary. By Theorem 2.12 each point in K can lie in at most three disks of \mathcal{D} . Therefore, $m \geq \lceil \frac{n-1}{3} \rceil$, which implies that at least $\lceil \frac{n-1}{3} \rceil$ points are necessary to block all the edges of \mathcal{T} and hence 0-GG(P). \square

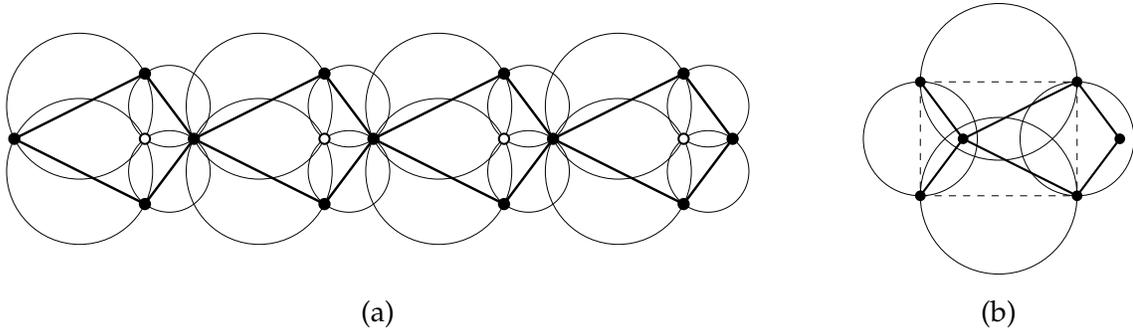


Figure 2.10: (a) 0-GG graph of $n = 13$ points (in bold edges) which is blocked by $\lceil \frac{n-1}{3} \rceil = 4$ white points, (b) dashed edges do not belong to 0-GG.

Figure 2.10(a) shows a 0-GG with $n = 13$ (black) points which is blocked by $\lceil \frac{n-1}{3} \rceil = 4$ (white) points. Note that all the disks, corresponding to the edges of every cycle, intersect at the same point in the plane (where we have placed the white points). As shown in Figure 2.10(b), the dashed edges do not belong to 0-GG. Thus, the lower bound provided by Theorem 2.22 is tight. It is easy to generalize the result of Theorem 2.22 to higher-order Gabriel graphs. Since in a k -GG we need at least $k + 1$ points to block an edge of \mathcal{T} and each point can be inside at most three disks in \mathcal{D} , we have the following corollary:

COROLLARY 2.23. *For every set P of n points, at least $\lceil \frac{(k+1)(n-1)}{3} \rceil$ points are necessary to block k -GG(P).*

In [3] the authors showed that every Gabriel graph can be blocked by a set K of $n - 1$ points by putting a point slightly to the right of each point of P , except for the rightmost one. Every disk with diameter determined by two points of P will contain a point of K . Using a similar argument one can block a k -GG by putting $k + 1$ points slightly to the right of each point of P , except for the rightmost one. Thus,

COROLLARY 2.24. *For every set P of n points, there exists a set of $(k + 1)(n - 1)$ points that blocks k -GG(P).*

Note that this upper bound is tight, because if the points of P are on a line, the disks representing the minimum spanning tree are disjoint and each disk needs $k + 1$ points to block the corresponding edge.

2.6 CONCLUSIONS

In this chapter, we considered the bottleneck and perfect matching admissibility of higher-order Gabriel graphs. We proved that

- 9-GG contains a Euclidean bottleneck matching of P and 8-GG may not have any.
- 0-GG has a matching of size at least $\frac{n-1}{4}$ and this bound is tight.
- 1-GG has a matching of size at least $\frac{2(n-1)}{5}$.
- 2-GG has a perfect matching.
- At least $\lceil \frac{n-1}{3} \rceil$ points are necessary to block 0-GG and this bound is tight.
- $\lceil \frac{(k+1)(n-1)}{3} \rceil$ points are necessary and $(k + 1)(n - 1)$ points are sufficient to block k -GG.

BIBLIOGRAPHY

- [1] M. Abellanas, P. Bose, J. García-López, F. Hurtado, C. M. Nicolás, and P. Ramos. On structural and graph theoretic properties of higher order Delaunay graphs. *International Journal of Computational Geometry & Applications*, 19(6):595–615, 2009.
- [2] O. Aichholzer, R. F. Monroy, T. Hackl, M. J. van Kreveld, A. Pilz, P. Ramos, and B. Vogtenhuber. Blocking Delaunay triangulations. *Computational Geometry: Theory and Applications*, 46(2):154–159, 2013.
- [3] B. Aronov, M. Dulieu, and F. Hurtado. Witness Gabriel graphs. *Computational Geometry: Theory and Applications*, 46(7):894–908, 2013.
- [4] C. Berge. Sur le couplage maximum d'un graphe. *C. R. Acad. Sci. Paris*, 247:258–259, 1958.
- [5] A. Biniáz, A. Maheshwari, and M. Smid. Bottleneck matchings and Hamiltonian cycles in higher-order Gabriel graphs. In *Proceedings of the 32nd European Workshop on Computational Geometry (EuroCG'16)*, pages 179–182, 2016.

- [6] A. Biniarz, A. Maheshwari, and M. H. M. Smid. Matchings in higher-order Gabriel graphs. *Theoretical Computer Science*, 596:67–78, 2015.
- [7] P. Bose, S. Collette, F. Hurtado, M. Korman, S. Langerman, V. Sacristan, and M. Saumell. Some properties of k -Delaunay and k -Gabriel graphs. *Computational Geometry: Theory and Applications*, 46(2):131–139, 2013.
- [8] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. Solving the Euclidean bottleneck matching problem by k -relative neighborhood graphs. *Algorithmica*, 8(3):177–194, 1992.
- [9] M. B. Dillencourt. Toughness and Delaunay triangulations. *Discrete & Computational Geometry*, 5:575–601, 1990.
- [10] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [11] F. Fodor. The densest packing of 12 congruent circles in a circle. *Beiträge Algebra Geom*, 41:401–409, 2000.
- [12] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278, 1969.
- [13] T. Kaiser, M. Saumell, and N. V. Cleemput. 10-Gabriel graphs are hamiltonian. *Information Processing Letters*, 115(11):877–881, 2015.
- [14] D. W. Matula and R. R. Sokal. Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis*, 12(3):205–222, 1980.
- [15] S. Micali and V. V. Vazirani. An $O(\sqrt{|V||E|})$ algorithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science, FOCS*, pages 17–27, 1980.
- [16] M. Mucha and P. Sankowski. Maximum matchings in planar graphs via gaussian elimination. *Algorithmica*, 45(1):3–20, 2006.
- [17] T.-H. Su and R.-C. Chang. The k -Gabriel graphs and their applications. In *SIGAL International Symposium on Algorithms*, pages 66–75, 1990.
- [18] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 22(2):107–111, 1947.
- [19] V. V. Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 887–898, 2012.

We consider an extension of the triangular-distance Delaunay graphs (TD-Delaunay) on a set P of points in general position in the plane. In TD-Delaunay, the convex distance is defined by a fixed-oriented equilateral triangle ∇ , and there is an edge between two points in P if and only if there is an empty homothet of ∇ having the two points on its boundary. We consider higher-order triangular-distance Delaunay graphs, namely k -TD, which contains an edge between two points if the interior of the smallest homothet of ∇ having the two points on its boundary contains at most k points of P . We consider the connectivity, Hamiltonicity and perfect-matching admissibility of k -TD. Finally we consider the problem of blocking the edges of k -TD.

This chapter was first published in the proceedings of the First International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2015) [7], and was subsequently published in the journal of Computational Geometry: Theory and Applications [8].

We also obtained lower and upper bounds on the size of maximum matching in 0-TD graphs, which are not included in this thesis. The results were first published in the proceedings of the 7th International Workshop on Algorithms and Computation (WALCOM 2013) [4], and was subsequently invited and accepted to a special issue of Theoretical Computer Science [5].

3.1 INTRODUCTION

The *triangular-distance Delaunay graph* of a point set P in the plane, TD-Delaunay for short, was introduced by Chew [15]. A TD-Delaunay is a graph whose convex distance function is defined by a fixed-oriented equilateral triangle. Let ∇ be a downward equilateral triangle whose barycenter is the origin and one of whose vertices is on the negative y -axis. A *homothet* of ∇ is obtained by scaling ∇ with respect to the origin by some factor $\mu \geq 0$, followed by a translation to a point b in the plane: $b + \mu\nabla = \{b + \mu a : a \in \nabla\}$. In the TD-Delaunay graph of P , there is a straight-line edge between two points p and q if and only if there exists a homothet of ∇ having p and q on its boundary and whose interior does not contain any point of P . In other words, (p, q) is an edge of TD-Delaunay graph if and only if there exists an empty downward equilateral triangle having p and q on its boundary. In this case, we say that the edge (p, q) has the *empty triangle property*.

We say that P is in general position if the line passing through any two points from P does not make angles 0° , 60° , and 120° with horizontal. In this chapter we consider point sets in general position and our results assume this pre-condition. If P is in general position, then the TD-Delaunay graph is a planar graph, see [10]. We define $t(p, q)$ as the smallest homothet of ∇ having p and q on its boundary. See Figure 3.1(a). Note that $t(p, q)$ has one of p and q at a vertex, and the other one on the opposite side. Thus,

OBSERVATION 3.1. *Each side of $t(p, q)$ contains either p or q .*

A graph G is *connected* if there is a path between any pair of vertices in G . Moreover, G is *k -connected* if there does not exist a set of at most $k - 1$ vertices whose removal disconnects G . In case $k = 2$, G is called *biconnected*. In other words a graph G is biconnected iff there is a simple

cycle between any pair of its vertices. A *matching* in G is a set of edges in G without common vertices. A *perfect matching* is a matching which matches all the vertices of G . A *Hamiltonian cycle* in G is a cycle (i.e., closed loop) through G that visits each vertex of G exactly once. For $H \subseteq G$ we denote the *bottleneck* of H , i.e., the length of the longest edge in H , by $\lambda(H)$.

Let $K_n(P)$ be a complete edge-weighted geometric graph on a point set P which contains a straight-line edge between any pair of points in P . For an edge (p, q) in $K_n(P)$ let $w(p, q)$ denote the weight of (p, q) . A *bottleneck matching* (resp. *bottleneck Hamiltonian cycle*) in P is defined to be a perfect matching (resp. Hamiltonian cycle) in $K_n(P)$, in which the weight of the maximum-weight edge is minimized. A *bottleneck biconnected spanning subgraph* of P is a spanning subgraph, $G(P)$, of $K_n(P)$ which is biconnected and the weight of the longest edge in $G(P)$ is minimized.

A tight lower bound on the size of a maximum matching in a TD-Delaunay graph, i.e. 0-TD, is presented in [5]. In this chapter we study higher-order TD-Delaunay graphs. The *order- k TD-Delaunay graph* of a point set P , denoted by k -TD, is a geometric graph which has an edge (p, q) iff the interior of $t(p, q)$ contains at most k points of P ; see Figure 3.1(b). The standard TD-Delaunay graph corresponds to 0-TD. We consider graph-theoretic properties of higher-order TD-Delaunay graphs, such as connectivity, Hamiltonicity, and perfect-matching admissibility. We also consider the problem of blocking TD-Delaunay graphs.

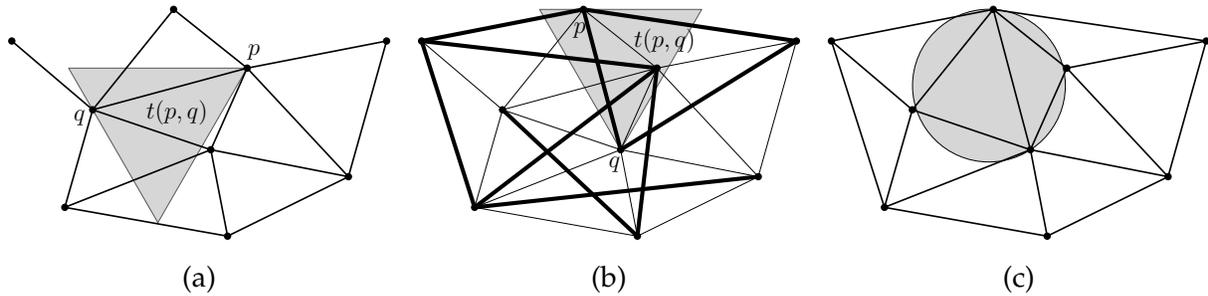


Figure 3.1: (a) Triangular-distance Delaunay graph (0-TD), (b) 1-TD graph, the light edges belong to 0-TD as well, and (c) Delaunay triangulation.

3.1.1 Previous Work

A *Delaunay triangulation* (DT) of P (which does not have any four co-circular points) is a graph whose distance function is defined by a fixed circle \odot centered at the origin. DT has an edge between two points p and q iff there exists a homothet of \odot having p and q on its boundary and whose interior does not contain any point of P ; see Figure 3.1(c). In this case the edge (p, q) is said to have the *empty circle property*. The *order- k Delaunay Graph* on P , denoted by k -DG, is defined to have an edge (p, q) iff there exists a homothet of \odot having p and q on its boundary and whose interior contains at most k points of P . The standard Delaunay triangulation corresponds to 0-DG.

For each pair of points $p, q \in P$ let $D[p, q]$ be the closed disk having pq as diameter. A *Gabriel Graph* on P is a geometric graph which has an edge between two points p and q iff $D[p, q]$ does not contain any point of $P \setminus \{p, q\}$. The *order- k Gabriel Graph* on P , denoted by k -GG, is defined to have an edge (p, q) iff $D[p, q]$ contains at most k points of $P \setminus \{p, q\}$.

For each pair of points $p, q \in P$, let $L(p, q)$ be the intersection of the two open disks with radius $|pq|$ centered at p and q , where $|pq|$ is the Euclidean distance between p and q . A *Relative*

Neighborhood Graph on P is a geometric graph which has an edge between two points p and q iff $L(p, q)$ does not contain any point of P . The *order- k Relative Neighborhood Graph* on P , denoted by k -RNG, is defined to have an edge (p, q) iff $L(p, q)$ contains at most k points of P . It is obvious that for a fixed point set, k -RNG is a subgraph of k -GG, and k -GG is a subgraph of k -DG.

The problem of determining whether an order- k geometric graph always has a (bottleneck) perfect matching or a (bottleneck) Hamiltonian cycle is of interest. In order to show the importance of this problem we provide the following example. Gabow and Tarjan [18] showed that a bottleneck matching of maximum cardinality in a graph can be computed in $O(m \cdot (n \log n)^{0.5})$ time, where m is the number of edges in the graph. Using their algorithm, a bottleneck perfect matching of a point set can be computed in $O(n^2 \cdot (n \log n)^{0.5})$ time; note that the complete graph on n points has $\Theta(n^2)$ edges. Chang et al. [14] showed that a bottleneck perfect matching of a point set is contained in 16-DG; this graph has $\Theta(n)$ edges and can be computed in $O(n \log n)$ time. Thus, by running the algorithm of Gabow and Tarjan on 16-DG, a bottleneck perfect matching of a point set can be computed in $O(n \cdot (n \log n)^{0.5})$ time.

If for each edge (p, q) in $K_n(P)$, $w(p, q)$ is equal the Euclidean distance between p and q , then Chang et al. [12, 13, 14] proved that a bottleneck biconnected spanning graph, bottleneck perfect matching, and bottleneck Hamiltonian cycle of P are contained in 1-RNG, 16-RNG, 19-RNG, respectively. This implies that 16-RNG has a perfect matching and 19-RNG is Hamiltonian. Since k -RNG is a subgraph of k -GG, the same results hold for 16-GG and 19-GG. It is known that k -GG is $(k + 1)$ -connected [11] and 10-GG (and hence 10-DG) is Hamiltonian [19]. Dillencourt showed that every Delaunay triangulation (0-DG) admits a perfect matching [17] but it can fail to be Hamiltonian [16].

Given a geometric graph $G(P)$ on a set P of n points, we say that a set K of points *blocks* $G(P)$ if in $G(P \cup K)$ there is no edge connecting two points in P . Actually P is an independent set in $G(P \cup K)$. Aichholzer et al. [2] considered the problem of blocking the Delaunay triangulation (i.e. 0-DG) for a given point set P in which no four points are co-circular. They show that $\frac{3n}{2}$ points are sufficient to block 0-DG and $n - 1$ points are necessary. To block a Gabriel graph, $n - 1$ points are sufficient [3].

3.1.2 Our Results

We consider some graph-theoretical properties of higher-order triangular distance Delaunay graphs on a given set P of n points in general position in the plane. We show for which values of k , k -TD contains a bottleneck biconnected spanning graph, a bottleneck Hamiltonian cycle, and a (bottleneck) perfect-matching; for the bottleneck structures we assume that the weight of any edge (p, q) in $K_n(P)$ is equal to the area of the smallest homothet of ∇ having p and q on its boundary. In Section 3.3 we prove that every k -TD graph is $(k + 1)$ -connected. In addition we show that a bottleneck biconnected spanning graph of P is contained in 1-TD. Using a similar approach as in [1, 12], in Section 3.4 we show that a bottleneck Hamiltonian cycle of P is contained in 7-TD. We also show a configuration of a point set P such that 5-TD fails to have a bottleneck Hamiltonian cycle. In Section 3.5 we prove that a bottleneck perfect matching of P is contained in 6-TD, and we show that for some point set P , 5-TD does not have a bottleneck perfect matching. In Section 3.5.2 we prove that 2-TD has a perfect matching and 1-TD has a matching of size at least $\frac{2(n-1)}{5}$. In Section 3.6 we consider the problem of blocking k -TD. We show that at least $\lceil \frac{n-1}{2} \rceil$ points are necessary and $n - 1$ points are sufficient to block a 0-TD. The open problems and concluding remarks are presented in Section 3.7.

3.2 PRELIMINARIES

Bonichon et al. [9] showed that the half- Θ_6 graph of a point set P in the plane is equal to the TD-Delaunay graph of P . They also showed that every plane triangulation is TD-Delaunay realizable.

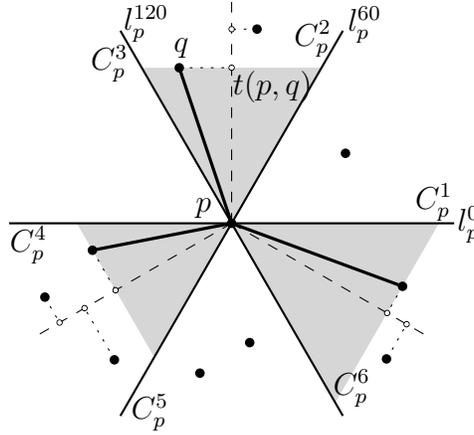


Figure 3.2: The construction of the TD-Delaunay graph.

The half- Θ_6 graph (or equivalently a TD-Delaunay graph) on a point set P can be constructed in the following way. For each point p in P , let l_p be the horizontal line through p . Define l_p^γ as the line obtained by rotating l_p by γ -degrees in counter-clockwise direction around p . Actually $l_p^0 = l_p$. Consider three lines l_p^0 , l_p^{60} , and l_p^{120} which partition the plane into six disjoint cones with apex p . Let C_p^1, \dots, C_p^6 be the cones in counter-clockwise order around p as shown in Figure 3.2. C_p^1, C_p^3, C_p^5 will be referred to as *odd cones*, and C_p^2, C_p^4, C_p^6 will be referred to as *even cones*. For each even cone C_p^i , connect p to the “nearest” point q in C_p^i . The *distance* between p and q , $d(p, q)$, is defined as the Euclidean distance between p and the orthogonal projection of q onto the bisector of C_p^i . See Figure 3.2. The resulting graph is the half- Θ_6 graph which is defined by even cones [9]. Moreover, the resulting graph is the TD-Delaunay graph defined with respect to homothets of ∇ . By considering the odd cones, another half- Θ_6 graph is obtained. The well-known Θ_6 graph is the union of half- Θ_6 graphs defined by odd and even cones.

Recall that $t(p, q)$ is the smallest homothet of ∇ having p and q on its boundary. In other words, $t(p, q)$ is the smallest downward equilateral triangle through p and q . Similarly we define $t'(p, q)$ as the smallest upward equilateral triangle having p and q on its boundary. It is obvious that the even cones correspond to downward triangles and odd cones correspond to upward triangles. We define an order on the equilateral triangles: for each two equilateral triangles t_1 and t_2 we say that $t_1 \prec t_2$ if the area of t_1 is less than the area of t_2 . Since the area of $t(p, q)$ is directly related to $d(p, q)$,

$$d(p, q) < d(r, s) \quad \text{if and only if} \quad t(p, q) \prec t(r, s).$$

For a set $\{t_1, \dots, t_m\}$ of equilateral triangles we define $\max\{t_1, \dots, t_m\}$ to be the triangle with the largest area. As shown in Figure 3.3 we have the following observation:

OBSERVATION 3.2. *If $t(p, q)$ contains a point r , then $t(p, r)$ and $t(q, r)$ are contained in $t(p, q)$.*

As a direct consequence of Observation 3.2, if a point r is contained in $t(p, q)$, then $\max\{t(p, r), t(q, r)\} \prec t(p, q)$. It is obvious that,

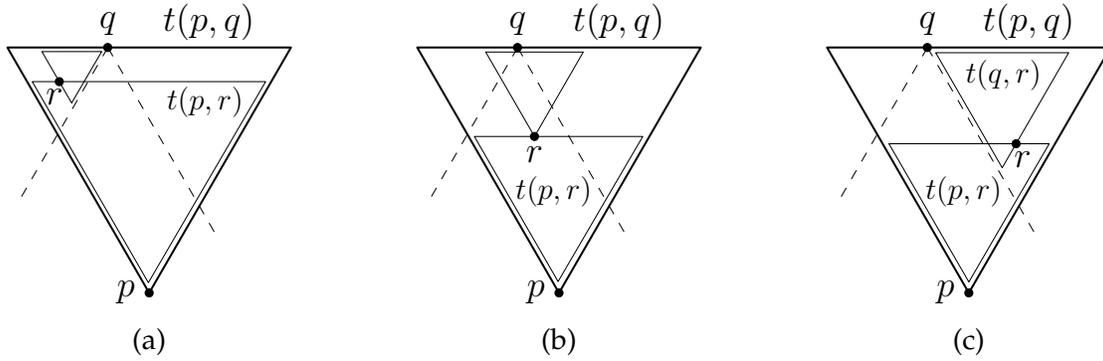


Figure 3.3: Illustration of Observation 3.2: the point r is contained in $t(p, q)$. The triangles $t(p, r)$ and $t(q, r)$ are inside $t(p, q)$.

OBSERVATION 3.3. For each two points $p, q \in P$, the area of $t(p, q)$ is equal to the area of $t'(p, q)$.

Thus, we define $X(p, q)$ as a regular hexagon centered at p which has q on its boundary, and its sides are parallel to l_p^0, l_p^{60} , and l_p^{120} .

OBSERVATION 3.4. If $X(p, q)$ contains a point r , then $t(p, r) \prec t(p, q)$.

We construct k -TD as follows. For each point $p \in P$, imagine the six cones having their apex at p , as described earlier. Then connect p to its $(k + 1)$ nearest neighbors in each even cone around p . For each edge (p, q) in k -TD we define its *weight*, $w(p, q)$, to be equal to the area of $t(p, q)$. The resulting graph is k -TD, which has $O(kn)$ edges. The k -TD can be constructed in $O(n \log n + kn \log \log n)$ -time, using the algorithm introduced by Lukovszki [20] for computing fault tolerant spanners.

For a graph $G = (V, E)$ and $K \subseteq V$, let $G - K$ be the subgraph obtained from G by removing the vertices in K , and let $o(G - K)$ be the number of odd components in $G - K$. The following theorem by Tutte [21] gives a characterization of the graphs which have a perfect matching:

THEOREM 3.5 (Tutte [21]). G has a perfect matching if and only if $o(G - K) \leq |K|$ for all $K \subseteq V$.

Berge [6] extended Tutte's theorem to a formula (known as Tutte-Berge formula) for the maximum size of a matching in a graph. In a graph G , the *deficiency*, $\text{def}_G(K)$, is $o(G - K) - |K|$. Let $\text{def}(G) = \max_{K \subseteq V} \text{def}_G(K)$.

THEOREM 3.6 (Tutte-Berge formula; Berge [6]). The size of a maximum matching in G is

$$\frac{1}{2}(n - \text{def}(G)).$$

For an edge-weighted graph G we define the *weight sequence* of G , $WS(G)$, as the sequence containing the weights of the edges of G in non-increasing order. For two graphs G_1 and G_2 we say that $WS(G_1) \prec WS(G_2)$ if $WS(G_1)$ is lexicographically smaller than $WS(G_2)$. A graph G_1 is said to be less than a graph G_2 if $WS(G_1) \prec WS(G_2)$.

3.3 CONNECTIVITY

In this section we consider the connectivity of higher-order triangular-distance Delaunay graphs.

3.3.1 $(k + 1)$ -connectivity

For a set P of points in the plane, the TD-Delaunay graph, i.e., 0-TD, is not necessarily a triangulation [15], but it is connected and internally triangulated [5], i.e., all internal faces are triangles. As shown in Figure 3.1(a), 0-TD may not be biconnected. As a warm up exercise we show that every k -TD is $(k + 1)$ -connected.

THEOREM 3.7. *For every point set P in general position in the plane, k -TD is $(k + 1)$ -connected. In addition, for every k , there exists a point set P such that k -TD is not $(k + 2)$ -connected.*

Proof. We prove the first part of this theorem by contradiction. Let K be the set of (at most) k vertices removed from k -TD, and let $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$, where $m > 1$, be the resulting maximal connected components. Let T be the set of all triangles defined by any pair of points belonging to different components, i.e., $T = \{t(a, b) : a \in C_i, b \in C_j, i \neq j\}$. Consider the smallest triangle $t_{\min} \in T$. Assume that t_{\min} is defined by two points a and b , i.e., $t_{\min} = t(a, b)$, where $a \in C_i, b \in C_j$, and $i \neq j$.

Claim 1: t_{\min} does not contain any point of $P \setminus K$ in its interior. By contradiction, suppose that t_{\min} contains a point $c \in P \setminus K$ in its interior. Three cases arise: (i) $c \in C_i$, (ii) $c \in C_j$, (iii) $c \in C_l$, where $l \neq i$ and $l \neq j$. In case (i) the triangle $t(c, b)$ between C_i and C_j is contained in $t(a, b)$. In case (ii) the triangle $t(a, c)$ between C_i and C_j is contained in $t(a, b)$. In case (iii) both triangles $t(a, c)$ and $t(c, b)$ are contained in $t(a, b)$. All cases contradict the minimality of $t(a, b) = t_{\min}$. Thus, t_{\min} contains no point of $P \setminus K$ in its interior, proving Claim 1.

By Claim 1, $t_{\min} = t(a, b)$ may only contain points of K . Since $|K| \leq k$, there must be an edge between a and b in k -TD. This contradicts that a and b belong to different components C_i and C_j in \mathcal{C} . Therefore, k -TD is $(k + 1)$ -connected.

We present a constructive proof for the second part of theorem. Let $P = A \cup B \cup K$, where $|A|, |B| \geq 1$ and $|K| = k + 1$. Place the points of A in the plane. Let $C_A^4 = \bigcap_{p \in A} C_p^4$. Place the points of K in C_A^4 . Let $C_K^4 = \bigcap_{p \in K} C_p^4$. Place the points of B in C_K^4 . Consider any pair (a, b) of points where $a \in A$ and $b \in B$. It is obvious that any path between a and b in k -TD goes through the vertices in K . Thus by removing the vertices in K , a and b become disconnected. Therefore, k -TD of P is not $(k + 2)$ -connected. \square

3.3.2 Bottleneck Biconnected Spanning Graph

As shown in Figure 3.1(a), 0-TD may not be biconnected. By Theorem 3.7, 1-TD is biconnected. In this section we show that a bottleneck biconnected spanning graph of P is contained in 1-TD.

THEOREM 3.8. *For every point set P in general position in the plane, 1-TD contains a bottleneck biconnected spanning graph of P .*

Proof. Let \mathcal{G} be the set of all biconnected spanning graphs with vertex set P . We define a total order on the elements of \mathcal{G} by their weight sequence. If two elements have the same weight sequence, we break the ties arbitrarily to get a total order. Let $G^* = (P, E)$ be a graph in \mathcal{G} with minimal weight sequence. Clearly, G^* is a bottleneck biconnected spanning graph of P . We will show that all edges of G^* are in 1-TD. By contradiction suppose that some edges in E do not belong to 1-TD, and let $e = (a, b)$ be the longest one (by the area of the triangle $t(a, b)$). If the graph $G^* - \{e\}$ is biconnected, then by removing e , we obtain a biconnected spanning graph

G with $WS(G) \prec WS(G^*)$; this contradicts the minimality of G^* . Thus, there is a pair $\{p, q\}$ of points such that any cycle between p and q in G^* goes through e . Since $(a, b) \notin 1\text{-TD}$, $t(a, b)$ contains at least two points of P , say x and y . Let G be the graph obtained from G^* by removing the edge (a, b) and adding the edges (a, x) , (b, x) , (a, y) , (b, y) . We show that in G there is a cycle C between p and q which does not go through e . Consider a cycle C^* in G^* between two points p and q (which goes through e). If none of x and y belong to C^* , then $C = (C^* - \{(a, b)\}) \cup \{(a, x), (b, x)\}$ is a cycle in G between p and q . If one of x or y , say x , belongs to C^* , then $C = (C^* - \{(a, b)\}) \cup \{(a, y), (b, y)\}$ is a cycle in G between p and q . If both x and y belong to C^* , w.l.o.g. assume that x is between b and y in the path $C^* - \{(a, b)\}$. Consider the partition of C^* into four parts: (a) edge (a, b) , (b) path δ_{bx} between b and x , (c) path δ_{xy} between x and y , and (d) path δ_{ya} between y and a . There are four cases:

1. None of p and q are on δ_{xy} . Let $C = \delta_{bx} \cup \delta_{ya} \cup \{(a, x), (b, y)\}$.
2. Both p and q are on δ_{xy} . Let $C = \delta_{xy} \cup \{(a, x), (a, y)\}$.
3. One of p, q is on δ_{xy} while the other is on δ_{bx} . Let $C = \delta_{bx} \cup \delta_{xy} \cup \{(b, y)\}$.
4. One of p, q is on δ_{xy} while the other is on δ_{ya} . Let $C = \delta_{xy} \cup \delta_{ya} \cup \{(a, x)\}$.

In all cases, C is a cycle in G between p and q . Thus, between any pair of points in G there exists a cycle, and hence G is biconnected. Since x and y are inside $t(a, b)$, by Observation 3.2, $\max\{t(a, x), t(a, y), t(b, x), t(b, y)\} \prec t(a, b)$. Therefore, $WS(G) \prec WS(G^*)$; this contradicts the minimality of G^* . \square

3.4 HAMILTONICITY

In this section we show that 7-TD contains a bottleneck Hamiltonian cycle. In addition, we will show that for some point sets, 5-TD does not contain any bottleneck Hamiltonian cycle.

THEOREM 3.9. *For every point set P in general position in the plane, 7-TD contains a bottleneck Hamiltonian cycle.*

Proof. Let \mathcal{H} be the set of all Hamiltonian cycles through the points of P . Define a total order on the elements of \mathcal{H} by their weight sequence. If two elements have exactly the same weight sequence, break ties arbitrarily to get a total order. Let $H^* = a_0, a_1, \dots, a_{n-1}, a_0$ be a cycle in \mathcal{H} with minimal weight sequence. It is obvious that H^* is a bottleneck Hamiltonian cycle of P . We will show that all the edges of H^* are in 7-TD. Consider any edge $e = (a_i, a_{i+1})$ in H^* and let $t(a_i, a_{i+1})$ be the triangle corresponding to e (all the index manipulations are modulo n).

Claim 1: None of the edges of H^* can be completely in the interior $t(a_i, a_{i+1})$. Suppose there is an edge $f = (a_j, a_{j+1})$ inside $t(a_i, a_{i+1})$. Let H be a cycle obtained from H^* by deleting e and f , and adding (a_i, a_j) and (a_{i+1}, a_{j+1}) . By Observation 3.2, $t(a_i, a_{i+1}) \succ \max\{t(a_i, a_j), t(a_{i+1}, a_{j+1})\}$, and hence $WS(H) \prec WS(H^*)$. This contradicts the minimality of H^* .

Therefore, we may assume that no edge of H^* lies completely inside $t(a_i, a_{i+1})$. Suppose there are w points of P inside $t(a_i, a_{i+1})$. Let $U = u_1, u_2, \dots, u_w$ represent these points indexed in the order we would encounter them on H^* starting from a_i . Let $R = \{r_1, r_2, \dots, r_w\}$ represent the vertices where r_i is the vertex succeeding u_i in the cycle. All the vertices in R , probably except r_w , are different from a_i (and a_{i+1}). Without loss of generality assume that $a_i \in C_{a_{i+1}}^4$, and $t(a_i, a_{i+1})$ is anchored at a_{i+1} , as shown in Figure 3.4.

Claim 2: For each $r_j \in R$, $t(r_j, a_{i+1}) \succeq \max\{t(a_i, a_{i+1}), t(u_j, r_j)\}$. Suppose there is a point $r_j \in R$ such that $t(r_j, a_{i+1}) \prec \max\{t(a_i, a_{i+1}), t(u_j, r_j)\}$. Construct a new cycle H by removing the edges (u_j, r_j) , (a_i, a_{i+1}) and adding the edges (a_{i+1}, r_j) and (a_i, u_j) . Since the two new edges have length strictly less than $\max\{t(a_i, a_{i+1}), t(u_j, r_j)\}$, $WS(H) \prec WS(H^*)$; which is a contradiction.

Claim 3: For each $r_j, r_k \in R$, $t(r_j, r_k) \succeq \max\{t(a_i, a_{i+1}), t(u_j, r_j), t(u_k, r_k)\}$. Suppose there is a pair r_j and r_k such that $t(r_j, r_k) \prec \max\{t(a_i, a_{i+1}), t(u_j, r_j), d(u_k, r_k)\}$. Construct a cycle H from H^* by first deleting (u_j, r_j) , (u_k, r_k) , (a_i, a_{i+1}) . This results in three paths. One of the paths must contain both a_i and either r_j or r_k . W.l.o.g. suppose that a_i and r_k are on the same path. Add the edges (a_i, u_j) , (a_{i+1}, u_k) , (r_j, r_k) . Since $\max\{t(u_j, r_j), t(u_k, r_k), d(a_i, a_{i+1})\} \succ \max\{t(a_i, u_j), t(a_{i+1}, u_k), t(r_j, r_k)\}$, $WS(H) \prec WS(H^*)$; we get a contradiction.

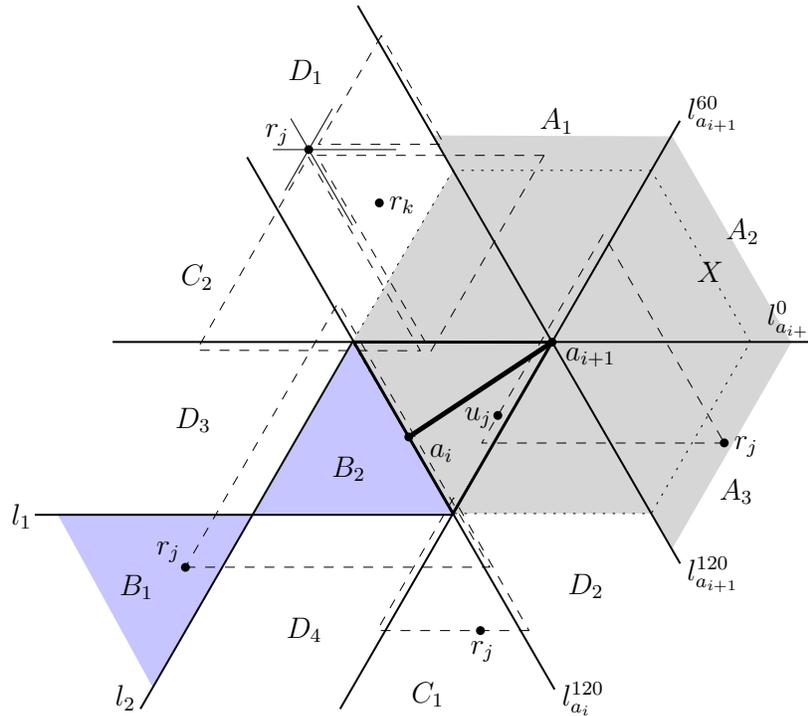


Figure 3.4: Illustration of Theorem 3.9.

We use Claim 2 and Claim 3 to show that the size of R is at most seven, and consequently $w \leq 7$. Consider the lines $l_{a_{i+1}}^0$, $l_{a_{i+1}}^{60}$, $l_{a_{i+1}}^{120}$, and $l_{a_i}^{120}$ as shown in Figure 3.4. Let l_1 and l_2 be the rays starting at the corners of $t(a_i, a_{i+1})$ opposite to a_{i+1} and parallel to $l_{a_{i+1}}^0$ and $l_{a_{i+1}}^{60}$ respectively. These lines and rays partition the plane into 12 regions, as shown in Figure 3.4. We will show that each of the regions $D_1, D_2, D_3, D_4, C_1, C_2$, and $B = B_1 \cup B_2$ contains at most one point of R , and the other regions do not contain any point of R . Consider the hexagon $X(a_{i+1}, a_i)$. By Claim 2 and Observation 3.4, no point of R can be inside $X(a_{i+1}, a_i)$. Moreover, no point of R can be inside the cones A_1, A_2 , or A_3 , because if $r_j \in \{A_1 \cup A_2 \cup A_3\}$, the (upward) triangle $t'(u_j, r_j)$ contains a_{i+1} . Then by Observation 3.4, $t(r_j, a_{i+1}) \prec t(u_j, r_j)$; which contradicts Claim 2.

We show that each of the regions D_1, D_2, D_3, D_4 contains at most one point of R . Consider the region D_1 ; by similar reasoning we can prove the claim for D_2, D_3, D_4 . Using contradiction, let r_j and r_k be two points in D_1 , and w.l.o.g. assume that r_j is the farthest to $l_{a_{i+1}}^{60}$. Then r_k can lie inside any of the cones $C_{r_j}^1, C_{r_j}^5$, and $C_{r_j}^6$ (but not in X). If $r_k \in C_{r_j}^1$, then $t'(r_j, r_k)$ is smaller than

$t'(a_i, a_{i+1})$ which means that $t(r_j, r_k) \prec t(a_i, a_{i+1})$. If $r_k \in C_{r_j}^5$, then $t'(u_j, r_j)$ contains r_k , that is $t(r_j, r_k) \prec t(u_j, r_j)$. If $r_k \in C_{r_j}^6$, then $t(u_j, r_j)$ contains r_k , that is $t(r_j, r_k) \prec t(u_j, r_j)$. All cases contradict Claim 3.

Now consider the region C_1 (or C_2). By contradiction assume that it contains two points r_j and r_k . Let r_j be the farthest from $l_{a_{i+1}}^0$. It is obvious that $t'(u_j, r_j)$ contains r_k , that is $t(r_j, r_k) \prec t(u_j, r_j)$; which contradicts Claim 3.

Consider the region $B = B_1 \cup B_2$. Note that it is possible for r_j or r_k to be a_i . If both r_j and r_k belong to B_2 , then $t'(r_j, r_k)$ is smaller than $t(a_i, a_{i+1})$. If $r_j \in B_1$ and $r_k \in B_2$, then $t'(u_j, r_j)$ contains r_k , and hence $t(r_j, r_k) \prec t(u_j, r_j)$. If both r_j and r_k belong to B_1 , let r_j be the farthest from $l_{a_i}^{120}$. Clearly, $t(u_j, r_j)$ contains r_k and hence $t(r_j, r_k) \prec t(u_j, r_j)$. All cases contradict Claim 3.

Therefore, any of the regions $D_1, D_2, D_3, D_4, C_1, C_2$, and $B = B_1 \cup B_2$ contains at most one point of R . Thus, $|R| \leq 7$ and $w \leq 7$, and $t(a_i, a_{i+1})$ contains at most 7 points of P . Therefore, $e = (a_i, a_{i+1})$ is an edge of 7-TD. \square

As a direct consequence of Theorem 3.9 we have shown that:

COROLLARY 3.10. 7-TD is Hamiltonian.

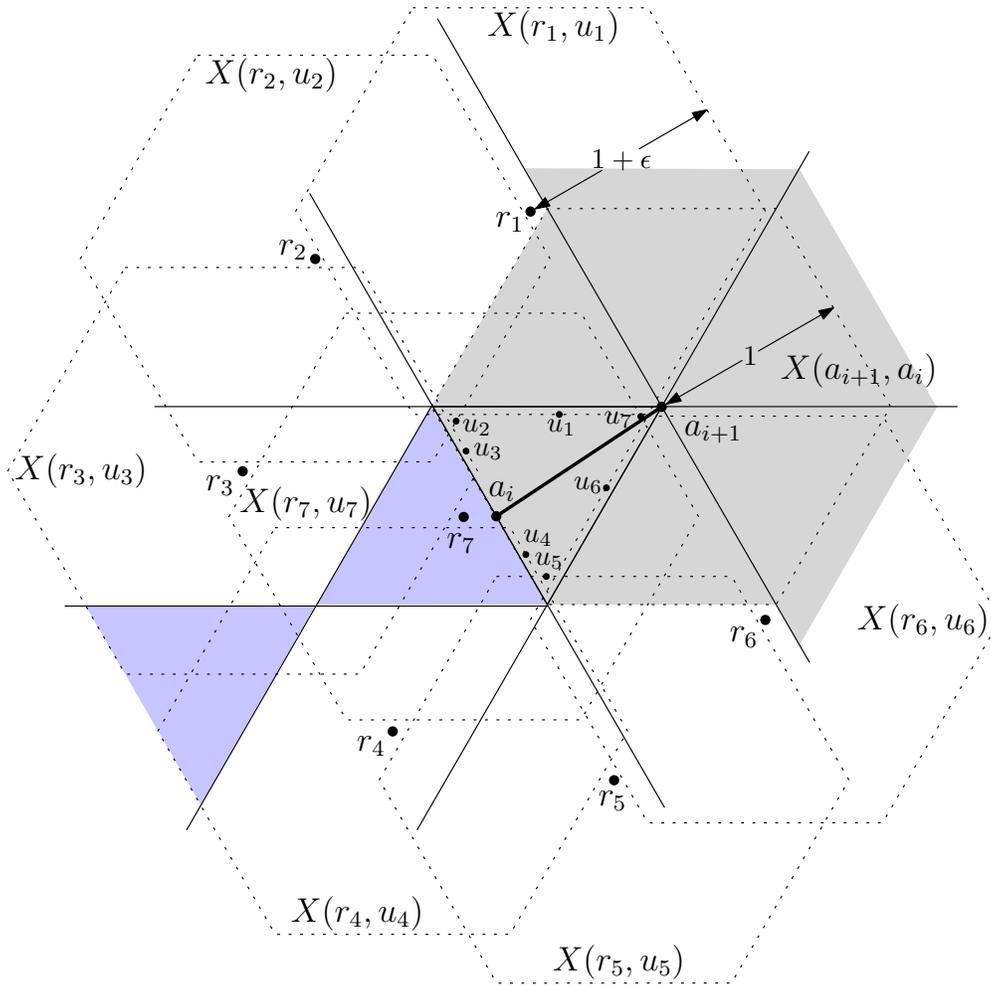


Figure 3.5: $t(a_i, a_{i+1})$ contains 7 points while the conditions in the proof of Theorem 3.9 hold.

An interesting question is to determine if k -TD contains a bottleneck Hamiltonian cycle for $k < 7$. Figure 3.5 shows a configuration where $t(a_i, a_{i+1})$ contains 7 points while the conditions of Claim 1, Claim 2, and Claim 3 in the proof of Theorem 3.9 hold. In Figure 3.5, $d(a_i, a_{i+1}) = 1$, $d(r_i, u_i) = 1 + \epsilon$, $d(r_i, r_j) > 1 + \epsilon$, $d(r_i, a_{i+1}) > 1 + \epsilon$ for $i, j = 1, \dots, 7$ and $i \neq j$.

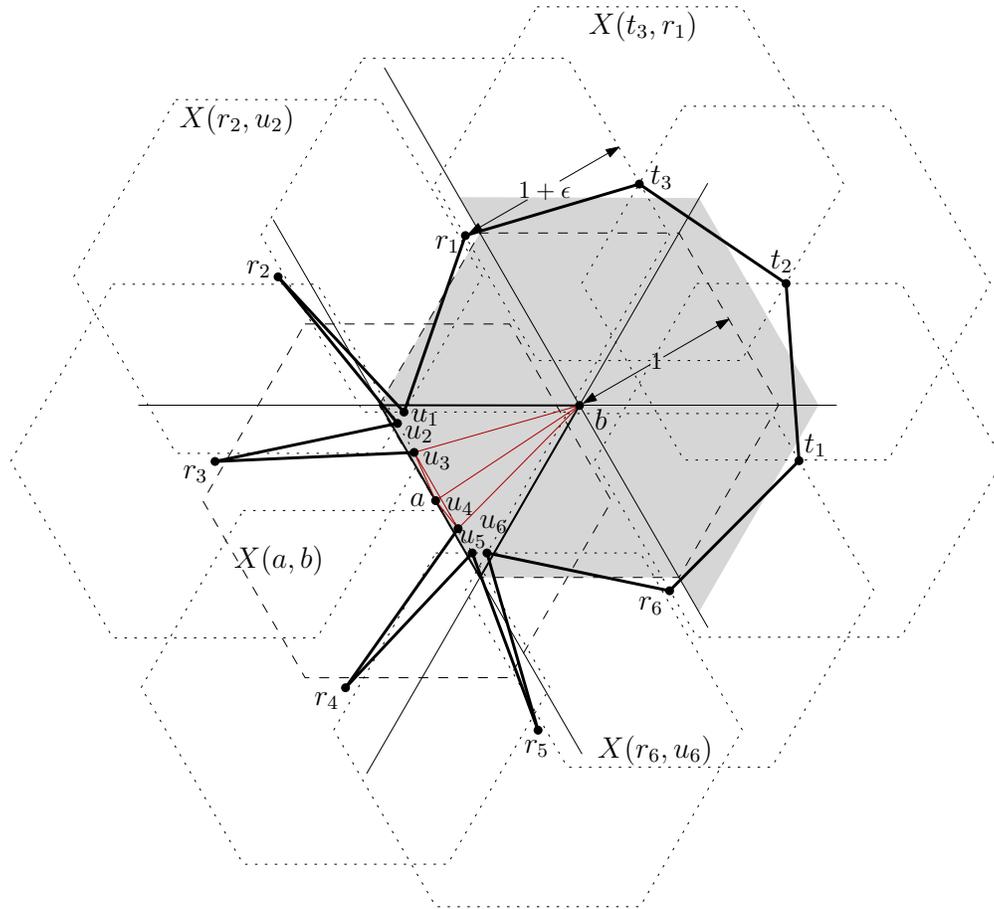


Figure 3.6: The points $\{r_1, \dots, r_6, t_1, t_2, t_3\}$ are connected to their first and second closest point (the bold edges). The edge (a, b) should be in any bottleneck Hamiltonian cycle, while $t(a, b)$ contains 6 points.

THEOREM 3.11. *There exists an arbitrary large point set such that its 5-TD does not contain any bottleneck Hamiltonian cycle.*

Proof. In order to prove the theorem, we provide such a point set. Figure 3.6 shows a configuration of P with 17 points such that 5-TD does not contain a bottleneck Hamiltonian cycle. In Figure 3.6, $d(a, b) = 1$ and $t(a, b)$ contains 6 points $U = \{u_1, \dots, u_6\}$. In addition $d(r_i, u_i) = 1 + \epsilon$, $d(r_i, r_j) > 1 + \epsilon$, $d(r_i, b) > 1 + \epsilon$ for $i, j = 1, \dots, 6$ and $i \neq j$. Let $R = \{t_1, t_2, t_3, r_1, \dots, r_6\}$. The dashed hexagons are centered at a and b and have diameter 1. The dotted hexagons are centered at vertices in R and have diameter $1 + \epsilon$. Each point in R is connected to its first and second closest points by edges of length $1 + \epsilon$ (the bold edges). Let B be the set of these edges. Let H be a cycle formed by $B \cup \{(u_3, b), (b, a), (a, u_5)\}$, i.e., $H = (u_4, r_4, u_5, r_5, u_6, r_6, t_1, t_2, t_3, r_1, u_1, r_2, u_2, r_3, u_3, a, b, u_4)$. It is obvious that H is a Hamiltonian cycle for P and $\lambda(H) = 1 + \epsilon$. Thus, the bottleneck of any bottleneck Hamiltonian cycle for P is at most $1 + \epsilon$. We will show that any bottleneck Hamiltonian cycle for P contains the edge (a, b) which does not belong to 5-TD. By

contradiction, let H^* be a bottleneck Hamiltonian cycle which does not contain (a, b) . In H^* , b is connected to two vertices b_l and b_r , where $b_l \neq a$ and $b_r \neq a$. Since the distance between b and any vertex in R is strictly bigger than $1 + \epsilon$ and $\lambda(H^*) \leq 1 + \epsilon$, $b_l \notin R$ and $b_r \notin R$. Thus b_l and b_r belong to U . Let $U' = \{u_1, u_2, u_5, u_6\}$. Consider two cases:

- $b_l \in U'$ or $b_r \in U'$. W.l.o.g. assume that $b_l \in U'$ and $b_l = u_1$. Since u_1 is the first/second closest point of r_1 and r_2 , in H^* one of r_1 and r_2 must be connected by an edge e to a point that is farther than its second closet point; e has length strictly greater than $1 + \epsilon$.
- $b_l \notin U'$ and $b_r \notin U'$. Thus, both b_l and b_r belong to $\{u_3, u_4\}$. That is, in H^* , a should be connected to a point c where $c \in R \cup U'$. If $c \in R$ then the edge (a, c) has length more than $1 + \epsilon$. If $c \in U'$, w.l.o.g. assume $c = u_1$; by the same argument as in the previous case, one of r_1 and r_2 must be connected by an edge e to a point that is farther than its second closet point; e has length strictly greater than $1 + \epsilon$.

Since $e \in H^*$, both cases contradicts that $\lambda(H^*) \leq 1 + \epsilon$. Therefore, every bottleneck Hamiltonian cycle contains edge (a, b) . Since (a, b) is not an edge in 5-TD, a bottleneck Hamiltonian cycle of P is not contained in 5-TD. We can construct larger point sets by adding new points very close to t_2 , and at distance at least $1 + 2\epsilon$ from b . \square

3.5 PERFECT MATCHING ADMISSIBILITY

In this section we consider the matching problem in higher-order triangular-distance Delaunay graphs. In Subsection 3.5.1 we show that 6-TD contains a bottleneck perfect matching. We also show that for some point sets P , 5-TD does not contain any bottleneck perfect matching. In Subsection 3.5.2 we prove that every 2-TD has a perfect matching when P has an even number of points, and 1-TD contains a matching of size at least $\frac{2(n-1)}{5}$.

3.5.1 Bottleneck Perfect Matching

THEOREM 3.12. *For a set P of an even number of points in general position in the plane, 6-TD contains a bottleneck perfect matching.*

Proof. Let \mathcal{M} be the set of all perfect matchings through the points of P . Define a total order on the elements of \mathcal{M} by their weight sequence. If two elements have exactly the same weight sequence, break ties arbitrarily to get a total order. Let $M^* = \{(a_1, b_1), \dots, (a_{\frac{n}{2}}, b_{\frac{n}{2}})\}$ be a perfect matching in \mathcal{M} with minimal weight sequence. It is obvious that M^* is a bottleneck perfect matching for P . We will show that all edges of M^* are in 6-TD. Consider any edge $e = (a_i, b_i)$ in M^* and its corresponding triangle $t(a_i, b_i)$.

Claim 1: None of the edges of M^* can be inside $t(a_i, b_i)$. Suppose there is an edge $f = (a_j, b_j)$ inside $t(a_i, b_i)$. Let M be a perfect matching obtained from M^* by deleting $\{e, f\}$, and adding $\{(a_i, a_j), (b_i, b_j)\}$. By Observation 3.2, the two new edges are smaller than the old ones. Thus, $WS(M) \prec WS(M^*)$ which contradicts the minimality of M^* .

Therefore, we may assume that no edge of M^* lies completely inside $t(a_i, b_i)$. Suppose there are w points of P inside $t(a_i, b_i)$. Let $U = u_1, u_2, \dots, u_w$ represent the points inside $t(a_i, b_i)$, and $R = r_1, r_2, \dots, r_w$ represent the points where $(r_i, u_i) \in M^*$. W.l.o.g. assume that $a_i \in C_{b_i}^4$, and $t(a_i, b_i)$ is anchored at b_i as shown in Figure 3.7.

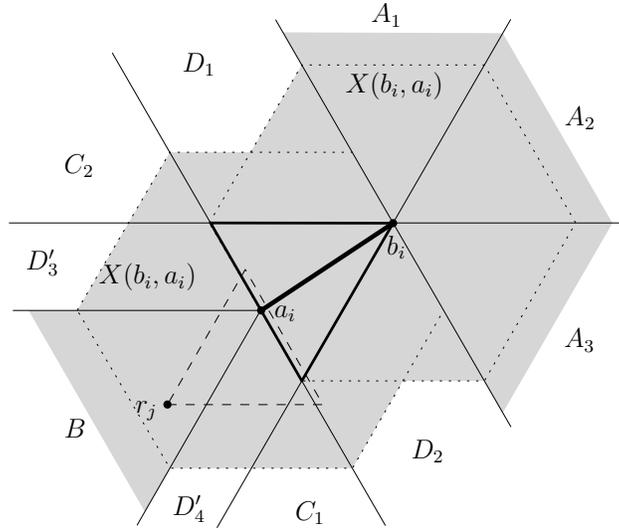


Figure 3.7: Proof of Theorem 3.12.

Claim 2: For each $r_j \in R$, $\min\{t(r_j, a_i), t(r_j, b_i)\} \succeq \max\{t(a_i, b_i), t(u_j, r_j)\}$. Otherwise, by a similar argument as in the proof of Claim 2 in Theorem 3.9 we can either match r_j with a_i or b_i to obtain a smaller matching M ; which is a contradiction.

Claim 3: For each pair r_j and r_k of points in R , $t(r_j, r_k) \succeq \max\{t(a_i, b_i), t(r_j, u_j), t(r_k, u_k)\}$. The proof is similar to the proof of Claim 3 in Theorem 3.9.

Consider Figure 3.7 which partitions the plane into eleven regions. As a direct consequence of Claim 2, the hexagons $X(b_i, a_i)$ and $X(a_i, b_i)$ do not contain any point of R . By a similar argument as in the proof of Theorem 3.9, the regions A_1, A_2, A_3 do not contain any point of R . In addition, the region B does not contain any point r_j of R , because otherwise $t'(r_j, u_j)$ contains a_i , that is $t(r_j, a_i) \prec t(u_j, r_j)$ which contradicts Claim 2. As shown in the proof of Theorem 3.9 each of the regions $D_1, D_2, D'_3, D'_4, C_1,$ and C_2 contains at most one point of R (note that $D'_3 \subset D_3$ and $D'_4 \subset D_4$). Thus, $w \leq 6$, and $t(a_i, b_i)$ contains at most 6 points of P . Therefore, $e = (a_i, b_i)$ is an edge of 6-TD. \square

As a direct consequence of Theorem 3.12 we have shown that:

COROLLARY 3.13. *For a set P of even number of points in general position in the plane, 6-TD has a perfect matching.*

In the following theorem, we show that the bound $k = 6$ proved in Theorem 3.12 is tight.

THEOREM 3.14. *There exists an arbitrarily large point set such that its 5-TD does not contain any bottleneck perfect matching.*

Proof. In order to prove the theorem, we provide such a point set. Figure 3.8 shows a configuration of a set P with 14 points such that $d(a, b) = 1$ and $t(a, b)$ contains six points $U = \{u_1, \dots, u_6\}$. In addition $d(r_i, u_i) = 1 + \epsilon$, $d(r_i, x) > 1 + \epsilon$ where $x \neq u_i$, for $i = 1, \dots, 6$. Let $R = \{r_1, \dots, r_6\}$. In Figure 3.8, the dashed hexagons are centered at a and b , each of diameter 1, and the dotted hexagons centered at vertices in R , each of diameter $1 + \epsilon$. Consider a perfect matching $M = \{(a, b)\} \cup \{(r_i, u_i) : i = 1, \dots, 6\}$ where each point $r_i \in R$ is matched to its closest point u_i . It is obvious that $\lambda(M) = 1 + \epsilon$, and hence the bottleneck of any bottleneck perfect matching is at most $1 + \epsilon$. We will show that any bottleneck perfect matching for P contains the edge (a, b)

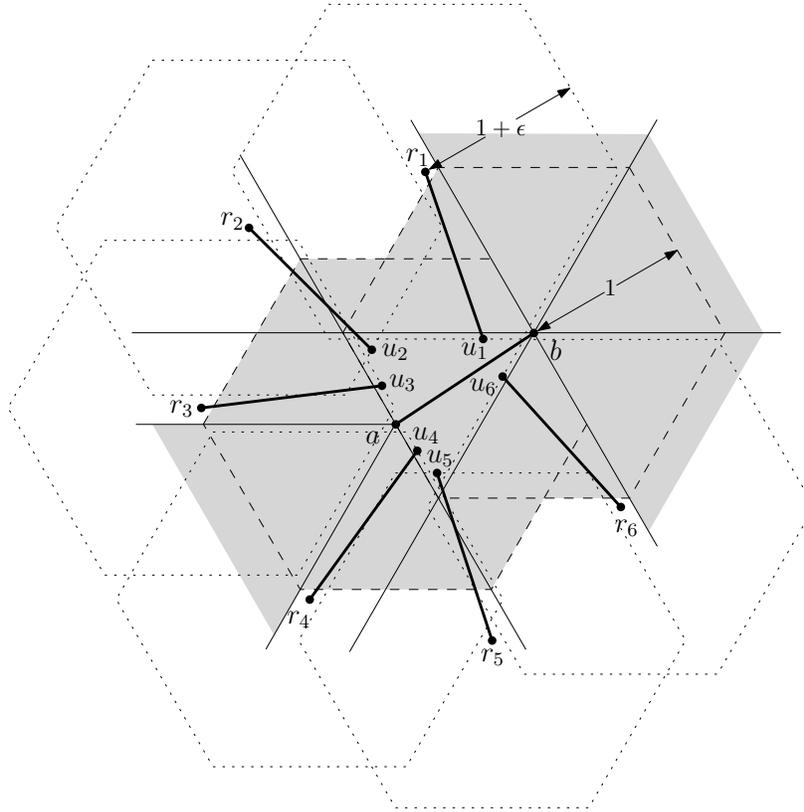


Figure 3.8: The points $\{r_1, \dots, r_6\}$ are matched to their closest point. The edge (a, b) should be an edge in any bottleneck perfect matching, while $t(a, b)$ contains 6 points.

which does not belong to 5-TD. By contradiction, let M^* be a bottleneck perfect matching which does not contain (a, b) . In M^* , b is matched to a point $c \in R \cup U$. If $c \in R$, then $d(b, c) > 1 + \epsilon$. If $c \in U$, w.l.o.g. assume $c = u_1$. Thus, in M^* the point r_1 is matched to a point d where $d \neq u_1$. Since u_1 is the unique closest point to r_1 and $d(r_1, u_1) = 1 + \epsilon$, $d(r_1, d) > 1 + \epsilon$. Both cases contradict that $\lambda(M^*) \leq 1 + \epsilon$. Therefore, every bottleneck perfect matching contains (a, b) . Since (a, b) is not an edge in 5-TD, a bottleneck perfect matching of P is not contained in 5-TD. We can construct larger point sets by adding new points—which are within distance $1 + \epsilon$ from each other—at distance at least $1 + 2\epsilon$ from the current point set. \square

3.5.2 Perfect Matching

In [5] the authors proved a tight lower bound of $\lceil \frac{n-1}{3} \rceil$ on the size of a maximum matching in 0-TD. In this section we prove that 1-TD has a matching of size $\frac{2(n-1)}{5}$ and 2-TD has a perfect matching when P has an even number of points.

For a triangle $t(a, b)$ through the points a and b , let $\text{top}(a, b)$, $\text{left}(a, b)$, and $\text{right}(a, b)$ respectively denote the top, left, and right sides of $t(a, b)$. Refer to Figure 3.9(a) for the following lemma.

LEMMA 3.15. *Let $t(a, b)$ and $t(p, q)$ intersect a horizontal line ℓ , and $t(a, b)$ intersects $\text{top}(p, q)$ in such a way that $t(p, q)$ contains the lowest corner of $t(a, b)$. Let a (resp. p) lie on $\text{right}(a, b)$ (resp. $\text{right}(p, q)$). If a and b lie above $\text{top}(p, q)$, and p and q lie above ℓ , then, $\max\{t(a, p), t(b, q)\} \prec \max\{t(a, b), t(p, q)\}$.*

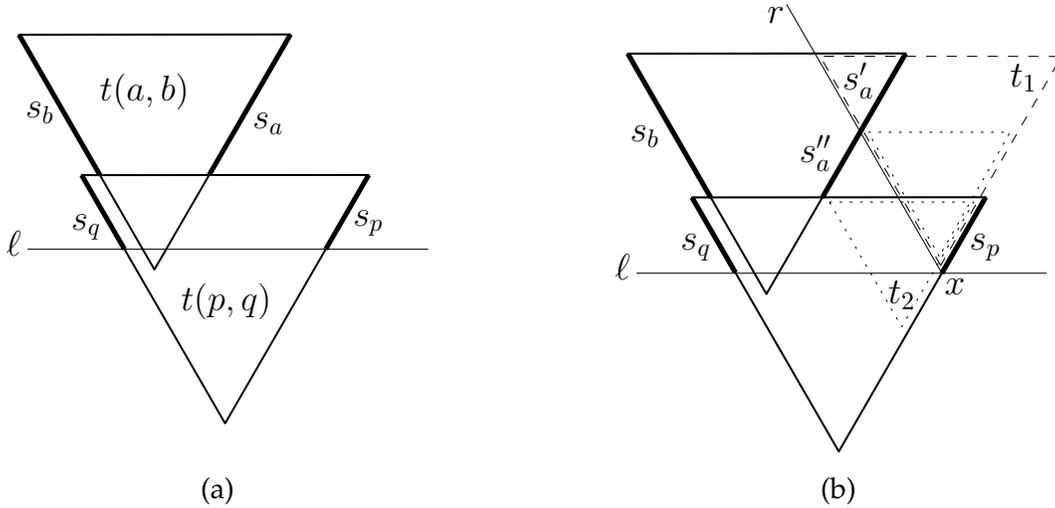


Figure 3.9: (a) Illustration of Lemma 3.15, and (b) proof of Lemma 3.15.

Proof. Recall that $t(a, b)$ is the smallest downward triangle through a and b . By Observation 3.1 each side of $t(a, b)$ contains either a or b . In Figure 3.9(a) the set of potential positions for point a on the boundary of $t(a, b)$ is shown by the line segment s_a ; and similarly by s_b, s_p, s_q for b, p, q , respectively. We will show that $t(a, p) \prec \max\{t(a, b), t(p, q)\}$. By similar reasoning we can show that $t(b, q) \prec \max\{t(a, b), t(p, q)\}$. Let x denote the intersection of ℓ and $\text{right}(p, q)$. Consider a ray r initiated at x and parallel to $\text{left}(p, q)$ which divides s_a into (at most) two parts s'_a and s''_a as shown in Figure 3.9(b). Two cases may appear:

- $a \in s'_a$. Let t_1 be a downward triangle anchored at x which has its top side on the line through $\text{top}(a, b)$ (the dashed triangle in Figure 3.9(b)). The top side of t_1 and $t(a, b)$ lie on the same horizontal line. The bottommost corner of t_1 is on ℓ while the bottommost corner of $t(a, b)$ is below ℓ . Thus, $t_1 \prec t(a, b)$. In addition, t_1 contains s'_a and s_p , thus, for any two points $a \in s'_a$ and $p \in s_p$, $t(a, p) \preceq t_1$. Therefore, $t(a, p) \prec t(a, b)$.
- $a \in s''_a$. Let t_2 be a downward triangle anchored at the intersection of $\text{right}(a, b)$ and $\text{top}(p, q)$ which has one side on the line through $\text{right}(p, q)$ (the dotted triangle in Figure 3.9(b)). This triangle is contained in $t(p, q)$, and has s_p on its right side. If we slide t_2 upward while its top-left corner remains on s''_a , the segment s_p remains on the right side of t_2 . Thus, any triangle connecting a point $a \in s''_a$ to a point $p \in s_p$ has the same size as t_2 . That is, $t(a, p) = t_2 \prec t(p, q)$.

Therefore, we have $t(a, p) \prec \max\{t(a, b), t(p, q)\}$. By similar argument we conclude that $t(b, q) \prec \max\{t(a, b), t(p, q)\}$. \square

Let $\mathcal{P} = \{P_1, P_2, \dots\}$ be a partition of the points in P . Let $G(\mathcal{P})$ be the complete graph with vertex set \mathcal{P} . For each edge $e = (P_i, P_j)$ in $G(\mathcal{P})$, let $w(e)$ be equal to the area of the smallest triangle between a point in P_i and a point in P_j , i.e. $w(e) = \min\{t(a, b) : a \in P_i, b \in P_j\}$. That is, the weight of an edge $e \in G(\mathcal{P})$ corresponds to the size of the smallest triangle $t(e)$ defined by the endpoints of e . Let \mathcal{T} be a minimum spanning tree of $G(\mathcal{P})$. Let T be the set of triangles corresponding to the edges of \mathcal{T} , i.e. $T = \{t(e) : e \in \mathcal{T}\}$.

LEMMA 3.16. *The interior of any triangle in T does not contain any point of P .*

Proof. By contradiction, suppose there is a triangle $\tau \in \mathcal{T}$ which contains a point $c \in P$. Let $e = (P_i, P_j)$ be the edge in \mathcal{J} which corresponds to τ . Let a and b respectively be the points in P_i and P_j which define τ , i.e. $\tau = t(a, b)$ and $w(e) = t(a, b)$. Three cases arise: (i) $c \in P_i$, (ii) $c \in P_j$, (iii) $c \in P_l$ where $l \neq i$ and $l \neq j$. In case (i) the triangle $t(c, b)$ between $c \in P_i$ and $b \in P_j$ is smaller than $t(a, b)$; contradicts that $w(e) = t(a, b)$ in $G(\mathcal{P})$. In case (ii) the triangle $t(a, c)$ between $a \in P_i$ and $c \in P_j$ is smaller than $t(a, b)$; contradicts that $w(e) = t(a, b)$ in $G(\mathcal{P})$. In case (iii) the triangle $t(a, c)$ (resp. $t(c, b)$) between P_i and P_l (resp. P_l and P_j) is smaller than $t(a, b)$; contradicts that e is an edge in \mathcal{J} . \square

LEMMA 3.17. *Each point in the plane can be in the interior of at most three triangles in \mathcal{T} .*

Proof. For each $t(a, b) \in \mathcal{T}$, the sides $\text{top}(a, b)$, $\text{right}(a, b)$, and $\text{left}(a, b)$ contains at least one of a and b . In addition, by Lemma 3.16, $t(a, b)$ does not contain any point of P in its interior. Thus, none of $\text{top}(a, b)$, $\text{right}(a, b)$, and $\text{left}(a, b)$ is completely inside the other triangles. Therefore, the only possible way that two triangles $t(a, b)$ and $t(p, q)$ can share a point is that one triangle, say $t(p, q)$, contains a corner of $t(a, b)$ in such a way that a and b are outside $t(p, q)$. In other words $t(a, b)$ intersects $t(p, q)$ through one of the sides $\text{top}(p, q)$, $\text{right}(p, q)$, or $\text{left}(p, q)$. If $t(a, b)$ intersects $t(p, q)$ through a direction $d \in \{\text{top}, \text{right}, \text{left}\}$ we say that $t(p, q) \prec_d t(a, b)$.

By contradiction, suppose there is a point c in the plane which is inside four triangles $\{t_1, t_2, t_3, t_4\} \subseteq \mathcal{T}$. Out of these four, either (i) three of them are like $t_i \prec_d t_j \prec_d t_k$ or (ii) there is a triangle t_l such that $t_l \prec_{\text{top}} t_i, t_l \prec_{\text{right}} t_j, t_l \prec_{\text{left}} t_k$, where $1 \leq i, j, k, l \leq 4$ and $i \neq j \neq k \neq l$. Figure 3.10 shows the two possible configurations (note that all other configurations obtained by changing the indices of triangles and/or the direction are symmetric to Figure 3.10(a) or Figure 3.10(b)).

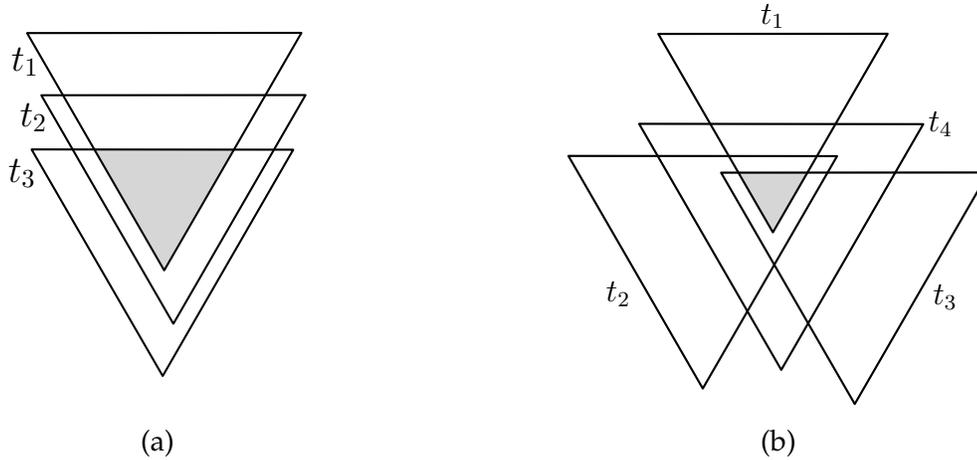


Figure 3.10: Two possible configurations: (a) $t_3 \prec_{\text{top}} t_2 \prec_{\text{top}} t_1$, (b) $t_4 \prec_{\text{top}} t_1, t_4 \prec_{\text{left}} t_2, t_4 \prec_{\text{right}} t_3$.

Recall that each of t_1, t_2, t_3, t_4 corresponds to an edge in \mathcal{J} . In the configuration of Figure 3.10(a) consider t_1, t_2 , and $\text{top}(t_3)$ which is shown in more detail in Figure 3.11(a). Suppose t_1 (resp. t_2) is defined by points a and b (resp. p and q). By Lemma 3.16, p and q are above $\text{top}(t_3)$, a and b are above $\text{top}(t_2)$. By Lemma 3.15, $\max\{t(a, p), t(b, q)\} \prec \max\{t(a, b), t(p, q)\}$. This contradicts the fact that both of the edges representing $t(a, b)$ and $t(p, q)$ are in \mathcal{J} , because by replacing $\max\{t(a, b), t(p, q)\}$ with $t(a, p)$ or $t(b, q)$, we obtain a tree \mathcal{J}' which is smaller than \mathcal{J} . In the configuration of Figure 3.10(b), consider all pairs of potential positions for two points defining t_4 which is shown in more detail in Figure 3.11(b). The pairs of potential positions on the boundary of t_4 are shown in red, green, and orange. Consider the red pair, and look at t_2, t_4 , and $\text{left}(t_1)$.

By Lemma 3.15 and the same reasoning as for the previous configuration, we obtain a smaller tree \mathcal{T}' ; which contradicts the minimality of \mathcal{T} . By symmetry, the green and orange pairs lead to a contradiction. Therefore, all configurations are invalid; which proves the lemma.

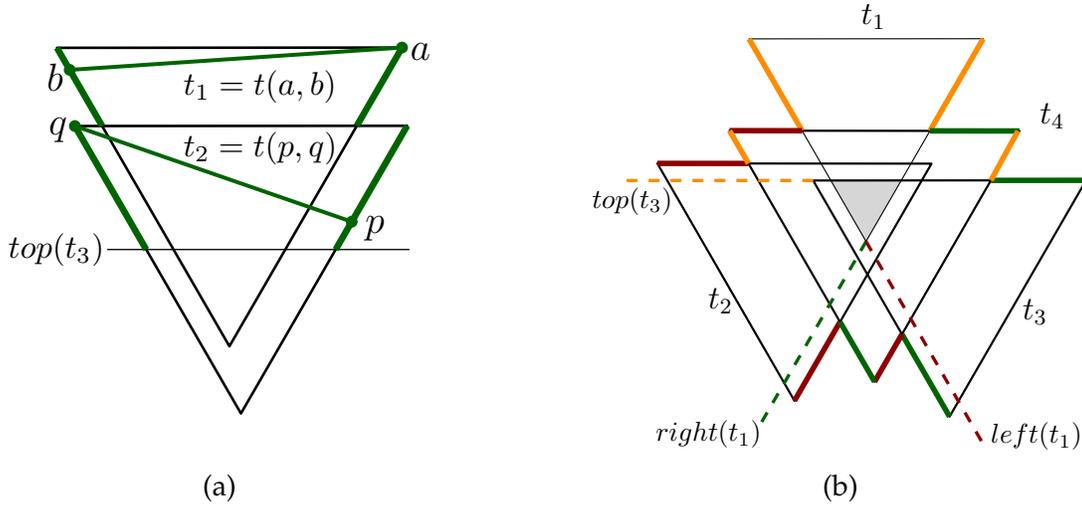


Figure 3.11: Illustration of Lemma 3.17.

□

Our results in this section are based on Lemma 3.16, Lemma 3.17, Theorem 3.5, and Theorem 3.6.

Now we prove that 2-TD has a perfect matching.

THEOREM 3.18. *For a set P of an even number of points in general position in the plane, 2-TD has a perfect matching.*

Proof. First we show that by removing a set K of points from 2-TD, at most $|K| + 1$ components are generated. Then we show that at least one of these components must be even. Finally by Theorem 3.5 we conclude that 2-TD has a perfect matching.

Let K be a set of vertices removed from 2-TD, and let $\mathcal{C} = \{C_1, \dots, C_{m(K)}\}$ be the resulting $m(K)$ components. Then, $\mathcal{P} = \{V(C_1), \dots, V(C_{m(K)})\}$ is a partition of the vertices in $P \setminus K$.

CLAIM 1. $m(K) \leq |K| + 1$.

Let $G(\mathcal{P})$ be the complete graph with vertex set \mathcal{P} which is constructed as described above. Let \mathcal{T} be a minimum spanning tree of $G(\mathcal{P})$ and let T be the set of triangles corresponding to the edges of \mathcal{T} . It is obvious that \mathcal{T} contains $m(K) - 1$ edges and hence $|T| = m(K) - 1$. Let $F = \{(p, t) : p \in K, t \in T, p \in t\}$ be the set of all (point, triangle) pairs where $p \in K$, $t \in T$, and p is inside t . By Lemma 3.17 each point in K can be inside at most three triangles in T . Thus, $|F| \leq 3 \cdot |K|$. Now we show that each triangle in T contains at least three points of K . Consider any triangle $\tau \in T$. Let $e = (V(C_i), V(C_j))$ be the edge of \mathcal{T} which is corresponding to τ , and let $a \in V(C_i)$ and $b \in V(C_j)$ be the points defining τ . By Lemma 3.16, τ does not contain any point of $P \setminus K$ in its interior. Therefore, τ contains at least three points of K , because otherwise (a, b) is an edge in 2-TD which contradicts the fact that a and b belong to different components in \mathcal{C} . Thus, each triangle in T contains at least three points of K in its interior. That is, $3 \cdot |T| \leq |F|$. Therefore, $3(m(K) - 1) \leq |F| \leq 3|K|$, and hence $m(K) \leq |K| + 1$.

CLAIM 2. $o(\mathcal{C}) \leq |K|$.

By Claim 1, $|\mathcal{C}| = m(K) \leq |K| + 1$. If $|\mathcal{C}| \leq |K|$, then $o(\mathcal{C}) \leq |K|$. Assume that $|\mathcal{C}| = |K| + 1$. Since $P = K \cup \{\bigcup_{i=1}^{|K|+1} V(C_i)\}$, the total number of vertices of P can be defined as $n = |K| + \sum_{i=1}^{|K|+1} |V(C_i)|$. Consider two cases where (i) $|K|$ is odd, (ii) $|K|$ is even. In both cases if all the components in \mathcal{C} are odd, then n is odd; this contradicts our assumption that P has an even number of vertices. Thus, \mathcal{C} contains at least one even component, which implies that $o(\mathcal{C}) \leq |K|$.

Finally, by Claim 2 and Theorem 3.5, we conclude that 2-TD has a perfect matching. \square

THEOREM 3.19. *For every set P of points in general position in the plane, 1-TD has a matching of size $\frac{2(n-1)}{5}$.*

Proof. Let K be a set of vertices removed from 1-TD, and let $\mathcal{C} = \{C_1, \dots, C_{m(K)}\}$ be the resulting $m(K)$ components. Then, $\mathcal{P} = \{V(C_1), \dots, V(C_{m(K)})\}$ is a partition of the vertices in $P \setminus K$. Note that $o(\mathcal{C}) \leq m(K)$. Let M^* be a maximum matching in 1-TD. By Theorem 3.6,

$$|M^*| = \frac{1}{2}(n - \text{def}(1\text{-TD})), \quad (3.1)$$

where

$$\text{def}(1\text{-TD}) = \max_{K \subseteq P} (o(\mathcal{C}) - |K|) \leq \max_{K \subseteq P} (|\mathcal{C}| - |K|) = \max_{0 \leq |K| \leq n} (m(K) - |K|). \quad (3.2)$$

Define $G(\mathcal{P})$, \mathcal{T} , T , and F as in the proof of Theorem 3.18. Then, by Lemma 3.17, $|F| \leq 3 \cdot |K|$. By the same reasoning as in the proof of Theorem 3.18, each triangle in T has at least two points of K in its interior. Thus, $2|T| \leq |F|$. Therefore, $2(m(K) - 1) \leq |F| \leq 3|K|$, and hence

$$m(K) \leq \frac{3|K|}{2} + 1. \quad (3.3)$$

In addition, $|K| + m(K) = |K| + |\mathcal{C}| \leq |P| = n$, and hence

$$m(K) \leq n - |K|. \quad (3.4)$$

By Inequalities (3.3) and (3.4),

$$m(K) \leq \min \left\{ \frac{3|K|}{2} + 1, n - |K| \right\}. \quad (3.5)$$

Thus, by (3.2) and (3.5)

$$\begin{aligned} \text{def}(1\text{-TD}) &\leq \max_{0 \leq |K| \leq n} (m(K) - |K|) \leq \max_{0 \leq |K| \leq n} \left\{ \min \left\{ \frac{3|K|}{2} + 1, n - |K| \right\} - |K| \right\} \\ &= \max_{0 \leq |K| \leq n} \left\{ \min \left\{ \frac{|K|}{2} + 1, n - 2|K| \right\} \right\} = \frac{n+4}{5}, \end{aligned} \quad (3.6)$$

where the last equation is achieved by setting $\frac{|K|}{2} + 1$ equal to $n - 2|K|$, which implies $|K| = \frac{2(n-1)}{5}$. Finally by substituting (3.6) in Equation (3.1) we have

$$|M^*| \geq \frac{2(n-1)}{5}.$$

\square

3.6 BLOCKING TD-DELAUNAY GRAPHS

In this section we consider the problem of blocking TD-Delaunay graphs. Let P be a set of n points in the plane such that no pair of points of P is collinear in the l^0 , l^{60} , and l^{120} directions. Recall that a point set K blocks k -TD(P) if in k -TD($P \cup K$) there is no edge connecting two points in P . That is, P is an independent set in k -TD($P \cup K$).

THEOREM 3.20. *At least $\lceil \frac{(k+1)(n-1)}{3} \rceil$ points are necessary to block k -TD(P).*

Proof. Let K be a set of m points which blocks k -TD(P). Let $G(\mathcal{P})$ be the complete graph with vertex set $\mathcal{P} = P$. Let \mathcal{T} be a minimum spanning tree of $G(\mathcal{P})$ and let T be the set of triangles corresponding to the edges of \mathcal{T} . It is obvious that $|T| = n - 1$. By Lemma 3.16 the triangles in T are empty, thus, the edges of \mathcal{T} belong to any k -TD(P) where $k \geq 0$. To block each edge, corresponding to a triangle in T , at least $k + 1$ points are necessary. By Lemma 3.17 each point in K can lie in at most three triangles of T . Therefore, $m \geq \lceil \frac{(k+1)(n-1)}{3} \rceil$, which implies that at least $\lceil \frac{(k+1)(n-1)}{3} \rceil$ points are necessary to block all the edges of \mathcal{T} and hence k -TD(P). \square

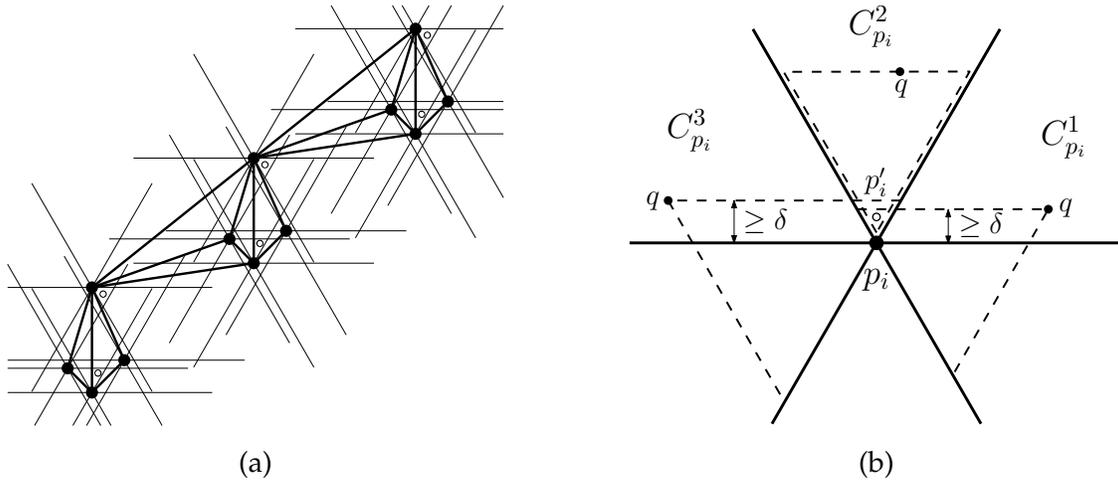


Figure 3.12: (a) A 0-TD graph which is shown in bold edges is blocked by $\lceil \frac{n-1}{2} \rceil$ white points, (b) p'_i blocks all the edges connecting p_i to the vertices above $l^0_{p_i}$.

Theorem 3.20 gives a lower bound on the number of points that are necessary to block a TD-Delaunay graph. By this theorem, at least $\lceil \frac{n-1}{3} \rceil$, $\lceil \frac{2(n-1)}{3} \rceil$, $n - 1$ points are necessary to block 0-, 1-, 2-TD(P) respectively. Now we introduce another formula which gives a better lower bound for 0-TD. For a point set P , let $\nu_k(P)$ and $\alpha_k(P)$ respectively denote the size of a maximum matching and a maximum independent set in k -TD(P). For every edge in the maximum matching, at most one of its endpoints can be in the maximum independent set. Thus,

$$\alpha_k(P) \leq |P| - \nu_k(P). \tag{3.7}$$

Let K be a set of m points which blocks k -TD(P). By definition there is no edge between points of P in k -TD($P \cup K$). That is, P is an independent set in k -TD($P \cup K$). Thus,

$$n \leq \alpha_k(P \cup K). \tag{3.8}$$

By (3.7) and (3.8) we have

$$n \leq \alpha_k(P \cup K) \leq (n + m) - \nu_k(P \cup K). \tag{3.9}$$

THEOREM 3.21. *At least $\lceil \frac{n-1}{2} \rceil$ points are necessary to block 0-TD(P).*

Proof. Let K be a set of m points which blocks 0-TD(P). Consider 0-TD(P \cup K). It is known that $\nu_0(P \cup K) \geq \lceil \frac{n+m-1}{3} \rceil$; see [5]. By Inequality (3.9),

$$n \leq (n+m) - \left\lceil \frac{n+m-1}{3} \right\rceil \leq \frac{2(n+m)+1}{3},$$

and consequently $m \geq \lceil \frac{n-1}{2} \rceil$ (note that m is an integer number). \square

Figure 3.12(a) shows a 0-TD graph on a set of 12 points which is blocked by 6 points. By removing the topmost point we obtain a set with odd number of points which can be blocked by 5 points. Thus, the lower bound provided by Theorem 3.21 is tight.

Now let $k = 1$. By Theorem 3.19 we have $\nu_1(P \cup K) \geq \frac{2((n+m)-1)}{5}$, and by Inequality (3.9)

$$n \leq (n+m) - \frac{2((n+m)-1)}{5} = \frac{3(n+m)+2}{5},$$

and consequently $m \geq \lceil \frac{2(n-1)}{3} \rceil$; the same lower bound as in Theorem 3.20.

Now let $k = 2$. By Theorem 3.18 we have $\nu_2(P \cup K) = \lfloor \frac{n+m}{2} \rfloor$ (note that $n+m$ may be odd). By Inequality (3.9)

$$n \leq (n+m) - \left\lfloor \frac{n+m}{2} \right\rfloor = \left\lceil \frac{n+m}{2} \right\rceil,$$

and consequently $m \geq n$, where $n+m$ is even, and $m \geq n-1$, where $n+m$ is odd.

THEOREM 3.22. *There exists a set K of $n-1$ points that blocks 0-TD(P).*

Proof. Let $d^0(p, q)$ be the Euclidean distance between l_p^0 and l_q^0 . Let $\delta = \min\{d^0(p, q) : p, q \in P\}$. For each point $p \in P$ let $p(x)$ and $p(y)$ respectively denote the x and y coordinates of p in the plane. Let p_1, \dots, p_n be the points of P in the increasing order of their y -coordinate. Let $K = \{p'_i : p'_i(x) = p_i(x), p'_i(y) = p_i(y) + \epsilon, \epsilon < \delta, 1 \leq i \leq n-1\}$. See Figure 3.12(b). For each point p_i , let E_{p_i} (resp. \bar{E}_{p_i}) denote the edges of 0-TD(P) between p_i and the points above $l_{p_i}^0$ (resp. below $l_{p_i}^0$). It is easy to see that the downward triangle between p_i and any point q above $l_{p_i}^0$ (i.e. any point $q \in C_{p_i}^1 \cup C_{p_i}^2 \cup C_{p_i}^3$) contains p'_i . Thus, p'_i blocks all the edges in E_{p_i} . In addition, the edges in \bar{E}_{p_i} are blocked by p'_1, \dots, p'_{i-1} . Therefore, all the edges of 0-TD(P) are blocked by the $n-1$ points in K . \square

We can extend the result of Theorem 3.22 to k -TD(P) where $k \geq 1$. For each point p_i we put $k+1$ copies of p'_i very close to p_i . Thus,

THEOREM 3.23. *There exists a set K of $(k+1)(n-1)$ points that blocks k -TD(P).*

This bound is tight. Consider the case where $k = 0$. In this case 0-TD(P) can be a path representing $n-1$ disjoint triangles and for each triangle we need at least one point to block its corresponding edge.

3.7 CONCLUSIONS

In this chapter, we considered some combinatorial properties of higher order triangular-distance Delaunay graphs of a point set P . We proved that

- k -TD is $(k + 1)$ connected.
- 1-TD contains a bottleneck biconnected spanning graph of P .
- 7-TD contains a bottleneck Hamiltonian cycle and 5-TD may not have any.
- 6-TD contains a bottleneck perfect matching and 5-TD may not have any.
- 1-TD has a matching of size at least $\frac{2(n-1)}{5}$.
- 2-TD has a perfect matching when P has an even number of points.
- $\lceil \frac{n-1}{2} \rceil$ points are necessary to block 0-TD.
- $\lceil \frac{(k+1)(n-1)}{3} \rceil$ points are necessary and $(k + 1)(n - 1)$ points are sufficient to block k -TD.

We leave a number of open problems:

- What is a tight lower bound for the size of maximum matching in 1-TD?
- Does 6-TD contain a bottleneck Hamiltonian cycle?
- As shown in Figure 3.1(a) 0-TD may not have a Hamiltonian cycle. For which values of $k = 1, \dots, 6$, is the graph k -TD Hamiltonian?

BIBLIOGRAPHY

- [1] M. Abellanas, P. Bose, J. García-López, F. Hurtado, C. M. Nicolás, and P. Ramos. On structural and graph theoretic properties of higher order Delaunay graphs. *International Journal of Computational Geometry & Applications*, 19(6):595–615, 2009.
- [2] O. Aichholzer, R. F. Monroy, T. Hackl, M. J. van Kreveld, A. Pilz, P. Ramos, and B. Vogtenhuber. Blocking Delaunay triangulations. *Computational Geometry: Theory and Applications*, 46(2):154–159, 2013.
- [3] B. Aronov, M. Dulieu, and F. Hurtado. Witness Gabriel graphs. *Computational Geometry: Theory and Applications*, 46(7):894–908, 2013.
- [4] J. Babu, A. Biniiaz, A. Maheshwari, and M. H. M. Smid. Fixed-orientation equilateral triangle matching of point sets. In *Proceedings of the 7th International Workshop on Algorithms and Computation, WALCOM'13*, pages 17–28, 2013.
- [5] J. Babu, A. Biniiaz, A. Maheshwari, and M. H. M. Smid. Fixed-orientation equilateral triangle matching of point sets. *Theoretical Computer Science*, 555:55–70, 2014.
- [6] C. Berge. Sur le couplage maximum d'un graphe. *C. R. Acad. Sci. Paris*, 247:258–259, 1958.

- [7] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. In *Proceedings of the First International Conference on Algorithms and Discrete Applied Mathematics, CALDAM'15*, pages 89–100, 2015.
- [8] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. *Computational Geometry: Theory and Applications*, 48(9):646–660, 2015.
- [9] N. Bonichon, C. Gavoille, N. Hanusse, and D. Ilcinkas. Connections between theta-graphs, Delaunay triangulations, and orthogonal surfaces. In *WG*, pages 266–278, 2010.
- [10] P. Bose, P. Carmi, S. Collette, and M. H. M. Smid. On the stretch factor of convex Delaunay graphs. *JoCG*, 1(1):41–56, 2010.
- [11] P. Bose, S. Collette, F. Hurtado, M. Korman, S. Langerman, V. Sacristan, and M. Saumell. Some properties of k -Delaunay and k -Gabriel graphs. *Computational Geometry: Theory and Applications*, 46(2):131–139, 2013.
- [12] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. 20-relative neighborhood graphs are Hamiltonian. *Journal of Graph Theory*, 15(5):543–557, 1991.
- [13] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. Solving the Euclidean bottleneck biconnected edge subgraph problem by 2-relative neighborhood graphs. *Discrete Applied Mathematics*, 39(1):1–12, 1992.
- [14] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. Solving the Euclidean bottleneck matching problem by k -relative neighborhood graphs. *Algorithmica*, 8(3):177–194, 1992.
- [15] P. Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.
- [16] M. B. Dillencourt. A non-Hamiltonian, nondegenerate Delaunay triangulation. *Information Processing Letters*, 25(3):149–151, 1987.
- [17] M. B. Dillencourt. Toughness and Delaunay triangulations. *Discrete & Computational Geometry*, 5:575–601, 1990.
- [18] H. N. Gabow and R. E. Tarjan. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9(3):411–417, 1988.
- [19] T. Kaiser, M. Saumell, and N. V. Cleemput. 10-Gabriel graphs are hamiltonian. *Information Processing Letters*, 115(11):877–881, 2015.
- [20] T. Lukovszki. New results of fault tolerant geometric spanners. In *WADS*, pages 193–204, 1999.
- [21] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 22(2):107–111, 1947.

Let P be a set of n points in general position in the plane. Given a convex geometric shape S , a geometric graph $G_S(P)$ on P is defined to have an edge between two points if and only if there exists an empty homothet of S having the two points on its boundary. A matching in $G_S(P)$ is said to be *strong*, if the homothets of S representing the edges of the matching, are pairwise disjoint, i.e., do not share any point in the plane. We consider the problem of computing a strong matching in $G_S(P)$, where S is a diametral-disk, an equilateral-triangle, or a square. We present an algorithm which computes a strong matching in $G_S(P)$; if S is a diametral-disk, then it computes a strong matching of size at least $\lceil \frac{n-1}{17} \rceil$, and if S is an equilateral-triangle, then it computes a strong matching of size at least $\lceil \frac{n-1}{9} \rceil$. If S can be a downward or an upward equilateral-triangle, we compute a strong matching of size at least $\lceil \frac{n-1}{4} \rceil$ in $G_S(P)$. When S is an axis-aligned square we compute a strong matching of size $\lceil \frac{n-1}{4} \rceil$ in $G_S(P)$, which improves the previous lower bound of $\lceil \frac{n}{5} \rceil$.

The results that are presented in this chapter are accepted to be published in the journal of Computational Geometry: Theory and Applications, special issue in memoriam: Ferran Hurtado [6].

4.1 INTRODUCTION

Let S be a compact and convex set in the plane that contains the origin in its interior. A *homothet* of S is obtained by scaling S with respect to the origin by some factor $\mu \geq 0$, followed by a translation to a point b in the plane: $b + \mu S = \{b + \mu a : a \in S\}$. For a point set P in the plane, we define $G_S(P)$ as the geometric graph on P which has a straight-line edge between two points p and q if and only if there exists a homothet of S having p and q on its boundary and whose interior does not contain any point of P . If P is in “general position”, i.e., no four points of P lie on the boundary of any homothet of S , then $G_S(P)$ is plane [10]. Hereafter, we assume that P is a set of n points in the plane, which is in general position with respect to S (we will define the general position in Section 4.2). If S is a disk \circ whose center is the origin, then $G_\circ(P)$ is the Delaunay triangulation of P . If S is an equilateral triangle ∇ whose barycenter is the origin, then $G_\nabla(P)$ is the triangular-distance Delaunay graph of P which is introduced by Chew [11].

A *matching* in a graph G is a set of edges which do not share any vertices. A *maximum matching* is a matching with maximum cardinality. A *perfect matching* is a matching which matches all the vertices of G . Let \mathcal{M} be a matching in $G_S(P)$. \mathcal{M} is referred to as a *matching of points with shape S* , e.g., a matching in $G_\circ(P)$ is a matching of points with disks. Let $\mathcal{S}_\mathcal{M}$ be a set of homothets of S representing the edges of \mathcal{M} . \mathcal{M} is called a *strong matching* if there exists a set $\mathcal{S}_\mathcal{M}$ whose elements are pairwise disjoint, i.e., the objects in $\mathcal{S}_\mathcal{M}$ do not share any point in the plane. Otherwise, \mathcal{M} is a *weak matching*. See Figure 4.1. To be consistent with the definition of the matching in the graph theory, we use the term “matching” to refer to a weak matching. Given a point set P in the plane and a shape S , the (*strong*) *matching problem* is to compute a (strong) matching of maximum cardinality in $G_S(P)$. In this chapter we consider the strong matching problem of points in general position in the plane with respect to a given shape $S \in \{\circ, \nabla, \square\}$ (see

Section 4.2 for the definition), where by \ominus we mean the line segment between the two points on the boundary of the disk is a diameter of that disk.

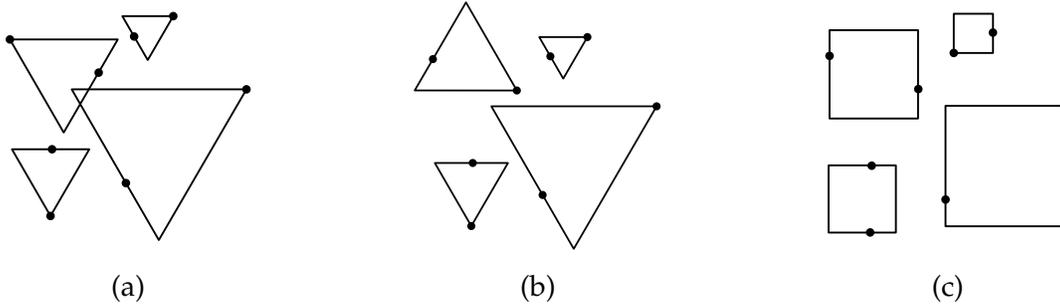


Figure 4.1: Point set P and (a) a perfect weak matching in $G_{\nabla}(P)$, (b) a perfect strong matching in $G_{\star}(P)$, and (c) a perfect strong matching in $G_{\square}(P)$.

4.1.1 Previous Work

The problem of computing a maximum matching in $G_S(P)$ is one of the fundamental problems in computational geometry and graph theory [1, 2, 3, 5, 8, 7, 12]. Dillencourt [12] and Ábrego et al. [1] considered the problem of matching points with disks. Let S be a closed disk \circ whose center is the origin, and let P be a set of n points in the plane which is in general position with respect to \circ . Then, $G_{\circ}(P)$ is the graph which has an edge between two points $p, q \in P$ if there exists a homothet of \circ having p and q on its boundary and does not contain any point of $P \setminus \{p, q\}$. $G_{\circ}(P)$ is equal to the Delaunay triangulation on P , $DT(P)$. Dillencourt [12] proved that $G_{\circ}(P)$ contains a perfect (weak) matching. Ábrego et al. [1] proved that $G_{\circ}(P)$ has a strong matching of size at least $\lceil (n-1)/8 \rceil$. They also showed that there exists a set P of n points in the plane with arbitrarily large n , such that $G_{\circ}(P)$ does not contain a strong matching of size more than $\frac{36}{73}n$.

For two points p and q , the disk which has the line segment pq as its diameter is called the diametral-disk between p and q . We denote a diametral-disk by \ominus . Let $G_{\ominus}(P)$ be the graph which has an edge between two points $p, q \in P$ if the diametral-disk between p and q does not contain any point of $P \setminus \{p, q\}$. $G_{\ominus}(P)$ is equal to the Gabriel graph on P , $GG(P)$. Biniáz et al. [8] proved that $G_{\ominus}(P)$ has a matching of size at least $\lceil (n-1)/4 \rceil$, and this bound is tight.

The problem of matching of points with equilateral triangles has been considered by Babu et al. [3]. Let S be a downward equilateral triangle ∇ whose barycenter is the origin and one of its vertices is on the negative y -axis. Let P be a set of n points in the plane which is in general position with respect to ∇ . Let $G_{\nabla}(P)$ be the graph which has an edge between two points $p, q \in P$ if there exists a homothet of ∇ having p and q on its boundary and does not contain any point of $P \setminus \{p, q\}$. $G_{\nabla}(P)$ is equal to the triangular-distance Delaunay graph on P , which was introduced by Chew [11]. Bonichon et al. [9] showed that $G_{\nabla}(P)$ is equal to the half-theta six graph on P , $\frac{1}{2}\Theta_6(P)$. Babu et al. [3] proved that $G_{\nabla}(P)$ has a matching of size at least $\lceil (n-1)/3 \rceil$, and this bound is tight. If we consider an upward triangle \triangle , then $G_{\triangle}(P)$ is defined similarly. Let $G_{\star}(P)$ be the graph on P which is the union of $G_{\nabla}(P)$ and $G_{\triangle}(P)$. Bonichon et al. [9] showed that $G_{\star}(P)$ is equal to the theta six graph on P , $\Theta_6(P)$. Since $G_{\nabla}(P)$ is a subgraph of $G_{\star}(P)$, the lower bound of $\lceil (n-1)/3 \rceil$ on the size of maximum matching in $G_{\nabla}(P)$ holds for $G_{\star}(P)$.

The problem of strong matching of points with axis-aligned rectangles is trivial. An obvious algorithm is to repeatedly match the two leftmost points. The problem of matching points with

Table 4.1: Lower bounds on the size of weak and strong matchings in $G_S(P)$.

| S | Weak match. | Ref. | Strong match. | Ref. |
|-------------------------|-------------------------------|--------|--|------------------------|
| \circ | $\lfloor \frac{n}{2} \rfloor$ | [12] | $\lceil \frac{n-1}{8} \rceil$ | [1] |
| \ominus | $\lceil \frac{n-1}{4} \rceil$ | [8] | $\lceil \frac{n-1}{17} \rceil$ | Theorem 4.14 |
| ∇ | $\lceil \frac{n-1}{3} \rceil$ | [3] | $\lceil \frac{n-1}{9} \rceil$ | Theorem 4.26 |
| ∇ or \triangle | $\lceil \frac{n-1}{3} \rceil$ | [3] | $\lceil \frac{n-1}{4} \rceil$ | Theorem 4.27 |
| \square | $\lfloor \frac{n}{2} \rfloor$ | [1, 2] | $\lceil \frac{n}{5} \rceil$ $\lceil \frac{n-1}{4} \rceil$ | [1, 2] Theorem 4.28 |

axis-aligned squares was considered by Ábrego et al. [2]. Let S be an axis-aligned square \square whose center is the origin. Let P be a set of n points in the plane which is in general position with respect to \square . Let $G_{\square}(P)$ be the graph which has an edge between two points $p, q \in P$ if there exists a homothet of \square having p and q on its boundary and does not contain any point of $P \setminus \{p, q\}$. $G_{\square}(P)$ is equal to the L_{∞} -Delaunay graph on P . Ábrego et al. [1, 2] proved that $G_{\square}(P)$ has a perfect (weak) matching and a strong matching of size at least $\lceil n/5 \rceil$. Further, they showed that there exists a set P of n points in the plane with arbitrarily large n , such that $G_{\square}(P)$ does not contain a strong matching of size more than $\frac{5}{11}n$. Table 4.1 summarizes the results.

Bereg et al. [5] concentrated on matching points of P with axis-aligned rectangles and squares, where P is not necessarily in general position. They proved that any set of n points in the plane has a strong rectangle matching of size at least $\lfloor \frac{n}{3} \rfloor$, and such a matching can be computed in $O(n \log n)$ time. As for squares, they presented a $\Theta(n \log n)$ time algorithm that decides whether a given matching has a weak square realization, and an $O(n^2 \log n)$ time algorithm for the strong square matching realization. They also proved that it is NP-hard to decide whether a given point set has a perfect strong square-matching.

4.1.2 Our results

In this chapter we consider the problem of computing a strong matching in $G_S(P)$, where $S \in \{\ominus, \nabla, \square\}$. In Section 4.2, we provide some observations and prove necessary Lemmas. Given a point set P in which is in general position with respect to a given shape S , in Section 4.3, we present an algorithm which computes a strong matching in $G_S(P)$. In Section 4.4, we prove that if S is a diametral-disk, then the algorithm of Section 4.3 computes a strong matching of size at least $\lceil (n-1)/17 \rceil$ in $G_{\ominus}(P)$. In Section 4.5, we prove that if S is an equilateral triangle, then the algorithm of Section 4.3 computes a strong matching of size at least $\lceil (n-1)/9 \rceil$ in $G_{\nabla}(P)$. In Section 4.6, we compute a strong matching of size at least $\lceil (n-1)/4 \rceil$ in $G_{\triangle}(P)$. In Section 4.7, we compute a strong matching of size at least $\lceil (n-1)/4 \rceil$ in $G_{\square}(P)$; this improves the previous lower bound of $\lceil n/5 \rceil$. A summary of the results is given in Table 4.1. In Section 4.8 we discuss a possible way to further improve upon the result obtained for diametral-disks in Section 4.4. Concluding remarks and open problems are given in Section 4.9.

4.2 PRELIMINARIES

Let $S \in \{\ominus, \nabla, \square\}$, and let S_1 and S_2 be two homothets of S . We say that S_1 is *smaller than* S_2 if the area of S_1 is smaller than the area of S_2 . For two points $p, q \in P$, let $S(p, q)$ be a smallest homothet of S having p and q on its boundary. If S is a diametral-disk, a downward equilateral-triangle, or a square, then we denote $S(p, q)$ by $D(p, q)$, $t(p, q)$, or $Q(p, q)$, respectively. If S is a diametral-disk, then $D(p, q)$ is uniquely defined by p and q . If S is an equilateral-triangle or a square, then S has the *shrinkability* property: if there exists a homothet S' of S that contains two points p and q , then there exists a homothet S'' of S such that $S'' \subseteq S'$, and p and q are on the boundary of S'' . If S is an equilateral-triangle, then we can shrink S'' further, such that each side of S'' contains either p or q . If S is a square, then we can shrink S'' further, such that p and q are on opposite sides of S'' . Thus, we have the following observation:

OBSERVATION 4.1. For two points $p, q \in P$,

- $D(p, q)$ is uniquely defined by p and q , and it has the line segment pq as a diameter.
- $t(p, q)$ is uniquely defined by p and q , and it has one of p and q on a corner and the other point is on the side opposite to that corner.
- $Q(p, q)$ has p and q on opposite sides.

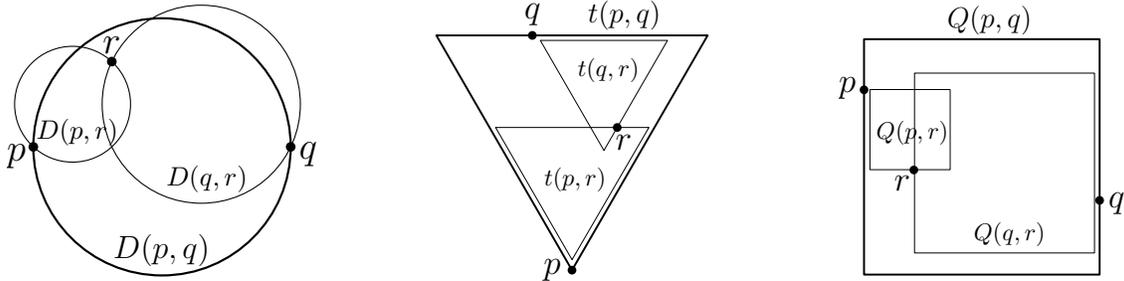


Figure 4.2: Illustration of Observation 4.2.

Given a shape $S \in \{\ominus, \nabla, \square\}$, we define an order on the homothets of S . Let S_1 and S_2 be two homothets of S . We say that $S_1 \prec S_2$ if the area of S_1 is less than the area of S_2 . Similarly, $S_1 \preceq S_2$ if the area of S_1 is less than or equal to the area of S_2 . We denote the homothet with the larger area by $\max\{S_1, S_2\}$. As illustrated in Figure 4.2, if $S(p, q)$ contains a point r , then both $S(p, r)$ and $S(q, r)$ have smaller area than $S(p, q)$. Thus, we have the following observation:

OBSERVATION 4.2. If $S(p, q)$ contains a point r , then $\max\{S(p, r), S(q, r)\} \prec S(p, q)$.

DEFINITION 4.3. Given a point set P and a shape $S \in \{\ominus, \nabla, \square\}$, we say that P is in “general position” with respect to S if

$S = \ominus$: no four points of P lie on the boundary of any diametral disk defined by any two points of P .

$S = \nabla$: the line passing through any two points of P does not make angles 0° , 60° , or 120° with the horizontal. This implies that no four points of P are on the boundary of any homothet of ∇ .

$S = \square$: (i) no two points in P have the same x -coordinate or the same y -coordinate, and (ii) no four points of P lie on the boundary of any homothet of \square .

Given a point set P which is in general position with respect to a given shape $S \in \{\ominus, \nabla, \square\}$, let $K_S(P)$ be the complete edge-weighted geometric graph on P . For each edge $e = (p, q)$ in $K_S(P)$, we define $S(e)$ to be the shape $S(p, q)$, i.e., a smallest homothet of S having p and q on its boundary. We say that $S(e)$ *represents* e , and vice versa. Furthermore, we assume that the weight $w(e)$ (resp. $w(p, q)$) of e is equal to the area of $S(e)$. Thus,

$$w(p, q) < w(r, s) \quad \text{if and only if} \quad S(p, q) \prec S(r, s).$$

Note that $G_S(P)$ is a subgraph of $K_S(P)$, and has an edge (p, q) iff $S(p, q)$ does not contain any point of $P \setminus \{p, q\}$.

LEMMA 4.4. *Let P be a set of n points in the plane which is in general position with respect to a given shape $S \in \{\ominus, \nabla, \square\}$. Then, any minimum spanning tree of $K_S(P)$ is a subgraph of $G_S(P)$.*

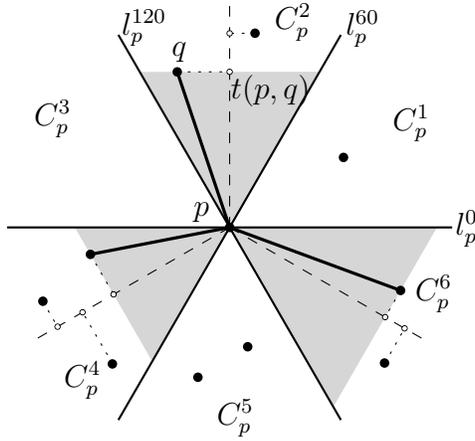
Proof. The proof is by contradiction. Assume there exists an edge $e = (p, q)$ in a minimum spanning tree T of $K_S(P)$ such that $e \notin G_S(P)$. Since (p, q) is not an edge in $G_S(P)$, $S(p, q)$ contains a point r such that $r \in P \setminus \{p, q\}$. By Observation 4.2, $\max\{S(p, r), S(q, r)\} \prec S(p, q)$. Thus, $w(p, r) < w(p, q)$ and $w(q, r) < w(p, q)$. By replacing the edge (p, q) in T with either (p, r) or (q, r) , we obtain a spanning tree in $K_S(P)$ which is smaller than T . This contradicts the minimality of T . \square

LEMMA 4.5. *Let G be an edge-weighted graph with edge set E and edge-weight function $w : E \rightarrow \mathbb{R}^+$. For any cycle C in G , if the maximum-weight edge in C is unique, then that edge is not in any minimum spanning tree of G .*

Proof. The proof is by contradiction. Let $e = (u, v)$ be the unique maximum-weight edge in a cycle C in G , such that e is in a minimum spanning tree T of G . Let T_u and T_v be the two trees obtained by removing e from T . Let $e' = (x, y)$ be an edge in C which connects a vertex $x \in T_u$ to a vertex $y \in T_v$. By assumption, $w(e') < w(e)$. Thus, in T , by replacing e with e' , we obtain a tree $T' = T_u \cup T_v \cup \{(x, y)\}$ in G such that $w(T') < w(T)$. This contradicts the minimality of T . \square

Recall that $t(p, q)$ is the smallest homothet of ∇ which has p and q on its boundary. Similarly, let $t'(p, q)$ denote the smallest upward equilateral-triangle \triangle having p and q on its boundary. Note that $t'(p, q)$ is uniquely defined by p and q , and it has one of p and q on a corner and the other point is on the side opposite to that corner. In addition the area of $t'(p, q)$ is equal to the area of $t(p, q)$.

$G_{\nabla}(P)$ is equal to the triangular-distance Delaunay graph $TD-DG(P)$, which is in turn equal to a half theta-six graph $\frac{1}{2}\Theta_6(P)$ [9]. A half theta-six graph on P , and equivalently $G_{\nabla}(P)$, can be constructed in the following way. For each point p in P , let l_p be the horizontal line through p . Define l_p^γ as the line obtained by rotating l_p by γ -degrees in counter-clockwise direction around p . Thus, $l_p^0 = l_p$. Consider three lines l_p^0 , l_p^{60} , and l_p^{120} which partition the plane into six disjoint cones with apex p . Let C_p^1, \dots, C_p^6 be the cones in counter-clockwise order around p as shown in Figure 4.3. C_p^1, C_p^3, C_p^5 will be referred to as *odd cones*, and C_p^2, C_p^4, C_p^6 will be referred to as *even cones*. For each even cone C_p^i , connect p to the “nearest” point q in C_p^i . The *distance* between p and q , is defined as the Euclidean distance between p and the orthogonal projection of q onto the bisector of C_p^i . See Figure 4.3. In other words, the nearest point to P in C_p^i is a point q in C_p^i which minimizes the area of $t(p, q)$. The resulting graph is the half theta-six graph which is defined by even cones [9]. Moreover, the resulting graph is $G_{\nabla}(P)$ which is defined with respect to the homothets of ∇ . By considering the odd cones, $G_{\triangle}(P)$ is obtained. By considering the odd

Figure 4.3: The construction of $G_{\nabla}(P)$.

cones and the even cones, $G_{\nabla}(P)$ —which is equal to $\Theta_6(P)$ —is obtained. Note that $G_{\nabla}(P)$ is the union of $G_{\nabla}(P)$ and $G_{\Delta}(P)$.

Let $X(p, q)$ be the regular hexagon centered at p which has q on its boundary, and its sides are parallel to l_p^0 , l_p^{60} , and l_p^{120} . Then, we have the following observation:

OBSERVATION 4.6. *If $X(p, q)$ contains a point r , then $t(p, r) \prec t(p, q)$.*

4.3 STRONG MATCHING IN $G_S(P)$

Given a point set P in the plane which is in general position with respect to a given shape $S \in \{\Theta, \nabla, \square\}$, in this section we present an algorithm which computes a strong matching in $G_S(P)$. Recall that $K_S(P)$ is the complete edge-weighted graph on P with the weight of each edge e is equal to the area of $S(e)$, where $S(e)$ is a smallest homothet of S representing e . Let T be a minimum spanning tree of $K_S(P)$. By Lemma 4.4, T is a subgraph of $G_S(P)$. For each edge $e \in T$ we denote by $T(e^+)$ the set of all edges in T whose weight is at least $w(e)$. Moreover, we define the *influence set* of e , as the set of all edges in $T(e^+)$ whose representing shapes overlap with $S(e)$, i.e.,

$$\text{Inf}(e) = \{e' : e' \in T(e^+), S(e') \cap S(e) \neq \emptyset\}.$$

Note that $\text{Inf}(e)$ is not empty, as $e \in \text{Inf}(e)$. Consequently, we define the *influence number* of T to be the maximum size of a set among the influence sets of edges in T , i.e.,

$$\text{Inf}(T) = \max\{|\text{Inf}(e)| : e \in T\}.$$

Algorithm 2 receives $G_S(P)$ as input and computes a strong matching in $G_S(P)$ as follows. The algorithm starts by computing a minimum spanning tree T of $G_S(P)$, where the weight of each edge is equal to the area of its representing shape. Then it initializes a forest F by T , and a matching \mathcal{M} by an empty set. Afterwards, as long as F is not empty, the algorithm adds to \mathcal{M} , the smallest edge e in F , and removes the influence set of e from F . Finally, it returns \mathcal{M} .

THEOREM 4.7. *Given a set P of n points in the plane and a shape $S \in \{\Theta, \nabla, \square\}$, Algorithm 2 computes a strong matching of size at least $\lceil \frac{n-1}{\text{Inf}(T)} \rceil$ in $G_S(P)$, where T is a minimum spanning tree of $G_S(P)$.*

Algorithm 2 *Strong-matching*($G_S(P)$)

```

1:  $T \leftarrow \text{MST}(G_S(P))$ 
2:  $F \leftarrow T$ 
3:  $\mathcal{M} \leftarrow \emptyset$ 
4: while  $F \neq \emptyset$  do
5:    $e \leftarrow$  smallest edge in  $F$ 
6:    $\mathcal{M} \leftarrow \mathcal{M} \cup \{e\}$ 
7:    $F \leftarrow F - \text{Inf}(e)$ 
8: return  $\mathcal{M}$ 

```

Proof. Let \mathcal{M} be the matching returned by Algorithm 2. First we show that \mathcal{M} is a strong matching. If \mathcal{M} contains one edge, then trivially, \mathcal{M} is a strong matching. Consider any two edges e_1 and e_2 in \mathcal{M} . Without loss of generality assume that e_1 is considered before e_2 in the while loop. At the time e_1 is added to \mathcal{M} , the algorithm removes from F , the edges in $\text{Inf}(e_1)$, i.e., all the edges whose representing shapes intersect $S(e_1)$. Since e_2 remains in F after the removal of $\text{Inf}(e_1)$, $e_2 \notin \text{Inf}(e_1)$. This implies that $S(e_1) \cap S(e_2) = \emptyset$, and hence \mathcal{M} is a strong matching.

In each iteration of the while loop we select e as the smallest edge in F , where F is a subgraph of T . Then, all edges in F have weight at least $w(e)$. Thus, $F \subseteq T(e^+)$; which implies that the set of edges in F whose representing shapes intersect $S(e)$ is a subset of $\text{Inf}(e)$. Therefore, in each iteration of the while loop, out of at most $|\text{Inf}(e)|$ -many edges of T , we add one edge to \mathcal{M} . Since $|\text{Inf}(e)| \leq \text{Inf}(T)$ and T has $n - 1$ edges, we conclude that $|\mathcal{M}| \geq \lceil \frac{n-1}{\text{Inf}(T)} \rceil$. \square

REMARK Let T be the minimum spanning tree computed by Algorithm 2. Let $e = (u, v)$ be an edge in T . Recall that $T(e^+)$ contains all the edges of T whose weight is at least $w(e)$. We define the *degree* of e as $\text{deg}(e) = \text{deg}(u) + \text{deg}(v) - 1$, where $\text{deg}(u)$ and $\text{deg}(v)$ are the number of edges incident on u and v in $T(e^+)$, respectively. Note that all the edges incident on u or v in $T(e^+)$ are in the influence set of e . Thus, $|\text{Inf}(e)| \geq \text{deg}(e)$, and consequently $\text{Inf}(T) \geq \text{deg}(e)$.

4.4 STRONG MATCHING IN $G_{\ominus}(P)$

In this section we consider the case where S is a diametral-disk \ominus . Recall that $G_{\ominus}(P)$ is an edge-weighted geometric graph, where the weight of an edge (p, q) is equal to the area of $D(p, q)$. $G_{\ominus}(P)$ is equal to the Gabriel graph, $\text{GG}(P)$. We prove that $G_{\ominus}(P)$, and consequently $\text{GG}(P)$, has a strong diametral-disk matching of size at least $\lceil \frac{n-1}{17} \rceil$.

We run Algorithm 2 on $G_{\ominus}(P)$ to compute a matching \mathcal{M} . By Theorem 4.7, \mathcal{M} is a strong matching of size at least $\lceil \frac{n-1}{\text{Inf}(T)} \rceil$, where T is a minimum spanning tree in $G_{\ominus}(P)$. By Lemma 4.4, T is a minimum spanning tree of the complete graph $K_{\ominus}(P)$. Observe that T is a Euclidean minimum spanning tree for P as well. In order to prove the desired lower bound, we show that $\text{Inf}(T) \leq 17$. Since $\text{Inf}(T)$ is the maximum size of a set among the influence sets of edges in T , it suffices to show that for every edge e in T , the influence set of e contains at most 17 edges.

LEMMA 4.8. *Let T be a minimum spanning tree of $G_{\ominus}(P)$, and let e be any edge in T . Then, $|\text{Inf}(e)| \leq 17$.*

We will prove this lemma in the rest of this section. Recall that, for each two points $p, q \in P$, $D(p, q)$ is the closed diametral-disk with diameter pq . Let \mathcal{D} denote the set of diametral-disks representing the edges in T . Since T is a subgraph of $G_{\ominus}(P)$, we have the following observation:

OBSERVATION 4.9. Each disk in \mathcal{D} does not contain any point of P in its interior.

We have proved the following lemma in Chapter 2.

LEMMA 4.10. For each pair D_i and D_j of disks in \mathcal{D} , D_i (resp. D_j) does not contain the center of D_j (resp D_i).

Let $e = (u, v)$ be an edge in T . Without loss of generality, we suppose that $D(u, v)$ has radius 1 and centered at the origin $o = (0, 0)$ such that $u = (-1, 0)$ and $v = (1, 0)$. For any point p in the plane, let $\|p\|$ denote the distance of p from o . Let $\mathcal{D}(e^+)$ be the disks in \mathcal{D} representing the edges of $T(e^+)$. Recall that $T(e^+)$ contains the edges of T whose weight is at least $w(e)$, where $w(e)$ is equal to the area of $D(u, v)$. Since the area of any circle is directly related to its radius, we have the following observation:

OBSERVATION 4.11. The disks in $\mathcal{D}(e^+)$ have radius at least 1.

Let $C(x, r)$ (resp. $D(x, r)$) be the circle (resp. closed disk) of radius r which is centered at a point x in the plane. Let $\mathcal{J}(e^+) = \{D_1, \dots, D_k\}$ be the set of disks in $\mathcal{D}(e^+) \setminus \{D(u, v)\}$ intersecting $D(u, v)$. We show that $\mathcal{J}(e^+)$ contains at most sixteen disks, i.e., $k \leq 16$.

For $i \in \{1, \dots, k\}$, let c_i denote the center of the disk D_i . In addition, let c'_i be the intersection point between $C(o, 2)$ and the ray with origin at o which passing through c_i . Let the point p_i be c_i , if $\|c_i\| < 2$, and c'_i , otherwise. See Figure 4.4. Finally, let $P' = \{o, u, v, p_1, \dots, p_k\}$.

OBSERVATION 4.12. Let c_j be the center of a disk D_j in $\mathcal{J}(e^+)$, where $\|c_j\| \geq 2$. Then, the disk $D(c_j, \|c_j\| - 1)$ is contained in the disk D_j . Moreover, the disk $D(p_j, 1)$ is contained in the disk $D(c_j, \|c_j\| - 1)$. See Figure 4.4.

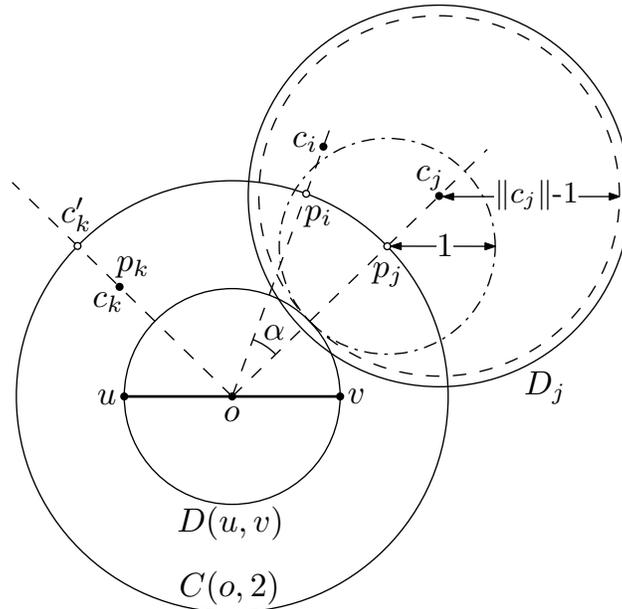


Figure 4.4: Proof of Lemma 4.13; $p_i = c'_i$, $p_j = c'_j$, and $p_k = c_k$.

The main idea in the proof of the following lemma is similar to the main idea in the proof of Lemma 2.8 in Chapter 2. Since the number of points and the constants involved in the calculations are different, for the sake of completeness we present a full proof for the following lemma.

LEMMA 4.13. *The distance between any pair of points in P' is at least 1.*

Proof. Let x and y be two points in P' . We are going to prove that $|xy| \geq 1$. We distinguish between the following three cases.

- $x, y \in \{o, u, v\}$. In this case the claim is trivial.
- $x \in \{o, u, v\}, y \in \{p_1, \dots, p_k\}$. If $\|y\| = 2$, then y is on $C(o, 2)$, and hence $|xy| \geq 1$. If $\|y\| < 2$, then y is the center of a disk D_i in $\mathcal{J}(e^+)$. By Observation 4.9, D_i does not contain u and v , and by Lemma 4.10, D_i does not contain o . Since D_i has radius at least 1, we conclude that $|xy| \geq 1$.
- $x, y \in \{p_1, \dots, p_k\}$. Without loss of generality assume $x = p_i$ and $y = p_j$, where $1 \leq i < j \leq k$. We differentiate between three cases:
 - $\|p_i\| < 2$ and $\|p_j\| < 2$. In this case p_i and p_j are the centers of D_i and D_j , respectively. By Lemma 4.10 and Observation 4.11, we conclude that $|p_i p_j| \geq 1$.
 - $\|p_i\| < 2$ and $\|p_j\| = 2$. By Observation 4.12 the disk $D(p_j, 1)$ is contained in the disk D_j . By Lemma 4.10, p_i is not in the interior of D_j , and consequently, it is not in the interior of $D(p_j, 1)$. Therefore, $|p_i p_j| \geq 1$.
 - $\|p_i\| = 2$ and $\|p_j\| = 2$. Recall that c_i and c_j are the centers of D_i and D_j , such that $\|c_i\| \geq 2$ and $\|c_j\| \geq 2$. Without loss of generality assume $\|c_i\| \leq \|c_j\|$. For the sake of contradiction assume that $|p_i p_j| < 1$. Then, for the angle $\alpha = \angle c_i o c_j$ we have $\sin(\alpha/2) < \frac{1}{4}$. Then, $\cos(\alpha) > 1 - 2\sin^2(\alpha/2) = \frac{7}{8}$. By the law of cosines in the triangle $\triangle c_i o c_j$, we have

$$|c_i c_j|^2 < \|c_i\|^2 + \|c_j\|^2 - \frac{14}{8} \|c_i\| \|c_j\|. \quad (4.1)$$

By Observation 4.12 the disk $D(c_j, \|c_j\| - 1)$ is contained in D_j ; see Figure 4.4. By Lemma 4.10, c_i is not in the interior of D_j , and consequently, is not in the interior of $D(c_j, \|c_j\| - 1)$. Thus, $|c_i c_j| \geq \|c_j\| - 1$. In combination with Inequality (4.1), this gives

$$\|c_j\| \left(\frac{14}{8} \|c_i\| - 2 \right) < \|c_i\|^2 - 1. \quad (4.2)$$

In combination with the assumption that $\|c_i\| \leq \|c_j\|$, Inequality (4.2) gives

$$\frac{6}{8} \|c_i\|^2 - 2 \|c_i\| + 1 < 0.$$

To satisfy this inequality, we should have $\|c_i\| < 2$, contradicting the fact that $\|c_i\| \geq 2$. This completes the proof. □

By Lemma 4.13, the points in P' has mutual distance 1. Moreover, the points in P' lie in (including the boundary) $C(o, 2)$. Bateman and Erdős [4] proved that it is impossible to have 20 points in (including the boundary) a circle of radius 2 such that one of the points is at the center and all of the mutual distances are at least 1. Therefore, P' contains at most 19 points, including o , u , and v . This implies that $k \leq 16$, and hence $\mathcal{J}(e^+)$ contains at most sixteen edges. This completes the proof of Lemma 4.8.

THEOREM 4.14. *Algorithm 2 computes a strong matching of size at least $\lceil \frac{n-1}{17} \rceil$ in $G_{\ominus}(P)$.*

4.5 STRONG MATCHING IN $G_{\nabla}(P)$

In this section we consider the case where S is a downward equilateral triangle ∇ , whose barycenter is the origin and one of its vertices is on the negative y -axis. In this section we assume that P is in general position, i.e., for each point $p \in P$, there is no point of $P \setminus \{p\}$ on l_p^0 , l_p^{60} , and l_p^{120} . In combination with Observation 4.1, this implies that for two points $p, q \in P$, no point of $P \setminus \{p, q\}$ are on the boundary of $t(p, q)$ (resp. $t'(p, q)$). Recall that $t(p, q)$ is the smallest homothet of ∇ having of p and q on a corner and the other point on the side opposite to that corner. We prove that $G_{\nabla}(P)$, and consequently $\frac{1}{2}\Theta_6(P)$, has a strong triangle matching of size at least $\lceil \frac{n-1}{9} \rceil$.

We run Algorithm 2 on $G_{\nabla}(P)$ to compute a matching \mathcal{M} . Recall that $G_{\nabla}(P)$ is an edge-weighted graph with the weight of each edge (p, q) is equal to the area of $t(p, q)$. By Theorem 4.7, \mathcal{M} is a strong matching of size at least $\lceil \frac{n-1}{\text{Inf}(T)} \rceil$, where T is a minimum spanning tree in $G_{\nabla}(P)$. In order to prove the desired lower bound, we show that $\text{Inf}(T) \leq 9$. Since $\text{Inf}(T)$ is the maximum size of a set among the influence sets of edges in T , it suffices to show that for every edge e in T , the influence set of e has at most nine edges.

LEMMA 4.15. *Let T be a minimum spanning tree of $G_{\nabla}(P)$, and let e be any edge in T . Then, $|\text{Inf}(e)| \leq 9$.*

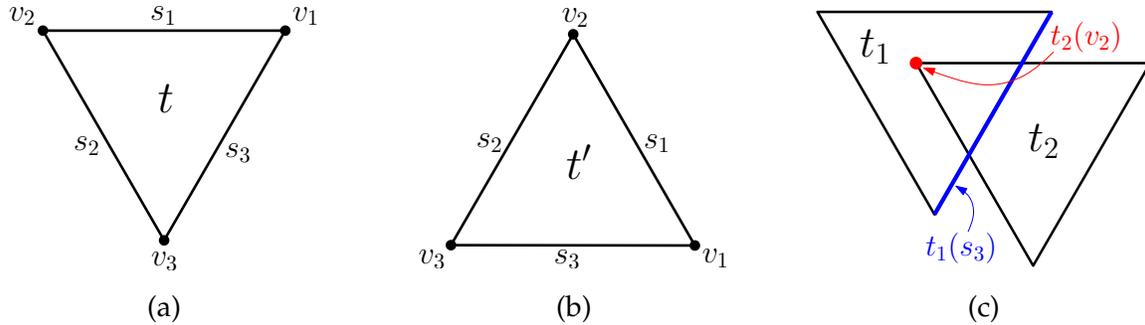


Figure 4.5: (a) Labeling the vertices and the sides of a downward triangle. (b) Labeling the vertices and the sides of an upward triangle. (c) Two intersecting triangles.

We will prove this lemma in the rest of this section. We label the vertices and the sides of a downward equilateral-triangle, t , and an upward equilateral-triangle, t' , as depicted in Figures 4.5(a) and 4.5(b). We refer to a vertex v_i and a side s_i of a triangle t by $t(v_i)$ and $t(s_i)$, respectively.

Recall that F is a subgraph of the minimum spanning tree T in $G_{\nabla}(P)$. In each iteration of the while loop in Algorithm 2, let \mathcal{T} denote the set of triangles representing the edges in F . By Lemma 4.4 and the general position assumption we have

OBSERVATION 4.16. *Each triangle $t(p, q)$ in \mathcal{T} does not contain any point of $P \setminus \{p, q\}$ in its interior or on its boundary.*

Consider two intersecting triangles $t_1(p_1, q_1)$ and $t_2(p_2, q_2)$ in \mathcal{T} . By Observation 4.1, each side of t_1 contains either p_1 or q_1 , and each side of t_2 contains either p_2 or q_2 . Thus, by Observation 4.16, we argue that no side of t_1 is completely in the interior of t_2 , and vice versa. Therefore, either exactly one vertex (corner) of t_1 is in the interior of t_2 , or exactly one vertex of t_2 is in the interior of t_1 . Without loss of generality assume that a corner of t_2 is in the interior of t_1 ,

as shown in Figure 4.5(c). In this case we say that t_1 intersects t_2 through the vertex $t_2(v_2)$, or symmetrically, t_2 intersects t_1 through the side $t_1(s_3)$.

The following two lemmas have been proved by Biniáz et al. [7]:

LEMMA 4.17 (Biniáz et al. [7]). *Let t_1 be a downward triangle which intersects a downward triangle t_2 through $t_2(s_1)$, and let a horizontal line ℓ intersects both t_1 and t_2 . Let p_1 and q_1 be two points on $t_1(s_2)$ and $t_1(s_3)$, respectively, which are above $t_2(s_1)$. Let p_2 and q_2 be two points on $t_2(s_2)$ and $t_2(s_3)$, respectively, which are above ℓ . Then, $\max\{t(p_1, p_2), t(q_1, q_2)\} \prec \max\{t_1, t_2\}$. See Figure 4.6(b).*

LEMMA 4.18 (Biniáz et al. [7]). *For every four triangles $t_1, t_2, t_3, t_4 \in \mathcal{T}$, $t_1 \cap t_2 \cap t_3 \cap t_4 = \emptyset$.*

As a consequence of Lemma 4.17, we have the following corollary:

COROLLARY 4.19. *Let t_1, t_2, t_3 be three triangles in \mathcal{T} . Then t_1, t_2 , and t_3 cannot make a chain configuration, such that t_2 intersects t_3 through $t_3(s_1)$ and t_1 intersects both t_2 and t_3 through $t_2(s_1)$ and $t_3(s_1)$. See Figure 4.6(b).*

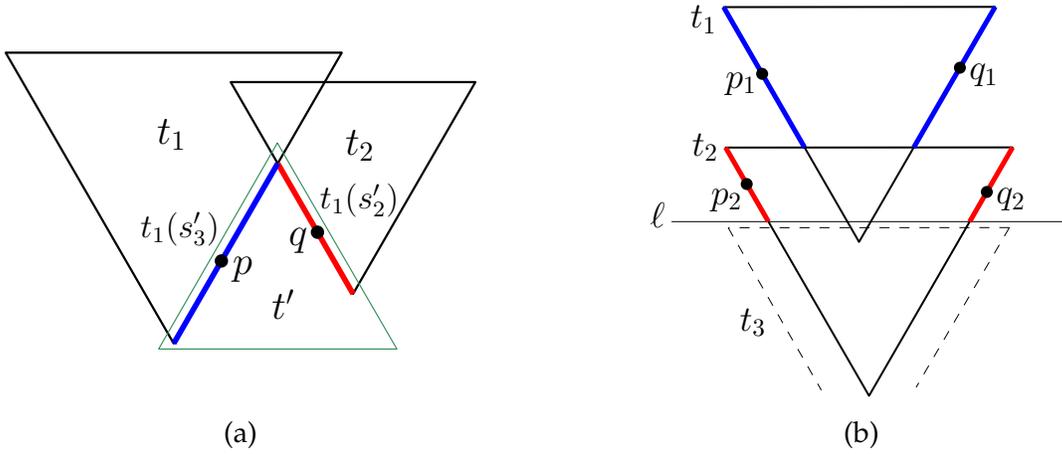


Figure 4.6: (a) Illustration of Lemma 4.20. (b) Illustration of Lemma 4.17.

LEMMA 4.20. *Let t_1 be a downward triangle which intersects a downward triangle t_2 through $t_2(v_2)$. Let p be a point on $t_1(s_3)$ and to the left of $t_2(s_2)$, and let q be a point on $t_2(s_2)$ and to the right of $t_1(s_3)$. Then, $t(p, q) \prec \max\{t_1, t_2\}$.*

Proof. Refer to Figure 4.6(a). Let $t_1(s'_3)$ be the part of the line segment $t_1(s_3)$ which is to the left of $t_2(s_2)$, and let $t_2(s'_2)$ be the part of the line segment $t_2(s_2)$ which is to the right of $t_1(s_3)$. Without loss of generality assume that $t_1(s'_3)$ is larger than $t_2(s'_2)$. Let t' be an upward triangle having $t_1(s'_3)$ as its left side. Then, $t' \prec t_1$, which implies that $t' \prec \max\{t_1, t_2\}$. Since t' has both p and q on its boundary, the area of the downward triangle $t(p, q)$ is smaller than the area of t' . Therefore, $t(p, q) \preceq t'$; which completes the proof. \square

Because of the symmetry, the statement of Lemma 4.20 holds even if p is above $t_2(s_1)$ and q is on $t_2(s_1)$. Consider the six cones with apex at p , as shown in Figure 4.3.

LEMMA 4.21. *Let T be a minimum spanning tree in $G_{\nabla}(P)$. Then, in T , every point p is adjacent to at most one point in each cone C_p^i , where $1 \leq i \leq 6$.*

Proof. If i is even, then by the construction of $G_{\nabla}(P)$, which is given in Section 4.2, p is adjacent to at most one point in C_p^i . Assume i is odd. For the sake of contradiction, assume in T , the point p is adjacent to two points q and r in a cone C_p^i . Then, $t(p, q)$ has q on a corner, and $t(p, r)$ has r on a corner. Without loss of generality assume $t(p, r) \prec t(q, r)$. Then, the hexagon $X(q, p)$ has r in its interior. Thus, $t(q, r) \prec t(p, q)$. Then the cycle r, p, q, r contradicts Lemma 4.5. Therefore, p is adjacent to at most one point in each of the six cones. \square

In Algorithm 2, in each iteration of the while loop, let $\mathcal{T}(e^+)$ be the triangles representing the edges of F . Recall that e is the smallest edge in F , and hence, $t(e)$ is a smallest triangle in $\mathcal{T}(e^+)$. Let $e = (p, q)$ and let $\mathcal{J}(e^+)$ be the set of triangles in $\mathcal{T}(e^+)$ (excluding $t(e)$) which intersect $t(e)$. We show that $\mathcal{J}(e^+)$ contains at most eight triangles. We partition the triangles in $\mathcal{J}(e^+)$ into $\{\mathcal{J}_1, \mathcal{J}_2\}$, such that every triangle $\tau \in \mathcal{J}_1$ shares only p or q with $t = t(e)$, i.e., $\mathcal{J}_1 = \{\tau : \tau \in \mathcal{J}(e^+), \tau \cap t \in \{p, q\}\}$, and every triangle $\tau \in \mathcal{J}_2$ intersects t either through a side or through corner which is not p nor q .

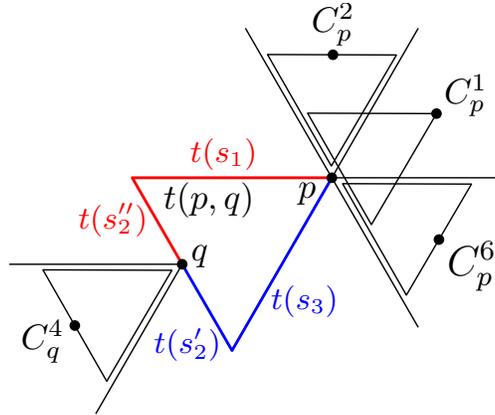


Figure 4.7: Illustration of the triangles in \mathcal{J}_1 .

By Observation 4.1, for each triangle $t(p, q)$, one of p and q is on a corner of $t(p, q)$ and the other one is on the side opposite to that corner. Without loss of generality assume that p is on the corner $t(v_1)$, and hence, q is on the side $t(s_2)$. See Figure 4.7. Note that the other cases, where p is on $t(v_2)$ or on $t(v_3)$ are similar. Since the intersection of t with any triangle $\tau \in \mathcal{J}_1$ is either p or q , τ has either p or q on its boundary. In combination with Observation 4.16, this implies that τ represent an edge e' in T , and hence, either p or q is an endpoint of e' . As illustrated in Figure 4.7, the other endpoint of e' can be either in C_p^1, C_p^2, C_p^6 , or in C_q^4 , because otherwise $\tau \cap t \notin \{p, q\}$. By Lemma 4.21, p has at most one neighbor in each of C_p^1, C_p^2, C_p^6 , and q has at most one neighbor in C_q^4 . Therefore, \mathcal{J}_1 contains at most four triangles. We are going to show that \mathcal{J}_2 also contains at most four triangles.

The point q divides $t(s_2)$ into two parts. Let $t(s_2')$ and $t(s_2'')$ be the parts of $t(s_2)$ which are below and above q , respectively; see Figure 4.7. The triangles in \mathcal{J}_2 intersect t either through $t(s_1) \cup t(s_2'')$ or through $t(s_3) \cup t(s_2')$; which are shown by red and blue polylines in Figure 4.7. We show that most two triangles in \mathcal{J}_2 intersect t through each of $t(s_1) \cup t(s_2'')$ or $t(s_3) \cup t(s_2')$. Because of symmetry, we only prove for $t(s_3) \cup t(s_2')$. When a triangle t' intersects t through both $t(s_3)$ and $t(s_2')$ we say t' intersects t through $t(v_3)$. In the next lemma, we prove that at most one triangle in \mathcal{J}_2 intersects t through each of $t(s_3), t(s_2')$. Again, because of symmetry, we only prove for $t(s_3)$.

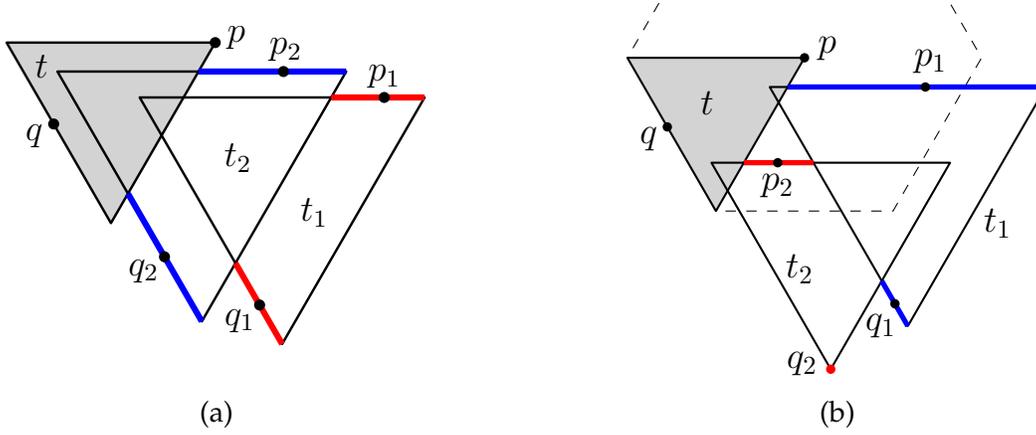


Figure 4.8: Illustration of Lemma 4.22: (a) $t_1(v_2) \in t_2$. (b) $t_1(v_2) \notin t_2$ and $t_2(v_2) \notin t_1$.

LEMMA 4.22. *At most one triangle in \mathcal{J}_2 intersects t through $t(s_3)$.*

Proof. The proof is by contradiction. Assume two triangles $t_1(p_1, q_1)$ and $t_2(p_2, q_2)$ in \mathcal{J}_2 intersect t through $t(s_3)$. Without loss of generality assume that p_i is on $t_i(s_1)$ and q_i is on $t_i(s_2)$ for $i = 1, 2$. Recall that the area of t_1 and the area of t_2 are at least the area of t . If $t_1(v_2)$ is in the interior of t_2 (as shown in Figure 4.8(a)) or $t_2(v_2)$ is in the interior of t_1 , then we get a contradiction to Corollary 4.19. Thus, assume that $t_1(v_2) \notin t_2$ and $t_2(v_2) \notin t_1$.

Without loss of generality assume that $t_1(s_1)$ is above $t_2(s_1)$; see Figure 4.8(b). By Lemma 4.20, we have $t(p, p_1) \prec \max\{t, t_1\} \preceq t_1$. If q_1 is in $X(p, q)$, then by Observation 4.6, $t(p, q_1) \prec t$. Then, the cycle p, p_1, q_1, p contradicts Lemma 4.5. Thus, assume that $q_1 \notin X(p, q)$. In this case $t_2(s_3)$ is to the left of $t_1(s_3)$, because otherwise q_1 lies in t_2 which contradicts Observation 4.16. Since both t_1 and t_2 are larger than t , t_2 intersects t_1 through $t_1(s_2)$, and hence $t_2(v_1)$ is in the interior of t_1 . This implies that q_2 is on $t_2(v_3)$. In addition, p_2 is on the part of $t_2(s_1)$ which lies in the interior of $X(p, q)$. By Observation 4.6 and Lemma 4.20, we have $t(p, p_2) \prec t$ and $t(q_1, q_2) \prec \max\{t_1, t_2\}$, respectively. Thus, the cycle p, p_1, q_1, q_2, p_2, p contradicts Lemma 4.5. \square

LEMMA 4.23. *At most two triangles in \mathcal{J}_2 intersect t through $t(v_3)$.*

Proof. For the sake of contradiction assume three triangles $t_1, t_2, t_3 \in \mathcal{J}_2$ intersect t through $t(v_3)$. This implies that $t(v_3)$ belongs to four triangles t, t_1, t_2, t_3 , which contradicts Lemma 4.18. \square

LEMMA 4.24. *If two triangles in \mathcal{J}_2 intersect t through $t(v_3)$, then no other triangle in \mathcal{J}_2 intersects t through $t(s_3)$ or through $t(s'_2)$.*

Proof. The proof is by contradiction. Assume two triangles $t_1(p_1, q_1)$ and $t_2(p_2, q_2)$ in \mathcal{J}_2 intersect t through $t(v_3)$, and a triangle $t_3(p_3, q_3)$ in \mathcal{J}_2 intersects t through $t(s_3)$ or $t(s'_2)$. Let p_i be the point which lies on $t_i(s_1)$ for $i = 1, 2, 3$. By Lemma 4.23, t_3 cannot intersect both $t(s_3)$ and $t(s'_2)$. Thus, t_3 intersects t either through $t(s_3)$ or through $t(s'_2)$. We prove the former case; the proof for the latter case is similar. Assume that t_3 intersects t through $t(s_3)$. By Lemma 4.20, $t(p, p_3) \prec t_3$. See Figure 4.9. In addition, both $t_1(s_3)$ and $t_2(s_3)$ are to the left of $t_3(s_3)$, because otherwise q_3 lies in $t_1 \cup t_2 \cup X(p, q)$. If $q_3 \in t_1 \cup t_2$ we get a contradiction to Observation 4.16. If $q_3 \in X(p, q)$ then by Observation 4.6, we have $t(p, q_3) \prec t$, and hence, the cycle p, p_3, q_3, p contradicts Lemma 4.5.

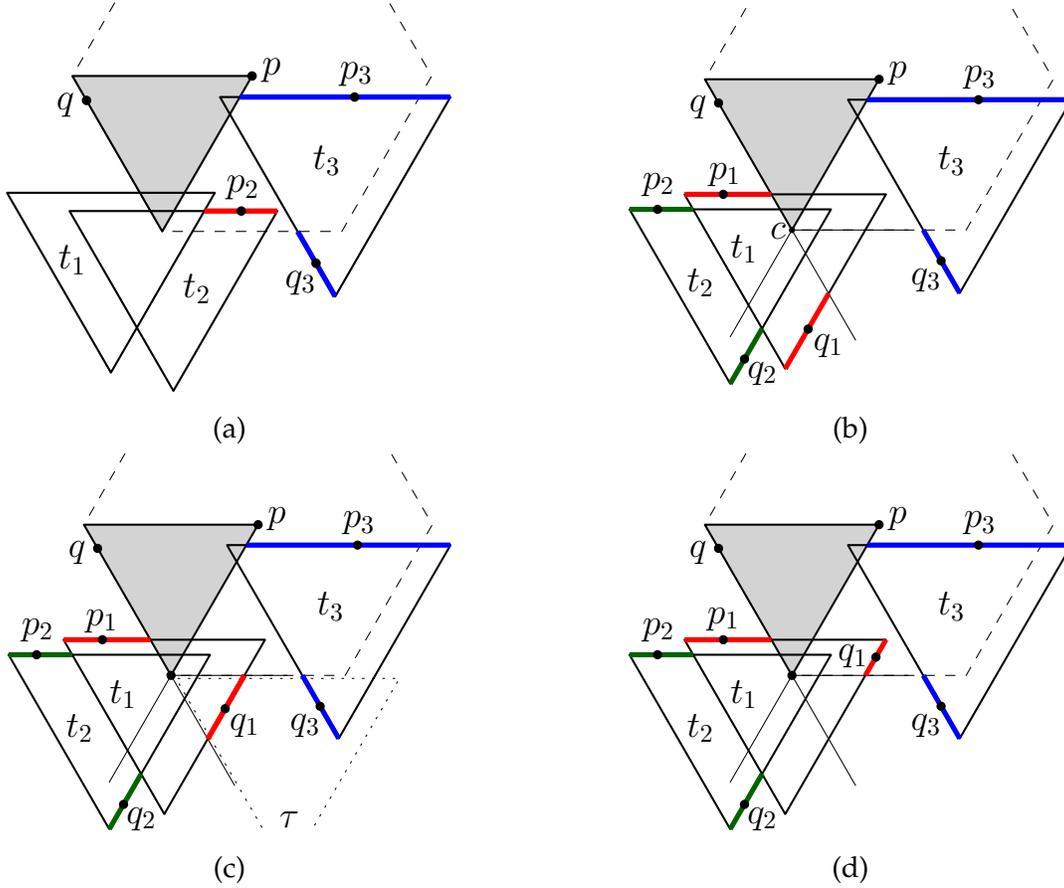


Figure 4.9: Illustration of Lemma 4.24: (a) p_2 is to the right of $t_1(s_3)$, (b) $q_1 \in C^5_{t(v_3)}$, (c) $q_1 \in C^6_{t(v_3)}$, and (d) $q_1 \in C^1_{t(v_3)}$.

Without loss of generality assume that $t_1(s_1)$ is above $t_2(s_1)$; see Figure 4.9. If $t_1(v_3)$ is in t_2 or $t_2(v_3)$ is in t_1 , then we get a contradiction to Corollary 4.19. Thus, assume that $t_1(v_3) \notin t_2$ and $t_2(v_3) \notin t_1$. This implies that either (i) $t_2(s_3)$ is to the right of $t_1(s_3)$ or (ii) $t_2(s_2)$ is to the left of $t_1(s_2)$. We show that both cases lead to a contradiction.

In case (i), p_2 lies in the interior of $X(p, q)$, and then by Observation 4.6, we have $t(p, p_2) \prec t$; see Figure 4.9(a). In addition, Lemma 4.20 implies that $t(p_2, q_3) \prec \max\{t, t_3\} \preceq t_3$. Thus, the cycle p, p_3, q_3, p_2, p contradicts Lemma 4.5.

Now consider case (ii) where $t_1(s_1)$ is above $t_2(s_1)$ and $t_2(s_2)$ is to the left of $t_1(s_2)$. If p_1 is to the right of t , then as in case (i), the cycle p, p_3, q_3, p_1, p contradicts Lemma 4.5. Thus, assume that p_1 is to the left of t , as shown in Figure 4.9(b). By Lemma 4.20, we have $t(q, p_1) \prec \max\{t, t_1\} \preceq t_1$. Each side of t_1 contains either p_1 or q_1 , while p_1 is on the part of $t_1(s_1)$ which is to the left of t , thus, q_1 is on $t_1(s_3)$. Consider the six cones around $t(v_3)$; see Figure 4.9(b). We have three cases: (a) $q_1 \in C^5_{t(v_3)}$, (b) $q_1 \in C^6_{t(v_3)}$ or (c) $q_1 \in C^1_{t(v_3)}$.

In case (a), which is shown in Figure 4.9(b), by Lemma 4.17, we have $\max\{t(p_1, p_2), t(q_1, q_2)\} \prec \max\{t_1, t_2\}$. Thus, the cycle p_1, p_2, q_2, q_1, p_1 contradicts Lemma 4.5. In Case (b), which is shown in Figure 4.9(c), we have $t(q_1, q_3) \prec t_3$, because if we map t_3 to a downward triangle τ —of area equal to the area of t_3 —which has $\tau(v_2)$ on $t(v_3)$, then τ contains both q_1 and q_3 . Therefore, the

cycle $p, p_3, q_3, q_1, p_1, q, p$ contradicts Lemma 4.5. In Case (c), which is shown in Figure 4.9(d), by Observation 4.6, $t(p, q_1) \prec t$, and then, the cycle p, q_1, p_1, q, p contradicts Lemma 4.5. \square

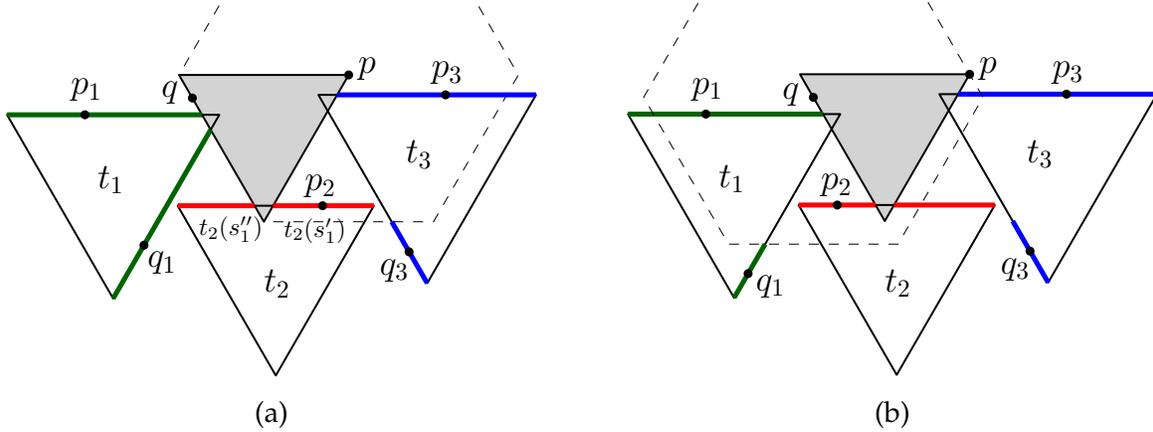


Figure 4.10: Illustration of Lemma 4.25: (a) $p_2 \in t_2(s'_1)$, and (b) $p_2 \in t_2(s''_1)$.

LEMMA 4.25. *If three triangles intersect t through $t(s'_2)$, $t(v_3)$ and $t(s_3)$. Then, at least one of the three triangles is not in \mathcal{J}_2 .*

Proof. The proof is by contradiction. Assume that three triangles $t_1(p_1, q_1)$, $t_2(p_2, q_2)$, $t_3(p_3, q_3)$ in \mathcal{J}_2 intersect t through $t(s'_2)$, $t(v_3)$, $t(s_3)$, respectively. Let p_i be the point which lies on $t_i(s_1)$ for $i = 1, 2, 3$. See Figure 4.10(a). By Lemma 4.20, we have $t(p, p_3) \prec t_3$ and $t(q, p_1) \prec t_1$. If q_3 is in the interior of $X(p, q)$, then by Observation 4.6, $t(p, q_3) \prec t$, and hence, the cycle p, p_3, q_3, p contradicts Lemma 4.5. If q_1 is in $X(q, p)$, then by Observation 4.6, $t(q, q_1) \prec t$, and hence, the cycle q, q_1, p_1, q contradicts Lemma 4.5; see Figure 4.10(b). Thus, assume that $q_3 \notin X(p, q)$ and $q_1 \notin X(q, p)$. Let $t_2(s'_1)$ and $t_2(s''_1)$ be the parts of $t_2(s_1)$ which are to the right of $t(s_3)$ and to the left of $t(s_2)$, respectively. Consider the point p_2 which lies on $t_2(s_1)$. If $p_2 \in t_2(s'_1)$, then $p_2 \in X(p, q)$ and by Observation 4.6, $t(p, p_2) \prec t$. In addition, Lemma 4.20 implies that $t(p_2, q_3) \prec t_3$. Thus, the cycle p, p_3, q_3, p_2, p contradicts Lemma 4.5; see Figure 4.10(a). If $p_2 \in t_2(s''_1)$, then $p_2 \in X(q, p)$ and by Observation 4.6, $t(q, p_2) \prec t$. In addition, Lemma 4.20 implies that $t(p_2, q_1) \prec t_2$. Thus, the cycle q, p_2, q_1, p_1, q contradicts Lemma 4.5; see Figure 4.10(b). \square

Putting Lemmas 4.22, 4.23, 4.24, and 4.25 together, implies that at most two triangles in \mathcal{J}_2 intersect t through $t(s_3) \cup t(s'_2)$, and consequently, at most two triangles in \mathcal{J}_2 intersect t through $t(s_1) \cup t(s''_2)$. Thus, \mathcal{J}_2 contains at most four triangles. Recall that \mathcal{J}_1 contains at most four triangles. Then, $\mathcal{J}(e^+)$ has at most eight triangles. Therefore, the influence set of e , contains at most 9 edges (including e itself). This completes the proof of Lemma 4.15.

THEOREM 4.26. *Algorithm 2 computes a strong matching of size at least $\lceil \frac{n-1}{9} \rceil$ in $G_{\nabla}(P)$.*

The bound obtained by Lemma 4.15 is tight. Figure 4.11 shows a configuration of 10 points in general position such that the influence set of a minimal edge is 9. In Figure 4.11, $t = t(p, q)$ represents a smallest edge of weight 1; the minimum spanning tree is shown in bold-green line segments. The weight of all edges—the area of the triangles representing these edges—is at least 1. The red triangles are in \mathcal{J}_1 and share either p or q with t . The blue triangles are in \mathcal{J}_2 and intersect t through $t(s_1) \cup t(s''_2)$ or through $t(s_3) \cup t(s'_2)$; as show in Figure 4.11, two of them share only the points $t(v_2)$ and $t(v_3)$.

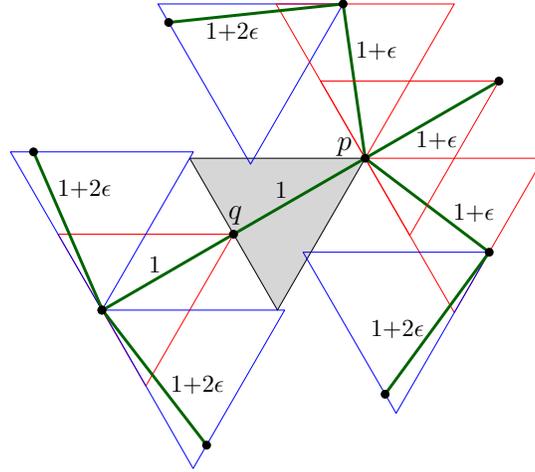


Figure 4.11: Four triangles in \mathcal{J}_1 (in red) and four triangles in \mathcal{J}_2 (in blue) intersect with $t(p, q)$.

4.6 STRONG MATCHING IN $G_{\nabla\triangle}(p)$

In this section we consider the problem of computing a strong matching in $G_{\nabla\triangle}(P)$. Recall that $G_{\nabla\triangle}(P)$ is the union of $G_{\nabla}(P)$ and $G_{\triangle}(P)$, and is equal to the graph $\Theta_6(P)$. We assume that P is in general position, i.e., for each point $p \in P$, there is no point of $P \setminus \{p\}$ on l_p^0 , l_p^{60} , and l_p^{120} . A matching \mathcal{M} in $G_{\nabla\triangle}(P)$ is a strong matching if for each edge e in \mathcal{M} there is a homothet of ∇ or a homothet of \triangle representing e , such that these homothets are pairwise disjoint. See Figure 4.1(b). Using a similar approach as in [2], we prove the following theorem:

THEOREM 4.27. *Let P be a set of n points in general position in the plane. Let S be an upward or a downward equilateral-triangle that contains P . Then, it is possible to find a strong matching of size at least $\lceil \frac{n-1}{4} \rceil$ for $G_{\nabla\triangle}(P)$ in S .*

Proof. The proof is by induction. Assume that any point set of size $n' \leq n - 1$ in a triangle S' , has a strong matching of size $\lceil \frac{n'-1}{4} \rceil$ in S' . Without loss of generality, assume S is an upward equilateral-triangle. If n is 0 or 1, then there is no matching in S , and if $n \in \{2, 3, 4, 5\}$, then by shrinking S , it is possible to find a strongly matched pair; the statement of the theorem holds. Suppose that $n \geq 6$, and $n = 4m + r$, where $r \in \{0, 1, 2, 3\}$. If $r \in \{0, 1, 3\}$, then $\lceil \frac{n-1}{4} \rceil = \lceil \frac{(n-1)-1}{4} \rceil$, and by induction we are done. Suppose that $n = 4m + 2$, for some $m \geq 1$. We prove that there are $\lceil \frac{n-1}{4} \rceil = m + 1$ disjoint equilateral-triangles (upward or downward) in S , each of them matches a pair of points in P . Partition S into four equal area equilateral triangles S_1, S_2, S_3, S_4 containing n_1, n_2, n_3, n_4 points, respectively; see Figure 4.12(a). Let $n_i = 4m_i + r_i$, where $r_i \in \{0, 1, 2, 3\}$. By induction, in $S_1 \cup S_2 \cup S_3 \cup S_4$, we have a strong matching of size at least

$$A = \left\lceil \frac{n_1 - 1}{4} \right\rceil + \left\lceil \frac{n_2 - 1}{4} \right\rceil + \left\lceil \frac{n_3 - 1}{4} \right\rceil + \left\lceil \frac{n_4 - 1}{4} \right\rceil. \tag{4.3}$$

CLAIM 1. $A \geq m$.

Proof. By Equation (4.3), we have

$$A = \sum_{i=1}^4 \left\lceil \frac{n_i - 1}{4} \right\rceil \geq \sum_{i=1}^4 \frac{n_i - 1}{4} = \frac{n}{4} - 1 = \frac{4m + 2}{4} - 1 = m - \frac{1}{2}.$$

Since A is an integer, we argue that $A \geq m$. \square

If $A > m$, then we are done. Assume that $A = m$; in fact, by the induction hypothesis we have an strong matching of size m for P . In order to complete the proof, we have to get one more strongly matched pair. Let R be the multiset $\{r_1, r_2, r_3, r_4\}$.

CLAIM 2. *If $A = m$, then either (i) one element in R is equal to 3 and the other elements are equal to 1, or (ii) two elements in R are equal to 0 and the other elements are equal to 1.*

Proof. Let $\alpha = r_1 + r_2 + r_3 + r_4$, where $0 \leq r_i \leq 3$. Then $n = 4(m_1 + m_2 + m_3 + m_4) + \alpha$. Since $n = 4m + 2$, $\alpha = 4k + 2$, for some $0 \leq k \leq 2$. Thus, $n = 4(m_1 + m_2 + m_3 + m_4 + k) + 2$, where $m = m_1 + m_2 + m_3 + m_4 + k$.

By induction, in S_i , we get a matching of size at least $\lceil \frac{(4m_i + r_i) - 1}{4} \rceil = m_i + \lceil \frac{r_i - 1}{4} \rceil$. Hence, in $S_1 \cup S_2 \cup S_3 \cup S_4$, we get a matching of size at least

$$A = m_1 + m_2 + m_3 + m_4 + \left\lceil \frac{r_1 - 1}{4} \right\rceil + \left\lceil \frac{r_2 - 1}{4} \right\rceil + \left\lceil \frac{r_3 - 1}{4} \right\rceil + \left\lceil \frac{r_4 - 1}{4} \right\rceil.$$

Since $A = m$ and $m = m_1 + m_2 + m_3 + m_4 + k$, we have

$$k = \left\lceil \frac{r_1 - 1}{4} \right\rceil + \left\lceil \frac{r_2 - 1}{4} \right\rceil + \left\lceil \frac{r_3 - 1}{4} \right\rceil + \left\lceil \frac{r_4 - 1}{4} \right\rceil. \quad (4.4)$$

Note that $0 \leq k \leq 2$. We go through some case analysis: (i) $k = 0$, (ii) $k = 1$, (iii) $k = 2$. In case (i), we have $\alpha = 4k + 2 = r_1 + r_2 + r_3 + r_4 = 2$. In order to have k equal to 0 in Equation (4.4), no element in R should be more than 1; this happens only if two elements in R are equal to 0 and the other two elements are equal to 1. In case (ii), we have $\alpha = r_1 + r_2 + r_3 + r_4 = 6$. In order to have k equal to 1 in Equation (4.4), at most one element in R should be greater than 1; this happens only if three elements in R are equal to 1 and the other element is equal to 3 (note that all elements in R are smaller than 4). In case (iii), we have $\alpha = r_1 + r_2 + r_3 + r_4 = 10$. In order to have k equal to 2 in Equation (4.4), at most two elements in R should be greater than 1; which is not possible. \square

We show how to find one more matched pair in each case of Claim 2.

We define S_1^{-x} as the smallest upward equilateral-triangle contained in S_1 and anchored at the top corner of S_1 , which contains all the points in S_1 except x points. If S_1 contains less than x points, then the area of S_1^{-x} is zero. We also define S_1^{+x} as the smallest upward equilateral-triangle that contains S_1 and anchored at the top corner of S_1 , which has all the points in S_1 plus x other points of P . Similarly we define upward triangles S_2^{-x} and S_2^{+x} which are anchored at the left corner of S_2 . Moreover, we define upward triangles S_4^{-x} and S_4^{+x} which are anchored at the right corner of S_4 . We define downward triangles S_{3l}^{-x} , S_{3r}^{-x} , S_{3b}^{-x} which are anchored at the top-left corner, top-right corner, and bottom corner of S_3 , respectively. See Figure 4.12(a).

Case 1: One element in R is equal to 3 and the other elements are equal to 1.

In this case, we have $m = m_1 + m_2 + m_3 + m_4 + 1$. Because of the symmetry, we have two cases: (i) $r_3 = 3$, (ii) $r_j = 3$ for some $j \in \{1, 2, 4\}$.

- $r_3 = 3$.

In this case $n_3 = 4m_3 + 3$. We differentiate between two cases, where all the elements of the multiset $\{m_1, m_2, m_4\}$ are equal to zero, or some of them are greater than zero.

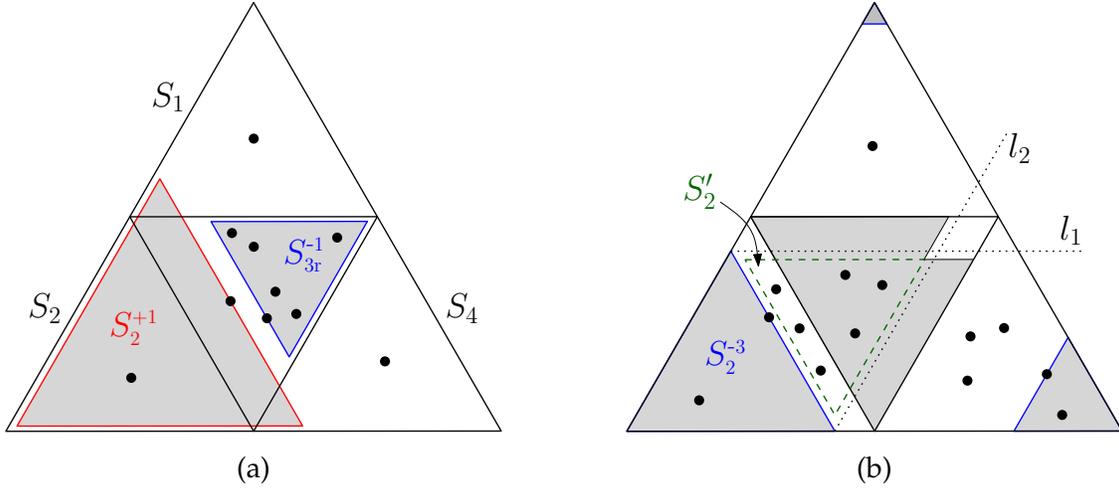


Figure 4.12: (a) Split S into four equal area triangles. (b) S_2^{-3} is larger than S_1^{-3} and S_4^{-3} .

- All elements of $\{m_1, m_2, m_4\}$ are equal zero. In this case, we have $m = m_3 + 1$. Consider the triangles S_2^{+1} and S_{3r}^{-1} . See Figure 4.12(a). Note that S_2^{+1} and S_{3r}^{-1} are disjoint, S_2^{+1} contains two points, and S_{3r}^{-1} contains $4m_3 + 2$ points. By induction, we get a matched pair in S_2^{+1} and a matching of size at least $m_3 + 1$ in S_{3r}^{-1} . Thus, in total, we get a matching of size at least $1 + (m_3 + 1) = m + 1$ in S .
- Some elements of $\{m_1, m_2, m_4\}$ are greater than zero. Consider the triangles S_1^{-3}, S_2^{-3} , and S_4^{-3} . Note that the area of some of these triangles—but not all—may be equal to zero. See Figure 4.12(b). By induction, we get matchings of size m_1, m_2 , and m_4 in S_1^{-3}, S_2^{-3} , and S_4^{-3} , respectively. Without loss of generality, assume S_2^{-3} is larger than S_1^{-3} and S_4^{-3} . Consider the half-lines l_1 and l_2 which are parallel to l^0 and l^{60} axis, and have their endpoints on the top corner and right corner of S_2^{-3} , respectively. We define S_2' as the downward equilateral-triangle which is bounded by l_1, l_2 , and the right side of S_2^{-3} ; the dashed triangle in Figure 4.12(b). Note that l_1 and l_2 do not intersect S_1^{-3} and S_4^{-3} . In addition, $S_1^{-3}, S_2^{-3}, S_4^{-3}$, and S_2' are pairwise disjoint. If any point of $S_1 \cup S_2 \cup S_3$ is to the right of l_2 , then consider S_4^{+1} and S_{3l}^{-1} . By induction, we get a matching of size $m_1 + m_2 + (m_3 + 1) + (m_4 + 1)$ in $S_1^{-3} \cup S_2^{-3} \cup S_{3l}^{-1} \cup S_4^{+1}$, and hence a matching of size $m + 1$ in S . If any point of $S_2 \cup S_3 \cup S_4$ is above l_1 , then consider S_1^{+1} and S_{3b}^{-1} . By induction, we get a matching of size $(m_1 + 1) + m_2 + (m_3 + 1) + m_4$ in $S_1^{+1} \cup S_2^{-3} \cup S_{3b}^{-1} \cup S_4^{-3}$, and hence a matching of size $m + 1$ in S . Otherwise, S_2' contains $n_3 + 3 = 4(m_3 + 1) + 2$ points. Thus, by induction, we get a matching of size $m_1 + m_2 + (m_3 + 2) + m_4$ in $S_1 \cup S_2^{-3} \cup S_2' \cup S_4$, and hence a matching of size $m + 1$ in S .

- $r_j = 3$, for some $j \in \{1, 2, 4\}$.

Without loss of generality, assume that $r_j = r_2$. Then, $n_2 = 4m_2 + 3$. Consider the triangles S_1^{-3}, S_2^{-1} , and S_4^{-3} . See Figure 4.13(a). By induction, we get matchings of size $m_1, m_2 + 1$, and m_4 in S_1^{-3}, S_2^{-1} , and S_4^{-3} , respectively. Now we consider the largest triangle among S_1^{-3}, S_2^{-1} , and S_4^{-3} . Because of the symmetry, we have two cases: (i) S_2^{-1} is the largest, or (ii) S_4^{-3} is the largest.

- S_2^{-1} is larger than S_1^{-3} and S_4^{-3} . Define the half-lines l_1, l_2 , and the triangle S_2' as in the previous case. See Figure 4.13(a). If any point of $S_1 \cup S_2 \cup S_3$ is to the right of l_2 , then

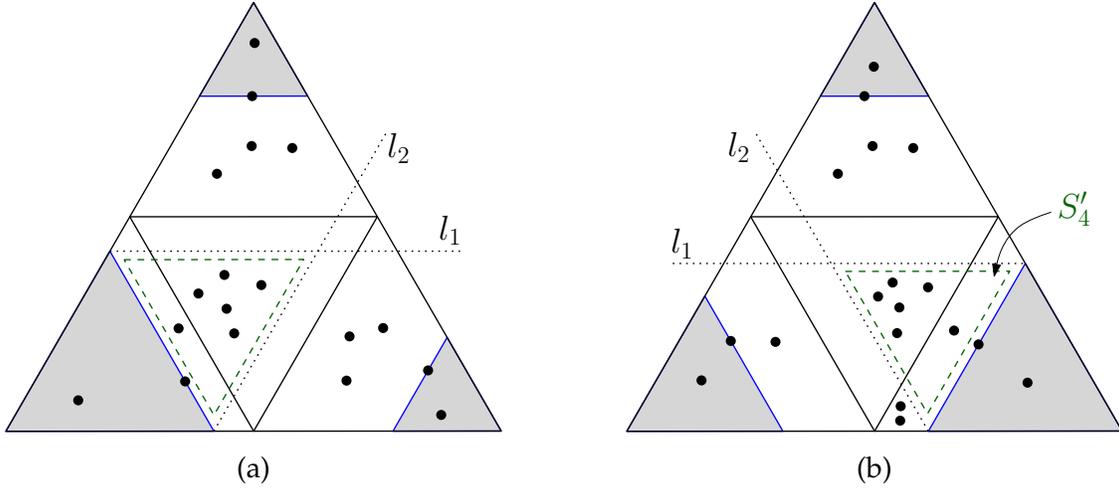


Figure 4.13: (a) S_2^{-1} is larger than S_1^{-3} and S_4^{-3} . (b) S_4^{-3} is larger than S_1^{-3} and S_2^{-1} .

consider S_4^{+1} and S_{3l}^{-1} . By induction, we get a matching of size $m_1 + (m_2 + 1) + m_3 + (m_4 + 1)$ in $S_1^{-3} \cup S_2^{-1} \cup S_{3l}^{-1} \cup S_4^{+1}$. If any point of $S_2 \cup S_3 \cup S_4$ is above l_1 , then consider S_1^{+1} and S_{3b}^{-1} . By induction, we get a matching of size $(m_1 + 1) + (m_2 + 1) + m_3 + m_4$ in $S_1^{+1} \cup S_2^{-1} \cup S_{3b}^{-1} \cup S_4^{-3}$. Otherwise, S_2' contains $n_3 + 1 = 4m_3 + 2$ points. Thus, by induction, we get a matching of size $m_1 + (m_2 + 1) + (m_3 + 1) + m_4$ in $S_1 \cup S_2^{-1} \cup S_2' \cup S_4$. As a result, in all cases we get a matching of size $m + 1$ in S .

- S_4^{-3} is larger than S_1^{-3} and S_2^{-1} . Define the half-lines l_1 , l_2 , and the triangle S_4' as in Figure 4.13(b). If any point of $S_1 \cup S_3 \cup S_4$ is above l_1 , then by induction, we get a matching of size $(m_1 + 1) + (m_2 + 1) + m_3 + m_4$ in $S_1^{+1} \cup S_2^{-1} \cup S_{3b}^{-1} \cup S_4^{+3}$. If at least three points of $S_1 \cup S_3 \cup S_4$ are to the left of l_2 , then consider S_2^{+3} and S_{3r}^{-3} . Note that S_2^{+3} contains $n_2 + 3 = 4(m_2 + 1) + 2$ points. By induction, we get a matching of size $m_1 + (m_2 + 2) + m_3 + m_4$ in $S_1^{-3} \cup S_2^{+3} \cup S_{3r}^{-3} \cup S_4^{-3}$. Otherwise, S_4' contains at least $n_3 + 1 = 4m_3 + 2$ points. Thus, by induction, we get a matching of size $m_1 + (m_2 + 1) + (m_3 + 1) + m_4$ in $S_1 \cup S_2 \cup S_4' \cup S_4^{-3}$. As a result, in all cases we get a matching of size $m + 1$ in S .

Case 2: Two elements in R are equal to 0 and the other elements are equal to 1.

In this case, we have $m = m_1 + m_2 + m_3 + m_4$. Again, because of the symmetry, we have two cases: (i) $r_3 = 0$, (ii) $r_3 \neq 0$.

- $r_3 = 0$.

Without loss of generality assume that $r_2 = 0$ and $r_1 = r_4 = 1$. Thus, $n_1 = 4m_1 + 1$, $n_2 = 4m_2$, $n_3 = 4m_3$, and $n_4 = 4m_4 + 1$. If all elements of $\{m_1, m_2, m_4\}$ are equal to zero, then we have $m = m_3$, where $m_3 \geq 1$. Consider the triangles S_4^{+1} and S_{3l}^{-1} , which are disjoint. By induction, we get a matched pair in S_4^{+1} and a matching of size at least m_3 in S_{3l}^{-1} . Thus, in total, we get a matching of size at least $1 + m_3 = m + 1$ in S . Assume some elements in $\{m_1, m_2, m_4\}$ are greater than zero. Consider the triangles S_1^{-3} , S_2^{-2} , and S_4^{-3} . See Figure 4.14(a). By induction, we get a matching of size m_1 , m_2 , and m_4 in S_1^{-3} , S_2^{-2} , and S_4^{-3} , respectively. Now we consider the largest triangle among S_1^{-3} , S_2^{-2} , and S_4^{-3} . Because of the symmetry, we have two cases: (i) S_2^{-2} is the largest, or (ii) S_4^{-3} is the largest.

- S_2^{-2} is larger than S_1^{-3} and S_4^{-3} . Define l_1 , l_2 , S_2' as in Figure 4.14(a). If any point of $S_1 \cup S_2 \cup S_3$ is to the right of l_2 , then by induction, we get a matching of size $m_1 + m_2 + m_3 +$

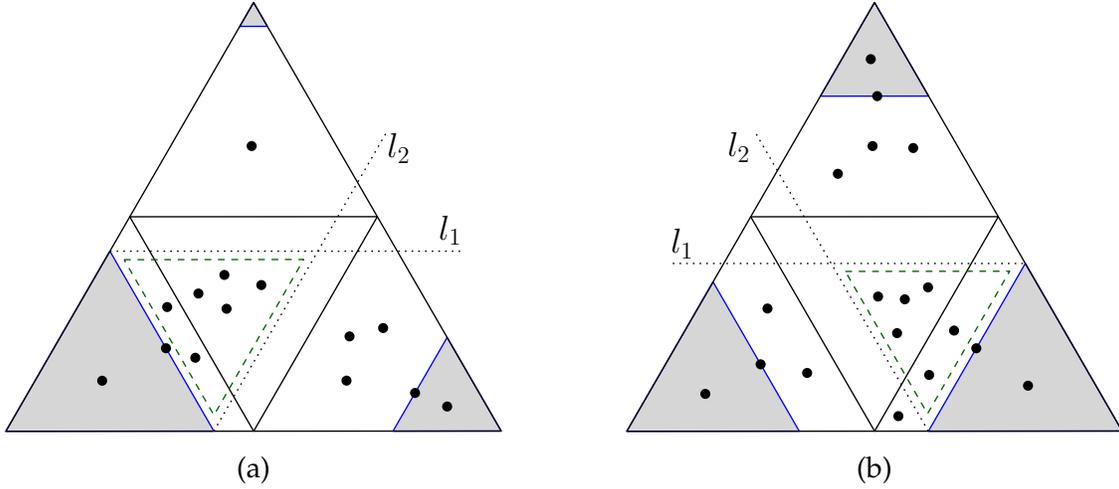


Figure 4.14: (a) S_2^{-2} is larger than S_1^{-3} and S_4^{-3} . (b) S_4^{-3} is larger than S_1^{-3} and S_2^{-2} .

($m_4 + 1$) in $S_1^{-3} \cup S_2^{-2} \cup S_{3l}^{-1} \cup S_4^{+1}$. If any point of $S_2 \cup S_3 \cup S_4$ is above l_1 , then by induction, we get a matching of size $(m_1 + 1) + m_2 + m_3 + m_4$ in $S_1^{+1} \cup S_2^{-2} \cup S_{3b}^{-1} \cup S_4^{-3}$. Otherwise, S_2' contains $n_3 + 2 = 4m_3 + 2$ points. Thus, by induction, we get a matching of size $m_1 + m_2 + (m_3 + 1) + m_4$ in $S_1 \cup S_2^{-2} \cup S_2' \cup S_4$. In all cases we get a matching of size $m + 1$ in S .

- S_4^{-3} is larger than S_1^{-3} and S_2^{-2} . Define l_1, l_2, S_4' as in Figure 4.14(b). If any point of $S_1 \cup S_3 \cup S_4$ is above l_1 , then by induction, we get a matching of size $(m_1 + 1) + m_2 + m_3 + m_4$ in $S_1^{+1} \cup S_2^{-2} \cup S_{3b}^{-1} \cup S_4^{+3}$. If at least two points of $S_1 \cup S_3 \cup S_4$ are to the left of l_2 , then by induction, we get a matching of size $m_1 + (m_2 + 1) + m_3 + m_4$ in $S_1^{-3} \cup S_2^{+2} \cup S_{3r}^{-2} \cup S_4^{-3}$. Otherwise, S_4' contains at least $n_3 + 2 = 4m_3 + 2$ points. Thus, by induction, we get a matching of size $m_1 + m_2 + (m_3 + 1) + m_4$ in $S_1 \cup S_2 \cup S_4' \cup S_4^{-3}$. In all cases we get a matching of size $m + 1$ in S .

- $r_3 \neq 0$.

In this case $r_3 = 1$, and without loss of generality, assume that $r_2 = 1$; which means $r_1 = r_4 = 0$. Thus, $n_1 = 4m_1, n_2 = 4m_2 + 1, n_3 = 4m_3 + 1$, and $n_4 = 4m_4$. If all elements of $\{m_1, m_2, m_4\}$ are equal to zero, then we have $m = m_3$, where $m_3 \geq 1$. Consider the triangles S_2^{+1} and S_{3r}^{-1} , which are disjoint. By induction, we get a matched pair in S_2^{+1} and a matching of size at least m_3 in S_{3r}^{-1} . Thus, in total, we get a matching of size at least $1 + m_3 = m + 1$ in S . Assume some elements in $\{m_1, m_2, m_4\}$ are greater than zero. Consider the triangles S_1^{-2}, S_2^{-3} , and S_4^{-2} . See Figure 4.15(a). By induction, we get matchings of size m_1, m_2 , and m_4 in S_1^{-2}, S_2^{-3} , and S_4^{-2} , respectively. Now we consider the largest triangle among S_1^{-2}, S_2^{-3} , and S_4^{-2} . Because of symmetry, we have two cases: (i) S_2^{-3} is the largest, or (ii) S_4^{-2} is the largest.

- S_2^{-3} is larger than S_1^{-2} and S_4^{-2} . Define l_1, l_2, S_2' as in Figure 4.15(a). If at least two points of $S_1 \cup S_2 \cup S_3$ are to the right of l_2 , then by induction, we get a matching of size $m_1 + m_2 + m_3 + (m_4 + 1)$ in $S_1^{-2} \cup S_2^{-3} \cup S_{3l}^{-2} \cup S_4^{+2}$. If at least two points of $S_2 \cup S_3 \cup S_4$ are above l_1 , then by induction, we get a matching of size $(m_1 + 1) + m_2 + m_3 + m_4$ in $S_1^{+2} \cup S_2^{-3} \cup S_{3b}^{-2} \cup S_4^{-2}$. Otherwise, S_2' contains $n_3 + 1 = 4m_3 + 2$ points, and we get a matching of size $m_1 + m_2 + (m_3 + 1) + m_4$ in $S_1 \cup S_2^{-3} \cup S_2' \cup S_4$. In all cases we get a matching of size $m + 1$ in S .

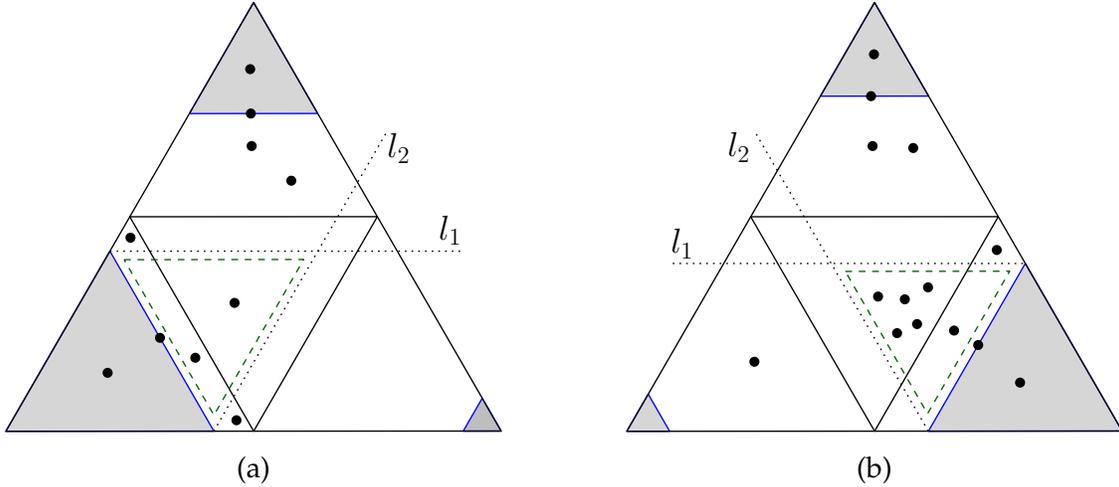


Figure 4.15: (a) S_2^{-3} is larger than S_1^{-2} and S_4^{-2} . (b) S_4^{-2} is larger than S_1^{-2} and S_2^{-3} .

- S_4^{-2} is larger than S_1^{-2} and S_2^{-3} . Define l_1, l_2, S'_4 as in Figure 4.15(b). If at least two points of $S_2 \cup S_3 \cup S_4$ are above l_1 , then by induction, we get a matching of size $(m_1 + 1) + m_2 + m_3 + m_4$ in $S_1^{+2} \cup S_2^{-3} \cup S_{3b}^{-2} \cup S_4^{-2}$. If any point of $S_1 \cup S_3 \cup S_4$ is to the left of l_2 , then by induction, we get a matching of size $m_1 + (m_2 + 1) + m_3 + m_4$ in $S_1^{-2} \cup S_2^{+1} \cup S_{3r}^{-1} \cup S_4^{-2}$. Otherwise, S'_4 contains at least $n_3 + 1 = 4m_3 + 2$ points, and we get a matching of size $m_1 + m_2 + (m_3 + 1) + m_4$ in $S_1 \cup S_2 \cup S'_4 \cup S_4^{-2}$. In all cases we get a matching of size $m + 1$ in S .

□

4.7 STRONG MATCHING IN $G_{\square}(P)$

In this section we consider the problem of computing a strong matching in $G_{\square}(P)$, where \square is an axis-aligned square whose center is the origin. We assume that P is in general position, i.e., (i) no two points have the same x -coordinate or the same y -coordinate, and (ii) no four points are on the boundary of any homothet of \square . Recall that $G_{\square}(P)$ is equal to the L_{∞} -Delaunay graph on P . Ábrego et al. [1, 2] proved that $G_{\square}(P)$ has a strong matching of size at least $\lceil n/5 \rceil$. Using a similar approach as in Section 4.6, we prove that $G_{\square}(P)$ has a strong matching of size at least $\lceil \frac{n-1}{4} \rceil$.

THEOREM 4.28. *Let P be a set of n points in general position in the plane. Let S be an axis-parallel square that contains P . Then, it is possible to find a strong matching of size at least $\lceil \frac{n-1}{4} \rceil$ for $G_{\square}(P)$ in S .*

Proof. The proof is by induction. Assume that any point set of size $n' \leq n - 1$ in an axis-parallel square S' , has a strong matching of size $\lceil \frac{n'-1}{4} \rceil$ in S' . If n is 0 or 1, then there is no matching in S , and if $n \in \{2, 3, 4, 5\}$, then by shrinking S , it is possible to find a strongly matched pair. Suppose that $n \geq 6$, and $n = 4m + r$, where $r \in \{0, 1, 2, 3\}$. If $r \in \{0, 1, 3\}$, then $\lceil \frac{n-1}{4} \rceil = \lceil \frac{(n-1)-1}{4} \rceil$, and by induction we are done. Suppose that $n = 4m + 2$, for some $m \geq 1$. We prove that there are $\lceil \frac{n-1}{4} \rceil = m + 1$ disjoint squares in S , each of them matches a pair of points in P . Partition S into four equal area squares S_1, S_2, S_3, S_4 which contain n_1, n_2, n_3, n_4

points, respectively; see Figure 4.16(a). Let $n_i = 4m_i + r_i$ for $1 \leq i \leq 4$, where $r_i \in \{0, 1, 2, 3\}$. Let R be the multiset $\{r_1, r_2, r_3, r_4\}$. By induction, in $S_1 \cup S_2 \cup S_3 \cup S_4$, we have a strong matching of size at least

$$A = \left\lceil \frac{n_1 - 1}{4} \right\rceil + \left\lceil \frac{n_2 - 1}{4} \right\rceil + \left\lceil \frac{n_3 - 1}{4} \right\rceil + \left\lceil \frac{n_4 - 1}{4} \right\rceil.$$

In the proof of Theorem 4.27, we have shown the following two claims:

CLAIM 1. $A \geq m$.

CLAIM 2. If $A = m$, then either (i) one element in R is equal to 3 and the other elements are equal to 1, or (ii) two elements in R are equal to 0 and the other elements are equal to 1.

If $A > m$, then we are done. Assume that $A = m$; in fact, by the induction hypothesis we have an strong matching of size m in S . We show how to find one more strongly matched pair in each case of Claim 2.

We define S_1^x as the smallest axis-parallel square contained in S_1 and anchored at the top-left corner of S_1 , which contains all the points in S_1 except x points. If S_1 contains less than x points, then the area of S_1^x is zero. We also define S_1^{+x} as the smallest axis-parallel square that contains S_1 and anchored at the top-left corner of S_1 , which has all the points in S_1 plus x other points of P . See Figure 4.16(a). Similarly we define the squares S_2^x, S_2^{+x} , and S_3^x, S_3^{+x} , and S_4^x, S_4^{+x} which are anchored at the top-right corner of S_2 , and the bottom-left corner of S_3 , and the bottom-right corner of S_4 , respectively.

Case 1: One element in R is equal to 3 and the other elements are equal to 1.

In this case, we have $m = m_1 + m_2 + m_3 + m_4 + 1$. Without loss of generality, assume that $r_1 = 3$ and $r_2 = r_3 = r_4 = 1$. Consider the squares $S_1^{-1}, S_2^{-3}, S_3^{-3}$, and S_4^{-3} . Note that the area of some of these squares—but not all—may be equal to zero. See Figure 4.16(b). By induction, we get matchings of size $m_1 + 1, m_2, m_3$, and m_4 , in $S_1^{-1}, S_2^{-3}, S_3^{-3}$, and S_4^{-3} , respectively. Now consider the largest square among $S_1^{-1}, S_2^{-3}, S_3^{-3}$, and S_4^{-3} . Because of the symmetry, we have only three cases: (i) S_1^{-1} is the largest, (ii) S_2^{-3} is the largest, and (iii) S_3^{-3} is the largest.

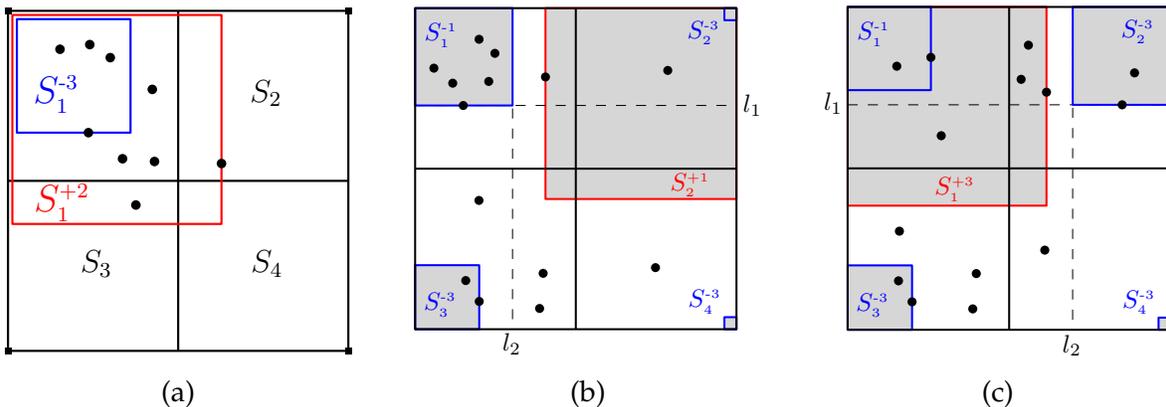


Figure 4.16: (a) Split S into four equal area squares. (b) S_1^{-1} is larger than S_2^{-3}, S_3^{-3} , and S_4^{-3} . (c) S_2^{-3} is larger than S_1^{-1}, S_3^{-3} , and S_4^{-3} .

- S_1^{-1} is the largest square. Consider the lines l_1 and l_2 which contain the bottom side and right side of S_1^{-1} , respectively; the dashed lines in Figure 4.16(b). Note that l_1 and l_2 do not

intersect any of S_2^{-3} , S_3^{-3} , and S_4^{-3} . If any point of S_1 is to the right of l_2 , then by induction, we get a matching of size $(m_1 + 1) + (m_2 + 1) + m_3 + m_4$ in $S_1^{-1} \cup S_2^{+1} \cup S_3^{-3} \cup S_4^{-3}$. Otherwise, by induction, we get a matching of size $(m_1 + 1) + m_2 + (m_3 + 1) + m_4$ in $S_1^{-1} \cup S_2^{-3} \cup S_3^{+1} \cup S_4^{-3}$. In all cases we get a matching of size $m + 1$ in S .

- S_2^{-3} is the largest square. Consider the lines l_1 and l_2 which contain the bottom side and left side of S_2^{-3} , respectively; the dashed lines in Figure 4.16(c). Note that l_1 and l_2 do not intersect any of S_1^{-1} , S_3^{-3} , and S_4^{-3} . If any point of S_2 is below l_1 , then by induction, we get a matching of size $(m_1 + 1) + m_2 + m_3 + (m_4 + 1)$ in $S_1^{-1} \cup S_2^{-3} \cup S_3^{-3} \cup S_4^{+1}$. Otherwise, by induction, we get a matching of size $(m_1 + 2) + m_2 + m_3 + m_4$ in $S_1^{+3} \cup S_2^{-3} \cup S_3^{-3} \cup S_4^{-3}$; see Figure 4.16(c). In all cases we get a matching of size $m + 1$ in S .
- S_4^{-3} is the largest square. Consider the lines l_1 and l_2 which contain the top side and left side of S_4^{-3} , respectively. If any point of S_4 is above l_1 , then by induction, we get a matching of size $(m_1 + 1) + (m_2 + 1) + m_3 + m_4$ in $S_1^{-1} \cup S_2^{+1} \cup S_3^{-3} \cup S_4^{-3}$. Otherwise, by induction, we get a matching of size $(m_1 + 1) + m_2 + (m_3 + 1) + m_4$ in $S_1^{-1} \cup S_2^{-3} \cup S_3^{+1} \cup S_4^{-3}$. In all cases we get a matching of size $m + 1$ in S .

Case 2: Two elements in R are equal to 0 and two elements are equal to 1.

In this case, we have $m = m_1 + m_2 + m_3 + m_4$. Because of the symmetry, only two cases may arise: (i) $r_1 = r_2 = 1$ and $r_3 = r_4 = 0$, (ii) $r_1 = r_4 = 1$ and $r_2 = r_3 = 0$.

- $r_1 = r_2 = 1$ and $r_3 = r_4 = 0$. Consider the squares S_1^{-3} , S_2^{-3} , S_3^{-2} , and S_4^{-2} . By induction, we get matchings of size m_1 , m_2 , m_3 , and m_4 , in S_1^{-3} , S_2^{-3} , S_3^{-2} , and S_4^{-2} , respectively. Now consider the largest square among S_1^{-3} , S_2^{-3} , S_3^{-2} , and S_4^{-2} . Because of the symmetry, we have only two cases: (a) S_1^{-3} is the largest, (b) S_3^{-2} is the largest. In case (a) we get one more matched pair either in S_2^{+1} or in S_3^{+2} . In case (b) we get one more matched pair either in S_1^{+1} or in S_4^{+2} .
- $r_1 = r_4 = 1$ and $r_2 = r_3 = 0$. Consider the squares S_1^{-3} , S_2^{-2} , S_3^{-2} , and S_4^{-3} . By induction, we get matchings of size m_1 , m_2 , m_3 , and m_4 , in S_1^{-3} , S_2^{-2} , S_3^{-2} , and S_4^{-3} , respectively. Now consider the largest square among S_1^{-3} , S_2^{-2} , S_3^{-2} , and S_4^{-3} . Because of the symmetry, we have only two cases: (a) S_1^{-3} is the largest, (b) S_2^{-2} is the largest. In case (a) we get one more matched pair either in S_2^{+2} or in S_3^{+2} . In case (b) we get one more matched pair either in S_1^{+1} or in S_4^{+1} .

□

4.8 A CONJECTURE ON STRONG MATCHING IN $G_{\ominus}(P)$

In this section, we discuss a possible way to further improve upon Theorem 4.14, as well as a construction leading to the conjecture that Algorithm 2 computes a strong matching of size at least $\lceil \frac{n-1}{8} \rceil$; unfortunately we are not able to prove this.

In Section 4.4 we proved that $\mathcal{J}(e^+)$ contains at most 16 edges. In order to achieve this upper bound we used the fact that the centers of the disks in $\mathcal{J}(e^+)$ should be far apart. We did not consider the endpoints of the edges representing these disks. By Observation 4.9, the disks representing the edges in $\mathcal{J}(e^+)$ cannot contain any of the endpoints. We applied this observation only on u and v . Unfortunately, our attempts to apply this observation on the endpoints of edges in $\mathcal{J}(e^+)$ have been so far unsuccessful.

Recall that T is a Euclidean minimum spanning tree of P , and for every edge $e = (u, v)$ in T , $\deg(e)$ is the degree of e in $T(e^+)$, where $T(e^+)$ is the set of all edges of T with weight at least

$w(e)$. Note that $w(e)$ is directly related to the Euclidean distance between u and v . Observe that the discs representing the edges adjacent to e intersect $D(u, v)$. Thus, these edges are in $\text{Inf}(e)$. We call an edge e in T a *minimal edge* if e is not longer than any of its adjacent edges. We observed that the maximum degree of a minimal edge is an upper bound for $\text{Inf}(e)$. We conjecture that,

CONJECTURE 1. $\text{Inf}(T)$ is at most the maximum degree of a minimal edge.

Monma and Suri [13] showed that for every point set P there exists a Euclidean minimum spanning tree, $\text{MST}(P)$, of maximum vertex degree five. Thus, the maximum edge degree in $\text{MST}(P)$ is 9. We show that for every point set P , there exists a Euclidean minimum spanning tree, $\text{MST}(P)$, such that the degree of each node is at most five and the degree of each minimal edge is at most eight. This implies the conjecture that $\text{Inf}(\text{MST}(P)) \leq 8$. That is, Algorithm 2 returns a strong matching of size at least $\lceil \frac{n-1}{8} \rceil$.

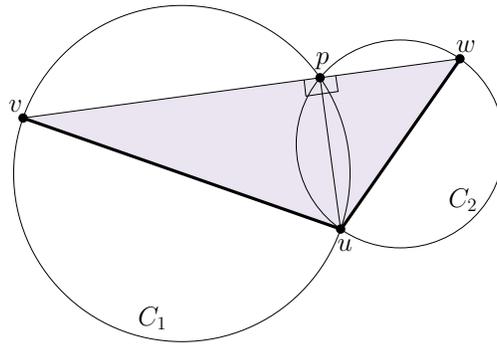


Figure 4.17: In $\text{MST}(P)$, the triangle $\triangle uvw$ formed by two adjacent edges uv and uw , is empty.

LEMMA 4.29. *If uv and uw are two adjacent edges in $\text{MST}(P)$, then the triangle $\triangle uvw$ has no point of $P \setminus \{u, v, w\}$ in its interior or on its boundary.*

Proof. If the angle between uv and uw is equal to π , then there is no other point of P on uv and uw . Assume that $\angle vuw < \pi$. Refer to Figure 4.17. Since $\text{MST}(P)$ is a subgraph of the Gabriel graph, the circles C_1 and C_2 with diameters uv and uw are empty. Since $\angle vuw < \pi$, C_1 and C_2 intersect each other at two points, say u and p . Connect u , v and w to p . Since uv and uw are the diameters of C_1 and C_2 , $\angle upv = \angle wpu = \pi/2$. This means that vw is a straight line segment. Since C_1 and C_2 are empty and $\triangle uvw \subset C_1 \cup C_2$, it follows that $\triangle uvw \cap P = \{u, v, w\}$. \square

LEMMA 4.30. *Follow Figure 4.18. For a convex-quadrilateral $Q = a, b, c, d$ with $|ab| \leq |bc| \leq |ad|$, if $\min\{\angle abc, \angle bad\} \geq \pi/3$ and $\angle abc + \angle bad \leq \pi$, then $|cd| \leq |ad|$.*

Proof. Let $\alpha_1 = \angle cad$, $\alpha_2 = \angle bac$, $\beta_1 = \angle cbd$, $\beta_2 = \angle abd$, $\gamma_1 = \angle acd$, $\gamma_2 = \angle acb$, $\delta_1 = \angle bdc$, and $\delta_2 = \angle adb$; see Figure 4.18. Since $|ab| \leq |bc| \leq |ad|$,

$$\gamma_2 \leq \alpha_2 \text{ and } \delta_2 \leq \beta_2.$$

Let ℓ be a line passing through c which is parallel to ad . Since $\angle abc + \angle bad \leq \pi$, ℓ intersects the line segment ab . This implies that $\alpha_1 \leq \gamma_2$. If $\beta_1 < \delta_1$, then $|cd| < |bc|$, and hence $|cd| < |ad|$ and we are done. Assume that $\delta_1 \leq \beta_1$. In this case, $\delta \leq \beta$. Now consider the two triangles $\triangle abc$ and $\triangle acd$. Since $\delta \leq \beta$ and $\alpha_1 \leq \gamma_2$, $\alpha_2 \leq \gamma_1$. Then we have

$$\alpha_1 \leq \gamma_2 \leq \alpha_2 \leq \gamma_1.$$

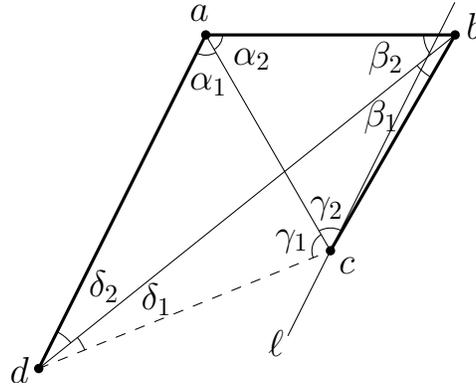


Figure 4.18: Illustration of Lemma 4.30: $|ab| \leq |bc| \leq |ad|$, $\angle abc \geq \pi/3$, $\angle bad \geq \pi/3$, and $\angle abc + \angle bad \leq \pi$.

Since $\alpha_1 \leq \gamma_1$, $|cd| \leq |ad|$, where the equality holds only if $\alpha_1 = \gamma_2 = \alpha_2 = \gamma_1$, i.e., Q is a diamond. This completes the proof. \square

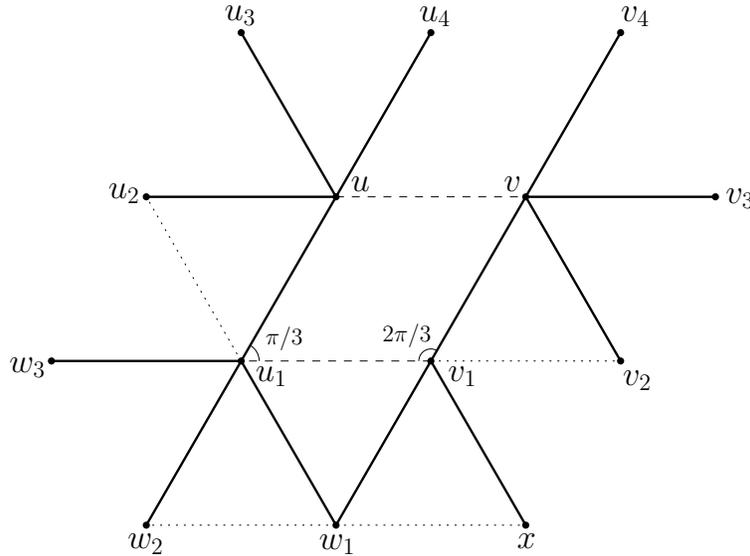


Figure 4.19: Solid segments represent the edges of $MST(P)$. Dashed segments represent the swapped edges. Dotted segments represent the edges which cannot exist.

LEMMA 4.31. *Every finite set of points P in the plane admits a minimum spanning tree whose node degree is at most five and whose minimal-edge degree is at most nine.*

Proof. Consider a minimum spanning tree, $MST(P)$, of maximum vertex degree 5. The maximum edge degree in $MST(P)$ is 9. Consider any minimal edge, uv . If the degree of uv is 8, then $MST(P)$ satisfies the statement of the lemma. Assume that the degree of uv is 9. Let u_1, u_2, u_3, u_4 and v_1, v_2, v_3, v_4 be the neighbors of u and v in clockwise and counterclockwise orders, respectively. See Figure 4.19. In $MST(P)$, the angles between two adjacent edges are at least $\pi/3$. Since $\angle u_i u u_{i+1} \geq \pi/3$ and $\angle v_i v v_{i+1} \geq \pi/3$ for $i = 1, 2, 3$, either $\angle v u u_1 + \angle u v v_1 \leq \pi$ or $\angle v u u_4 + \angle u v v_4 \leq \pi$. Without loss of generality assume that $\angle v u u_1 + \angle u v v_1 \leq \angle v u u_4 + \angle u v v_4$ and $\angle v u u_1 + \angle u v v_1 \leq \pi$. We prove that the spanning tree obtained by swapping the edge uv

with u_1v_1 is also a minimum spanning tree, and it has one fewer minimal-edge of degree 9. By repeating this procedure at each minimal-edge of degree 9, we obtain a minimum spanning tree which satisfies the statement of the lemma. Let $Q = u, v, v_1, u_1$. By Lemma 4.29, v_1 is outside the triangle Δu_1uv , and u_1 is outside the triangle Δuvv_1 . In addition, u_1 and v_1 are on the same side of the line subtended from uv . Thus, Q is a convex quadrilateral. Without loss of generality assume that $|vv_1| \leq |uu_1|$. By Lemma 4.30, $|u_1v_1| \leq |uu_1|$. If $|u_1v_1| < |uu_1|$, we get a contradiction to Lemma 4.5. Thus, assume that $|u_1v_1| = |uu_1|$. As shown in the proof of Lemma 4.30, this case happens only when Q is a diamond. This implies that $\angle vu_1u + \angle uvv_1 = \pi$, and consequently $\angle vu_1u + \angle uvv_1 = \pi$. In addition, $\angle u_iuu_{i+1} = \pi/3$ and $\angle v_ivv_{i+1} = \pi/3$ for $i = 1, 2, 3$. To establish the validity of our edge-swap, observe that the nine edges incident to u and v are all equal in length. Therefore, swapping uv with u_1v_1 does not change the cost of the spanning tree and, furthermore, the resulting tree is a valid spanning tree since u_1v_1 is not an edge of the original spanning tree $MST(P)$; otherwise u, v, v_1 , and u_1 would form a cycle. We have removed a minimal edge uv of degree 9, but it remains to show that the degree of u_1 and v_1 does not increase to six and new minimal edge of degree 9 is not generated. Note that u_1u_2 and v_1v_2 are not the edges of $MST(P)$, and hence, $\deg(u_1)$ and $\deg(v_1)$ are still less than six. In order to show that no new minimal edge is generated, we differentiate between two cases:

- $\min\{\angle vv_1u_1, \angle v_1u_1u\} > \pi/3$. Since $\angle v_1u_1u > \pi/3$ and $\angle uu_1u_2 = \pi/3$, u_1 can be adjacent to at most two vertices other than u and v_1 , and hence $\deg(u_1) \leq 4$; similarly $\deg(v_1) \leq 4$. Thus, u, v, u_1 , and v_1 are of degree at most four, and hence no new minimal edge of degree 9 is generated.
- $\min\{\angle vv_1u_1, \angle v_1u_1u\} = \pi/3$. W.l.o.g. assume that $\angle v_1u_1u = \pi/3$. This implies that $\angle vv_1u_1 = 2\pi/3$. Since $\angle v_1u_1u = \pi/3$ and $\angle uu_1u_2 = \pi/3$, u_1 is adjacent to at most three vertices other than u and v_1 . Let u, v_1, w_1, w_2, w_3 be the neighbors of u_1 in clockwise order. Note that v_1 is not adjacent to u, v_2 nor w_1 . But v_1 can be connected to another vertex, say x , which implies that $\deg(v_1) \leq 3$. We prove that the spanning tree obtained by swapping the edge u_1v_1 with v_1w_1 is also a minimum spanning tree of node degree at most five, which has one fewer minimal edge of degree 9. The new tree is a legal minimum spanning tree for P , because $|v_1w_1| = |v_1u_1|$. In addition, $\deg(u_1) \leq 4$ and $\deg(v_1) \leq 4$. Since w_1w_2 and w_1x are illegal edges, $\deg(w_1) \leq 4$. Thus, u, v, u_1, v_1 , and w_1 are of degree at most four and no new minimal edge of degree 9 is generated. This completes the proof that our edge-swap reduces the number of minimal-edges of degree nine by one.

□

4.9 CONCLUSIONS

Given a set of n points in general position in the plane, we considered the problem of strong matching of points with convex geometric shapes. A matching is strong if the objects representing whose edges are pairwise disjoint. In this chapter we presented algorithms which compute strong matchings of points with diametral-disks, equilateral-triangles, and squares. Specifically we showed that:

- There exists a strong matching of points with diametral-disks of size at least $\lceil \frac{n-1}{17} \rceil$.
- There exists a strong matching of points with downward equilateral-triangles of size at least $\lceil \frac{n-1}{9} \rceil$.

- There exists a strong matching of points with downward/upward equilateral-triangles of size at least $\lceil \frac{n-1}{4} \rceil$.
- There exists a strong matching of points with axis-parallel squares of size at least $\lceil \frac{n-1}{4} \rceil$.

The existence of a downward/upward equilateral-triangle matching of size at least $\lceil \frac{n-1}{4} \rceil$, implies the existence of either a downward equilateral-triangle matching of size at least $\lceil \frac{n-1}{8} \rceil$ or an upward equilateral-triangle matching of size at least $\lceil \frac{n-1}{8} \rceil$. This does not imply a lower bound better than $\lceil \frac{n-1}{9} \rceil$ for downward equilateral-triangle matching (or any fixed oriented equilateral-triangle).

A natural open problem is to improve any of the provided lower bounds, or extend these results for other convex shapes. A specific open problem is to prove that Algorithm 2 computes a strong matching of points with diametral-disks of size at least $\lceil \frac{n-1}{8} \rceil$ as discussed in Section 4.8. An alternative for Algorithm 2 is to pick an edge with the smallest influence set in each iteration. It remains open to derive lower bounds on the size of matchings reported by this new algorithm.

BIBLIOGRAPHY

- [1] B. M. Ábrego, E. M. Arkin, S. Fernández-Merchant, F. Hurtado, M. Kano, J. S. B. Mitchell, and J. Urrutia. Matching points with circles and squares. In *Proceedings of the Japanese Conference Discrete and Computational Geometry, JCDCG*, pages 1–15, 2004.
- [2] B. M. Ábrego, E. M. Arkin, S. Fernández-Merchant, F. Hurtado, M. Kano, J. S. B. Mitchell, and J. Urrutia. Matching points with squares. *Discrete & Computational Geometry*, 41(1):77–95, 2009.
- [3] J. Babu, A. Biniáz, A. Maheshwari, and M. H. M. Smid. Fixed-orientation equilateral triangle matching of point sets. *Theoretical Computer Science*, 555:55–70, 2014.
- [4] P. Bateman and P. Erdős. Geometrical extrema suggested by a lemma of Besicovitch. *Amer. Math. Monthly*, 58:306–314, 1951.
- [5] S. Bereg, N. Mutsanas, and A. Wolff. Matching points with rectangles and squares. *Computational Geometry: Theory and Applications*, 42(2):93–108, 2009.
- [6] A. Biniáz, A. Maheshwari, and M. Smid. Strong matching of points with geometric shapes. *Accepted to Computational Geometry: Theory and Applications, special issue in memoriam: Ferran Hurtado*.
- [7] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Higher-order triangular-distance Delaunay graphs: Graph-theoretical properties. *Computational Geometry: Theory and Applications*, 48(9):646–660, 2015.
- [8] A. Biniáz, A. Maheshwari, and M. H. M. Smid. Matchings in higher-order Gabriel graphs. *Theoretical Computer Science*, 596:67–78, 2015.
- [9] N. Bonichon, C. Gavoille, N. Hanusse, and D. Ilcinkas. Connections between theta-graphs, Delaunay triangulations, and orthogonal surfaces. In *WG*, pages 266–278, 2010.
- [10] P. Bose, P. Carmi, S. Collette, and M. H. M. Smid. On the stretch factor of convex Delaunay graphs. *JoCG*, 1(1):41–56, 2010.

- [11] P. Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.
- [12] M. B. Dillencourt. Toughness and Delaunay triangulations. *Discrete & Computational Geometry*, 5:575–601, 1990.
- [13] C. L. Monma and S. Suri. Transitions in geometric minimum spanning trees. *Discrete & Computational Geometry*, 8:265–293, 1992.

A bottleneck plane perfect matching of a set of n points in \mathbb{R}^2 is defined to be a perfect non-crossing matching that minimizes the length of the longest edge; the length of this longest edge is known as *bottleneck*. The problem of computing a bottleneck plane perfect matching has been proved to be NP-hard. We present an algorithm that computes a bottleneck plane matching of size at least $\frac{n}{5}$ in $O(n \log^2 n)$ -time. Then we extend our idea toward an $O(n \log n)$ -time approximation algorithm which computes a plane matching of size at least $\frac{2n}{5}$ whose edges have length at most $\sqrt{2} + \sqrt{3}$ times the bottleneck.

This chapter is published in the journal of Computational Geometry: Theory and Applications [1].

5.1 INTRODUCTION

We study the problem of computing a bottleneck non-crossing matching of points in the plane. For a given set P of n points in the plane, where n is even, let $K(P)$ denote the complete Euclidean graph with vertex set P . The *bottleneck plane matching* problem is to find a perfect non-crossing matching of $K(P)$ that minimizes the length of the longest edge. We denote such a matching by M^* . The bottleneck, λ^* , is the length of the longest edge in M^* . The problem of computing M^* has been proved to be NP-hard [2]. Figure 1 in [2] and [4] shows that the longest edge in the minimum weight matching (which is planar) can be unbounded with respect to λ^* . On the other hand the weight of the bottleneck matching can be unbounded with respect to the weight of the minimum weight matching, see Figure 5.1.

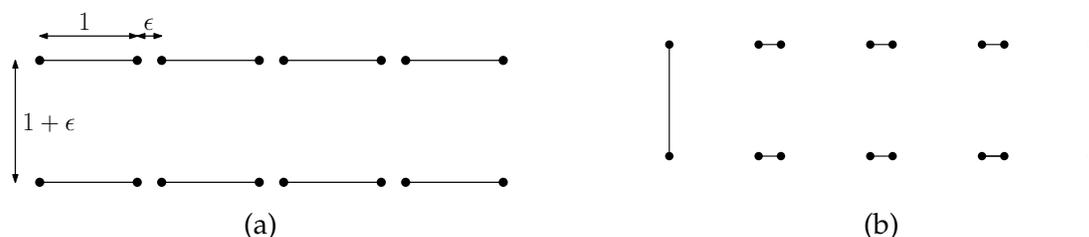


Figure 5.1: (a) bottleneck matching, (b) minimum weight matching.

Matching and bottleneck matching problems play an important role in graph theory, and thus, they have been studied extensively, e.g., [2, 3, 5, 9, 10, 13, 14]. Self-crossing configurations are often undesirable and may even imply an error condition; for example, a potential collision between moving objects, or inconsistency in a layout of a circuit. In particular, non-crossing matchings are especially important in the context of VLSI circuit layouts [11] and operations research.

Table 5.1: Summary of results.

| Algorithm | running time | bottleneck (λ) | match. size |
|----------------------|----------------------------|----------------------------------|-------------|
| A.-Affash et al. [2] | $O(n^{1.5} \sqrt{\log n})$ | $2\sqrt{10}\lambda^*$ | $n/2$ |
| Section 5.4.1 | $O(n \log^2 n)$ | λ^* | $n/5$ |
| Section 5.4.2 | $O(n \log n)$ | $(\sqrt{2} + \sqrt{3})\lambda^*$ | $2n/5$ |

5.1.1 Previous Work

It is desirable to compute a perfect matching of a point set in the plane, which is optimal with respect to some criterion such as: (a) *minimum-cost matching* which minimizes the sum of the lengths of all edges; also known as *minimum-weight matching* or *min-sum matching*, and (b) *bottleneck matching* which minimizes the length of the longest edge; also known as *min-max matching* [10]. For the minimum-cost matching, Vaidya [13] presented an $O(n^{2.5} \log^4 n)$ time algorithm, which was improved to $O(n^{1.5} \log^5 n)$ by Varadarajan [14]. As for bottleneck matching, Chang et al. [5] proved that such kind of matching is a subset of 17-RNG (relative neighborhood graph). They presented an algorithm, running in $O(n^{1.5} \sqrt{\log n})$ -time to compute a bottleneck matching of maximum cardinality. The matching computed by their algorithm may be crossing. Efrat and Katz [10] extended the result of Chang et al. [5] to higher dimensions. They proved that a bottleneck matching in any constant dimension can be computed in $O(n^{1.5} \sqrt{\log n})$ -time under the L_∞ -norm.

Note that a plane perfect matching of a point set can be computed in $O(n \log n)$ -time, e.g., by matching the two leftmost points recursively.

Abu-Affash et al. [2] showed that the bottleneck plane perfect matching problem is NP-hard and presented an algorithm that computes a plane perfect matching whose edges have length at most $2\sqrt{10}$ times the bottleneck, i.e., $2\sqrt{10}\lambda^*$. They also showed that this problem does not admit a PTAS (Polynomial Time Approximation Scheme), unless $P=NP$. Carlsson et al. [4] showed that the bottleneck plane perfect matching problem for a Euclidean bipartite complete graph is also NP-hard.

5.1.2 Our Results

The main results of this chapter are summarized in Table 5.1. We use the unit disk graph as a tool for our approximations. First, we present an $O(n \log n)$ -time algorithm in Section 5.3, that computes a plane matching of size at least $\frac{n-1}{5}$ in a connected unit disk graph. Then in Section 5.4.1 we describe how one can use this algorithm to obtain a bottleneck plane matching of size at least $\frac{n}{5}$ with edges of length at most λ^* in $O(n \log^2 n)$ -time. In Section 5.4.2 we present an $O(n \log n)$ -time approximation algorithm that computes a plane matching of size at least $\frac{2n}{5}$ whose edges have length at most $(\sqrt{2} + \sqrt{3})\lambda^*$. Finally we conclude this chapter in Section 5.5.

5.2 PRELIMINARIES

Let P denote a set of n points in the plane, where n is even, and let $K(P)$ denote the complete Euclidean graph over P . A *matching*, M , is a subset of edges of $K(P)$ without common vertices. Let $|M|$ denote the *cardinality* of M , which is the number of edges in M . M is a *perfect matching* if

it covers all the vertices of P , i.e., $|M| = \frac{n}{2}$. The *bottleneck* of M is defined as the longest edge in M . We denote its length by λ_M . A *bottleneck perfect matching* is a perfect matching that minimizes the bottleneck length. A *plane matching* is a matching with non-crossing edges. We denote a plane matching by M_{\neq} and a crossing matching by M_{\times} . The *bottleneck plane perfect matching*, M^* , is a perfect plane matching which minimizes the length of the longest edge. Let λ^* denote the length of the bottleneck edge in M^* . In this chapter we consider the problem of computing a bottleneck plane matching of P .

The Unit Disk Graph, $UDG(P)$, is defined to have the points of P as its vertices and two vertices p and q are connected by an edge if their Euclidean distance $|pq|$ is at most 1. The *maximum plane matching* of $UDG(P)$ is the maximum cardinality matching of $UDG(P)$, which has no pair of crossing edges. The following lemma states a folklore result, however, for the sake of completeness we provide a simple proof.

LEMMA 5.1. *If the maximum plane matching in unit disk graphs can be computed in polynomial time, then the bottleneck plane perfect matching problem for point sets can also be solved in polynomial time.*

Proof. Let $D = \{|pq| : p, q \in P\}$ be the set of all distances determined by pairs of points in P . Note that $\lambda^* \in D$. For each $\lambda \in D$, define the “unit” disk graph $DG(\lambda, P)$, in which two points p and q are connected by an edge if and only if $|pq| \leq \lambda$. Then λ^* is the minimum λ in D such that $DG(\lambda, P)$ has a plane matching of size $\frac{n}{2}$. \square

The Gabriel Graph, $GG(P)$, is defined to have the points of P as its vertices and two vertices p and q are connected by an edge if the disk with diameter pq does not contain any point of $P \setminus \{p, q\}$ in its interior and on its boundary.

LEMMA 5.2. *If the unit disk graph $UDG(P)$ of a point set P is connected, then $UDG(P)$ and $K(P)$ have the same minimum spanning tree.*

Proof. By running Kruskal’s algorithm on $UDG(P)$, we get a minimum spanning tree, say T . All the edges of T have length at most one, and the edges of $K(P)$ which do not belong to $UDG(P)$ all have length greater than one. Hence, T is also a minimum spanning tree of $K(P)$. \square

As a direct consequence of Lemma 5.2 we have the following corollary:

COROLLARY 5.3. *Consider the unit disk graph $UDG(P)$ of a point set P . We can compute the minimum spanning forest of $UDG(P)$, by first computing the minimum spanning tree of P and then removing the edges whose length is more than one.*

LEMMA 5.4. *For each pair of crossing edges (u, v) and (x, y) in $UDG(P)$, the four endpoints $u, v, x,$ and y are in the same component of $UDG(P)$.*

Proof. Note that the quadrilateral Q formed by the end points $u, v, x,$ and y is convex. W.l.o.g. assume that the angle $\angle xuy$ is the largest angle in Q . Clearly $\angle xuy \geq \pi/2$, and hence, in triangle $\triangle xuy$, the angles $\angle yxu$ and $\angle uyx$ are both less than $\pi/2$. Thus, the edges (u, x) and (u, y) are both less than (x, y) . This means that (u, x) and (u, y) are also edges of $UDG(P)$, thus, their four endpoints belong to the same component. \square

As a direct consequence of Lemma 5.4 we have the following corollary:

COROLLARY 5.5. *Any two edges that belong to different components of $UDG(P)$ do not cross.*

Let $\text{MST}(P)$ denote the Euclidean minimum spanning tree of P . We have prove the following lemma in Chapter 4.

LEMMA 5.6. *If (u, v) and (u, w) are two adjacent edges in $\text{MST}(P)$, then the triangle $\triangle uvw$ has no point of $P \setminus \{u, v, w\}$ inside or on its boundary.*

COROLLARY 5.7. *Consider $\text{MST}(P)$ and let N_v be the set of neighbors of a vertex v in $\text{MST}(P)$. Then the convex hull of N_v contains no point of P except v and the set N_v .*

The shaded area in Figure 5.2 shows the union of all these convex hulls.

5.3 PLANE MATCHING IN UNIT DISK GRAPHS

In this section we present two approximation algorithms for computing a maximum plane matching in a unit disk graph $\text{UDG}(P)$. In Section 5.3.1 we present a straight-forward $\frac{1}{3}$ -approximation algorithm. For the case when $\text{UDG}(P)$ is connected, we present a $\frac{2}{5}$ -approximation algorithm in Section 5.3.2.

5.3.1 $\frac{1}{3}$ -approximation algorithm

Given a possibly disconnected unit disk graph $\text{UDG}(P)$, we start by computing a (possibly crossing) maximum matching M_\times of $\text{UDG}(P)$ using Edmonds algorithm [7]. Then we transform M_\times to another (possibly crossing) matching M'_\times with some properties, and then pick at least one-third of its edges which satisfy the non-crossing property. Consider a pair of crossing edges (p, q) and (r, s) in M_\times , and let c denote the intersection point. If their smallest intersection angle is at most $\pi/3$, we replace these two edges with new ones. For example if $\angle pcr \leq \pi/3$, we replace (p, q) and (r, s) with new edges (p, r) and (q, s) . Since the angle between them is at most $\pi/3$, the new edges are not longer than the older ones, i.e. $\max\{|pr|, |qs|\} \leq \max\{|pq|, |rs|\}$, and hence the new edges belong to the unit disk graph. On the other hand the total length of the new edges is strictly less than the older ones; i.e. $|pr| + |qs| < |pq| + |rs|$. For each pair of intersecting edges in M_\times , with angle at most $\pi/3$, we apply this replacement. We continue this process until we have a matching M'_\times with the property that if two matched edges intersect, each of the angles incident on c is larger than $\pi/3$.

For each edge in M'_\times , consider the counter clockwise angle it makes with the positive x -axis; this angle is in the range $[0, \pi)$. Using these angles, we partition the edges of M'_\times into three subsets, one subset for the angles $[0, \pi/3)$, one subset for the angles $[\pi/3, 2\pi/3)$, and one subset for the angles $[2\pi/3, \pi)$. Observe that edges within one subset are non-crossing. Therefore, if we output the largest subset, we obtain a non-crossing matching of size at least $|M'_\times|/3 = |M_\times|/3$.

Since in each step (each replacement) the total length of the matched edges decreases, the replacement process converges and the algorithm will stop. Bonnet and Miltzow [8] showed that this process stops after $O(n^3)$ steps, where n is the number of points in P . Thus the running time of this algorithm is polynomial.

5.3.2 $\frac{2}{5}$ -approximation algorithm for connected unit disk graphs

In this section we assume that the unit disk graph $\text{UDG}(P)$ is connected. Monma et al. [12] proved that every set of points in the plane admits a minimum spanning tree of degree at most five which

can be computed in $O(n \log n)$ time. By Lemma 5.2, the same claim holds for $\text{UDG}(P)$. Here we present an algorithm which extracts a plane matching M from $\text{MST}(P)$. Consider a minimum spanning tree T of $\text{UDG}(P)$ with vertices of degree at most five. We define the *skeleton tree*, T' , as the tree obtained from T by removing all its leaves; see Figure 5.2. Clearly $T' \subseteq T \subseteq \text{UDG}(P)$. For clarity we use u and v to refer to the leaves of T and T' respectively. In addition, let v and v' , respectively, refer to the copies of a vertex v in T and T' . In each step, pick an arbitrary leaf $v' \in T'$. By the definition of T' , it is clear that the copy of v' in T , i.e. v , is connected to vertices u_1, \dots, u_k , for some $1 \leq k \leq 4$, which are leaves of T (if T' has one vertex then $k \leq 5$). Pick an arbitrary leaf u_i and add (v, u_i) as a matched pair to M . For the next step we update T by removing v and all its adjacent leaves. We also compute the new skeleton tree and repeat this process. In the last iteration, T' is empty and we may be left with a tree T consisting of one single vertex or one single edge. If T consists of one single vertex, we disregard it, otherwise we add its only edge to M .

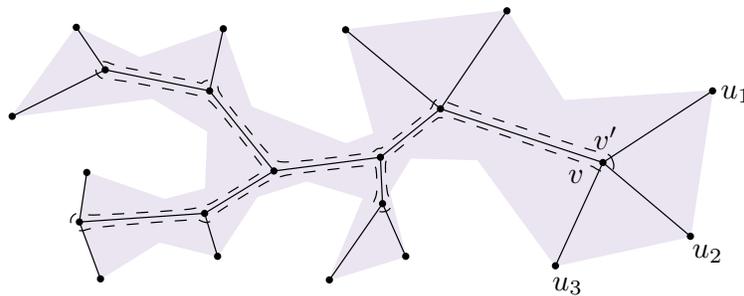


Figure 5.2: Minimum spanning tree T with union of empty convex hulls. The skeleton tree T' is surrounded by dashed line, and v' is a leaf in T' .

The formal algorithm is given as `PLANEMATCHING`, which receives a point set P —whose unit disk graph is connected—as input and returns a matching M as output. The function $\text{MST}_5(P)$ returns a Euclidean minimum spanning tree of P with degree at most five, and the function $\text{Neighbor}(v', T')$ returns the only neighbor of leaf v' in T' .

LEMMA 5.8. *Given a set P of n points in the plane such that $\text{UDG}(P)$ is connected, algorithm `PLANEMATCHING` returns a plane matching M of $\text{MST}(P)$ of size $|M| \geq \frac{n-1}{5}$. Furthermore, M can be computed in $O(n \log n)$ time.*

Proof. In each iteration an edge $(v, u_i) \in T$ is added to M . Since T is plane, M is also plane and M is a matching of $\text{MST}(P)$.

Line 5 picks $v' \in T'$ which is a leaf, so its analogous vertex $v \in T$ is connected to at least one leaf. In each iteration we select an edge incident to one of the leaves and add it to M , then disregard all other edges connected to v (line 9). So for the next iteration T loses at most five edges. Since T has $n - 1$ edges initially and we add one edge to M out of each five edges of T , we have $|M| \geq \frac{n-1}{5}$.

According to Corollary 5.3 and by [12], line 2 takes $O(n \log n)$ time. The while-loop is iterated $O(n)$ times and in each iteration, lines 5-13 take constant time. So, the total running time of algorithm `PLANEMATCHING` is $O(n \log n)$. \square

The size of a maximum matching can be at most $\frac{n}{2}$. Therefore, algorithm `PLANEMATCHING` computes a matching of size at least $\frac{2(n-1)}{5n}$ times the size of a perfect matching, and hence, when n is large enough, `PLANEMATCHING` is a $\frac{2}{5}$ -approximation algorithm. On the other hand

Algorithm 3 PLANEMATCHING(P)**Input:** set P of n points in the plane, such that $\text{UDG}(P)$ is connected.**Output:** plane matching M of $\text{MST}(P)$ with $|M| \geq \frac{n-1}{5}$.

```

1:  $M \leftarrow \emptyset$ 
2:  $T \leftarrow \text{MST}_5(P)$ 
3:  $T' \leftarrow \text{SkeletonTree}(T)$ 
4: while  $T' \neq \emptyset$  do
5:    $v' \leftarrow$  a leaf of  $T'$ 
6:    $L_v \leftarrow$  set of leaves connected to  $v$  in  $T$ 
7:    $u \leftarrow$  an element of  $L_v$ 
8:    $M \leftarrow M \cup \{(v, u)\}$ 
9:    $T \leftarrow T \setminus (\{v\} \cup L_v)$ 
10:  if  $\text{deg}(\text{Neighbor}(v', T')) = 2$  then
11:     $T' \leftarrow T' \setminus \{v', \text{Neighbor}(v', T')\}$ 
12:  else
13:     $T' \leftarrow T' \setminus \{v'\}$ 
14: if  $T$  consists of one single edge then
15:    $M \leftarrow M \cup T$ 
16: return  $M$ 

```

there are unit disk graphs whose maximum matchings have size $\frac{n-1}{5}$; see Figure 5.3. In this case PLANEMATCHING returns a maximum matching. In addition, when $\text{UDG}(P)$ is a tree or a cycle, PLANEMATCHING returns a maximum matching.

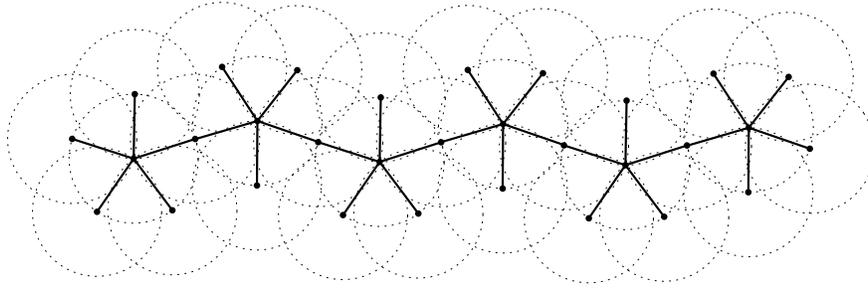


Figure 5.3: Unit disk graph with all edges of unit length, and maximum matching of size $\frac{n-1}{5}$.

In Section 5.4.1 we will show how one can use a modified version of algorithm PLANEMATCHING to compute a bottleneck plane matching of size at least $\frac{n}{5}$ with bottleneck length at most λ^* . Recall that λ^* is the length of the bottleneck edge in the bottleneck plane perfect matching M^* . Section 5.4.2 extends this idea to an algorithm which computes a plane matching of size at least $\frac{2n}{5}$ with edges of length at most $(\sqrt{2} + \sqrt{3})\lambda^*$.

5.4 APPROXIMATING BOTTLENECK PLANE PERFECT MATCHING

The general approach of our algorithms is to first compute a (possibly crossing) bottleneck perfect matching M_\times of $K(P)$ using the algorithm in [5]. Let λ_{M_\times} denote the length of the bottleneck edge in M_\times . It is obvious that the bottleneck length of any plane perfect matching is not less than λ_{M_\times} .

Therefore, $\lambda^* \geq \lambda_{M_\times}$. We consider a “unit” disk graph $DG(\lambda_{M_\times}, P)$ over P , in which there is an edge between two vertices p and q if $|pq| \leq \lambda_{M_\times}$. Note that $DG(\lambda_{M_\times}, P)$ is not necessarily connected. Let G_1, \dots, G_k be the connected components of $DG(\lambda_{M_\times}, P)$. For each component G_i , consider a minimum spanning tree T_i of degree at most five. We show how to extract from T_i a plane matching M_i of proper size and appropriate edge lengths.

LEMMA 5.9. *Each component of $DG(\lambda_{M_\times}, P)$ has an even number of vertices.*

Proof. This follows from the facts that M_\times is a perfect matching and both end points of each edge in M_\times belong to the same component of $DG(\lambda_{M_\times}, P)$. \square

5.4.1 First Approximation Algorithm

In this section we describe the process of computing a plane matching M of size $|M| \geq \frac{1}{5}n$ with bottleneck length at most λ^* . Consider the minimum spanning trees T_1, \dots, T_k of the k components of $DG(\lambda_{M_\times}, P)$. For $1 \leq i \leq k$, let P_i denote the set of vertices in T_i and n_i denote the number of vertices of P_i . Our approximation algorithm runs in two steps:

STEP 1: We start by running algorithm PLANEMATCHING on each of the point sets P_i . Let M_i be the output. Recall that algorithm PLANEMATCHING, from Section 5.3.2, picks a leaf $v' \in T_i'$, corresponding to a vertex $v \in T_i$, matches it to one of its neighboring leaves in T_i and disregards the other edges connected to v . According to Lemma 5.8, this gives us a plane matching M_i of size at least $\frac{n_i-1}{5}$. However, we are looking for a matching of size at least $\frac{n_i}{5}$.

The total number of edges of T_i is $n_i - 1$ and in each of the iterations, the algorithm picks one edge out of at most five candidates. If in at least one of the iterations of the while-loop, v has degree at most four (in T_i), then in that iteration algorithm PLANEMATCHING picks one edge out of at most four candidates. Therefore, the size of M_i satisfies

$$|M_i| \geq 1 + \frac{(n_i - 1) - 4}{5} = \frac{n_i}{5}.$$

If in all the iterations of the while-loop, v has degree five, we look at the angles between the consecutive leaves connected to v . Recall that in $MST(P)$ all the angles are greater than or equal to $\pi/3$. If in at least one of the iterations, v is connected to two consecutive leaves u_j and u_{j+1} for $1 \leq j \leq 3$, such that $\angle u_j v u_{j+1} = \pi/3$, we change M_i as follow. Remove from M_i the edge incident to v and add to M_i the edges (u_j, u_{j+1}) and (v, u_s) , where $u_s, s \notin \{j, j+1\}$, is one of the leaves connected to v . Clearly $\triangle v u_j u_{j+1}$ is equilateral and $|u_j u_{j+1}| = |u_j v| = |u_{j+1} v| \leq \lambda_{M_\times}$, and by Lemma 5.6, (u_j, u_{j+1}) does not cross other edges. In this case, the size of M_i satisfies

$$|M_i| = 2 + \frac{(n_i - 1) - 5}{5} = \frac{n_i + 4}{5} \geq \frac{n_i}{5}.$$

STEP 2: In this step we deal with the case that in all the iterations of the while-loop, v has degree five and the angle between any pair of consecutive leaves connected to v is greater than $\pi/3$. Recall that M_\times is a perfect matching and both end points of each edge in M_\times belong to the same T_i .

LEMMA 5.10. *In Step 2, at least two leaves of T_i are matched in M_\times .*

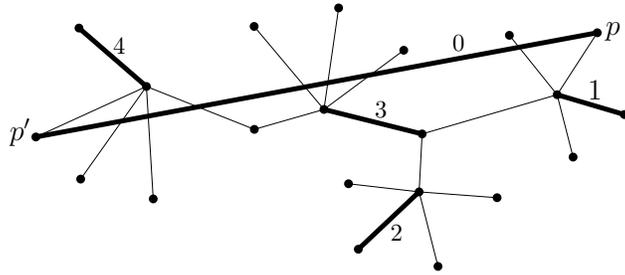


Figure 5.4: Resulted matching M_i by modified PLANEMATCHING. The numbers show the order in which the edges (bold edges) are added to M_i .

Proof. Let m_i and m'_i denote the number of external (leaves) and internal nodes of T_i , respectively. Clearly m'_i is equal to the number of vertices of T'_i and $m_i + m'_i = n_i$. Consider the reverse process in PLANEMATCHING. Start with a 5-star tree t_i , i.e. $t_i = K_{1,5}$, and in each iteration append a new $K_{1,5}$ to t_i until $t_i = T_i$. In the first step $m_i = 5$ and $m'_i = 1$. In each iteration a leaf of the appended $K_{1,5}$ is identified with a leaf of t_i ; the resulting vertex becomes an internal node. On the other hand, the “center” of the new $K_{1,5}$ becomes an internal node of t_i and its other four neighbors become leaves of t_i . So in each iteration, the number of leaves m_i increases by three, and the number of internal nodes m'_i increases by two. Hence, in all iterations (including the first step) we have $m_i \geq m'_i + 4$.

Again consider M_\times . In the worst case if all m'_i internal vertices of T_i are matched to leaves, we still have four leaves which have to be matched together. \square

According to Lemma 5.10 there is an edge $(p, p') \in M_\times$ where p and p' are leaves in T_i . We can find p and p' for all T_i 's by checking all the edges of M_\times once. We remove all the edges of M_i and initialize $M = \{(p, p')\}$. Again we run a modified version of PLANEMATCHING in such a way that in each iteration, in line 7 it selects the leaf u_i adjacent to v such that (v, u_i) is not intersected by (p, p') . In each iteration v has degree five and is connected to at least four leaf edges with angles greater than $\pi/3$. Thus, (p, p') can intersect at most three of the leaf edges and such kind of (v, u_i) exists. See Figure 5.4. In this case, M_i has size

$$|M_i| = 1 + \frac{n_i - 1}{5} = \frac{n_i + 4}{5} \geq \frac{n_i}{5}.$$

We run the above algorithm on each T_i and for each of them we compute a plane matching M_i . The final matching of point set P will be $M = \bigcup_{i=1}^k M_i$.

THEOREM 5.11. *Let P be a set of n points in the plane, where n is even, and let λ^* be the minimum bottleneck length of any plane perfect matching of P . In $O(n^{1.5} \sqrt{\log n})$ time, a plane matching of P of size at least $\frac{n}{5}$ can be computed, whose bottleneck length is at most λ^* .*

Proof. Proof of edge length: Let λ_{M_\times} be the length of the longest edge in M_\times and consider a component G_i of $DG(\lambda_{M_\times}, P)$. All the selected edges in Steps 1 and 2 belong to T_i except (u_j, u_{j+1}) and (p, p') . T_i is a subgraph of G_i , and the edge (u_j, u_{j+1}) belongs to G_i , and the edge (p, p') belongs to M_\times (which belongs to G_i as well). So all the selected edges belong to G_i , and $\lambda_{M_i} \leq \lambda_{M_\times}$. Since $\lambda_{M_\times} \leq \lambda^*$, we have $\lambda_{M_i} \leq \lambda_{M_\times} \leq \lambda^*$ for all $i, 1 \leq i \leq k$.

Proof of planarity: The edges of M_i belong to the minimum spanning forest of $DG(\lambda_{M_\times}, P)$ which is plane, except (u_j, u_{j+1}) and (p, p') . According to Corollary 5.5 and Lemma 5.6 the edge (u_j, u_{j+1}) does not cross the edges of the minimum spanning forest. In Step 2 we select edges

of T_i in such a way that avoid (p, p') . Note that (p, p') belongs to the component G_i and by Corollary 5.5 it does not cross any edge of the other components of $DG(\lambda_{M_\times}, P)$. So M is plane.

Proof of matching size: Since $M = M_1 \cup \dots \cup M_k$, and for each $1 \leq i \leq k$, $|M_i| \geq \frac{n_i}{5}$, hence

$$|M| \geq \sum_{i=1}^k |M_i| \geq \sum_{i=1}^k \frac{n_i}{5} = \frac{n}{5}.$$

Proof of complexity: The initial matching M_\times can be computed in time $O(n^{1.5} \sqrt{\log n})$ by using the algorithm of Chang et al. [5]. By Lemma 5.8 algorithm PLANEMATCHING runs in $O(n \log n)$ time. In Step 1 we spend constant time for checking the angles and the number of leaves connected to v during the while-loop. In Step 2, the matched leaves p and p' can be computed in $O(n)$ time for all T_i 's by checking all the edges of M_\times before running the algorithm again. So the modified PLANEMATCHING still runs in $O(n \log n)$ time, and the total running time of our method is $O(n^{1.5} \sqrt{\log n})$. \square

Since the running time of the algorithm is bounded by the time of computing the initial bottleneck matching M_\times , any improvement in computing M_\times leads to a faster algorithm for computing a plane matching M . In the next section we improve the running time.

5.4.1.1 Improving the Running Time

In this section we present an algorithm that improves the running time to $O(n \log^2 n)$. We first compute a forest F , such that each edge in F is of length at most λ^* and, for each tree $T \in F$, we have a leaf $p \in T$ and a point $p' \in T$ such that $|pp'| \leq \lambda^*$. Once we have this forest F , we apply Step 1 and Step 2 on F to obtain the matching M as in the previous section.

Let $MST(P)$ be a (five-degree) minimum spanning tree of P . Let $F_\lambda = \{T_1, T_2, \dots, T_k\}$ be the forest obtained from $MST(P)$ by removing all the edges whose length is greater than λ , i.e., $F_\lambda = \{e \in MST(P) : |e| \leq \lambda\}$. For a point $p \in P$, let $cl(p, P)$ be the point in P that is closest to p .

LEMMA 5.12. *For all $T \in F_{\lambda^*}$, it holds that*

(i) *the number of points in T is even, and*

(ii) *for each two leaves $p, q \in T$ that are incident to the same node v in T , let $p' = cl(p, P \setminus \{v\})$, let $q' = cl(q, P \setminus \{v\})$, and assume that $|pp'| \leq |qq'|$. Then, $\lambda^* \geq |pp'|$ and p' belongs to T .*

Proof. (i) Suppose that T has odd number of points. Thus in M^* one of the points in T should be matched to a point in a tree $T' \neq T$ by an edge e . Since $e \notin F_{\lambda^*}$, we have $|e| > \lambda^*$, which contradicts that λ^* is the minimum bottleneck. (ii) Note that v is the closest point to both p and q . In M^* , at most one of p and q can be matched to v , and the other one must be matched to a point which is at least as far as its second closest point. Thus, λ^* is at least $|pp'|$. The distance between any two trees in F_{λ^*} is greater than λ^* . Now if p' is not in T , then in any bottleneck perfect matching, either p or q is matched to a point of distance greater than λ^* , which contradicts that λ^* is the minimum bottleneck. \square

Let $E = (e_1, e_2, \dots, e_{n-1})$ be the edges of $MST(P)$ in sorted order of their lengths. Our algorithm performs a binary search on E , and for each considered edge e_i , it uses Algorithm 4 to decide whether $\lambda < \lambda^*$, where $\lambda = |e_i|$. The algorithm constructs the forest F_λ , and for each tree T in F_λ , it picks two leaves p and q from T and finds their second closest points p' and q' .

Assume w.l.o.g. that $|pp'| \leq |qq'|$. Then, the algorithm returns FALSE if p' does not belong to T . By Lemma 5.12, if the algorithm returns FALSE, then we know that $\lambda < \lambda^*$.

Let e_j be the shortest edge in $MST(P)$, for which Algorithm 4 does not return FALSE. This means that Algorithm 4 returns FALSE for e_{j-1} , and there is a tree T in $F_{|e_{j-1}|}$ and a leaf p in T , such that $|pp'| \geq |e_j|$. Thus $|e_j| \leq \lambda^*$ and for each tree T in the forest $F_{|e_j|}$, we stored a leaf p of T and a point $p' \in T$, such that $|pp'| \leq \lambda^*$. Since each tree in $F_{|e_j|}$ is a subtree of $MST(P)$, $F_{|e_j|}$ is planar and each tree in $F_{|e_j|}$ is of degree at most five.

Algorithm 4 COMPARETOOPT(λ)

```

1: compute  $F_\lambda$ 
2:  $L \leftarrow$  empty list
3: for each  $T \in F_\lambda$  do
4:   if  $T$  has an odd number of points then
5:     return FALSE
6:   if there exist two leaves  $p$  and  $q$  incident to a node  $v \in T$  then
7:      $p' \leftarrow cl(p, P \setminus \{v\})$ 
8:      $q' \leftarrow cl(q, P \setminus \{v\})$ 
9:     if  $|pp'| \leq |qq'|$  then
10:      if  $p'$  does not belong to  $T$  then
11:        return FALSE
12:      else
13:        add the triple  $(p, p', T)$  to  $L$ 
14:      else
15:        if  $q'$  does not belong to  $T$  then
16:          return FALSE
17:        else
18:          add the triple  $(q, q', T)$  to  $L$ 
19: return  $L$ 

```

Now we can apply Step 1 and Step 2 on $F_{|e_j|}$ as in the previous section. Note that in Step 2, for each tree T_i we have a pair (p, p') (or (q, q')) in the list L which can be matched. In Step 2, in each iteration v has degree five, thus, p' should be a vertex of degree two or a leaf in T_i . If p' is a leaf we run the modified version of PLANEMATCHING as in the previous section. If p' has degree two, we remove all the edges of M_i and initialize $M_i = (p, p')$. Then remove p' from T_i and run PLANEMATCHING on the resulted subtrees. Finally, set $M = \bigcup_{i=1}^k M_i$.

LEMMA 5.13. *The matching M is planar.*

Proof. Consider two edges $e = (p, p')$ and $e' = (q, q')$ in M . We distinguish between four cases:

1. $e \in F_{|e_j|}$ and $e' \in F_{|e_j|}$. In this case, both e and e' belong to $MST(P)$ and hence they do not cross each other.
2. $e \notin F_{|e_j|}$ and $e' \in F_{|e_j|}$. If e and e' cross each other, then this contradicts the selection of (q, q') in Step 2 (which prevents (p, p')).
3. $e \in F_{|e_j|}$ and $e' \notin F_{|e_j|}$. It leads to a contradiction as in the previous case.

4. $e \notin F_{|e_j|}$ and $e' \notin F_{|e_j|}$. If e and e' cross each other, then either $\min\{|pq|, |pq'|\} < |pp'|$ or $\min\{|qp|, |qp'|\} < |qq'|$, which contradicts the selection of p' or q' . Note that p cannot be the second closest point to q , because p and q are in different trees.

□

LEMMA 5.14. *The matching M can be computed in $O(n \log^2 n)$ time.*

Proof. Computing $\text{MST}(P)$ and sorting its edges take $O(n \log n)$ [6]. Since we performed a binary search on the edges of $\text{MST}(P)$, we need $\log n$ iterations. In each iteration, for an edge e_i , we compute the forest $F_{|e_i|}$ in $O(n)$ and the number of the trees in the forest can be $O(n)$ in the worst case. We compute in advance the second order Voronoi diagram of the points together with a corresponding point location data structure, in $O(n \log n)$ [6]. For each tree in the forest, we perform a point location query to find the closest points p' and q' , which takes $O(\log n)$ for each query. Therefore the total running time is $O(n \log^2 n)$. □

THEOREM 5.15. *Let P be a set of n points in the plane, where n is even, and let λ^* be the minimum bottleneck length of any plane perfect matching of P . In $O(n \log^2 n)$ time, a plane matching of P of size at least $\frac{n}{5}$ can be computed, whose bottleneck length is at most λ^* .*

5.4.2 Second Approximation Algorithm

In this section we present another approximation algorithm which gives a plane matching M of size $|M| \geq \frac{2}{5}n$ with bottleneck length $\lambda_M \leq (\sqrt{2} + \sqrt{3})\lambda^*$. Let $\text{DT}(P)$ denote the Delaunay triangulation of P . Let the edges of $\text{DT}(P)$ be, in sorted order of their lengths, e_1, e_2, \dots . Initialize a forest F consisting of n trees, each one being a single node for one point of P . Run Kruskal's algorithm on the edges of $\text{DT}(P)$ and terminate as soon as every tree in F has an even number of nodes. Let e_l be the last edge that is added by Kruskal's algorithm. Observe that e_l is the longest edge in F . Denote the trees in F by T_1, \dots, T_k and for $1 \leq i \leq k$, let P_i be the vertex set of T_i and let $n_i = |P_i|$.

LEMMA 5.16. $\lambda^* \geq |e_l|$.

Proof. Let i be such that e_l is an edge in T_i . Let T'_i and T''_i be the trees obtained by removing e_l from T_i . Let P'_i be the vertex set of T'_i . Then $|e_l| = \min\{|pq| : p \in P'_i, q \in P \setminus P'_i\}$. Consider the optimal matching M^* with bottleneck length λ^* . Since e_l is the last edge added, P'_i has odd size. The matching M^* contains an edge joining a point in P'_i with a point in $P \setminus P'_i$. This edge has length at least $|e_l|$. □

By Lemma 5.16 the length of the longest edge in F is at most λ^* . For each $T_i \in F$, where $1 \leq i \leq k$, our algorithm will compute a plane matching M_i of P_i of size at least $\frac{2n_i}{5}$ with edges of length at most $(\sqrt{2} + \sqrt{3})\lambda^*$ and returns $\bigcup_{i=1}^k M_i$. To describe the algorithm for tree T_i on vertex set P_i , we will write P, T, n, M instead of P_i, T_i, n_i, M_i , respectively. Thus, P is a set of n points, where n is even, and T is a minimum spanning tree of P_i .

Consider the minimum spanning tree T of P having degree at most five, and let T' be the skeleton tree of T . Suppose that T' has at least two vertices. We will use the following notation. Let v' be a leaf in T' , and let w' be the neighbor of v' . Recall that v' and w' are copies of vertices v and w in T . In T , we consider the clockwise ordering of the neighbors of v . Let this ordering be w, u_1, u_2, \dots, u_k for some $1 \leq k \leq 4$. Clearly u_1, \dots, u_k are leaves in T . Consider two leaves u_i

and u_j where $i < j$. We define $\text{cw}(u_i v u_j)$ as the clockwise angle from $\overline{vu_i}$ to $\overline{vu_j}$. We say that the leaf v' (or its copy v) is an *anchor* if $k = 2$ and $\text{cw}(u_1 v u_2) \geq \pi$. See Figure 5.5.

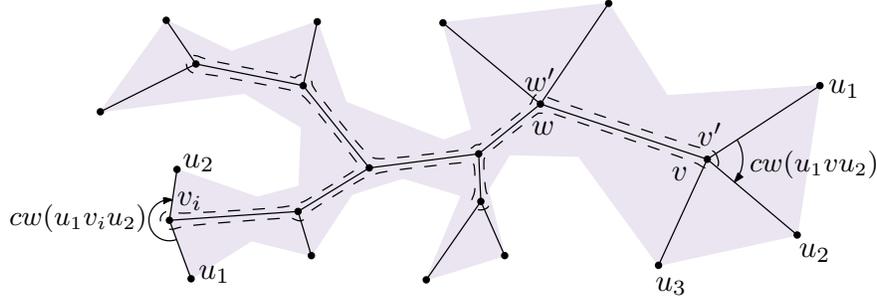


Figure 5.5: The vertices around v are sorted clockwise, and v_i is an anchor vertex.

Now we describe how one can iteratively compute a plane matching of proper size with bounded-length edges from T . We start with an empty matching M . Each iteration consists of two steps, during which we add edges to M . As we prove later, the output is a plane matching of P of size at least $\frac{2}{5}n$ with bottleneck at most $(\sqrt{2} + \sqrt{3})\lambda^*$.

5.4.2.1 Step 1

We keep applying the following process as long as T has more than six vertices and T' has some non-anchor leaf. Note that T' has at least two vertices.

Take a non-anchor leaf v' in T' and according to the number k of leaves connected to v in T do the following:

$k = 1$ add (v, u_1) to M , and set $T = T \setminus \{v, u_1\}$.

$k = 2$ since v' is not an anchor, $\text{cw}(u_1 v u_2) < \pi$. By Lemma 5.6 the triangle $\Delta_{u_1 v u_2}$ is empty. We add (u_1, u_2) to M , and set $T = T \setminus \{u_1, u_2\}$.

$k = 3$ in this case v has degree four and at least one of $\text{cw}(u_1 v u_2)$ and $\text{cw}(u_2 v u_3)$ is less than π . W.l.o.g. suppose that $\text{cw}(u_1 v u_2) < \pi$. By Lemma 5.6 the triangle $\Delta_{u_1 v u_2}$ is empty. Add (u_1, u_2) to M and set $T = T \setminus \{u_1, u_2\}$.

$k = 4$ this case is handled similarly as the case $k = 3$.

At the end of Step 1, T has at most six vertices or all the leaves of T' are anchors. In the former case, we add edges to M as will be described in Section 5.4.2.3 and after which the algorithm terminates. In the latter case we go to Step 2.

5.4.2.2 Step 2

In this step we deal with the case that T has more than six vertices and all the leaves of T' are anchors. We define the *second level skeleton tree* T'' to be the skeleton tree of T' . In other words, T'' is the tree which is obtained from T' by removing all the leaves. For clarity we use w to refer to a leaf of T'' , and we use $w, w',$ and w'' , respectively, to refer to the copies of vertex w in $T, T',$ and T'' . For now suppose that T'' has at least two vertices. Consider a leaf w'' and its neighbor y'' in T'' . Note that in T , w is connected to y , to at least one anchor, and possibly to some leaves of T . After Step 1, the copy of w'' in T' , i.e. w' , is connected to anchors v'_1, \dots, v'_k in T' (or v_1, \dots, v_k in

T) for some $1 \leq k \leq 4$, and connected to at most $4 - k$ leaves of T . In T , we consider the clockwise ordering of the non-leaf neighbors of w . Let this ordering be y, v_1, v_2, \dots, v_k . We denote the pair of leaves connected to anchor v_i by a_i and b_i in clockwise order around v_i ; see Figure 5.6.

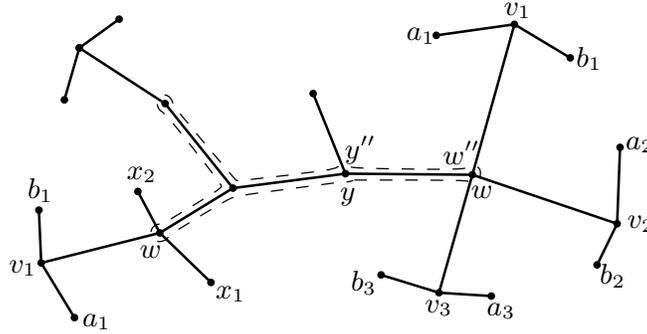


Figure 5.6: Second level skeleton tree T'' is surrounded by dashed line. v_i 's are ordered clockwise around leaf vertex w'' , as well as x_i 's. a_i and b_i are ordered clockwise around v_i .

In this step we pick an arbitrary leaf $w'' \in T''$ and according to the number of anchors incident to w'' , i.e. k , we add edges to M . Since $1 \leq k \leq 4$, four cases occur and we discuss each case separately. Before that, we state some lemmas.

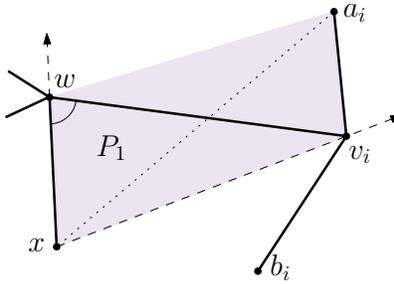


Figure 5.7: Illustrating Lemma 5.17.

LEMMA 5.17. *Let w'' be a leaf in T'' . Consider an anchor v_i which is adjacent to w in T . For any neighbor x of w for which $x \neq v_i$, if $\text{cw}(v_i wx) \leq \pi/2$ (resp. $\text{cw}(xwv_i) \leq \pi/2$), the polygon $P_1 = \{v_i, x, w, a_i\}$ (resp. $P_2 = \{v_i, b_i, w, x\}$) is convex and empty.*

Proof. We prove the case when $\text{cw}(v_i wx) \leq \pi/2$; see Figure 5.7. The proof for the second case is symmetric. To prove the convexity of P_1 we show that the diagonals $\overrightarrow{v_i w}$ and $\overrightarrow{a_i x}$ of P_1 intersect each other. To show the intersection we argue that a_i lies to the left of $\overrightarrow{xv_i}$ and to the right of \overrightarrow{xw} .

Consider $\overrightarrow{xv_i}$. According to Lemma 5.6, triangle $\Delta_{v_i x w}$ is empty so b_i lies to the right of $\overrightarrow{xv_i}$. On the other hand, v_i is an anchor, so $\text{cw}(a_i v_i b_i) \geq \pi$, and hence a_i lies to the left of $\overrightarrow{xv_i}$. Now consider \overrightarrow{xw} . For the sake of contradiction, suppose that a_i is to the left of \overrightarrow{xw} . Since $\text{cw}(v_i wx) \leq \pi/2$, the angle $\text{cw}(a_i w v_i)$ is greater than $\pi/2$. This means that $\overrightarrow{v_i a_i}$ is the largest side of $\Delta_{a_i w v_i}$, which contradicts that $\overrightarrow{v_i a_i}$ is an edge of $\text{MST}(P)$. So a_i lies to the right of \overrightarrow{xw} . Therefore, $\overrightarrow{v_i w}$ intersects $\overrightarrow{a_i x}$ and P_1 is convex. P_1 is empty because by Lemma 5.6, the triangles $\Delta_{v_i wx}$ and $\Delta_{v_i w a_i}$ are empty. \square

LEMMA 5.18. *Let w'' be a leaf in T'' and consider the clockwise sequence v_1, \dots, v_k of anchors that are incident on w . The sequence of vertices $a_1, v_1, b_1, \dots, a_k, v_k, b_k$ are angularly sorted in clockwise order around w .*

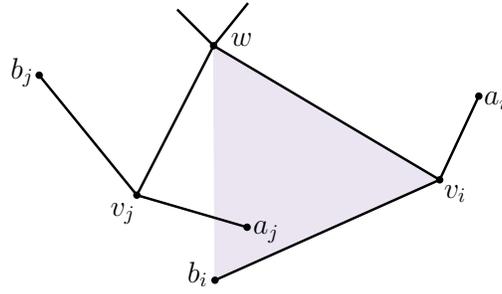


Figure 5.8: Proof of Lemma 5.18.

Proof. Using contradiction, consider two vertices v_i and v_j , and assume that v_i comes before v_j and a_j comes before b_i in the clockwise order; see Figure 5.8. Either a_j is in $\triangle wv_i b_i$ or b_i is in $\triangle wv_j a_j$. However, by Lemma 5.6, neither of these two cases can happen. \square

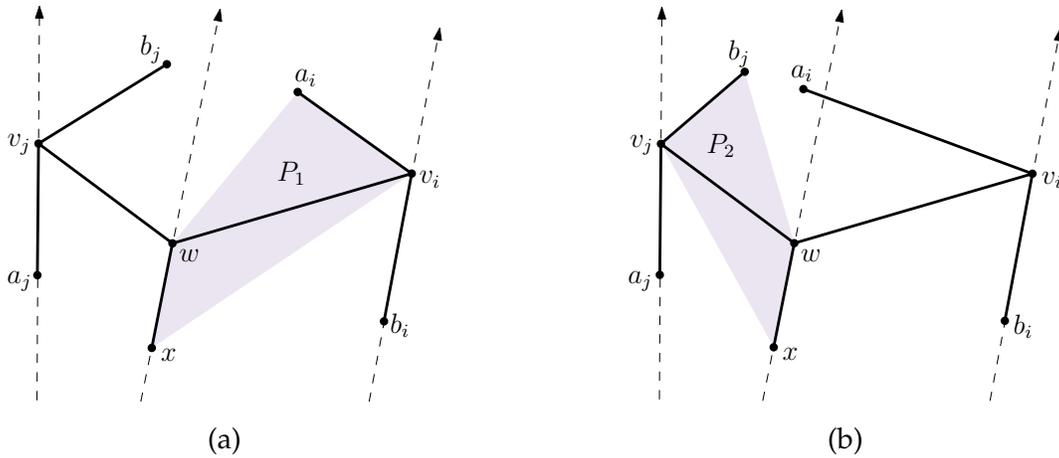


Figure 5.9: Illustrating Lemma 5.19.

LEMMA 5.19. *Let w'' be a leaf in T'' and consider the clockwise sequence v_1, \dots, v_k of anchors that are adjacent to w . Let $1 \leq i < j \leq k$ and let x be a neighbor of w for which $x \neq v_i$ and $x \neq v_j$.*

1. *If x is between v_i and v_j in the clockwise order:*
 - a) *if a_i is to the right of \overrightarrow{xw} , then $P_1 = \{x, w, a_i, v_i\}$ is convex and empty.*
 - b) *if a_i is not to the right of \overrightarrow{xw} , then $P_2 = \{w, x, v_j, b_j\}$ is convex and empty.*
2. *If v_j is between v_i and x , or v_i is between x and v_j in the clockwise order:*
 - a) *if b_i is to the left of \overrightarrow{xw} , then $P_3 = \{w, x, v_i, b_i\}$ is convex and empty.*
 - b) *if b_i is not to the left of \overrightarrow{xw} , then $P_4 = \{x, w, a_j, v_j\}$ is convex and empty.*

Proof. We only prove the first case, the proof for the second case is symmetric. Thus, we assume that x is between v_i and v_j in the clockwise order. First assume that a_i is to the right of \overrightarrow{xw} . See Figure 5.9(a). Consider $\overrightarrow{b_i v_i}$. Since v_i is an anchor, a_i cannot be to the right of $\overrightarrow{b_i v_i}$, and according to Lemma 5.6, x cannot be to the right of $\overrightarrow{b_i v_i}$. For the same reasons, both the vertices b_j and x cannot be to the left of $\overrightarrow{a_j v_j}$. Now consider \overrightarrow{xw} . By assumption, a_i is to the right of \overrightarrow{xw} . Therefore $\overrightarrow{x a_i}$ intersects $\overrightarrow{w v_i}$ and hence P_1 is convex.

Now assume that a_i is not to the right of \overrightarrow{xw} ; see Figure 5.9(b). By Lemma 5.18, b_j is to the left of \overrightarrow{xw} . Therefore, $\overrightarrow{xb_j}$ intersects $\overrightarrow{wv_j}$ and hence P_2 is convex. The emptiness of the polygons follows directly from Lemma 5.6. \square

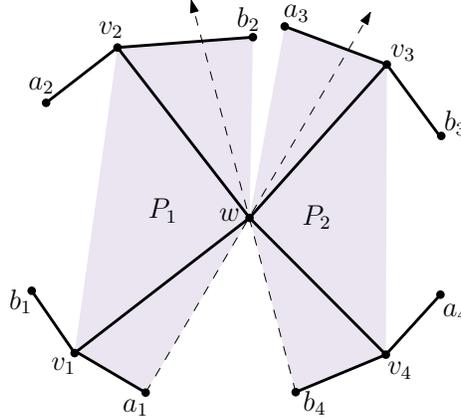


Figure 5.10: Illustrating Lemma 5.20.

LEMMA 5.20. *Let w'' be a leaf in T'' and consider the clockwise sequence v_1, \dots, v_k of anchors that are incident on w . If $4 \leq k \leq 5$, then at least one of the pentagons $P_1 = \{a_1, v_1, v_2, b_2, w\}$ and $P_2 = \{a_{k-1}, v_{k-1}, v_k, b_k, w\}$ is convex and empty.*

Proof. We prove the case when $k = 4$; the proof for $k = 5$ is analogous. By Lemma 5.18, a_1 comes before b_4 in the clockwise order. Consider $\overrightarrow{a_1w}$; see Figure 5.10. Let $l(a_1, b_2)$ denote the line through a_1 and b_2 . If b_2 is to the left of $\overrightarrow{a_1w}$, then $l(a_1, b_2)$ separates w from v_1 and v_2 . Otherwise, b_2 is to the right of $\overrightarrow{a_1w}$. Now consider $\overrightarrow{b_4w}$. If a_3 is to the right of $\overrightarrow{b_4w}$, then $l(a_3, b_4)$ separates w from v_3 and v_4 . The remaining case, i.e., when a_3 is to the left of $\overrightarrow{b_4w}$, cannot happen by Lemma 5.18 (because, otherwise, a_3 would come before b_2 in the clockwise order).

Thus, we have shown that (i) $l(a_1, b_2)$ separates w from v_1 and v_2 or (ii) $l(a_3, b_4)$ separates w from v_3 and v_4 . Assume w.l.o.g. that (i) holds. Now to prove the convexity of P_1 we show that all internal angles of P_1 are less than π . Since v_1 is an anchor, $\text{cw}(b_1v_1a_1) \leq \pi$. By Lemma 5.6, b_1 is to the left of $\overrightarrow{v_1v_2}$. Therefore, $\text{cw}(v_2v_1a_1) < \text{cw}(b_1v_1a_1) \leq \pi$. On the other hand $\text{cw}(wv_1a_1) \geq \pi/3$, so in Δa_1v_1w , $\text{cw}(v_1a_1w) \leq 2\pi/3$. By a similar analysis $\text{cw}(b_2v_2v_1)$ and $\text{cw}(wb_2v_2)$ are less than π . In addition, $\text{cw}(a_1wb_2)$ in Δa_1wb_2 is less than π . Thus, P_1 is convex. Its emptiness is assured from the emptiness of the triangles Δv_2wb_2 , Δv_1wa_1 and Δv_1wv_2 . \square

Now we are ready to present the details of Step 2. Recall that T has more than six vertices and all leaves of T' are anchors. In this case T'' has at least one vertex. If T'' has exactly one vertex, we add edges to M as will be described in Section 5.4.2.3 and after which the algorithm terminates. Assume that T'' has at least two vertices. Pick a leaf w'' in T'' . As before, let v_1, \dots, v_k where $1 \leq k \leq 4$ be the clockwise order of the anchors connected to w . Let x_1, \dots, x_ℓ where $\ell \leq 4 - k \leq 3$ be the clockwise order of the leaves of T connected to w ; see Figure 5.6. This means that $\text{deg}(w) = 1 + k + \ell$, where $k + \ell \leq 4$. Now we describe different configurations that may appear at w , according to k and ℓ .

CASE 1: assume $k = 1$. If $\ell = 0$ or 1, add (a_1, v_1) and (b_1, w) to M and set $T = T \setminus \{a_1, b_1, v_1, w\}$.

If $\ell = 1$ remove x_1 from T as well. If $\ell = 2$, consider α_i as the acute angle between segments

$\overline{wv_1}$ and $\overline{wx_i}$. W.l.o.g. assume $\alpha_1 = \min(\alpha_1, \alpha_2)$. Two cases may arise: (i) $\alpha_1 \leq \pi/2$, (ii) $\alpha_1 > \pi/2$. If (i) holds, w.l.o.g. assume that x_1 comes before v_1 in clockwise order around w . According to Lemma 5.17 polygon $P = \{v_1, b_1, w, x_1\}$ is convex and empty. So we add (v_1, a_1) , (b_1, x_1) and (x_2, w) to M . Other cases are handled in a similar way. If (ii) holds, according to Lemma 5.6 triangle Δx_1wx_2 is empty. So we add (v_1, a_1) , (b_1, w) and (x_1, x_2) to M . In both cases set $T = T \setminus \{a_1, b_1, v_1, x_1, x_2, w\}$. If $\ell = 3$, remove x_3 from T and handle the rest as $\ell = 2$.

CASE 2: assume $k = 2$. If $\ell = 0$, we add (v_1, a_1) , (b_1, w) , and (v_2, a_2) to M . If $\ell = 1$ suppose that x_1 comes before v_1 in clockwise ordering. According to Lemma 5.19 one of polygons P_3 and P_4 is empty; suppose it be $P_3 = \{w, x_1, v_1, b_1\}$ (where $i = 1$ and $j = 2$ in Lemma 5.19). Thus we add (v_1, a_1) , (b_1, x_1) , (v_2, a_2) and (b_2, w) to M . In both cases set $T = T \setminus \{a_1, b_1, v_1, a_2, b_2, v_2, w\}$ and if $\ell = 1$ remove x_1 from T as well. If $\ell = 2$, remove x_2 from T and handle the rest as $\ell = 1$.

CASE 3: assume $k = 3$. If $\ell = 0$ then set $M = M \cup \{(v_1, a_1), (b_1, w), (v_2, a_2), (v_3, a_3)\}$. If $\ell = 1$, consider β_i as the acute angle between segments $\overline{wx_1}$ and $\overline{wv_i}$. W.l.o.g. assume β_2 has minimum value among all β_i 's and x_1 comes after v_2 in clockwise order. According to Lemma 5.19 one of polygons P_1 and P_2 is empty; suppose it be $P_1 = \{w, x_1, v_2, a_2\}$ (where $i = 2$ and $j = 3$ in Lemma 5.19). Thus we set $M = M \cup \{(v_2, b_2), (a_2, x_1), (v_1, a_1), (b_1, w), (v_3, a_3)\}$. In both cases set $T = T \setminus \{a_1, b_1, v_1, a_2, b_2, v_2, a_3, b_3, v_3, w\}$ and if $\ell = 1$ remove x_1 from T as well. Other cases can be handled similarly.

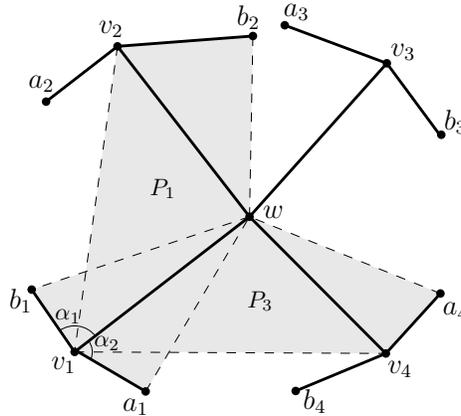


Figure 5.11: The vertex v_1 is shared between P_1 and P_3 .

CASE 4: assume $k = 4$. According to Lemma 5.20 one of $P_1 = \{a_1, v_1, v_2, b_2, w\}$ and $P_2 = \{a_3, v_3, v_4, b_4, w\}$ is convex and empty. Again by Lemma 5.20 one of $P_3 = \{a_4, v_4, v_1, b_1, w\}$ and $P_4 = \{a_2, v_2, v_3, b_3, w\}$ is also convex and empty. Without loss of generality assume that P_1 and P_3 are empty. See Figure 5.11. Clearly, these two polygons share a vertex (v_1 in Figure 5.11). Let $\alpha_1 = cw(b_1, v_1, w)$ which is contained in P_3 and $\alpha_2 = cw(w, v_1, a_1)$ which is contained in P_1 . We pick one of the polygons P_1 and P_3 which minimizes $\alpha_i, i = 1, 2$. Let P_1 be that polygon. So we set $M = M \cup \{(v_1, b_1), (v_2, a_2), (a_1, b_2), (v_3, a_3), (b_3, w), (v_4, a_4)\}$ and set $T = T \setminus \{a_1, b_1, v_1, a_2, b_2, v_2, a_3, b_3, v_3, a_4, b_4, v_4, w\}$.

This concludes Step 2. Go back to Step 1.

5.4.2.3 Base Cases

In this section we describe the base cases of our algorithm. As mentioned in Steps 1 and 2, we may have two base cases: (a) T has at most $t \leq 6$ vertices, (b) T'' has only one vertex.

(A) $t \leq 6$ Suppose that T has at most six vertices.

$t = 2$ it can happen only if $t = n = 2$, and we add the only edge to M .

$t = 4, 5$ in this case we match four vertices. If $t = 4$, T could be a star or a path of length three, and in both cases we match all the vertices. If $t = 5$, remove one of the leaves and match other four vertices.

$t = 6$ in this case we match all the vertices. If one of the leaves connected to a vertex of degree two, we match those two vertices and handle the rest as the case when $t = 4$, otherwise, each leaf of T is connected to a vertex of degree more than two, and hence T' has at most two vertices. Figure 5.12(a) shows the solution for the case when T' has only one vertex and T is a star; note that at least two angles are less than π . Now consider the case when T' has two vertices, v_1 and v_2 , which have degree three in T . Figure 5.12(b) shows the solution for the case when neither v_1 nor v_2 is an anchor. Figure 5.12(c) shows the solution for the case when v_1 is an anchor but v_2 is not. Figure 5.12(d) shows the solution for the case when both v_1 and v_2 are anchors. Since v_2 is an anchor in Figure 5.12(d), at least one of $cw(b_2v_2v_1)$ and $cw(v_1v_2a_2)$ is less than or equal to $\pi/2$. W.l.o.g. assume $cw(v_1v_2a_2) \leq \pi/2$. By Lemma 5.17 polygon $P = \{v_1, a_2, v_2, a_1\}$ is convex and empty. We add (v_1, b_1) , (v_2, b_2) , and (a_1, a_2) to M .

$t = 1, 3$ this case could not happen. Initially $t = n$ is even. Consider Step 1; before each iteration t is bigger than six and during the iteration two vertices are removed from T . So, at the end of Step 1, t is at least five. Now consider Step 2; before each iteration T'' has at least two vertices and during the iteration at most one vertex is removed from T'' . So, at the end of Step 2, T'' has at least one vertex that is connected to at least one anchor. This means that t is at least four. Thus, t could never be one or three before and during the execution of the algorithm.

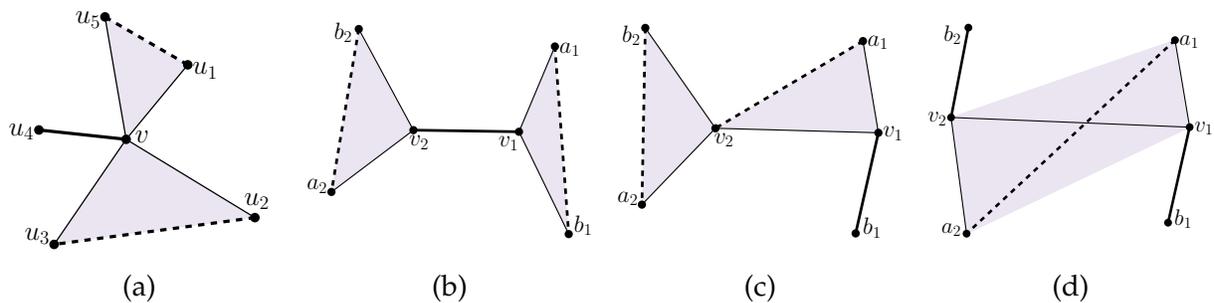


Figure 5.12: The bold (solid and dashed) edges are added to M and all vertices are matched. (a) a star, (b) no anchor, (c) one anchor, and (d) two anchors.

(B) T'' HAS ONE VERTEX In this case, the only vertex $w'' \in T''$ is connected to at least two anchors, otherwise w'' would have been matched in Step 1. So we consider different cases when w is connected to k , $2 \leq k \leq 5$ anchors and $\ell \leq 5 - k$ leaves of T :

$k = 2$ if $\ell = 0, 1, 2$ we handle it as Case 2 in Step 2. If $\ell = 3$, at least two leaves are consecutive, say x_1 and x_2 . Since $cw(x_1wx_2) < \pi$ we add (x_1, x_2) to M and handle the rest like the case when $\ell = 1$.

$k = 3$ if $\ell = 2$ remove x_2 from T . Handle the rest as Case 3 in Step 2.

$k = 4$ if $\ell = 1$ remove x_1 from T . Handle the rest as Case 4 in Step 2.

$k = 5$ add (v_5, a_5) to M , remove a_5, b_5, v_5 from T , and handle the rest as Case 4 in Step 2.

This concludes the algorithm.

LEMMA 5.21. *The convex empty regions that are considered in different iterations of the algorithm, do not overlap.*

Proof. In Step 1, Step 2 and the base cases, we used three types of convex empty regions; see Figure 5.13. Using contradiction, suppose that two convex regions P_1 and P_2 overlap. Since the regions are empty, no vertex of P_1 is in the interior of P_2 and vice versa. Then, one of the edges in $MST(P)$ that is shared with P_1 intersects some edge in $MST(P)$ that is shared with P_2 , which is a contradiction. \square

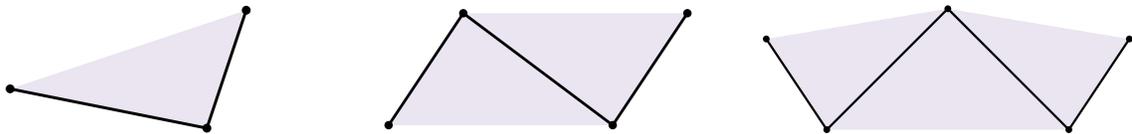


Figure 5.13: Empty convex regions. Bold edges belong to $MST(P)$.

THEOREM 5.22. *Let P be a set of n points in the plane, where n is even, and let λ^* be the minimum bottleneck length of any plane perfect matching of P . In $O(n \log n)$ time, a plane matching of P of size at least $\frac{2n}{5}$ can be computed, whose bottleneck length is at most $(\sqrt{2} + \sqrt{3})\lambda^*$.*

Proof. Proof of planarity: In each iteration, in Step 1, Step 2, and in the base cases, the edges added to M are edges of $MST(P)$ or edges inside convex empty regions. By Lemma 5.21 the convex empty regions in each iteration will not be intersected by the convex empty regions in the next iterations. Therefore, the edges of M do not intersect each other and M is plane.

Proof of matching size: In Step 1, in each iteration, all the vertices which are excluded from T are matched. In Step 2, when $k = \ell = 1$ we match four vertices out of five, and when $k = 3, \ell = 0$ we match eight vertices out of ten. In base case (a) when $t = 5$ we match four vertices out of five. In base case (b) when $k = 3, \ell = 0$ we match eight vertices out of ten. In all other cases of Step 2 and the base cases, the ratio of matched vertices is more than $4/5$. Thus, in each iteration at least $4/5$ of the vertices removed from T are matched and hence $|M_i| \geq \frac{2n_i}{5}$. Therefore,

$$|M| = \sum_{i=1}^k |M_i| \geq \sum_{i=1}^k \frac{2n_i}{5} = \frac{2n}{5}.$$

Proof of edge length: By Lemma 5.16 the length of edges of T is at most λ^* . Consider an edge $e \in M$ and the path δ between its end points in T . If e is added in Step 1, then $|e| \leq 2\lambda^*$ because δ has at most two edges. If e is added in Step 2, δ has at most three edges ($|e| \leq 3\lambda^*$) except in

Case 4. In this case we look at δ in more detail. We consider the worst case when all the edges of δ have maximum possible length λ^* and the angles between the edges are as big as possible; see Figure 5.14. Consider the edge $e = (a_1, b_2)$ added to M in Case 4. Since v_2 is an anchor and $\text{cw}(wv_2a_2) \geq \pi/3$, the angle $\text{cw}(b_2v_2w) \leq 2\pi/3$. As our choice between P_1 and P_3 in Case 4, $\text{cw}(w, v_1, a_1) \leq \pi/2$. Recall that e avoids w , and hence $|e| \leq (\sqrt{2} + \sqrt{3})\lambda^*$. The analysis for the base cases is similar.

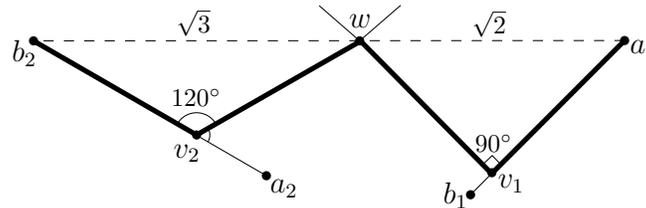


Figure 5.14: Path δ (in bold) with four edges of length λ^* between end points of edge (a_1, b_2) .

Proof of complexity: The Delaunay triangulation of P can be computed in $O(n \log n)$ time. Using Kruskal's algorithm, the forest F of even components can be computed in $O(n \log n)$ time. In Step 1 (resp. Step 2) in each iteration, we pick a leaf of T' (resp. T'') and according to the number of leaves (resp. anchors) connected to it we add some edges to M . Note that in each iteration we can update T, T' and T'' by only checking the two hop neighborhood of selected leaves. Since the two hop neighborhood is of constant size, we can update the trees in $O(1)$ time in each iteration. Thus, the total running time of Step 1, Step 2, and the base cases is $O(n)$ and the total running time of the algorithm is $O(n \log n)$. \square

5.5 CONCLUSIONS

We considered the NP-hard problem of computing a bottleneck plane perfect matching of a point set. Abu-Affash et al. [2] presented a $2\sqrt{10}$ -approximation for this problem. We used the maximum plane matching problem in unit disk graphs (UDG) as a tool for approximating a bottleneck plane perfect matching. In Section 5.3.1 we presented a polynomial time algorithm which computes a plane matching of size $\frac{n}{6}$ in UDG. We also presented a $\frac{2}{5}$ -approximation algorithm for computing a maximum matching in UDG. By extending this algorithm we showed how one can compute a bottleneck plane matching of size $\frac{n}{5}$ with edges of optimum-length. A modification of this algorithm gives us a plane matching of size at least $\frac{2n}{5}$ with edges of length at most $\sqrt{2} + \sqrt{3}$ times the optimum.

BIBLIOGRAPHY

- [1] A. K. Abu-Affash, A. Biniiaz, P. Carmi, A. Maheshwari, and M. H. M. Smid. Approximating the bottleneck plane perfect matching of a point set. *Computational Geometry: Theory and Applications*, 48(9):718–731, 2015.
- [2] A. K. Abu-Affash, P. Carmi, M. J. Katz, and Y. Trabelsi. Bottleneck non-crossing matching in the plane. *Computational Geometry: Theory and Applications*, 47(3):447–457, 2014.
- [3] G. Aloupis, J. Cardinal, S. Collette, E. D. Demaine, M. L. Demaine, M. Dulieu, R. F. Monroy, V. Hart, F. Hurtado, S. Langerman, M. Saumell, C. Seara, and P. Taslakian. Non-crossing

- matchings of points with geometric objects. *Computational Geometry: Theory and Applications*, 46(1):78–92, 2013.
- [4] J. Carlsson and B. Armbruster. A bottleneck matching problem with edge-crossing constraints. Manuscript, see <http://users.iems.northwestern.edu/~armbruster/2010matching.pdf>, 2010.
- [5] M.-S. Chang, C. Y. Tang, and R. C. T. Lee. Solving the Euclidean bottleneck matching problem by k -relative neighborhood graphs. *Algorithmica*, 8(3):177–194, 1992.
- [6] M. de Berg, O. Cheong, M. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.
- [7] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [8] Édouard Bonnet and T. Miltzow. Flip distance to a non-crossing perfect matching. In *Proceedings of the 32nd European Workshop on Computational Geometry (EuroCG'16)*, pages 79–82, 2016.
- [9] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.
- [10] A. Efrat and M. J. Katz. Computing Euclidean bottleneck matchings in higher dimensions. *Information Processing Letters*, 75(4):169–174, 2000.
- [11] T. Lengauer. *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons, New York, 1990.
- [12] C. L. Monma and S. Suri. Transitions in geometric minimum spanning trees. *Discrete & Computational Geometry*, 8:265–293, 1992.
- [13] P. M. Vaidya. Geometry helps in matching. *SIAM Journal on Computing*, 18(6):1201–1225, 1989.
- [14] K. R. Varadarajan. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In *FOCS*, pages 320–331, 1998.

Given a set of n red points and n blue points in the plane, we are interested to match the red points with the blue points by straight line segments in such a way that the segments do not cross each other and the length of the longest segment is minimized. In general, this problem is NP-hard. We give exact solutions for some special cases of the input point set.

This chapter is published in the proceedings of the 26th Canadian Conference on Computational Geometry (CCCG 2014) [3].

6.1 INTRODUCTION

We study the problem of computing a bottleneck non-crossing matching of red and blue points in the plane. Let $R = \{r_1, \dots, r_n\}$ be a set of n red points and $B = \{b_1, \dots, b_n\}$ be a set of n blue points in the plane. A *RB-matching* is a non-crossing perfect matching of the points by straight line segments in such a way that each segment has one endpoint in B and one in R . The length of the longest edge in an RB-matching M is known as *bottleneck* which we denote by λ_M . The *bottleneck bichromatic matching* (BBM) problem is to find a non-crossing matching M^* with minimum bottleneck λ^* . Carlsson et al. [4] showed that the bottleneck bichromatic matching problem is NP-hard. Moreover, when all the points have the same color, the bottleneck non-crossing perfect matching problem is NP-hard [1].

Notice that, the bottleneck (possibly crossing) perfect matching of red and blue points can be computed exactly in $O(n^{1.5} \log n)$ time [5]. In addition, a non-crossing perfect matching of red and blue points always exists and can be computed in $O(n \log n)$ time by applying the ham sandwich cut recursively. In [2] the authors considered the problem of non-crossing matching of points with different geometric objects.

In this chapter we present exact solutions for some special cases of the BBM problem when the points are arranged in convex position, boundary of a circle, and on a line. For simplicity, in the rest of the chapter we refer to a RB-matching as a “matching”.

6.2 POINTS IN CONVEX POSITION

In this section we deal with the case when $R \cup B$ form the vertices of a convex polygon. Carlsson et al. [4] presented an $O(n^4 \log n)$ -time algorithm for points on convex position. We improve their result to $O(n^3)$ time. Let P denote the union of R and B , that is $P = \{r_1, \dots, r_n, b_1, \dots, b_n\}$. We have the following observation:

OBSERVATION 6.1. *Let (r_i, b_j) be an edge in any RB-matching of P , then there are the same number of red and blue points on each side of the line passing through r_i and b_j .*

Using Observation 6.1, we present a dynamic programming algorithm which solves the BBM problem for P . For simplicity of notation, let $P = \{p_1, \dots, p_{2n}\}$ denote the sequence of the vertices of the convex polygon in counter clockwise order, starting at an arbitrary vertex p_1 ; see Figure 6.1. By Observation 6.1, we denote (p_i, p_j) as a *feasible edge* if p_i and p_j have different colors and

the sequence p_{i+1}, \dots, p_{j-1} contains the same number of red and blue points. In other words we say that p_j is a *feasible match* for p_i , and vice versa. Let F_i denote the set of feasible matches for p_i . Figure 6.1 shows that $F_1 = \{p_4, p_8, p_{10}\}$. Therefore, we define a weight function w which assigns a weight $w_{i,j}$ to each pair (p_i, p_j) , where

$$w_{i,j} = \begin{cases} |p_i p_j| & : \text{if } (p_i, p_j) \text{ is a feasible edge} \\ +\infty & : \text{otherwise} \end{cases}$$

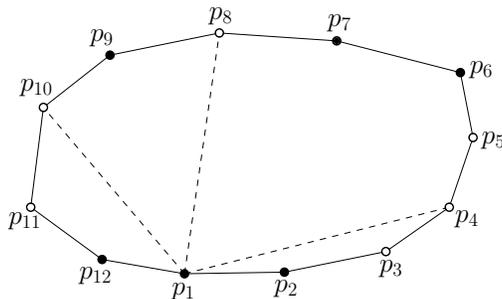


Figure 6.1: Points arranged on convex position.

Consider any subsequence $P_{i,j} = \{p_i, \dots, p_j\}$ of P , where $1 \leq i < j \leq 2n$. Let $A[i, j]$ denote the bottleneck of the optimal matching in $P_{i,j}$ if $P_{i,j}$ has an RB-matching; otherwise, $A[i, j] = +\infty$. So $A[1, 2n]$ denotes the optimal solution for P . We use dynamic programming to compute $A[1, 2n]$. We derive a recurrence for $A[i, j]$. For a feasible edge (p_i, p_k) where $i + 1 \leq k \leq j$ and $p_k \in F_i$, the values of the sub-problems to the left and right of (p_i, p_k) are $A[i + 1, k - 1]$ and $A[k + 1, j]$. We match p_i to a feasible point p_k which minimizes the bottleneck. Thus,

$$A[i, j] = \min_{\substack{i+1 \leq k \leq j \\ p_k \in F_i}} \{\max\{w_{i,k}, A[i + 1, k - 1], A[k + 1, j]\}\}.$$

The size of A (which is the total number of sub-problems) is $O(n^2)$. For each sub-problem $A[i, j]$ we have at most $k = j - i$ lookups in A . Therefore, the total running time is $O(n^3)$.

THEOREM 6.2. *Given a set B of n blue points and a set R of n red points in convex position, one can compute a bottleneck non-crossing RB-matching in time $O(n^3)$ and in space $O(n^2)$.*

Note that in [1] the authors showed that for points in convex position and when all the points have the same color, a bottleneck plane matching can be computed in $O(n^3)$ time and $O(n^2)$ space via dynamic programming. They obtained the same time and space complexities for the bichromatic set of points.

6.2.1 Points on Circle

In this section we consider the BBM problem when the points in R and B are arranged on the boundary of a circle. Clearly, we can use the same algorithm as for points in convex position to solve this problem in $O(n^3)$ time. But for points on a circle we can do better; we present an algorithm running in $O(n^2)$ time. Consider $P = \{p_1, \dots, p_{2n}\}$ as the sequence of the points in counter clockwise order on a circle. We prove that there is an optimal matching M^* , such that each point $p_i \in P$ is connected to its first feasible match in the clockwise or counter clockwise order from p_i .

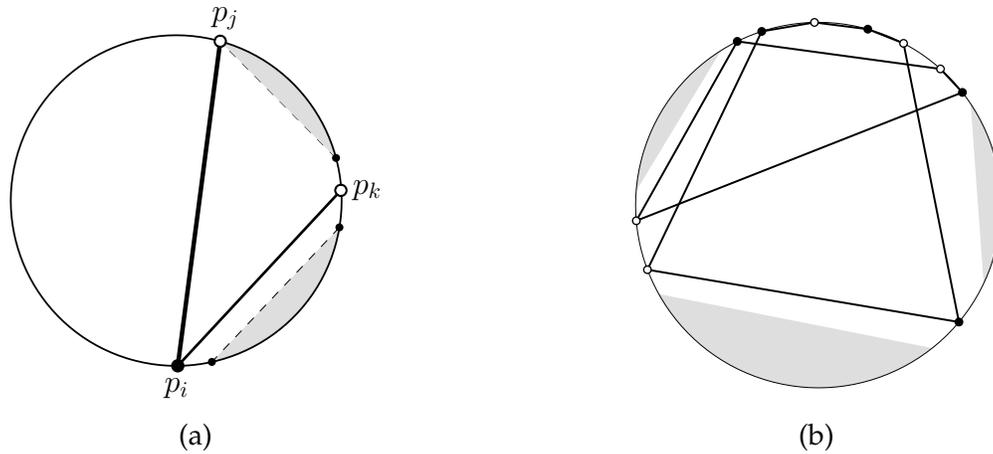


Figure 6.2: (a) illustrating the proof of Lemma 6.3, (b) the resulting graph of procedure CompareToOpt.

LEMMA 6.3. *There is an optimal RB-matching for a point set P on a circle, such that each $p_i \in P$ is connected to its first feasible match in the clockwise or counter clockwise order from p_i .*

Proof. Consider an optimal matching M^* with an edge (p_i, p_j) . Consider two arcs $\widehat{p_i p_j}$ and $\widehat{p_j p_i}$. W.l.o.g. let $\widehat{p_i p_j}$ be the smaller one. Clearly, the distance between any two points on $\widehat{p_i p_j}$ is at most $|p_i p_j|$. If p_{i+1}, \dots, p_{j-1} contains no feasible match for p_i , then p_j is the first feasible match to the right of p_i . Otherwise, let p_k be the first feasible match for p_i in $\widehat{p_i p_j}$; see Figure 6.2(a). By connecting p_i to p_k we have two smaller arcs $\widehat{p_{i+1} p_{k-1}}$ and $\widehat{p_{k+1} p_j}$. Obviously, $|p_i p_k| < |p_i p_j|$, and any matching of the vertices on $\widehat{p_{i+1} p_{k-1}}$ and $\widehat{p_{k+1} p_j}$ have the bottleneck smaller than $|p_i p_j|$. By repeating this process for all edges of M^* and all new edges, we obtain a matching M which satisfies the statement of the lemma and $\lambda_M \leq \lambda^*$. \square

As a result of Lemma 6.3, for each point $p_i \in P$, we consider at most two feasible matches in F_i . Thus, using the dynamic programming idea of the previous section, for each sub-problem $A[i, j]$ we have at most two lookups in A . Thus, it takes $O(n^2)$ time to fill the table A . By preprocessing P , for each point $p_i \in P$ we can find its first matched points in $O(n^2)$ time. Thus, the total running time of the algorithm is $O(n^2)$.

6.2.1.1 A Faster Algorithm

Let $R = \{r_1, \dots, r_n\}$ be a set of n red points and $B = \{b_1, \dots, b_n\}$ be a set of n blue points on the boundary of a circle C . Without loss of generality let $P = \{p_1, \dots, p_{2n}\}$ be the clockwise ordered set of all the points. In this section we present an $O(n \log n)$ time algorithm which solves the BBM problem for P .

Let F_i denote the first feasible matches of p_i in clockwise and counter-clockwise order. Note that $|F_i| \leq 2$. We describe how one can compute F_i for all points in P in linear time. First, consider the case that we are looking for the first clockwise-feasible match for each red point. We make a copy P' of P . Consider an empty stack, and start from an arbitrary red point r_{start} and walk on P' clockwise. If we see a red point, push it onto the stack. If we see a blue point p_j and the stack is not empty, we pop a red point p_i from the stack and add p_j to F_i , and delete p_i and p_j from P' . If we see a blue point p_j and stack is empty, we do nothing. The process stops as soon as we find the proper match for each red vertex. As we visit each point in P at most twice, this

step takes linear time. We can do the same process for the counter-clockwise order. Therefore, F_i for all $1 \leq i \leq 2n$ can be computed in $O(n)$ time.

Let F denote the set of all feasible edges, in sorted order of their lengths. Let G be the graph with vertex set P and edge set F . Note that the degree of each vertex in G is at most two and hence the total number of edges is $2n$. Let G_λ be the subgraph of G containing all the edges of length at most λ . Our algorithm performs a binary search on the edges in G and for each considered edge e , we use the following procedure to decide whether G_λ , where $\lambda = |e|$, has a non-crossing perfect matching. The running time of the algorithm is $O(n \log n)$.

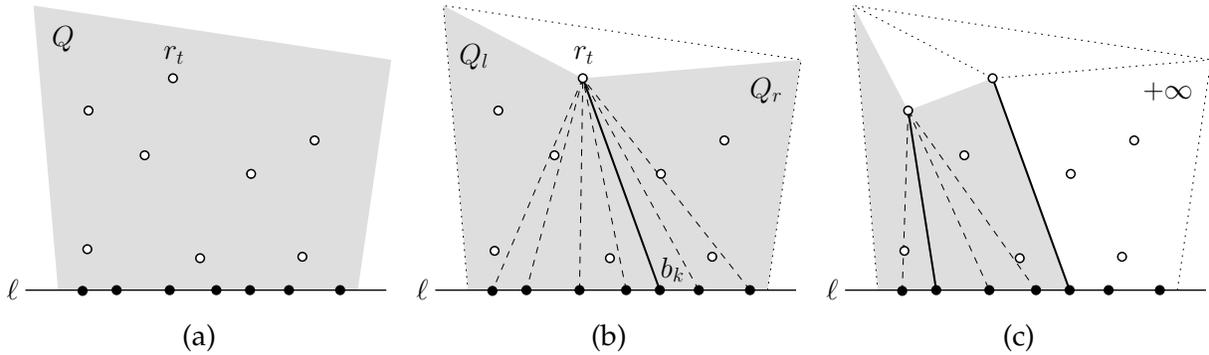


Figure 6.3: (a) definition of a sub-problem, (b) possible matching edges for r_t , and (c) Q_r returns $+\infty$ as it does not contain a matching; recurse on Q_l .

For each edge $e = (p_i, p_j)$ in G_λ let I_e be the set of all vertices of P in the smaller arc between p_i and p_j , including p_i and p_j . Let P_0 and P_1 be the lists of vertices of degree zero and one in G_λ , respectively. If P_0 is non-empty, then it is obvious that a perfect matching does not exist. If P_0 is empty and P_1 is non-empty, then for each point $p \in P_1$, do the following. Let $e = (p, q)$ be the only edge incident to p . It is obvious that any perfect matching in G_λ should contain e . In addition, (p, q) is a feasible edge, and then all the points in I_e can be matched properly. Thus, we can remove the points of I_e from G_λ . Note that this changes the lists P_0 and P_1 . The algorithm CompareToOpt receives G_λ as input and decides whether it has a perfect non-crossing matching.

Algorithm 5 CompareToOpt(G_λ)

Input: a graph G_λ

Output: TRUE, if G_λ has a non-crossing perfect matching, FALSE, otherwise

- 1: $P_0 \leftarrow$ vertices of degree zero in G_λ
 - 2: $P_1 \leftarrow$ vertices of degree one in G_λ
 - 3: **while** $P_0 \neq \emptyset$ or $P_1 \neq \emptyset$ **do**
 - 4: **if** $P_0 \neq \emptyset$ **then return** FALSE
 - 5: $p \leftarrow$ a vertex in P_1
 - 6: $q \leftarrow$ the vertex adjacent to p in G_λ
 - 7: **for each** r in $I_{(p,q)}$ **do**
 - 8: remove r and its adjacent edges from G_λ
 - 9: update P_0 and P_1
 - 10: **return** TRUE
-

The algorithm CompareToOpt consider each vertex and each edge once, so it executes in linear in the size of G_λ . At the end of the while loop, we have $P_0 = P_1 = \emptyset$. All the vertices of the

remaining part of G_λ have degree two and this case is the same as the problem that we started with (BBM problem) and by Lemma 6.3, it has a perfect non-crossing matching, thus we return TRUE. See Figure 6.2(b).

Notice that, if the procedure returns FALSE for some λ , then we know that $\lambda < \lambda^*$. Let e be the shortest edge for which the procedure returns TRUE. Thus $|e| \geq \lambda^*$, and a bottleneck RB-matching is contained in G_λ , where $\lambda = |e|$.

THEOREM 6.4. *Given a set B of n blue points and a set R of n red points on a circle, one can compute a bottleneck non-crossing RB-matching in time $O(n \log n)$ and in space $O(n)$.*

6.3 BLUE POINTS ON STRAIGHT LINE

In this section we deal with the case where the blue points are on a horizontal line and the red points are on one side of the line. Formally, given a sequence $B_{1,n} = b_1, \dots, b_n$ of n blue points on a horizontal line ℓ and n red points above ℓ , we are interested to find a non-crossing matching M between the points in R and B , such that the length of the longest edge in M is minimized. We show how to build dynamic programming algorithms that solve this problem. In Section 6.3.1 we present a bottom-up dynamic programming algorithm that solves this problem in $O(n^5)$ time. In Section 6.3.2 we present a top-down dynamic programming algorithm for this problem running in $O(n^4)$ time.

6.3.1 First algorithm

In this section we present a dynamic programming algorithm for the problem. We define a subproblem (R', B') in the following way: given a quadrilateral Q with one face on ℓ , we are looking for a bottleneck RB-matching in Q , where $R' = R \cap Q$ and $B' = B \cap Q$. For simplicity, we may refer to the sub-problem (R', B') as its bounding box Q . In the top level we imagine a bounding quadrilateral which contains all the points of R and B . See Figure 6.3(a). Let $b(Q)$ denote the bottleneck of the sub-problem Q . If Q is empty, we set $b(Q) = 0$. If Q is not empty but $|R'| \neq |B'|$, we set $b(Q) = +\infty$, as it is not possible to have a RB-matching for (R', B') . Otherwise, we have $|R'| = |B'| > 0$; let r_t be the topmost red point in R' in Q . It has at most $|B'|$ possible matching edges. Each of the matching edges defines two new independent sub-problems Q_l and Q_r to its left and right sides, respectively. See Figures 6.3(b) and 6.3(c). Thus, we can compute the bottleneck of a sub-problem Q , using the following recursion:

$$b(Q) = \min_{b_k \in B'} \{ \max\{|r_t b_k|, b(Q_l), b(Q_r)\} \}.$$

Note that the y -coordinate of all the red points in Q_l and Q_r are smaller than y -coordinate of r_t . If we recurse this process on Q_l and Q_r , it is obvious that each sub-problem (R', B') is bounded by the left and right sides of its corresponding quadrilateral. Thus, each sub-problem is defined by a pair of edges (or possibly the edges of the outer bounding box).

Note that the total number of edges is $n^2 + 2$ (including the edges of the outer box). The dynamic programming table contains $n^2 + 2$ rows and $n^2 + 2$ columns, each corresponds to an edge. The cells correspond to sub-problems. The dynamic programming table contains $O(n^4)$ cells, and for each we have at most n pairs of possible sub-problems, which implies at most $2n$ lookups in the table. Therefore, the algorithm runs in time $O(n^5)$ and space $O(n^4)$.

6.3.2 Second algorithm

In this section we present a top-down dynamic programming algorithm that improves the result of Section 6.3.1. Consider the problem (R, B) , where $B = B_{1,n} = \{b_1, \dots, b_n\}$. Let r_t be the topmost red point. In any solution M to the problem, consider the edge $(r_t, b_k) \in M$ which matches r_t to a point b_k in B , then there is no edge in M that intersects (r_t, b_k) . Thus, (r_t, b_k) is a feasible edge if on each side of (r_t, b_k) the number of red points equals the number of blue points. In this case, b_k is a feasible match for r_t . Recall that F_t denotes the set of all feasible matches for r_t . See Figure 6.4(a). In other words,

$$F_t = \{k : (r_t, b_k) \text{ is a feasible edge}\}.$$

For a feasible edge (r_t, b_k) , let R_l (resp. R_r) and B_l (resp. B_r) be the red and blue points to the left (resp. right) of (r_t, b_k) , respectively. That is, the edge (r_t, b_k) divides the (R, B) problem into two sub-problems (R_l, B_l) and (R_r, B_r) , where $|R_l| = |B_l|$ and $|R_r| = |B_r|$. Clearly, $B_l = B_{1,k-1} = \{b_1, \dots, b_{k-1}\}$ and $B_r = B_{k+1,n} = \{b_{k+1}, \dots, b_n\}$. We develop the following recurrence to solve the problem:

$$b(R, B) = \min_{k \in F_t} \{\max\{|r_t b_k|, b(B_l, R_l), b(B_r, R_r)\}\}.$$

Let l_t denote the horizontal line passing through r_t . Note that the y -coordinate of all red points in R_l and R_r is smaller than the y -coordinate of r_t , and hence they lie below l_t . This implies that the left (resp. right) sub-problem is contained in a trapezoidal region T_l (resp. T_r) with bounding edges ℓ , l_t , and (r_t, b_k) . See Figure 6.4(a). Since, in each step we have two sub-problems, in the rest of this section we describe the process for the right sub-problem; the process for the left sub-problem is symmetric. Note that r_t is the top-left corner of the right sub-problem. Thus, given B_r and r_t , we know that r_t is connected to a blue point immediately to the left of B_r . In addition, we can find the red points assigned to the right sub-problem in the following way. Stand at a blue point immediately to the right of B_r and scan the plane clockwise, starting from ℓ . Count the red points in T_r while scanning, and stop as soon as the number of red points seen equals the number of blue points in B_r . These red points form the set R_r . See Figures 6.4(b) and 6.4(c).

Since, r_t defines the right (resp. left) and top boundaries of T_l (resp. T_r) which contains the left (resp. right) sub-problem, we call r_t a “boundary vertex”. We define a sub-problem as a sequence $B_{i,j} = \{b_i, \dots, b_j\}$ of blue points, a boundary vertex, r_t , connected to b_{j+1} (resp. b_{i-1}) for the left (resp. right) sub-problem. More precisely, a sub-problem $(B_{i,j}, r_t, d)$ consists of an interval $B_{i,j}$, a boundary vertex r_t , and a direction $d = \{\text{left}, \text{right}\}$ which indicates that r_t is connected to a point immediately to the left or to the right of $B_{i,j}$. For a sub-problem $(B_{i,j}, t, d)$, where $d = \text{left}$ we find the vertex set $R_{i,j}$ in the following way. Scan the plane by a clockwise rotating line s anchored at b_{j+1} . Count the red points in trapezoidal region formed by ℓ , l_t , and (r_t, b_{i-1}) , and stop as soon as $j - i + 1$ red points have been encountered. These red points form the set $R_{i,j}$. See Figures 6.4(b) and 6.4(c).

In the top level, we add points b_0 and b_{n+1} on ℓ to the left and right of B , respectively. We add a point r_0 as the boundary vertex of the (R, B) problem in such a way that R and B are contained in the trapezoid formed by ℓ , l_0 , and the line segment $r_0 b_0$. Thus in the top level we have the sub-problem $(B_{1,n}, r_0, \text{left})$.

The dynamic programming table is a four-dimensional table $A[1..n, 1..n, 0..n, 1..2]$, where the first and second dimensions correspond to an interval of blue points, the third dimension corresponds a boundary vertex, and the fourth dimension corresponds to the directions. For simplicity

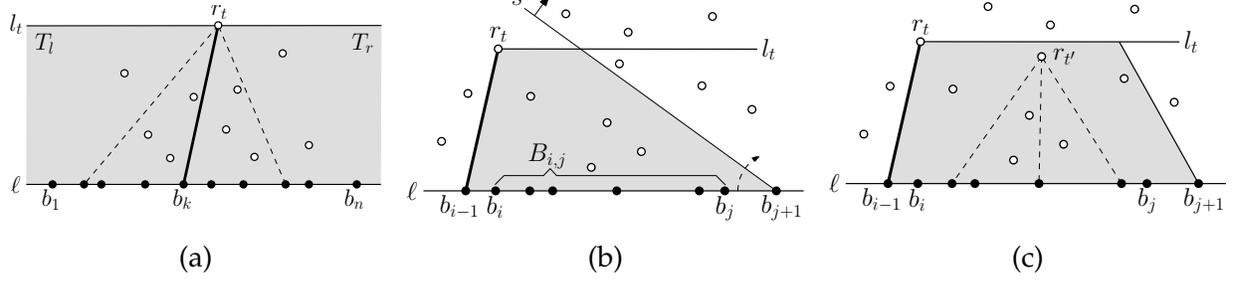


Figure 6.4: (a) feasible matches for r_t , (b) scanning the red points in the trapezoidal region (shaded area), and (c) the trapezoidal region which contains the same number of red and blue points.

we use l and r for left and right directions, respectively. Each cell $A[i, j, t, d]$ stores the bottleneck of the sub-problem $(B_{i,j}, r_t, d)$, and we are looking for $A[1, n, 0, l]$ which corresponds to the bottleneck of M^* . We fill A in the following way:

$$A[i, j, t, d] = \min_{k \in F_{t'}} \{ \max\{|r_t' b_k|, A[i, k-1, t', r], A[k+1, j, t', l]\} \},$$

where $r_{t'}$ is the topmost red point in the point set $R_{i,j}$ assigned to $(B_{i,j}, t, d)$.

Algorithm 6 computes the bottleneck of each subproblem using top-down dynamic programming. In the top level, we execute $\text{LineMatching}(1, n, 0, l)$. Before running algorithm LineMatching , for each point, we pre-sort the red points in the following way. For each red point r , we keep a sorted list of all the red points below l_r in clockwise order. For each blue point, we keep two sorted lists of red points in clockwise and counter-clockwise orders. This step takes $O(n^2 \log n)$ time.

Algorithm 6 $\text{LineMatching}(i, j, t, d)$

Input: sequence $B_{i,j}$, top point r_t , and direction d .

Output: bottleneck of M^* .

- 1: **if** $A[i, j, t, d] > 0$ **then**
 - 2: **return** $A[i, j, t, d]$
 - 3: **if** $i > j$ **then**
 - 4: **return** $A[i, j, t, d] \leftarrow 0$
 - 5: $R_{i,j} \leftarrow j - i + 1$ red points assigned to $B_{i,j}$
 - 6: $t' \leftarrow \text{top-index}(R_{i,j})$
 - 7: **if** $i = j$ **then**
 - 8: **return** $A[i, j, t, d] \leftarrow |r_{t'} b_i|$
 - 9: $b \leftarrow +\infty$
 - 10: $F_{t'} \leftarrow$ indices of feasible blue points for $r_{t'}$
 - 11: **for each** $k \in F_{t'}$ **do**
 - 12: $A[i, k-1, t', r] \leftarrow \text{LineMatching}(i, k-1, t', r)$
 - 13: $A[k+1, j, t', l] \leftarrow \text{LineMatching}(k+1, j, t', l)$
 - 14: $m \leftarrow \max\{|r_{t'} b_k|, A[i, k-1, t', r], A[k+1, j, t', l]\}$
 - 15: **if** $m < b$ **then**
 - 16: $b \leftarrow m$
 - 17: **return** $A[i, j, t, d] \leftarrow b$
-

LEMMA 6.5. *Algorithm LineMatching computes the bottleneck of M^* in $O(n^4)$ time.*

Proof. Each cell $A[i, j, t, d]$ corresponds to a sub-problem formed by an interval $B_{i,j}$, a boundary vertex r_t , and a direction d . The total number of possible $B_{i,j}$ intervals is $\binom{n}{2} + n$ (i can be equal to j). For each interval, any of the n red points can be the corresponding boundary vertex, which can be connected to the left or right side of the interval. Thus, the total number of subproblems is $2n\binom{n}{2} + 2n^2 = O(n^3)$. In order to compute $R_{i,j}$ for each sub-problem, we use the sorted lists assigned to b_{i-1} (or b_{j+1}) and scan for the red points in the trapezoidal region. To compute the feasible blue vertices for $r_{t'} \in R_{i,j}$, we use the sorted list assigned to $r_{t'}$ and keep track of feasible matches for $r_{t'}$ in $B_{i,j}$. Thus, for each sub-problem, we can compute $R_{i,j}$, $r_{t'}$, and $F_{t'}$ in linear time. Therefore, the total running time of the algorithm is $O(n^4)$. \square

Finally, we reconstruct M^* from A in linear time.

THEOREM 6.6. *Given a set B of n blue points on a horizontal line ℓ , a set R of n red points above ℓ , one can compute a bottleneck non-crossing RB-matching in time $O(n^4)$ and in space $O(n^3)$.*

BIBLIOGRAPHY

- [1] A. K. Abu-Affash, P. Carmi, M. J. Katz, and Y. Trabelsi. Bottleneck non-crossing matching in the plane. *Computational Geometry: Theory and Applications*, 47(3):447–457, 2014.
- [2] G. Aloupis, J. Cardinal, S. Collette, E. D. Demaine, M. L. Demaine, M. Dulieu, R. F. Monroy, V. Hart, F. Hurtado, S. Langerman, M. Saumell, C. Seara, and P. Taslakian. Non-crossing matchings of points with geometric objects. *Computational Geometry: Theory and Applications*, 46(1):78–92, 2013.
- [3] A. Biniaz, A. Maheshwari, and M. H. M. Smid. Bottleneck bichromatic plane matching of points. In *Proceedings of the 26th Canadian Conference on Computational Geometry, CCCG'14*, pages 431–435, 2014.
- [4] J. Carlsson and B. Armbruster. A bottleneck matching problem with edge-crossing constraints. Manuscript, see <http://users.iems.northwestern.edu/~armbruster/2010matching.pdf>, 2010.
- [5] A. Efrat, A. Itai, and M. J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.

Let P be a set of n points in general position in the plane which is partitioned into *color* classes. P is said to be *color-balanced* if the number of points of each color is at most $\lfloor n/2 \rfloor$. Given a color-balanced point set P , a *balanced cut* is a line which partitions P into two color-balanced point sets, each of size at most $2n/3 + 1$. A *colored matching* of P is a perfect matching in which every edge connects two points of distinct colors by a straight line segment. A *plane colored matching* is a colored matching which is non-crossing. In this chapter, we present an algorithm which computes a balanced cut for P in linear time. Consequently, we present an algorithm which computes a plane colored matching of P optimally in $\Theta(n \log n)$ time.

This chapter is published in the proceedings of the 14th International Symposium on Algorithms and Data Structures (WADS'15) [8].

We also extended the results of this chapter to the case where the points are in a simple polygon; the extension is not included in this thesis. However, the results have been published in the Proceedings of the First international conference on Topics in Theoretical Computer Science (TTCS'15) [6], and also published in the journal of Computational Geometry: Theory and Applications [7].

7.1 INTRODUCTION

Let P be a set of n points in general position (no three points on a line) in the plane. Assume P is partitioned into *color* classes, i.e., each point in P is colored by one of the given colors. P is said to be *color-balanced* if the number of points of each color is at most $\lfloor n/2 \rfloor$. In other words, P is color-balanced if no color is in strict majority. For a color-balanced point set P , we define a *feasible cut* as a line ℓ which partitions P into two point sets Q_1 and Q_2 such that both Q_1 and Q_2 are color-balanced. In addition, if the number of points in each of Q_1 and Q_2 is at most $2n/3 + 1$, then ℓ is said to be a *balanced cut*. The well-known ham-sandwich cut (see [13]) is a balanced cut: given a set of $2m$ red points and $2m$ blue points in general position in the plane, a ham-sandwich cut is a line ℓ which partitions the point set into two sets, each of them having m red points and m blue points. Feasible cuts and balanced cuts are useful for convex partitioning of the plane and for computing plane structures, e.g., plane matchings and plane spanning trees.

Assume n is an even number. Let $\{R, B\}$ be a partition of P such that $|R| = |B| = n/2$. Let $K_n(R, B)$ be the complete bipartite geometric graph on P which connects every point in R to every point in B by a straight-line edge. An *RB-matching* in P is a perfect matching in $K_n(R, B)$. Assume the points in R are colored red and the points in B are colored blue. An *RB-matching* in P is also referred to as a *red-blue matching* or a *bichromatic matching*. A *plane RB-matching* is an *RB-matching* in which no two edges cross. Let $\{P_1, \dots, P_k\}$, where $k \geq 2$, be a partition of P . Let $K_n(P_1, \dots, P_k)$ be the complete multipartite geometric graph on P which connects every point in P_i to every point in P_j by a straight-line edge, for all $1 \leq i < j \leq k$. Imagine the points in P to be colored, such that all the points in P_i have the same color, and for $i \neq j$, the points in P_i have a different color from the points in P_j . We say that P is a *k-colored* point set. A *colored matching* of

P is a perfect matching in $K_n(P_1, \dots, P_k)$. A *plane colored matching* of P is a perfect matching in $K_n(P_1, \dots, P_k)$ in which no two edges cross. See Figure 7.1(a).

In this chapter we consider the problem of computing a balanced cut for a given color-balanced point set in general position in the plane. We show how to use balanced cuts to compute plane matchings in multipartite geometric graphs.

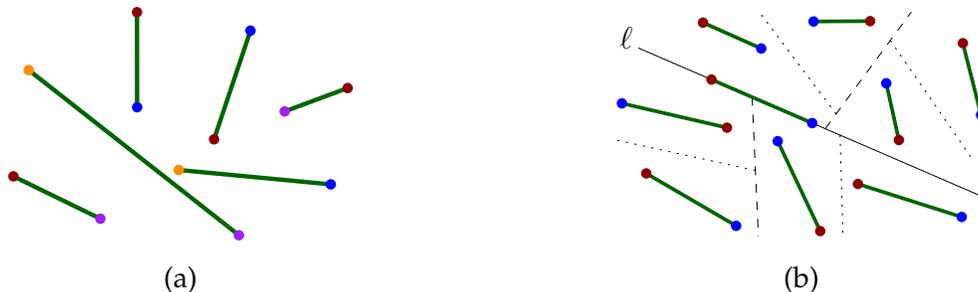


Figure 7.1: (a) A plane colored matching. (b) Recursive ham sandwich cuts.

7.1.1 Previous Work

7.1.1.1 2-Colored Point Sets

Let P be a set of $n = 2m$ points in general position in the plane. Let $\{R, B\}$ be a partition of P such that $|R| = |B| = m$. Assume the points in R are colored red and the points in B are colored blue. It is well-known that $K_n(R, B)$ has a plane RB-matching [1]. In fact, a minimum weight RB-matching, i.e., a perfect matching that minimizes the sum of Euclidean length of the edges, is plane. A minimum weight RB-matching in $K_n(R, B)$ can be computed in $O(n^{2.5} \log n)$ time [16], or even in $O(n^{2+\epsilon})$ time [2]. Consequently, a plane RB-matching can be computed in $O(n^{2+\epsilon})$ time. As a plane RB-matching is not necessarily a minimum weight RB-matching, one may compute a plane RB-matching faster than computing a minimum weight RB-matching. Hershberger and Suri [11] presented an $O(n \log n)$ time algorithm for computing a plane RB-matching. They also proved a lower bound of $\Omega(n \log n)$ time for computing a plane RB-matching, by providing a reduction from sorting.

Alternatively, one can compute a plane RB-matching by recursively applying the ham sandwich theorem; see Figure 7.1(b). We say that a line ℓ bisects a point set R if both sides of ℓ have the same number of points of R ; if $|R|$ is odd, then ℓ contains one point of R .

THEOREM 7.1 (Ham Sandwich Theorem). *For a point set P in general position in the plane which is partitioned into sets R and B , there exists a line that simultaneously bisects R and B .*

A line ℓ that simultaneously bisects R and B can be computed in $O(|R| + |B|)$ time, assuming $R \cup B$ is in general position in the plane [13]. By recursively applying Theorem 7.1, we can compute a plane RB-matching in $\Theta(n \log n)$ time.

7.1.1.2 3-Colored Point Sets

Let P be a set of $n = 3m$ points in general position in the plane. Let $\{R, G, B\}$ be a partition of P such that $|R| = |G| = |B| = m$. Assume the points in R are colored red, the points in G are colored green, and the points in B are colored blue. A lot of research has been done to generalize the ham sandwich theorem to 3-colored point sets, see e.g. [4, 5, 12]. It is easy to see that there exist configurations of P such that there exists no line which bisects R , G , and B , simultaneously.

Furthermore, for some configurations of P , for any $k \in \{1, \dots, m-1\}$, there does not exist any line ℓ such that an open half-plane bounded by ℓ contains k red, k green, and k blue points (see [5] for an example). For the special case, where the points on the convex hull of P are monochromatic, Bereg and Kano [5] proved that there exists an integer $1 \leq k \leq m-1$ and an open half-plane containing exactly k points from each color.

Bereg et al. [4] proved that if the points of P are on any closed Jordan curve γ , then for every integer k with $0 \leq k \leq m$ there exists a pair of disjoint intervals on γ whose union contains exactly k points of each color. In addition, they showed that if m is even, then there exists a double wedge that contains exactly $m/2$ points of each color.

Now, let P be a 3-colored point set of size n in general position in the plane, with n even. Assume the points in P are colored red, green, and blue such that P is color-balanced. Let R , G , and B denote the set of red, green, and blue points, respectively. Note that $|R|$, $|G|$, and $|B|$ are at most $\lfloor n/2 \rfloor$, but, they are not necessarily equal. Kano et al. [12] proved the existence of a feasible cut, when the points on the convex hull of P are monochromatic.

THEOREM 7.2 (Kano et al. [12]). *Let P be a 3-colored point set in general position in the plane, such that P is color-balanced and $|P|$ is even. If the points on the convex hull of P are monochromatic, then there exists a line ℓ which partitions P into Q_1 and Q_2 such that both Q_1 and Q_2 are color-balanced and have an even number of points and $2 \leq |Q_i| \leq |P| - 2$, for $i = 1, 2$.*

They also proved the existence of a plane perfect matching in $K_n(R, G, B)$ by recursively applying Theorem 7.2. Their proof is constructive. Although they did not analyze the running time, it can be shown that their algorithm runs in $O(n^2 \log n)$ time as follows. If the size of the largest color class is exactly $n/2$, then consider the points in the largest color class as R and the other points as B , then compute a plane RB -matching; and we are done. If there are two adjacent points of distinct colors on the convex hull, then match these two points and recurse on the remaining points. Otherwise, if the convex hull is monochromatic, pick a point $p \in P$ on the convex hull and sort the points in $P \setminus \{p\}$ around p . A line ℓ —partitioning the point set into two color-balanced point sets—is found by scanning the sorted list. Then recurse on each of the partitions. To find ℓ they spend $O(n \log n)$ time. The total running time of their algorithm is $O(n^2 \log n)$.

Based on the algorithm of Kano et al. [12], we can show that a plane perfect matching in $K_n(R, G, B)$ can be computed in $O(n \log^3 n)$ time. We can prove the existence of a feasible cut for P , even if the points on the convex hull of P are not monochromatic. To find feasible cuts recursively, we use the dynamic convex hull structure of Overmars and Leeuwen [14], which uses $O(\log^2 n)$ time for each insertion and deletion. Pick a point $p \in P$ on the convex hull of P and look for a point $q \in P \setminus \{p\}$, such that the line passing through p and q is a feasible cut. Search for q , alternatively, in clockwise and counterclockwise directions around p . To do this, we repeatedly check if the line passing through p and its (clockwise and counterclockwise in turn) neighbor on the convex hull, say r , is a feasible cut. If the line through p and r is not a feasible cut, then we delete r . At some point we find a feasible cut ℓ which divides P into Q_1 and Q_2 . Add the two points on ℓ to either Q_1 or Q_2 such that they remain color-balanced. Let $|Q_1| = k$ and $|Q_2| \geq k$. In order to compute the data structure for Q_2 , we use the current data structure and undo the deletions on the side of ℓ which contains Q_2 . We rebuild the data structure for Q_1 . Then, we recurse on Q_1 and Q_2 . The running time can be expressed by $T(n) = T(n-k) + T(k) + O(k \log^2 n)$, where $k \leq n-k$. This recurrence solves to $O(n \log^3 n)$. Notice that, because we undo the deletions on one side of ℓ and rebuild the data structure for the points on the other side of ℓ , any dynamic data structure that performs insertions and deletions in faster amortized time may not be feasible.

7.1.1.3 Multicolored Point Sets

Let $\{P_1, \dots, P_k\}$, where $k \geq 2$, be a partition of P and $K_n(P_1, \dots, P_k)$ be the complete multipartite geometric graph on P . A necessary and sufficient condition for the existence of a perfect matching in $K_n(P_1, \dots, P_k)$ follows from the following result of Sitton [15].

THEOREM 7.3 (Sitton [15]). *The size of a maximum matching in any complete multipartite graph K_{n_1, \dots, n_k} , with $n = n_1 + \dots + n_k$ vertices, where $n_1 \geq \dots \geq n_k$, is*

$$|M_{\max}| = \min \left\{ \sum_{i=2}^k n_i, \left\lfloor \frac{1}{2} \sum_{i=1}^k n_i \right\rfloor \right\}.$$

Theorem 7.3 implies that if n is even and $n_1 \leq \frac{n}{2}$, then K_{n_1, \dots, n_k} has a perfect matching. It is obvious that if $n_1 > \frac{n}{2}$, then K_{n_1, \dots, n_k} does not have any perfect matching. Therefore,

COROLLARY 7.4. *Let $k \geq 2$ and consider a partition $\{P_1, \dots, P_k\}$ of a point set P , where $|P|$ is even. Then, $K_n(P_1, \dots, P_k)$ has a colored matching if and only if P is color-balanced.*

Aichholzer et al. [3], and Kano et al. [12] show that the same condition as in Corollary 7.4 is necessary and sufficient for the existence of a plane colored matching in $K_n(P_1, \dots, P_k)$:

THEOREM 7.5 (Aichholzer et al. [3], and Kano et al. [12]). *Let $k \geq 2$ and consider a partition $\{P_1, \dots, P_k\}$ of a point set P , where $|P|$ is even. Then, $K_n(P_1, \dots, P_k)$ has a plane colored matching if and only if P is color-balanced.*

In fact, they show something stronger. Aichholzer et al. [3] show that a minimum weight colored matching in $K_n(P_1, \dots, P_k)$, which minimizes the total Euclidean length of the edges, is plane. Gabow [10] gave an implementation of Edmonds' algorithm which computes a minimum weight matching in general graphs in $O(n(m + n \log n))$ time, where m is the number of edges in G . Since P is color-balanced, $K_n(P_1, \dots, P_k)$ has $\Theta(n^2)$ edges. Thus, a minimum weight colored matching in $K_n(P_1, \dots, P_k)$, and hence a plane colored matching in $K_n(P_1, \dots, P_k)$, can be computed in $O(n^3)$ time. Kano et al. [12] extended their $O(n^2 \log n)$ -time algorithm for the 3-colored point sets to the multicolored case.

Since the problem of computing a plane RB-matching in $K_n(R, B)$ is a special case of the problem of computing a plane colored matching in $K_n(P_1, \dots, P_k)$, the $\Omega(n \log n)$ time lower bound for computing a plane RB-matching holds for computing a plane colored matching.

7.1.2 Our Contribution

Our main contribution, which is presented in Section 7.2, is the following: given any color-balanced point set P in general position in the plane, there exists a balanced cut for P . Further, we show that if n is even, then there exists a balanced cut which partitions P into two point sets each of even size, and such a balanced cut can be computed in linear time. In Section 7.3, we present a divide-and-conquer algorithm which computes a plane colored matching in $K_n(P_1, \dots, P_k)$ in $\Theta(n \log n)$ time, by recursively finding balanced cuts in color-balanced subsets of P . In case P is not color-balanced, then $K_n(P_1, \dots, P_k)$ does not admit a perfect matching; we describe how to find a plane colored matching with the maximum number of edges in Section 7.3.1. In addition, we show how to compute a maximum matching in any complete multipartite graph in linear time.

7.2 BALANCED CUT THEOREM

Given a color-balanced point set P with $n \geq 4$ points in general position in the plane, recall that a *balanced cut* is a line which partitions P into two point sets Q_1 and Q_2 , such that both Q_1 and Q_2 are color-balanced and $\max\{|Q_1|, |Q_2|\} \leq \frac{2n}{3} + 1$. Let $\{P_1, \dots, P_k\}$ be a partition of P , where the points in P_i are colored C_i . In this section we prove the existence of a balanced cut for P . Moreover, we show how to find such a balanced cut in $O(n)$ time.

If $k = 2$, the existence of a balanced cut follows from the ham sandwich cut theorem. If $k \geq 4$, we reduce the k -colored point set P to a three colored point set. Afterwards, we prove the statement for $k = 3$.

LEMMA 7.6. *Let P be a color-balanced point set of size n in the plane with $k \geq 4$ colors. In $O(n)$ time P can be reduced to a color-balanced point set P' with 3 colors such that any balanced cut for P' is also a balanced cut for P .*

Proof. We repeatedly merge the color families in P until we get a color-balanced point set P' with three colors. Afterwards, we show that any balanced cut for P' is also a balanced cut for P .

Without loss of generality assume that C_1, \dots, C_k is a non-increasing order of the color classes according to the number of points in each color class. That is, $\lfloor |P|/2 \rfloor \geq |P_1| \geq \dots \geq |P_k| \geq 1$ (note that P is color-balanced). In order to reduce the k -colored problem to a 3-colored problem, we repeatedly merge the two color families of the smallest cardinality. In each iteration we merge the two smallest color families, C_{k-1} and C_k , to get a new color class, C'_{k-1} , where $P'_{k-1} = P_{k-1} \cup P_k$. In order to prove that $P' = P_1 \cup \dots \cup P_{k-2} \cup P'_{k-1}$ is color-balanced with respect to the coloring $C_1, \dots, C_{k-2}, C'_{k-1}$ we have to show that $|P'_{k-1}| \leq \lfloor |P'|/2 \rfloor$. Note that before the merge we have $|P| = |P_1| + \dots + |P_{k-2}| + |P_{k-1}| + |P_k|$, while after the merge we have $|P'| = |P_1| + \dots + |P_{k-2}| + |P'_{k-1}|$, where $|P'_{k-1}| = |P_{k-1}| + |P_k|$. Since P_{k-1} and P_k are the two smallest and $k \geq 4$, $|P'_{k-1}| \leq |P_1| + \dots + |P_{k-2}|$. This implies that after the merge we have $|P'_{k-1}| \leq \lfloor |P'|/2 \rfloor$. Thus P' is color-balanced. By repeatedly merging the points of the two smallest color families, at some point we get a 3-colored point set P' which is color-balanced. Without loss of generality assume that P' is colored by R, G , and B . Consider any balanced cut ℓ for P' ; ℓ partitions P' into two sets Q_1 and Q_2 , each of size at most $\frac{2}{3}n + 1$, such that the number points of each color in Q_i is at most $\lfloor |Q_i|/2 \rfloor$, where $i = 1, 2$. Note that the set of points in P colored C_j , for $1 \leq j \leq k$, is a subset of points in P' colored either R, G , or B . Thus, the number of points colored C_j in Q_i is at most $\lfloor |Q_i|/2 \rfloor$, where $j = 1, \dots, k$ and $i = 1, 2$. Therefore, ℓ is a balanced cut for P .

In order to merge the color families, a monotone priority queue (see [9]) can be used, where the priority of each color C_j is the number of points colored C_j . The monotone priority queue offers insert and extract-min operations where the priority of an inserted element is greater than the priority of the last element extracted from the queue. We store the color families in a monotone priority queue of size $\frac{n}{2}$ (because all elements are in the range of 1 up to $\frac{n}{2}$). Afterwards, we perform a sequence of $O(k)$ extract-min and insert operations. Since $k \leq n$, the total time to merge k color families is $O(n)$. □ □

According to Lemma 7.6, from now on we assume that P is a color-balanced point set consisting of n points colored by three colors.

LEMMA 7.7. *Let P be a color-balanced point set of $n \geq 4$ points in general position in the plane with three colors. In $O(n)$ time we can compute a line ℓ such that*

1. ℓ does not contain any point of P .

2. ℓ partitions P into two point sets Q_1 and Q_2 , where
 - a) both Q_1 and Q_2 are color-balanced,
 - b) both Q_1 and Q_2 contains at most $\frac{2}{3}n + 1$ points.

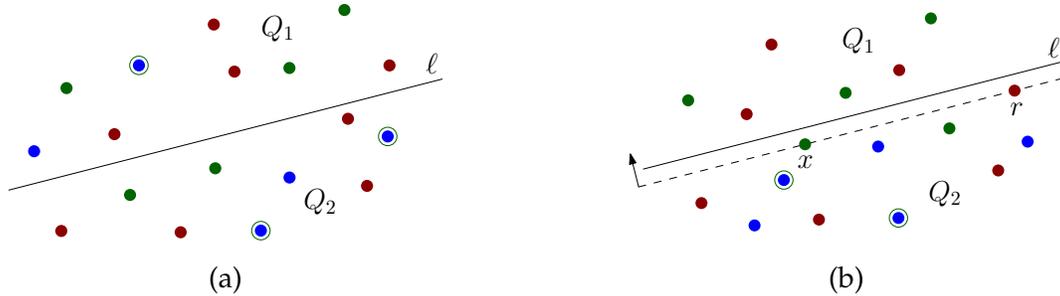


Figure 7.2: Illustrating the balanced cut theorem. The blue points in X are surrounded by circles. The line ℓ is a balanced cut where: (a) $|R|$ is even, and (b) $|R|$ is odd.

Proof. Assume that the points in P are colored red, green, and blue. Let R , G , and B denote the set of red, green, and blue points, respectively. Without loss of generality assume that $1 \leq |B| \leq |G| \leq |R|$. Since P is color-balanced, $|R| \leq \lfloor \frac{n}{2} \rfloor$. Let X be an arbitrary subset of B such that $|X| = |R| - |G|$; note that $X = \emptyset$ when $|R| = |G|$, and $X = B$ when $|R| = \frac{n}{2}$ (where n is even). Let $Y = B - X$. Let ℓ be a ham sandwich cut for R and $G \cup X$ (pretending that the points in $G \cup X$ have the same color). Let Q_1 and Q_2 denote the set of points on each side of ℓ ; see Figure 7.2(a). If $|R|$ is odd, then $|G \cup X|$ is also odd, and thus ℓ contains a point $r \in R$ and a point $x \in G \cup X$; see Figure 7.2(b). In this case without loss of generality assume that the number of blue points in Q_2 is at least the number of blue points in Q_1 ; slide ℓ slightly such that r and x lie in the same side as Q_2 , i.e. Q_2 is changed to $Q_2 \cup \{r, x\}$. We prove that ℓ satisfies the statement of the theorem. The line ℓ does not contain any point of P and by the ham sandwich cut theorem it can be computed in $O(n)$ time.

Now we prove that both Q_1 and Q_2 are color-balanced. Let R_1 , G_1 , and B_1 be the set of red, green, and blue points in Q_1 . Let $X_1 = X \cap Q_1$ and $Y_1 = Y \cap Q_1$. Note that $B_1 = X_1 \cup Y_1$. Similarly, define R_2 , G_2 , B_2 , X_2 , and Y_2 as subsets of Q_2 . Since $|R| = |G \cup X|$ and ℓ bisects both R and $G \cup X$, we have $|R_1| = \lfloor |R|/2 \rfloor$ and $|G_1| + |X_1| = |R_1|$. In the case that $|R|$ is odd, we add the points on ℓ to Q_2 (assuming that $|B_2| \geq |B_1|$). Thus, in either case ($|R|$ is even or odd) we have $|R_2| = \lceil |R|/2 \rceil$ and $|G_2| + |X_2| = |R_2|$. Therefore,

$$\begin{aligned}
 |Q_1| &\geq |R_1| + |G_1| + |X_1| = 2\lfloor |R|/2 \rfloor, \\
 |Q_2| &\geq |R_2| + |G_2| + |X_2| = 2\lceil |R|/2 \rceil.
 \end{aligned}
 \tag{7.1}$$

Let t_1 and t_2 be the total number of red and green points in Q_1 and Q_2 , respectively. Then, we have the following inequalities:

$$\begin{aligned}
t_1 &= |R_1| + |G_1| & t_2 &= |R_2| + |G_2| \\
&= 2|R_1| - |X_1| & &= 2|R_2| - |X_2| \\
&\geq 2|R_1| - |X| & &\geq 2|R_2| - |X| \\
&= 2\lfloor |R|/2 \rfloor - (|R| - |G|) & &= 2\lceil |R|/2 \rceil - (|R| - |G|) \\
&= \begin{cases} |G| & \text{if } |R| \text{ is even} \\ |G| - 1 & \text{if } |R| \text{ is odd,} \end{cases} & &= \begin{cases} |G| & \text{if } |R| \text{ is even} \\ |G| + 1 & \text{if } |R| \text{ is odd.} \end{cases}
\end{aligned} \tag{7.2}$$

In addition, we have the following equations:

$$|Q_1| = t_1 + |B_1| \quad \text{and} \quad |Q_2| = t_2 + |B_2|. \tag{7.3}$$

Note that $|R_1| = \lfloor |R|/2 \rfloor$ and $|G_1| \leq |Q_1 \cap (G \cup X)| = |R_1|$, thus, by Inequality (7.1) we have $|R_1| \leq \lfloor |Q_1|/2 \rfloor$ and $|G_1| \leq \lfloor |Q_1|/2 \rfloor$. Similarly, $|R_2| \leq \lfloor |Q_2|/2 \rfloor$ and $|G_2| \leq \lfloor |Q_2|/2 \rfloor$. Therefore, in order to argue that Q_1 and Q_2 are color-balanced, it only remains to show that $|B_1| \leq \lfloor |Q_1|/2 \rfloor$ and $|B_2| \leq \lfloor |Q_2|/2 \rfloor$. Note that $|B_1|, |B_2| \leq |B|$ and by initial assumption $|B| \leq |G|$. We differentiate between two cases where $|R|$ is even and $|R|$ is odd. If $|R|$ is even, by Inequalities (7.2) we have $t_1, t_2 \geq |G|$. Therefore, by the fact that $\max\{|B_1|, |B_2|\} \leq |B| \leq |G|$ and Equation (7.3), we have $|B_1| \leq \lfloor |Q_1|/2 \rfloor$ and $|B_2| \leq \lfloor |Q_2|/2 \rfloor$. If $|R|$ is odd, we slide ℓ towards Q_1 ; assuming that $|B_2| \geq |B_1|$. In addition, since $|B_1| + |B_2| = |B|$ and $|B| \geq 1$, $|B_2| \geq 1$. Thus, $|B_1| \leq |B| - 1 \leq |G| - 1$, while by Inequality (7.2), $t_1 \geq |G| - 1$. Therefore, Equality (7.3) implies that $|B_1| \leq \lfloor |Q_1|/2 \rfloor$. Similarly, by Inequality (7.2) we have $t_2 \geq |G| + 1$ while $|B_2| \leq |G|$. Thus, Equality (7.3) implies that $|B_2| \leq \lfloor |Q_2|/2 \rfloor$. Therefore, both Q_1 and Q_2 are color-balanced.

We complete the proof by providing the following upper bound on the size of Q_1 and Q_2 . Since we assume that R is the largest color class, $|R| \geq \lceil \frac{n}{3} \rceil$. By Inequality (7.1), $\min\{|Q_1|, |Q_2|\} \geq 2\lfloor |R|/2 \rfloor$, which implies that

$$\max\{|Q_1|, |Q_2|\} \leq n - 2 \left\lfloor \frac{|R|}{2} \right\rfloor \leq n - 2 \left(\frac{|R| - 1}{2} \right) \leq n - \frac{n}{3} + 1 = \frac{2n}{3} + 1.$$

□

□

Therefore, by Lemma 7.6 and Lemma 7.7, we have proved the following theorem:

THEOREM 7.8 (Balanced Cut Theorem). *Let P be a color-balanced point set of $n \geq 4$ points in general position in the plane. In $O(n)$ time we can compute a line ℓ such that*

1. ℓ does not contain any point of P .
2. ℓ partitions P into two point sets Q_1 and Q_2 , where
 - a) both Q_1 and Q_2 are color-balanced,
 - b) both Q_1 and Q_2 contains at most $\frac{2}{3}n + 1$ points.

By Theorem 7.5, if P has even number of points and no color is in strict majority, then P admits a plane perfect matching. By Theorem 7.8, we partition P into two sets Q_1 and Q_2 such that in each of them no point is in strict majority. But, in order to apply the balanced cut theorem,

recursively, to obtain a perfect matching on each side of the cut, we need both Q_1 and Q_2 to have an even number of points. Thus, we extend the result of Theorem 7.8 to a restricted version of the problem where $|P|$ is even and we are looking for a balanced cut which partitions P into Q_1 and Q_2 such that both $|Q_1|$ and $|Q_2|$ are even. The following theorem describes how to find such a balanced cut.

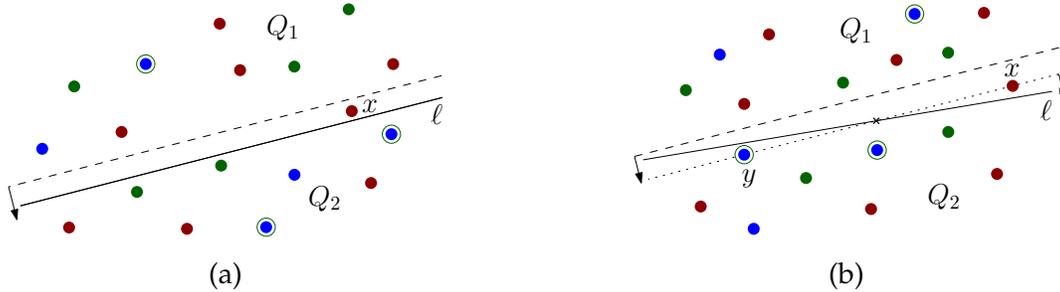


Figure 7.3: Updating ℓ to make $|Q_1|$ and $|Q_2|$ even numbers, where: (a) ℓ passes over one point, and (b) ℓ passes over two points.

THEOREM 7.9. *Let P be a color-balanced point set of $n \geq 4$ points in general position in the plane with n even and three colors. In $O(n)$ time we can compute a line ℓ such that*

1. ℓ does not contain any point of P .
2. ℓ partitions P into two point sets Q_1 and Q_2 , where
 - a) both Q_1 and Q_2 are color-balanced,
 - b) both Q_1 and Q_2 have even number of points,
 - c) both Q_1 and Q_2 contains at most $\frac{2}{3}n + 1$ points.

Proof. Let ℓ be the balanced cut obtained in the proof of Lemma 7.7, which divides P into Q_1 and Q_2 . Note that ℓ does not contain any point of P . If $|Q_1|$ is even, subsequently $|Q_2|$ is even, thus ℓ satisfies the statement of the theorem and we are done. Assume that $|Q_1|$ and $|Q_2|$ are odd. Let $R_1, G_1,$ and B_1 be the set of red, green, and blue points in Q_1 . Let $X_1 = X \cap Q_1$ and $Y_1 = Y \cap Q_1$. Note that $B_1 = X_1 \cup Y_1$. Similarly, define $R_2, G_2, B_2, X_2,$ and Y_2 as subsets of Q_2 . Note that $|Q_1| = |R_1| + |G_1| + |X_1| + |Y_1|$ and $|Q_2| = |R_2| + |G_2| + |X_2| + |Y_2|$. Recall that $|R_1| = |G_1| + |X_1| = \lfloor |R|/2 \rfloor$ and $|R_2| = |G_2| + |X_2| = \lceil |R|/2 \rceil$. Thus, $|R_1| + |G_1| + |X_1|$ and $|R_2| + |G_2| + |X_2|$ are even. In order to make $|Q_1|$ and $|Q_2|$ to be odd numbers, both $|Y_1|$ and $|Y_2|$ have to be odd numbers. Thus, $|Y_1| \geq 1$ and $|Y_2| \geq 1$, which implies that

$$\begin{aligned} |Q_1| &= |R_1| + |G_1| + |X_1| + |Y_1| \geq 2\lfloor |R|/2 \rfloor + 1, \\ |Q_2| &= |R_2| + |G_2| + |X_2| + |Y_2| \geq 2\lceil |R|/2 \rceil + 1. \end{aligned} \tag{7.4}$$

In addition,

$$\begin{aligned} |B_1| &= |B| - (|X_2| + |Y_2|) \leq |B| - 1, \\ |B_2| &= |B| - (|X_1| + |Y_1|) \leq |B| - 1. \end{aligned} \tag{7.5}$$

Note that Q_1 is color-balanced. That is, $|R_1|, |G_1|, |B_1| \leq \lfloor |Q_1|/2 \rfloor$, where $|Q_1|$ is odd. Thus, by addition of one point (of any color) to Q_1 , it still remain color-balanced. Therefore, we slide ℓ

slightly towards Q_2 and stop as soon as it passes over a point $x \in Q_2$; see Figure 7.3(a). If ℓ passes over two points x and y , rotate ℓ slightly, such that x lies on the same side as Q_1 and y remains on the other side; see Figure 7.3(b). We prove that ℓ satisfies the statement of the theorem. It is obvious that updating the position of ℓ takes $O(n)$ time. Let $Q'_1 = Q_1 \cup \{x\}$ and $Q'_2 = Q_2 - \{x\}$. By the previous argument Q'_1 is color-balanced. Now we show that Q'_2 is color-balanced as well. Note that $|Q'_2| = |Q_2| - 1$, thus, by Inequality (7.4) we have

$$|Q'_2| \geq 2\lceil |R|/2 \rceil.$$

Let R'_2 , G'_2 , and B'_2 be the set of red, green, and blue points in Q'_2 , and let t'_2 be the total number of red and green points in Q'_2 . Then,

$$|Q'_2| = t'_2 + |B'_2|. \quad (7.6)$$

To prove that Q'_2 is color-balanced we differentiate between three cases, where $x \in R_2$, $x \in G_2$, or $x \in B_2$:

- $x \in R_2$. In this case: (i) $|R'_2| = |R_2| - 1 = \lceil |R|/2 \rceil - 1 \leq \lfloor |Q'_2|/2 \rfloor$. (ii) $|G'_2| = |G_2| \leq |R_2| = \lceil |R|/2 \rceil \leq \lfloor |Q'_2|/2 \rfloor$. (iii) $t'_2 = t_2 - 1 \geq |G| - 1$, while $|B'_2| = |B_2| \leq |B| - 1 \leq |G| - 1$; Inequality (7.6) implies that $|B'_2| \leq \lfloor |Q'_2|/2 \rfloor$.
- $x \in G_2$. In this case: (i) $|R'_2| = |R_2| = \lceil |R|/2 \rceil \leq \lfloor |Q'_2|/2 \rfloor$. (ii) $|G'_2| = |G_2| - 1 \leq |R_2| - 1 = \lceil |R|/2 \rceil - 1 \leq \lfloor |Q'_2|/2 \rfloor$. (iii) $t'_2 = t_2 - 1 \geq |G| - 1$, while $|B'_2| = |B_2| \leq |B| - 1 \leq |G| - 1$; Inequality (7.6) implies that $|B'_2| \leq \lfloor |Q'_2|/2 \rfloor$.
- $x \in B_2$. In this case: (i) $|R'_2| = |R_2| = \lceil |R|/2 \rceil \leq \lfloor |Q'_2|/2 \rfloor$. (ii) $|G'_2| = |G_2| \leq |R_2| = \lceil |R|/2 \rceil \leq \lfloor |Q'_2|/2 \rfloor$. (iii) $t'_2 = t_2 \geq |G|$, while $|B'_2| = |B_2| - 1 \leq |B| - 2 \leq |G| - 2$; Inequality (7.6) implies that $|B'_2| \leq \lfloor |Q'_2|/2 \rfloor$.

In all cases $|R'_2|, |G'_2|, |B'_2| \leq \lfloor |Q'_2|/2 \rfloor$, which imply that Q'_2 is color-balanced.

As for the size condition,

$$\min\{|Q'_1|, |Q'_2|\} = \min\{|Q_1| + 1, |Q_2| - 1\} \geq 2\lceil |R|/2 \rceil,$$

where the last inequality resulted from Inequality (7.4). This implies that $\max\{|Q'_1|, |Q'_2|\} \leq \frac{2n}{3} + 1$. Thus, ℓ satisfies the statement of the theorem, with $Q_1 = Q'_1$ and $Q_2 = Q'_2$. \square \square

We note that the constant $\frac{2}{3}$ in Theorem 7.9 is tight. Consider an equilateral triangle with vertices a , b , and c . Consider a color-balanced point set P of $\frac{n}{3}$ red points, $\frac{n}{3}$ green points, and $\frac{n}{3}$ blue points such that the red points are located around a , the green points are located around b , and the red points are located around c . Then, any feasible cut for P partitions it into two point sets such that one of them has size at least $\frac{2}{3}n$.

Note that both Theorem 7.9 and Theorem 7.2 prove the existence of a line ℓ which partitions a color-balanced point set P into two color-balanced point sets Q_1 and Q_2 . But, there are two main differences: (i) Theorem 7.9 can be applied on any color-balanced point set P in general position. Theorem 7.2 is only applicable on color-balanced point sets in general position, where the points on the convex hull are monochromatic. (ii) Theorem 7.9 proves the existence of a balanced cut such that $\frac{n}{3} - 1 \leq |Q_i| \leq \frac{2n}{3} + 1$, while the cut computed by Theorem 7.2 is not necessarily balanced, as $2 \leq |Q_i| \leq n - 2$, where $i = 1, 2$. In addition, the balanced cut in Theorem 7.9 can be computed in $O(n)$ time, while the cut in Theorem 7.2 is computed in $O(n \log n)$ time.

7.3 PLANE COLORED MATCHING ALGORITHM

Let P be a color-balanced point set of n points in general position in the plane with respect to a partition $\{P_1, \dots, P_k\}$, where n is even and $k \geq 2$. In this section we present an algorithm which computes a plane colored matching in $K_n(P_1, \dots, P_k)$ in $\Theta(n \log n)$ time.

Let $\{C_1, \dots, C_k\}$ be a set of k colors. Imagine all the points in P_i are colored C_i for all $1 \leq i \leq k$. Without loss of generality, assume that $|P_1| \geq |P_2| \geq \dots \geq |P_k|$. If $k = 2$, then we can compute an RB-matching in $O(n \log n)$ time by recursively applying the ham sandwich theorem. If $k \geq 4$, as in Lemma 7.6, in $O(n)$ time, we compute a color-balanced point set P with three colors. Any plane colored matching for P with respect to the three colors, say (R, G, B) , is also a plane colored matching with respect to the coloring C_1, \dots, C_k . Hereafter, assume that P is a color-balanced point set which is colored by three colors.

By Theorem 7.9, in linear time we can find a line ℓ that partitions P into two sets Q_1 and Q_2 , where both Q_1 and Q_2 are color-balanced with an even number of points, such that $\max\{|Q_1|, |Q_2|\} \leq \frac{2n}{3} + 1$. Since Q_1 and Q_2 are color-balanced, by Corollary 7.4, both Q_1 and Q_2 admit plane colored matchings. Let $M(Q_1)$ and $M(Q_2)$ be plane colored matchings in Q_1 and Q_2 , respectively. Since Q_1 and Q_2 are separated by ℓ , $M(Q_1) \cup M(Q_2)$ is a plane colored matching for P . Thus, in order to compute a plane colored matching in P , one can compute plane colored matchings in Q_1 and Q_2 recursively, as described in Algorithm 7. The *RGB-matching* function receives a colored point set P of n points, where n is even and the points of P are colored by three colors, and computes a plane colored matching in P . The *BalancedCut* function partitions P into Q_1 and Q_2 where both are color-balanced and have even number of points.

Algorithm 7 *RGB-matching*(P)

Input: a color-balanced point set P with respect to (R, G, B) , where $|P|$ is even.

Output: a plane colored matching in P .

```

1: if  $P$  is 2-colored then
2:   return RB-matching( $P$ )
3: else
4:    $\ell \leftarrow$  BalancedCut( $P$ )
5:    $Q_1 \leftarrow$  points of  $P$  to the left of  $\ell$ 
6:    $Q_2 \leftarrow$  points of  $P$  to the right of  $\ell$ 
7:   return RGB-matching( $Q_1$ )  $\cup$  RGB-matching( $Q_2$ )

```

Now we analyze the running time of the algorithm. If $k = 2$, then in $O(n \log n)$ time we can find a plane RB-matching for P . If $k \geq 4$, then by Lemma 7.6, in $O(n)$ time we reduce the k -colored problem to a 3-colored problem. Then, the function *RGB-matching* computes a plane colored matching in P . Let $T(n)$ denote the running time of *RGB-matching* on the 3-colored point set P , where $|P| = n$. As described in Theorem 7.8 and Theorem 7.9, in linear time we can find a balanced cut ℓ in line 4 in Algorithm 7. The recursive calls to *RGB-matching* function in line 7 takes $T(|Q_1|)$ and $T(|Q_2|)$ time. Thus, the running time of *RGB-matching* can be expressed by the following recurrence:

$$T(n) = T(|Q_1|) + T(|Q_2|) + O(n).$$

Since $|Q_1|, |Q_2| \leq \frac{2n}{3} + 1$ and $|Q_1| + |Q_2| = n$, this recurrence solves to $T(n) = O(n \log n)$.

THEOREM 7.10. *Given a color-balanced point set P of size n in general position in the plane with n even, a plane colored matching in P can be computed in $\Theta(n \log n)$ time.*

7.3.1 Maximum Matching

If P is not color-balanced, then $K_n(P_1, \dots, P_k)$ does not admit a perfect matching. In this case we compute a maximum matching.

THEOREM 7.11. *Given a colored point set P of size n in general position in the plane, a maximum plane colored matching M in P can be computed optimally in $\Theta(n + |M| \log |M|)$ time.*

Proof. Let $\{P_1, \dots, P_k\}$, where $k \geq 2$, be a partition of the points in P such that the points in P_i are colored C_i for $1 \leq i \leq k$. Without loss of generality assume that $|P_1| \geq \dots \geq |P_k|$. If $|P_1| \leq \lfloor |P|/2 \rfloor$, then P is color-balanced, and hence, by Theorem 7.10 we can compute a plane colored matching in $\Theta(n \log n)$ time. Assume $|P_1| > \lfloor |P|/2 \rfloor$. Then P is not color-balanced, and hence, P does not admit a perfect matching. In this case, by Theorem 7.3, the size of any maximum matching, say M , is

$$|M| = \sum_{i=2}^k |P_i|.$$

Let P'_1 be any arbitrary subset of P_1 such that $|P'_1| = |P_2| + \dots + |P_k|$. Imagine the points in $P_2 \cup \dots \cup P_k$ are colored red and the points in P'_1 are colored blue. Let $P' = P'_1 \cup P_2 \cup \dots \cup P_k$. Any plane RB-matching in P' is a maximum plane colored matching in P , and has $|P_1| + \dots + |P_k| = |M|$ edges. An RB-matching of size $|M|$ can be computed in $\Theta(|M| \log |M|)$ time. \square \square

THEOREM 7.12. *Given any complete multipartite graph $K_n(V_1, \dots, V_k)$ on n vertices and $k \geq 2$, a maximum matching in $K_n(V_1, \dots, V_k)$ can be computed optimally in $\Theta(n)$ time.*

Proof. If n is odd, then by Theorem 7.3, we can remove a vertex from the largest vertex set, without changing the size of a maximum matching. Thus, assume that n is even. Without loss of generality assume that $|V_1| \geq |V_2| \geq \dots \geq |V_k|$. If $|V_1| \geq n/2$, then let R be an arbitrary subset of V_1 such that $|R| = |V_2| + \dots + |V_k|$, and let $B = V_2 \cup \dots \cup V_k$. Then, any maximal matching in $K_n(R, B)$ —which is also a perfect matching—is a maximum matching in $K_n(V_1, \dots, V_k)$.

If $|V_1| < n/2$, then by a similar argument as in Lemma 7.6, in $O(n)$ time we merge V_1, \dots, V_k to obtain a partition $\{R, G, B\}$ of vertices, such that $\max\{|R|, |G|, |B|\} \leq n/2$ and $K_n(R, G, B)$ is a subgraph of $K_n(V_1, \dots, V_k)$. Now we describe how to compute a perfect matching in $K_n(R, G, B)$. Without loss of generality assume that $|R| \geq |G| \geq |B|$. Let $m = n - 2 \cdot |R|$; observe that m is an even number. Since $m = n - 2 \cdot |R| \leq n - (|R| + |G|)$, we have $m \leq |B|$, and subsequently $m \leq |G|$. Let G' (resp. B') be an arbitrary subset of G (resp. B) of size $m/2$. Thus, $|B'| = |G'| = m/2$. Let $G'' = G \setminus G'$ and $B'' = B \setminus B'$. Thus,

$$|G'' \cup B''| = n - |R| - |G' \cup B'| = n - |R| - m = n - |R| - (n - 2|R|) = |R|.$$

Thus, both $K_m(G', B')$ and $K_{n-m}(R, G'' \cup B'')$ have perfect matchings. Therefore, the union of any maximal matching in $K_m(G', B')$ and any maximal matching in $K_{n-m}(R, G'' \cup B'')$ is a perfect matching in $K_n(R, G, B)$, and subsequently in $K_n(V_1, \dots, V_k)$.

Since a maximal matching in a complete bipartite graph is also a maximum matching and can be computed in linear time, the presented algorithm takes $O(n)$ time. \square \square

BIBLIOGRAPHY

- [1] The 1979 Putnam exam. In G. L. Alexanderson, L. F. Klosinski, and L. C. Larson, editors, *The William Lowell Putnam Mathematical Competition Problems and Solutions: 1965-1984*. Mathematical Association of America, USA, 1985.
- [2] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM Journal on Computing*, 29(3):912–953, 1999.
- [3] O. Aichholzer, S. Cabello, R. F. Monroy, D. Flores-Peñaloza, T. Hackl, C. Huemer, F. Hurtado, and D. R. Wood. Edge-removal and non-crossing configurations in geometric graphs. *Disc. Math. & Theo. Comp. Sci.*, 12(1):75–86, 2010.
- [4] S. Bereg, F. Hurtado, M. Kano, M. Korman, D. Lara, C. Seara, R. I. Silveira, J. Urrutia, and K. Verbeek. Balanced partitions of 3-colored geometric sets in the plane. *Discrete Applied Mathematics*, 181:21–32, 2015.
- [5] S. Bereg and M. Kano. Balanced line for a 3-colored point set in the plane. *The Electronic Journal of Combinatorics*, 19(1):P33, 2012.
- [6] A. Biniáz, P. Bose, A. Maheshwari, and M. H. M. Smid. Plane geodesic spanning trees, Hamiltonian cycles, and perfect matchings in a simple polygon. In *Proceedings of the first international conference on Topics in Theoretical Computer Science, TTCS'15*, pages 56–71, 2015.
- [7] A. Biniáz, P. Bose, A. Maheshwari, and M. H. M. Smid. Plane geodesic spanning trees, Hamiltonian cycles, and perfect matchings in a simple polygon. *Computational Geometry: Theory and Applications*, 57:27–39, 2016.
- [8] A. Biniáz, A. Maheshwari, S. C. Nandy, and M. H. M. Smid. An optimal algorithm for plane matchings in multipartite geometric graphs. In *Proceedings of the 14th International Symposium on Algorithms and Data Structures, WADS'15*, pages 66–78, 2015.
- [9] B. V. Cherkassky, A. V. Goldberg, and C. Silverstein. Buckets, heaps, lists, and monotone priority queues. *SIAM Journal on Computing*, 28(4):1326–1346, 1999.
- [10] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California.*, pages 434–443, 1990.
- [11] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32(2):249–267, 1992.
- [12] M. Kano, K. Suzuki, and M. Uno. Properly colored geometric matchings and 3-trees without crossings on multicolored points in the plane. In *16th Japanese Conference on Disc. Comp. Geom. and Graphs, JCDCCG*, pages 96–111, 2013.
- [13] C. Lo, J. Matousek, and W. L. Steiger. Algorithms for ham-sandwich cuts. *Discrete & Computational Geometry*, 11:433–452, 1994.
- [14] M. H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23(2):166–204, 1981.

- [15] D. Sitton. Maximum matchings in complete multipartite graphs. *Furman University Electronic Journal of Undergraduate Mathematics*, 2:6–16, 1996.
- [16] P. M. Vaidya. Geometry helps in matching. *SIAM Journal on Computing*, 18(6):1201–1225, 1989.

Given a set P of n points in the plane, where n is even, we consider the following question: How many plane perfect matchings can be packed into P ? For points in general position we prove the lower bound of $\lfloor \log_2 n \rfloor - 1$. For some special configurations of point sets, we give the exact answer. We also consider some restricted variants of this problem.

This chapter is published in the journal of Discrete Mathematics & Theoretical Computer Science [12].

8.1 INTRODUCTION

Let P be a set of n points in general position in the plane (no three points on a line). A *geometric graph* $G = (P, E)$ is a graph whose vertex set is P and whose edge set E is a set of straight-line segments with endpoints in P . We say that two edges of G *cross* each other if they have a point in common that is interior to both edges. Two edges are *disjoint* if they have no point in common. A subgraph S of G is said to be *plane* (*non-crossing* or *crossing-free*) if its edges do not cross. A *plane matching* is a plane graph consisting of pairwise disjoint edges. Two subgraphs S_1 and S_2 are *edge-disjoint* if they do not share any edge. A *complete geometric graph* $K(P)$ is a geometric graph on P which contains a straight-line edge between every pair of points in P .

We say that a set of subgraphs of $K(P)$ is *packed into* $K(P)$, if the subgraphs in the set are pairwise edge-disjoint. In a packing problem, we ask for the largest number of subgraphs of a given type that can be packed into $K(P)$. Among all subgraphs of $K(P)$, *plane perfect matchings*, *plane spanning trees*, and *plane spanning paths* are of interest [2, 3, 4, 5, 9, 13, 29, 35]. That is, one may look for the maximum number of plane spanning trees, plane Hamiltonian paths, or plane perfect matchings that can be packed into $K(P)$. Since $K(P)$ has $\frac{n(n-1)}{2}$ edges, at most $\frac{n}{2}$ spanning trees, at most $\frac{n}{2}$ spanning paths, and at most $n - 1$ perfect matchings can be packed into it. In this chapter we consider perfect matchings. A *perfect matching* in $K(P)$ is a set of edges that do not share any endpoint and cover all the points in P .

A long-standing open question is to determine if the edges of $K(P)$ (where n is even) can be partitioned into $\frac{n}{2}$ plane spanning trees. In other words, is it possible to pack $\frac{n}{2}$ plane spanning trees into $K(P)$? If P is in convex position, the answer in the affirmative follows from the result of Bernhart and Kanien [11]. For P in general position, Aichholzer et al. [5] prove that $\Omega(\sqrt{n})$ plane spanning trees can be packed into $K(P)$. They also show the existence of at least 2 edge-disjoint plane spanning paths.

In this chapter we consider a closely related question: How many plane perfect matchings can be packed into $K(P)$, where P is a set of n points in general position in the plane, with n even?

8.1.1 Previous Work

8.1.1.1 Existence of Plane Subgraphs

The existence of certain plane subgraphs in a geometric graph on a set P of n points is one of the classic problems in combinatorial and computational geometry.

One of the extremal problems in geometric graphs which was first studied by Avital and Hanani [10], Kuptiz [32], Erdős [20], and Perles (see reference [44]) is the following. What is the smallest number $e_k(n)$ such that any geometric graph with n vertices and more than $e_k(n)$ edges contains $k + 1$ pairwise disjoint edges, i.e., a plane matching of size at least $k + 1$. Note that $k \leq \lfloor n/2 \rfloor - 1$. By a result of Hopf and Pannwitz [28], Sutherland [42], and Erdős [20], $e_1(n) = n$, i.e., any geometric graph with $n + 1$ edges contains a pair of disjoint edges, and there are some geometric graphs with n edges which do not contain any pair of disjoint edges.

Alon and Erdős [8] proved that $e_2(n) < 6n - 5$, i.e., any geometric graph with n vertices and at least $6n - 5$ edges contains a plane matching of size three. This bound was improved to $e_2(n) \leq 3n$ by Goddard et al. [23]. Recently Černý [15] proved that $e_2(n) \leq \lceil 2.5n \rceil$; while the lower bound of $e_2(n) \geq \lceil 2.5n \rceil - 3$ is due to Perles (see [15]). For $e_3(n)$, Goddard et al. [23] showed that $3.5n - 6 \leq e_3(n) \leq 10n$, which was improved by Tóth and Valtr [44] to $4n - 9 \leq e_3(n) \leq 8.5n$.

For general values of k , Akiyama and Alon [7] gave the upper bound of $e_k(n) = O(n^{2-1/(k+1)})$. Goddard et al. [23] improved the bound to $e_k(n) = O(n(\log n)^{k-3})$. Pach and Törőcsik [37] obtained the upper bound of $e_k(n) \leq k^4 n$; which is the first upper bound that is linear in n . The upper bound was improved to $k^3(n + 1)$ by Tóth and Valtr [44]; they also gave the lower bound of $e_k(n) \geq \frac{3}{2}(k - 1)n - 2k^2$. Tóth [43] improved the upper bound to $e_k(n) \leq 2^9 k^2 n$, where the constant has been improved to 2^8 by Felsner [21]. It is conjectured that $e_k(n) \leq ckn$ for some constant c .

For the maximum value of k , i.e., $k = \frac{n}{2} - 1$, with n even, Aichholzer et al. [4] showed that $e_{n/2-1}(n) = \binom{n}{2} - \frac{n}{2} = \frac{n(n-2)}{2}$. That is, by removing $\frac{n}{2} - 1$ edges from any complete geometric graph, the resulting graph has $k + 1 = \frac{n}{2}$ disjoint edges, i.e., a plane perfect matching. This bound is tight; there exist complete geometric graphs, such that by removing $\frac{n}{2}$ edges, the resulting graph does not have any plane perfect matching. Similar bounds were obtained by Kupitz and Perles for complete convex graphs, i.e., complete graphs of point sets in convex position. Kupitz and Perles showed that any convex geometric graph with n vertices and more than kn edges contains $k + 1$ pairwise disjoint edges; see [23] (see also [7] and [8]). In particular, in the convex case, $2n + 1$ edges guarantee a plane matching of size three. In addition, Keller and Perles [30] gave a characterization of all sets of $\frac{n}{2}$ edges whose removal prevents the resulting graph from having a plane perfect matching.

Černý et al. [16] considered the existence of Hamiltonian paths in geometric graphs. They showed that after removing at most $\sqrt{n}/(2\sqrt{2})$ edges from any complete geometric graph of n vertices, the resulting graph still contains a plane Hamiltonian path. Aichholzer et al. [4] obtained tight bounds on the maximum number of edges that can be removed from a complete geometric graph, such that the resulting graph contains a certain plane subgraph; they considered plane perfect matchings, plane subtrees of a given size, and triangulations.

8.1.1.2 Counting Plane Graphs

The number of plane graphs of a given type in a set of n points is also of interest. In 1980, Newborn and Moser [36] asked for the maximal number of plane Hamiltonian cycles; they give

an upper bound of $2 \cdot 6^{n-2} \lfloor \frac{n}{2} \rfloor!$, but conjecture that it should be of the form c^n , for some constant c . In 1982, Ajtai et al. [6] proved that the number of plane graphs is at most 10^{13n} . Every plane graph is a subgraph of some triangulation (with at most $3n - 6$ edges). Since a triangulation has at most 2^{3n-6} plane subgraphs, as noted in [22], any bound of α^n on the number of triangulations implies a bound of $2^{3n-6} \alpha^n < (8\alpha)^n$ on the number of plane graphs. The best known upper bound of 30^n , for the number of triangulations is due to Sharir and Sheffer [38]. This implies the bound 240^n for plane graphs. As for plane perfect matchings, since a perfect matching has $\frac{n}{2}$ edges, Dumitrescu [19] obtained an upper bound of $\binom{3n-6}{n/2} \alpha^n \leq (3.87\alpha)^n$, where $\alpha = 30$. Sharir and Welzl [39] improved this bound to $O(10.05^n)$. They also showed that the number of all (not necessarily perfect) plane matchings is at most $O(10.43^n)$.

García et al. [22] showed that the number of plane perfect matchings of a fixed size set of points in the plane is minimum when the points are in convex position. Motzkin [34] showed that points in convex position have $C_{n/2}$ many perfect matchings (classically referred to as non-crossing configurations of chords on a circle), where $C_{n/2}$ is the $(n/2)^{\text{th}}$ Catalan number; $C_{n/2} = \Theta(n^{-3/2} 2^n)$. Thus, the number of plane perfect matchings of n points in the plane is at least $C_{n/2}$. García et al. [22] presented a configuration of n points in the plane which has $\Omega(n^{-4} 3^n)$ many plane perfect matchings. See Table 8.1.

8.1.1.3 Counting Edge-Disjoint Plane Graphs

The number of edge-disjoint plane graphs of a given type in a point set P of n points is also of interest. Nash-Williams [35] and Tutte [45] independently considered the number of (not necessarily plane) spanning trees. They obtained necessary and sufficient conditions for a graph to have k edge-disjoint spanning trees. Kundu [31] showed that any k -edge-connected graph contains at least $\lceil \frac{k-1}{2} \rceil$ edge-disjoint spanning trees.

As for the plane spanning trees a long-standing open question is to determine if the edges of $K(P)$ (where n is even) can be partitioned into $\frac{n}{2}$ plane spanning trees. In other words, is it possible to pack $\frac{n}{2}$ plane spanning trees into $K(P)$? If P is in convex position, the answer in the affirmative follows from the result of Bernhart and Kanien [11]. In [13], the authors characterize the partitions of the complete convex graph into plane spanning trees. They also describe a sufficient condition, which generalizes the convex case, for points in general position. Aichholzer et al. [5] showed that if the convex hull of P contains h vertices, then $K(P)$ contains at least $\lfloor \frac{h}{2} \rfloor$ edge-disjoint plane spanning trees, and if P is in a “regular wheel configuration”, $K(P)$ can be partitioned into $\frac{n}{2}$ spanning trees. For P in general position they showed that $K(P)$ contains $\Omega(\sqrt{n})$ edge-disjoint plane spanning trees. They obtained the following trade-off between the number of edge-disjoint plane spanning trees and the maximum vertex degree in each tree: For any $k \leq \sqrt{n/12}$, $K(P)$ has k edge-disjoint plane spanning trees with maximum vertex degree $O(k^2)$ and diameter $O(\log(n/k^2))$. They also showed the existence of at least 2 edge-disjoint plane Hamiltonian paths.

8.1.2 Our Results

Given a set P of n points in the plane, with n even, we consider the problem of packing plane perfect matchings into $K(P)$. From now on, a *matching* will be a *perfect matching*.

In Section 8.3 we prove bounds on the number of plane matchings that can be packed into $K(P)$. In Section 8.3.1 we show that if P is in convex position, then $\frac{n}{2}$ plane matchings can be packed into $K(P)$; this bound is tight.

Table 8.1: Number of plane perfect matchings in a point set P of n points (n is even).

| Matching | $\forall P : \geq$ | $\exists P : \leq$ | $\exists P : \geq$ | $\forall P : \leq$ |
|------------------|--------------------------------|-----------------------------|--------------------|--------------------|
| total | $2^n [22, 34]$ | $2^n [34]$ | $3^n [22]$ | $O(10.05^n) [39]$ |
| edge-disjoint | $\lfloor \log_2 n \rfloor - 1$ | $\lceil \frac{n}{3} \rceil$ | $\frac{n}{2}$ | $n - 1$ |
| plane edge-disj. | 2 | 2 | 5 | 5 |

The points in wheel configurations are considered in Section 8.3.2. We show that if P is in regular wheel configuration, then $\frac{n}{2} - 1$ edge-disjoint plane matchings can be packed into $K(P)$; this bound is tight as well. In addition, for a fixed size set of points, we give a wheel configuration of the points which contains at most $\lceil \frac{n}{3} \rceil$ edge-disjoint plane matchings.

Point sets in general position are considered in Section 8.3.3. We show how to find three edge-disjoint plane matchings in any set of at least 8 points. If n is a power of two, we prove that $K(P)$ contains at least $\log_2 n$ many edge-disjoint plane matchings. For the general case, where n is an even number, we prove that $K(P)$ contains at least $\lfloor \log_2 n \rfloor - 2$ edge-disjoint plane matchings.

In Section 8.3.4 we count the number of pairwise non-crossing plane matchings. Two plane matchings M_1 and M_2 are called *non-crossing* (or *compatible*) if the edges of M_1 and M_2 do not cross each other. We show that $K(P)$ contains at least two and at most five non-crossing plane matchings; these bounds are tight. Table 8.1 summarizes the results.

In Section 8.4 we study the concept of *matching persistency* in a graph. A graph G is called *matching-persistent*, if by removing any perfect matching M from G , the resulting graph, $G - M$, still contains a perfect matching. We define the *plane matching persistency* of a point set P , denoted by $\text{pmp}(P)$, to be the smallest number of edge-disjoint plane matchings such that, if we remove them from $K(P)$ the resulting graph does not have any plane perfect matching. In other words, $\text{pmp}(P) = |\mathcal{M}|$, where \mathcal{M} is the smallest set of edge-disjoint plane matchings such that $K(P) - \bigcup_{M \in \mathcal{M}} M$ does not have any plane perfect matching. Here, the challenge is to find point sets with high plane matching persistency. We show that $\text{pmp}(P) \geq 2$ for all point sets P . We give a configuration of P with $\text{pmp}(P) \geq 3$. Concluding remarks and open problems are presented in Section 8.5.

8.2 PRELIMINARIES

8.2.1 Graph-Theoretical Background

Consider a graph $G = (V, E)$ with vertex set V and edge set E . If G is a complete graph on a vertex set V of size n , then G is denoted by K_n . A k -factor is a regular graph of degree k . If G is the union of pairwise edge-disjoint k -factors, their union is called a k -factorization and G itself is k -factorable [25]. A *matching* in a graph G is a set of edges that do not share vertices. A perfect matching of G is a 1-factor of G . In this chapter only perfect matchings are considered and they are simply called matchings. Since a perfect matching is a regular graph of degree one, it is a 1-factor. It is well-known that for n even, the complete graph K_n is 1-factorable (See [25]). Note that K_n has $\frac{n(n-1)}{2}$ edges and every 1-factor has $\frac{n}{2}$ edges. Thus, K_n can be partitioned into at most $n - 1$ edge-disjoint perfect matchings.

On the other hand it is well-known that the edges of a complete graph K_n , where n is even, can be colored by $n - 1$ colors such that any two adjacent edges have a different color. Each color

is assigned to $\frac{n}{2}$ edges, so that each color defines a 1-factor. The following geometric construction of a coloring, which uses a “regular wheel configuration”, is provided in [41]. In a regular wheel configuration, $n - 1$ equally spaced points are placed on a circle and one point is placed at the center of the circle. For each color class, include an edge e from the center to one of the boundary vertices, and all of the edges perpendicular to the line through e , connecting pairs of boundary vertices.

The number of perfect matchings in a complete graph K_n (with n even), denoted by $M(n)$, is given by the double factorial; $M(n) = (n - 1)!!$ [14], where $(n - 1)!! = 1 \cdot 3 \cdot 5 \cdots (n - 3) \cdot (n - 1)$.

8.2.2 Plane Matchings in Colored Point Sets

Let P be a set of n colored points in general position in the plane with n even. A *colored matching* of P , is a perfect matching such that every edge connects two points of distinct colors. A *plane colored matching* is a colored matching which is non-crossing. A special case of a plane colored matching, where P is partitioned into a set R of $\frac{n}{2}$ red points and a set B of $\frac{n}{2}$ blue points, is called *plane bichromatic matching*, also known as *red-blue matching* (RB-matching). In other words, an RB-matching of P is a non-crossing perfect matching such that every edge connects a red point to a blue point. It is well-known that if no three points of P are collinear, then P has an RB-matching [1]. As shown in Figure 8.1(a), some point sets have a unique RB-matching. Hershberger and Suri [27] construct an RB-matching in $O(n \log n)$ time, which is optimal.

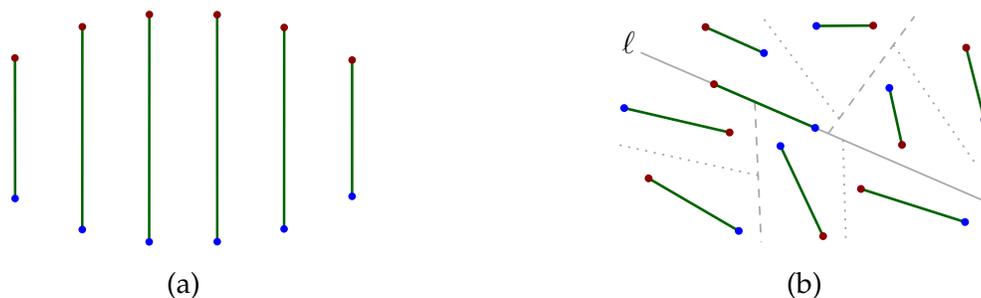


Figure 8.1: (a) A point set with a unique RB-matching, (b) Recursive ham sandwich cuts: first cut is in solid, second-level cuts are in dashed, and third-level cuts are in dotted lines.

We review some proofs for the existence of a plane perfect matching between R and B :

- $\text{Min}(R, B)$: Consider a matching M between R and B which minimizes the total Euclidean length of the edges. The matching M is plane. To prove this, suppose that two line segments $r_1 b_1$ and $r_2 b_2$ in M intersect. By the triangle inequality, $|r_1 b_2| + |r_2 b_1| < |r_1 b_1| + |r_2 b_2|$. This implies that by replacing $r_1 b_1$ and $r_2 b_2$ in M by $r_1 b_2$ and $r_2 b_1$, the total length of the matching is decreased; which is a contradiction.
- $\text{Cut}(R, B)$: The *ham sandwich theorem* implies that there is a line ℓ , known as a *ham sandwich cut*, that splits both R and B exactly in half; if the size of R and B is odd, the line passes through one of each. Match the two points on ℓ (if there are any) and recursively solve the problem on each side of ℓ ; the recursion stops when each subset has one red point and one blue point. By matching these two points in all subsets, a plane perfect matching for P is obtained. See Figure 8.1(b). A ham sandwich cut can be computed in $O(n)$ time [33], and hence the running time can be expressed as the recurrence $T(n) = O(n) + 2 \cdot T(\lfloor \frac{n}{2} \rfloor)$. Therefore, an RB-matching can be computed in $O(n \log n)$ time.

- **Tangent(R, B):** If R and B are separated by a line, we can compute an RB-matching in the following way. W.l.o.g. assume that R and B are separated by a vertical line ℓ . Let $\text{CH}(R)$ and $\text{CH}(B)$ denote the convex hulls of R and B. Compute the upper tangent rb of $\text{CH}(R)$ and $\text{CH}(B)$ where $r \in R$ and $b \in B$. Match r and b , and recursively solve the problem for $R - \{r\}$ and $B - \{b\}$; the recursion stops when the two subsets are empty. In each iteration, all the remaining points are below the line passing through r and b , thus, the line segments representing a matched pair in the successor iterations do not cross rb . Therefore, the resulting matching is plane.

Consider a set P of n points where n is even, and a partition $\{P_1, \dots, P_k\}$ of P into k color classes. Sufficient and necessary conditions for the existence of a colored matching in P follows from the following theorem by Sitton [40]:

THEOREM 8.1 (Sitton [40]). *Let K_{n_1, \dots, n_k} be a complete multipartite graph with n vertices, where $n_1 \leq \dots \leq n_k$. If $n_k \leq n_1 + \dots + n_{k-1}$, then K_{n_1, \dots, n_k} has a matching of size $\lfloor \frac{n}{2} \rfloor$.*

Aichholzer et al. [4] showed that if K_{n_1, \dots, n_k} is a geometric graph corresponding to a colored point set P , then the minimum-weight colored matching of P is non-crossing. Specifically, they extend the proof of 2-colored point sets to multi-colored point sets:

THEOREM 8.2 (Aichholzer et al. [4]). *Let P be a set of colored points in general position in the plane with $|P|$ even. Then P admits a non-crossing perfect matching such that every edge connects two points of distinct colors if and only if at most half the points in P have the same color.*

8.3 PACKING PLANE MATCHINGS INTO POINT SETS

Let P be a set of n points in the plane with n even. In this section we prove lower bounds on the number of plane matchings that can be packed into $K(P)$. It is obvious that every point set has at least one plane matching, because a minimum weight perfect matching in $K(P)$, denoted by $\text{Min}(P)$, is plane. A trivial lower bound of 2 (for $n \geq 4$) is obtained from a minimum weight Hamiltonian cycle in $K(P)$, because this cycle is plane and consists of two edge-disjoint matchings. We consider points in convex position (Section 8.3.1), wheel configuration (Section 8.3.2), and general position (Section 8.3.3).

8.3.1 Points in Convex Position

In this section we consider points in convex position. We show that if P is in convex position, $\frac{n}{2}$ plane matchings can be packed into $K(P)$; this bound is tight.

LEMMA 8.3. *If P is in convex position, where $|P|$ is even and $|P| \geq 4$, then every plane matching in P contains at least two edges of $\text{CH}(P)$.*

Proof. Let M be a plane matching in P . We prove this lemma by induction on the size of P . If $|P| = 4$, then $|M| = 2$. None of the diagonals of P can be in M , thus, the two edges in M belong to $\text{CH}(P)$. If $|P| > 4$ then $|M| \geq 3$. If all edges of M are edges of $\text{CH}(P)$, then the claim in the lemma holds. Assume that M contains a diagonal edge pq , where pq is not an edge of $\text{CH}(P)$. Let P_1 and P_2 be the sets of points of P on each side of $\ell(p, q)$ (both including p and q). Let M_1 and M_2 be the edges of M in P_1 and P_2 , respectively. It is obvious that P_1 (resp. P_2) is in convex position and M_1 (resp. M_2) is a plane matching in P_1 (resp. P_2). By the induction hypothesis M_1

(resp. M_2) contains two edges of $\text{CH}(P_1)$ (resp. $\text{CH}(P_2)$). Since $\text{CH}(P) = \text{CH}(P_1) \cup \text{CH}(P_2)$ and $|M_1 \cap M_2| = 1$, M contains at least two edges of $\text{CH}(P)$. \square

THEOREM 8.4. *For any set P of n points in convex position in the plane, with n even, the maximum number of plane matchings that can be packed into $K(P)$ is $\frac{n}{2}$.*

Proof. By Lemma 8.3, every plane matching in P contains at least two edges of $\text{CH}(P)$. On the other hand, $\text{CH}(P)$ has n edges. Therefore, the number of plane matchings that can be packed into $K(P)$ is at most $\frac{n}{2}$. Bernhart and Kainen [11, Theorem 3.4] showed that the book thickness of the complete graph is $\frac{n}{2}$. Their construction of $\frac{n}{2}$ edge-disjoint paths directly carries over to packing the same amount of plane matchings into $K(P)$.

In order to be self-contained, we show how to pack $\frac{n}{2}$ plane matchings into $K(P)$. Let $P = \{p_0, \dots, p_{n-1}\}$, and w.l.o.g. assume that p_0, p_1, \dots, p_{n-1} is the radial ordering of the points in P with respect to a fixed point in the interior of $\text{CH}(P)$. For each p_i in the radial ordering, where $0 \leq i < \frac{n}{2}$, let $M_i = \{p_{i+j-1}p_{n+i-j} : j = 1, \dots, \frac{n}{2}\}$ (all indices are modulo n). Informally speaking, M_i is a plane perfect matching obtained from edge $p_i p_{i-1}$ and all edges parallel to $p_i p_{i-1}$; see Figure 8.2. Let $\mathcal{M} = \{M_i : i = 0, \dots, \frac{n}{2} - 1\}$. The matchings in \mathcal{M} are plane and pairwise edge-disjoint. Thus, \mathcal{M} is a set of $\frac{n}{2}$ plane matchings that can be packed into $K(P)$. \square

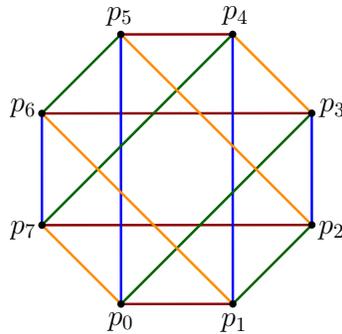


Figure 8.2: Points in convex position.

8.3.2 Points in Wheel Configurations

A point set P of n points is said to be in “regular wheel configuration” in the plane, if $n - 1$ points of P are equally spaced on a circle C and one point of P is at the center of C . We introduce a variation of the regular wheel configuration as follows. Let the point set P be partitioned into X and Y such that $|X| \geq 3$ and $|X|$ is an odd number. The points in X are equally spaced on a circle C . For any two distinct points $p, q \in X$ let $\ell(p, q)$ be the line passing through p and q . Since X is equally spaced on C and $|X|$ is an odd number, $\ell(p, q)$ does not contain the center of C . Let $H(p, q)$ and $H'(p, q)$ be the two half planes defined by $\ell(p, q)$ such that $H'(p, q)$ contains the center of C . Let $C' = \bigcap_{p, q \in X} H'(p, q)$. The points in Y are in the interior of C' ; see Figure 8.3(a). For any two points $p, q \in X$, the line segment pq does not intersect the interior of C' . The special case when $|Y| = 1$ is the regular wheel configuration.

LEMMA 8.5. *Let P be a set of points in the plane where $|P|$ is an even number and $|P| \geq 6$. Let $\{X, Y\}$ be a partition of the points in P such that $|X|$ is an odd number and $|Y| \leq 2 \lfloor \frac{|P|}{6} \rfloor - 1$. If P is in the wheel configuration described above, then any plane matching in P contains at least two edges of $\text{CH}(P)$.*

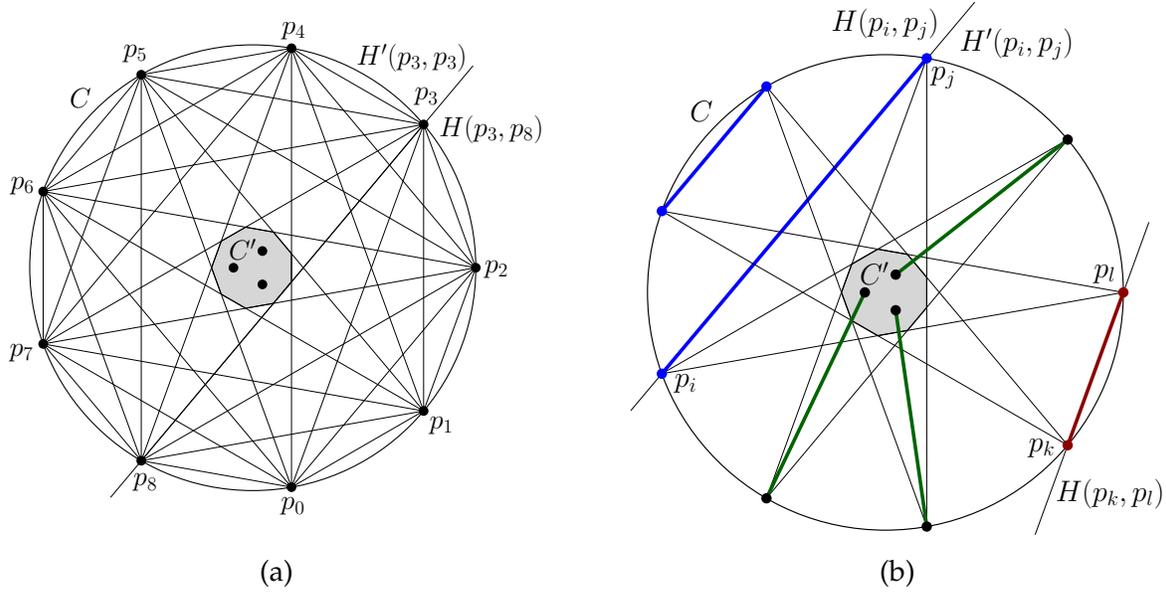


Figure 8.3: (a) A variation of the regular wheel configuration. (b) Illustration of Lemma 8.5. The points of A and the edges of $M(A)$ are in blue, and the points of B and the edges of $M(B)$ are in red.

Proof. Consider a plane matching M of P . It is obvious that $\text{CH}(P) = \text{CH}(X)$; we show that M contains at least two edges of $\text{CH}(X)$. Note that $|X| = |P| - |Y|$, and both $|X|$ and $|Y|$ are odd numbers. Observe that $|X| \geq 4 \lfloor \frac{|P|}{6} \rfloor + 1 = 2|Y| + 3 \geq 5$; which implies that $|Y| \leq \frac{|X|-1}{2} - 1$. Thus, $|X| > |Y|$, and hence there is at least one edge in M with both endpoints in X . Let $p_i p_j$ be the longest such edge. Recall that $C' \subset H'(p_i, p_j)$. Let A be the set of points of P in $H(p_i, p_j)$ (including p_i and p_j), and let A' be the set of points of P in $H'(p_i, p_j)$ (excluding p_i and p_j). By definition, $H(p_i, p_j) \cup \ell(p_i, p_j)$ does not contain any point of Y . Thus, $A \subset X$ and A is in convex position with $|A| \leq \frac{|X|-1}{2}$ (note that $|X|$ is an odd number). Let $M(A)$ and $M(A')$ be the edges of M induced by the points in A and A' , respectively. Clearly, $\{M(A), M(A')\}$ is a partition of the edges of M , and hence $M(A)$ (resp. $M(A')$) is a plane perfect matching for A (resp. A'). We show that each of $M(A)$ and $M(A')$ contains at least one edge of $\text{CH}(X)$. First we consider $M(A)$. If $|A| = 2$, then $p_i p_j$ is the only edge in $M(A)$ and it is an edge of $\text{CH}(X)$. Assume that $|A| \geq 4$. By Lemma 8.3, $M(A)$ contains at least two edges of $\text{CH}(A)$. On the other hand each edge of $\text{CH}(A)$, except for $p_i p_j$, is also an edge of $\text{CH}(X)$; see Figure 8.3(b). This implies that $M(A) - \{p_i p_j\}$ contains at least one edge of $\text{CH}(X)$. Now we consider $M(A')$. Let $X' = A' \cap X$, that is, $\{A, X'\}$ is a partition of the points in X . Since $|A| \leq \frac{|X|-1}{2}$, we have $|X'| \geq \frac{|X|+1}{2}$. Recall that $|Y| \leq \frac{|X|-1}{2} - 1$. Thus, $|Y| < |X'|$, and hence there is an edge $p_k p_l \in M(A')$ with both p_k and p_l in X' . Let B be the set of points of P in $H(p_k, p_l)$ (including p_k and p_l). By definition, $H(p_k, p_l) \cup \ell(p_k, p_l)$ does not contain any point of Y . Thus, $B \subset X$ and B is in convex position. On the other hand, by the choice of $p_i p_j$ as the longest edge, A cannot be a subset of B and hence $B \subset X'$. Let $M(B)$ be the edges of $M(A')$ induced by the points in B . We show that $M(B)$ contains at least one edge of $\text{CH}(X)$. If $|B| = 2$, then $p_k p_l$ is the only edge in $M(B)$ and it is an edge of $\text{CH}(X)$. Assume that $|B| \geq 4$. By Lemma 8.3, $M(B)$ contains at least two edges of $\text{CH}(B)$. On the other hand, each edge of $\text{CH}(B)$, except for $p_k p_l$, is also an edge of $\text{CH}(X)$; see Figure 8.3(b). This implies that $M(B) - \{p_k p_l\}$ contains at least one edge of $\text{CH}(X)$. This completes the proof. \square

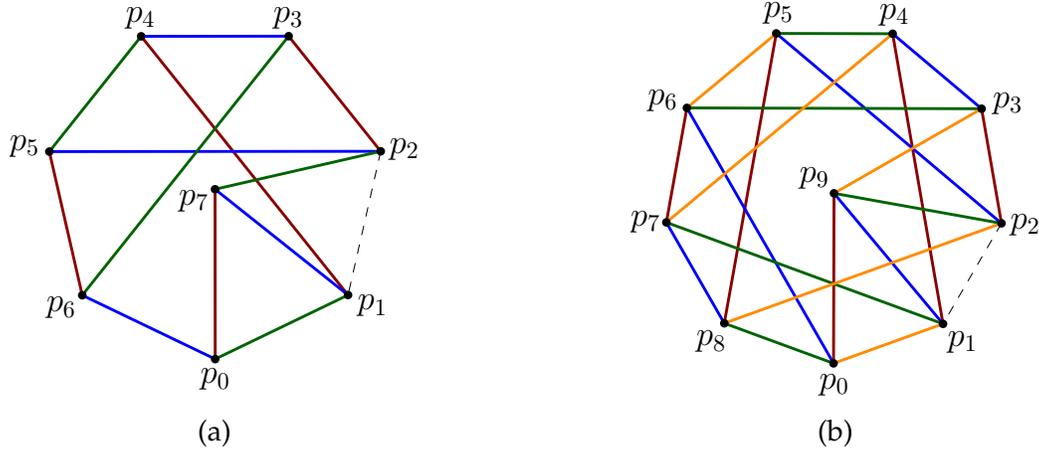


Figure 8.4: Points in the regular configuration with (a) $n = 4k$ and (b) $n = 4k + 2$; one of the edges in $\text{CH}(P)$ cannot be matched.

THEOREM 8.6. *For a set P of $n \geq 6$ points in the regular wheel configuration in the plane with n even, the maximum number of plane matchings that can be packed into $K(P)$ is $\frac{n}{2} - 1$.*

Proof. In the regular wheel configuration, P is partitioned into a point set X of size $n - 1$ and a point set Y of size 1. The points of X are equally spaced on a circle C and the (only) point of Y is the center of C . By Lemma 8.5, every plane matching in P contains at least two edges of $\text{CH}(P)$. On the other hand, $\text{CH}(P)$ has $n - 1$ edges. Therefore, the number of plane matchings that can be packed into $K(P)$ is at most $\frac{n-1}{2}$. Since n is an even number and the number of plane matchings is an integer, we can pack at most $\frac{n}{2} - 1$ plane matchings into $K(P)$.

Now we show how to pack $\frac{n}{2} - 1$ plane matchings into $K(P)$. Let $P = \{p_0, \dots, p_{n-1}\}$, and w.l.o.g. assume that p_{n-1} is the center of C . Let $P' = P - \{p_{n-1}\}$, and let p_0, p_1, \dots, p_{n-2} be the radial ordering of the points in P' with respect to p_{n-1} . For each p_i in the radial ordering, where $0 \leq i \leq \frac{n}{2} - 2$, let

$$R_i = \{p_{i+j}p_{i+2\lceil \frac{(n-2)}{4} \rceil - j + 1} : j = 1, \dots, \lceil \frac{(n-2)}{4} \rceil\},$$

and

$$L_i = \{p_{i-j}p_{i-2\lfloor \frac{(n-2)}{4} \rfloor + j - 1} : j = 1, \dots, \lfloor \frac{(n-2)}{4} \rfloor\}$$

(all indices are modulo $n - 1$). Let $M_i = R_i \cup L_i \cup \{p_i p_{n-1}\}$; informally speaking, M_i is a plane perfect matching obtained from edge $p_i p_{n-1}$ and edges parallel to $p_i p_{n-1}$. See Figure 8.4(a) for the case where $n = 4k$ and Figure 8.4(b) for the case where $n = 4k + 2$. Let $\mathcal{M} = \{M_i : i = 0, \dots, \frac{n}{2} - 2\}$. The matchings in \mathcal{M} are plane and pairwise edge-disjoint. Thus, \mathcal{M} is a set of $\frac{n}{2} - 1$ plane matchings that can be packed into $K(P)$. \square

In the following theorem we use the wheel configuration to show that for any even integer $n \geq 6$, there exists a set P of n points in the plane, such that no more than $\lceil \frac{n}{3} \rceil$ plane matchings can be packed into $K(P)$.

THEOREM 8.7. *For any even number $n \geq 6$, there exists a set P of n points in the plane such that no more than $\lceil \frac{n}{3} \rceil$ plane matchings can be packed into $K(P)$.*

Proof. The set P of n points is partitioned into X and Y , where $|Y| = 2\lfloor \frac{n}{6} \rfloor - 1$ and $|X| = n - |Y|$. The points in X are equally spaced on a circle C and the points in Y are in the interior $\bigcap_{p, q \in X} H'(p, q)$.

By Lemma 8.5, any plane matching in P contains at least two edges of $\text{CH}(P)$. Since $\text{CH}(P) = \text{CH}(X)$, any plane matching of P contains at least two edges of $\text{CH}(X)$. Thus, if \mathcal{M} denotes any set of plane matchings which can be packed into $K(P)$, we have (note that $|X|$ is odd)

$$|\mathcal{M}| \leq \frac{|X| - 1}{2} = \frac{n - 2\lfloor n/6 \rfloor}{2} = \frac{n}{2} - \lfloor \frac{n}{6} \rfloor \leq \frac{n}{2} - \frac{n-5}{6} \leq \lceil \frac{n}{3} \rceil.$$

□

8.3.3 Points in General Position

In this section we consider the problem of packing plane matchings for point sets in general position (no three points on a line) in the plane. Let P be a set of n points in general position in the plane, with n even. Let $M(P)$ denote the maximum number of plane matchings that can be packed into $K(P)$. As mentioned earlier, a trivial lower bound of 2 (when $n \geq 4$) is obtained from a minimum weight Hamiltonian cycle, which is plane and consists of two edge-disjoint perfect matchings.

In this section we show that at least $\lfloor \log_2 n \rfloor - 1$ plane matchings can be packed into $K(P)$. As a warm-up, we first show that if n is a power of two, then $\log_2 n$ plane matchings can be packed into $K(P)$. Then we extend this result to get a lower bound of $\lfloor \log_2 n \rfloor - 1$ for every point set with an even number of points. We also show that if $n \geq 8$, then at least three plane matchings can be packed into $K(P)$, which improves the result for $n = 10, 12$, and 14 . Note that, as a result of Theorem 8.7, there exists a set of $n = 6$ points such that no more than $\lceil \frac{n}{3} \rceil = 2$ plane matchings can be packed into $K(P)$. First consider the following observation.

OBSERVATION 8.8. *Let $\mathcal{P} = \{P_1, \dots, P_k\}$ be a partition of the point set P , such that $|P_i|$ is even and $\text{CH}(P_i) \cap \text{CH}(P_j) = \emptyset$ for all $1 \leq i, j \leq k$ where $i \neq j$. Let i be an index such that, $M(P_i) = \min\{M(P_j) : 1 \leq j \leq k\}$. Then, $M(P) \geq M(P_i)$.*

LEMMA 8.9. *For a set P of n points in general position in the plane, where n is a power of 2, at least $\log_2 n$ plane matchings can be packed into $K(P)$.*

Proof. We prove this lemma by induction. The statement of the lemma holds for the base case, where $n = 2$. Assume that $n \geq 4$. Recall that $M(P)$ denotes the maximum number of plane matchings that can be packed into $K(P)$. W.l.o.g. assume that a vertical line ℓ partitions P into sets R and B , each of size $\frac{n}{2}$. By the induction hypothesis, $M(R), M(B) \geq \log_2(\frac{n}{2})$. By Observation 8.8, $M(P) \geq \min\{M(R), M(B)\} \geq \log_2(\frac{n}{2})$. That is, by pairing a matching M_R in R with a matching M_B in B we get a plane matching M_P in $K(P)$, such that each edge in M_P has both endpoints in R or in B . If we consider the points in R as red and the points in B as blue, $\text{Cut}(R, B)$ (see Section 8.2.2) gives us a plane perfect matching M'_P in $K(P)$, such that each edge in M'_P has one endpoint in R and one endpoint in B . That is $M'_P \cap M_P = \emptyset$. Therefore, we obtain one more plane matching in $K(P)$, which implies that $M(P) \geq \log_2(\frac{n}{2}) + 1 = \log_2 n$. □

Let R and B be two point sets which are separated by a line. A *crossing tangent* between R and B is a line l touching $\text{CH}(R)$ and $\text{CH}(B)$ such that R and B lie on different sides of l . Note that l contains a point $r \in R$, a point $b \in B$, and consequently the line segment rb ; we say that l is subtended from rb . It is obvious that there are two (intersecting) crossing tangents between R and B ; see Figure 8.5.

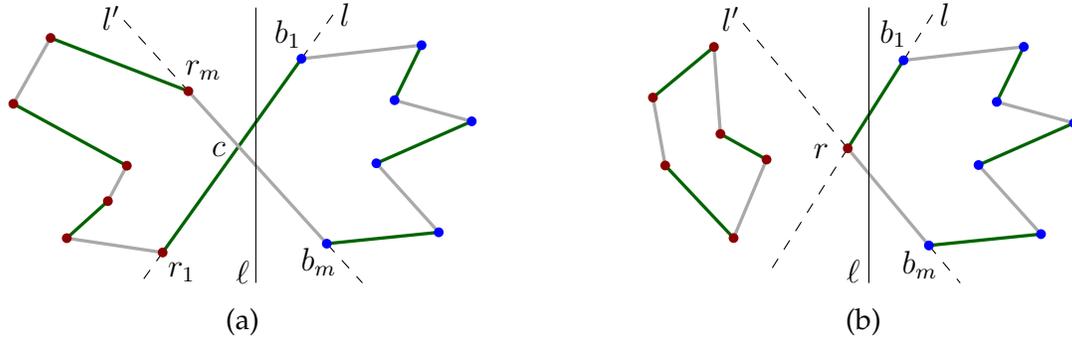


Figure 8.5: (a) The crossing tangents intersect at a point $c \notin P$: R and B are sorted clockwise around c , (b) The crossing tangents intersect at a point $r \in R$: B is sorted clockwise around r . M_1 and M_2 are shown by green and gray line segments.

LEMMA 8.10. *For a set P of $n \geq 8$ points in general position in the plane with n even, at least three plane matchings can be packed into $K(P)$.*

Proof. We describe how to extract three edge-disjoint plane matchings, M_1, M_2, M_3 , from $K(P)$. Let ℓ be a vertical line which splits P into sets R and B , each of size $\frac{n}{2}$. Consider the points in R as red and the points in B as blue. We differentiate between two cases: (a) $n = 4k$ and (b) $n = 4k + 2$, for some integer $k > 1$.

In case (a), both R and B have an even number of points. Let $M_1(R)$ and $M_2(R)$ (resp. $M_1(B)$ and $M_2(B)$) be two edge-disjoint plane matchings in R (resp. B) obtained by a minimum length Hamiltonian cycle in R (resp. B). Let $M_1 = M_1(R) \cup M_1(B)$ and $M_2 = M_2(R) \cup M_2(B)$. Clearly M_1 and M_2 are edge-disjoint plane matchings for P . Let $M_3 = \text{Cut}(R, B)$. It is obvious that M_3 is edge-disjoint from M_1 and M_2 , which completes the proof in the first case.

In case (b), both R and B have an odd number of points and we cannot get a perfect matching in each of them. Let l and l' be the two crossing tangents between R and B , subtended from rb and $r'b'$, respectively. We differentiate between two cases: (i) l and l' intersect in the interior of rb and $r'b'$, (ii) l and l' intersect at an endpoint of both rb and $r'b'$; see Figure 8.5.

- In case (i), let c be the intersection point; see Figure 8.5(a). Let r_1, r_2, \dots, r_m and b_1, b_2, \dots, b_m be the points of R and B , respectively, sorted clockwise around c , where $m = \frac{n}{2}$, $r_1 = r, r_m = r', b_1 = b, b_m = b'$. Consider the Hamiltonian cycle $H = \{r_i r_{i+1} : 1 \leq i < m\} \cup \{b_i b_{i+1} : 1 \leq i < m\} \cup \{r_1 b_1, r_m b_m\}$. Let M_1 and M_2 be the two edge-disjoint matchings obtained from H . Note that $r_1 b_1$ and $r_m b_m$ cannot be in the same matching, thus, M_1 and M_2 are plane. Let $M_3 = \text{Tangent}(R, B)$. As described in Section 8.2.2, M_3 is a plane matching for P . In order to prove that $M_3 \cap (M_1 \cup M_2) = \emptyset$, we show that rb and $r'b'$ —which are the only edges in $M_1 \cup M_2$ that connect a point in R to a point in B —do not belong to M_3 . Note that $\text{Tangent}(R, B)$ iteratively selects an edge which has the same number of red and blue points below its supporting line, whereas the supporting lines of rb and $r'b'$ have different numbers of red and blue points below them. Thus rb and $r'b'$ are not considered by $\text{Tangent}(R, B)$. Therefore M_3 is edge-disjoint from M_1 and M_2 .
- In case (ii), w.l.o.g. assume that l and l' intersect at the red endpoint of rb and $r'b'$, i.e., $r = r'$; See Figure 8.5(b). Let $R' = R \setminus \{r\}$ and $B' = B \cup \{r\}$. Note that both R' and B' have an even number of points and $|R'|, |B'| \geq 4$. Let $M_1(R')$ and $M_2(R')$ be two edge-disjoint plane matchings in R' obtained by a minimum length Hamiltonian cycle in R' . Let b_1, b_2, \dots, b_m

be the points of B sorted clockwise around r , where $m = \frac{n}{2}$, $b_1 = b$, $b_m = b'$. Consider the Hamiltonian cycle $H(R') = \{b_i b_{i+1} : 1 \leq i < m\} \cup \{rb_1, rb_m\}$. Let $M_1(B')$ and $M_2(B')$ be the two edge-disjoint plane matchings in B' obtained from $H(B')$. Let $M_1 = M_1(R') \cup M_1(B')$ and $M_2 = M_2(R') \cup M_2(B')$. Clearly M_1 and M_2 are edge-disjoint plane matchings in P . Let $M_3 = \text{Tangent}(R, B)$. As described in case (i), M_3 is a plane matching in P and $M_3 \cap (M_1 \cup M_2) = \emptyset$. Therefore, M_3 is edge-disjoint from M_1 and M_2 . □

As a direct consequence of Lemma 8.9 and Lemma 8.10 we have the following theorem.

THEOREM 8.11. *For a set P of $n = 2^i \cdot m$ points in general position in the plane with n even, $m \geq 4$, $i \geq 0$, at least $i + 2$ plane matchings can be packed into $K(P)$.*

Proof. If $i = 0$, then a minimum weight Hamiltonian cycle in $K(P)$ consists of two plane matchings. Assume $i \geq 1$. Partition P by vertical lines, into 2^{i-1} point sets, each of size $2m$. By Lemma 8.10, at least three plane matchings can be packed into each set. Considering these sets as the base cases in Lemma 8.9, we obtain $i - 1$ plane matchings between these sets. Thus, in total, $i + 2$ plane matchings can be packed into $K(P)$. □

THEOREM 8.12. *For a set P of n points in general position in the plane, with n even, at least $\lfloor \log_2 n \rfloor - 1$ plane matchings can be packed into $K(P)$.*

Proof. If n is a power of two, then by Lemma 8.9 at least $\log_2 n \geq \lfloor \log_2 n \rfloor - 1$ matchings can be packed into $K(P)$. Assume n is not a power of two. We describe how to pack a set \mathcal{M} of $\lfloor \log_2 n \rfloor - 1$ plane perfect matchings into $K(P)$. The construction consists of the following three main steps which we will describe in detail.

1. Building a binary tree T .
2. Assigning the points of P to the leaves of T .
3. Extracting \mathcal{M} from P using internal nodes of T .

1. building the tree t . In this step we build a binary tree T such that each node of T stores an even number, and each internal node of T has a left and a right child. For an internal node u , let $left(u)$ and $right(u)$ denote the left child and the right child of u , respectively. Given an even number n , we build T in the following way:

- The root of T stores n .
- If a node of T stores 2, then that node is a leaf.
- For a node u storing m , with m even and $m \geq 4$, we store the following even numbers into $left(u)$ and $right(u)$:
 - If m is divisible by 4, we store $\frac{m}{2}$ in both $left(u)$ and $right(u)$; see Figure 8.6(a).
 - If m is not divisible by 4 and u is the root or the left child of its parent then we store $2\lfloor \frac{m}{4} \rfloor$ in $left(u)$ and $m - 2\lfloor \frac{m}{4} \rfloor$ in $right(u)$; see Figure 8.6(b).
 - If m is not divisible by 4 and u is the right child of its parent then we store $m - 2\lfloor \frac{m}{4} \rfloor$ in $left(u)$ and $2\lfloor \frac{m}{4} \rfloor$ in $right(u)$; see Figure 8.6(c).

Note that in the last two cases—where m is not divisible by four—the absolute difference between the values stored in $left(u)$ and $right(u)$ is exactly 2. See Figure 8.7.

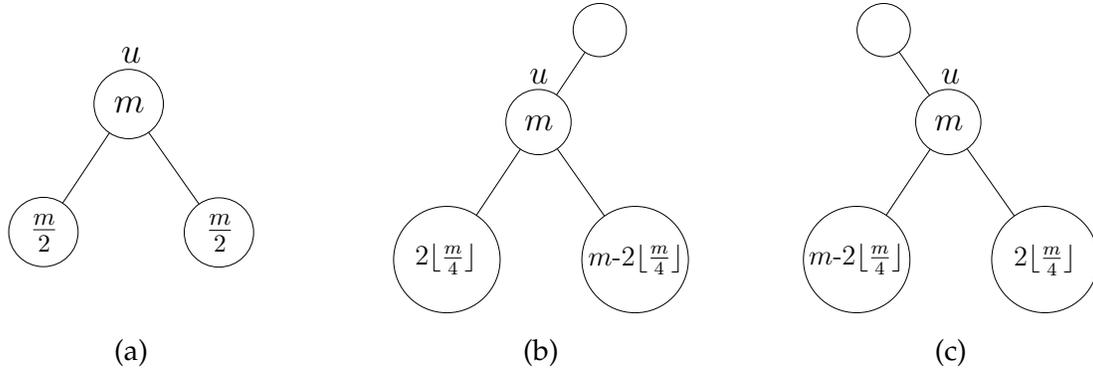


Figure 8.6: (a) m is divisible by four, (b) m is not divisible by four and u is a left child, and (c) m is not divisible by four and u is a right child.

2. *assigning the points to the leaves of the tree.* In this step we describe how to assign the points of P , in pairs, to the leaves of T . We may assume without loss of generality that no two points of P have the same x -coordinate. Sort the points of P in a increasing order of their x -coordinate. Assign the first two points to the leftmost leaf, the next two points to the second leftmost leaf, and so on. Note that T has $\frac{n}{2}$ leaves, and hence all the points of P are assigned to the leaves of T . See Figure 8.7.

3. *extracting the matchings.* Let L be the number of edges in a shortest path from the root to any leaf in T ; in Figure 8.7, $L = 3$. For an internal node $u \in T$, let T_u be the subtree rooted at u . Let L_u and R_u be the set of points assigned to the left and right subtrees of T_u , respectively, and let $P_u = L_u \cup R_u$. Consider the points in L_u as red and the points in R_u as blue. Since the points in L_u have smaller x -coordinates than the points in R_u , we say that L_u and R_u are separated by a vertical line $\ell(u)$. For each internal node u where u is in level $0 \leq i < L$ in T —assuming the root is in level 0—we construct a plane perfect matching M_u in P_u in the following way. Let m be the even number stored at u .

- If m is divisible by 4 (Figure 8.6(a)), then let $M_u = \text{Min}(L_u, R_u)$; see Section 8.2.2. Since $|L_u| = |R_u|$, M_u is a plane perfect matching for P_u . See vertices u_2, u_3 in Figure 8.7.
- If m is not divisible by 4 and u is the root or a left child (Figure 8.6(b)), then $|R_u| - |L_u| = 2$. Let a, b be the two points assigned to the rightmost leaf in T_u , and let $M_u = \{ab\} \cup \text{Min}(L_u, R_u - \{a, b\})$. Since $|L_u| = |R_u - \{a, b\}|$, M_u is a perfect matching in P_u . In addition, a and b are the two rightmost points in P_u , thus, ab does not intersect any edge in $\text{Min}(L_u, R_u - \{a, b\})$, and hence M_u is plane. See vertices u_0, u_1, u_5 in Figure 8.7.
- If m is not divisible by 4 and u is a right child (Figure 8.6(c)), then $|L_u| - |R_u| = 2$. Let a, b be the two points assigned to the leftmost leaf in T_u and let $M_u = \{ab\} \cup \text{Min}(L_u - \{a, b\}, R_u)$. Since $|L_u - \{a, b\}| = |R_u|$, M_u is a perfect matching in P_u . In addition, a and b are the two leftmost points in P_u , thus, ab does not intersect any edge in $\text{Min}(L_u - \{a, b\}, R_u)$, and hence M_u is plane. See vertices u_4, u_6 in Figure 8.7.

For each i , where $0 \leq i < L$, let $S(i)$ be the set of vertices of T in level i ; see Figure 8.7. For each level i let $M_i = \bigcup_{u \in S(i)} M_u$. Let $\mathcal{M} = \{M_i : 0 \leq i < L\}$. In the rest of the proof, we show that \mathcal{M} contains $\lfloor \log_2 n \rfloor - 1$ edge-disjoint plane matchings in P .

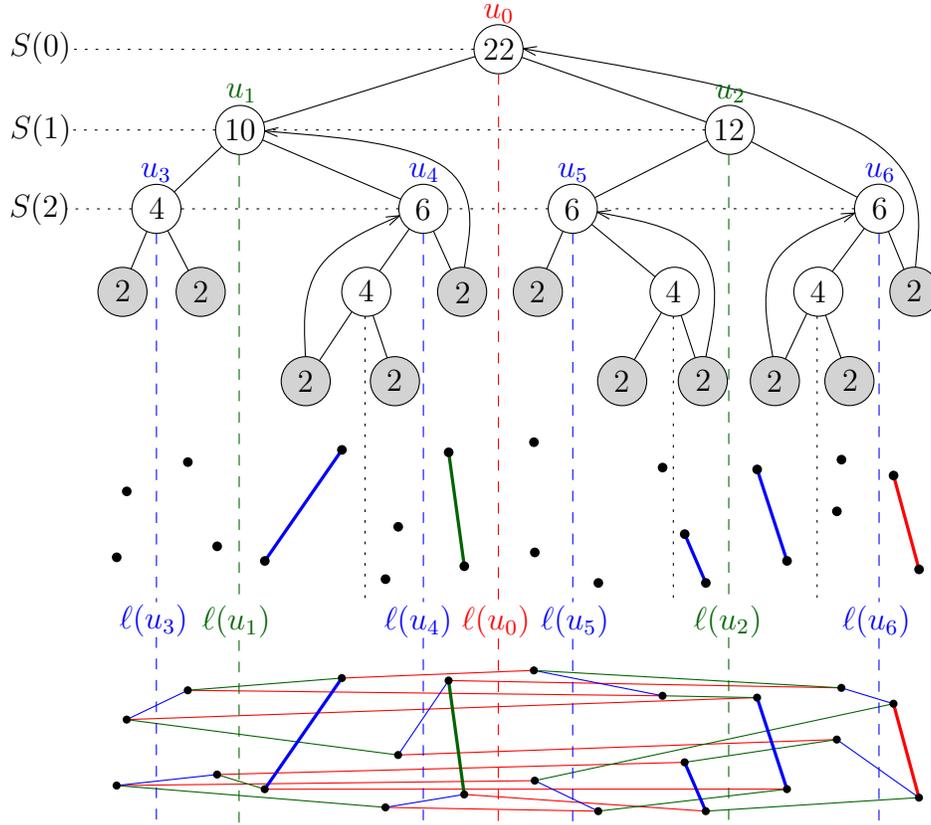


Figure 8.7: The points in P are assigned, in pairs, to the leaves of T , from left to right. The point set P with the edge-disjoint plane matchings is shown as well. M_0 contains the bold red edge and the red edges crossing $\ell(u_0)$. M_1 contains the bold green edge and the green edges crossing $\ell(u_1), \ell(u_2)$. M_2 contains the bold blue edges and the blue edges crossing $\ell(u_3), \ell(u_4), \ell(u_5), \ell(u_6)$.

CLAIM 1. For each i , where $0 \leq i < L$, M_i is a plane perfect matching in P .

Note that if u is the root of the tree, then $P_u = P$. In addition, for each internal node u (including the root), $\{L_u, R_u\}$ is a partition of the point set P_u . This implies that in each level i of the tree, where $0 \leq i < L$, we have $P = \bigcup_{u \in S(i)} P_u$. Moreover, the points in P are assigned to the leaves of T in non-decreasing order of their x -coordinate. Thus, $\mathcal{P}_i = \{P_u : u \in S(i)\}$ is a partition of the point set P ; the sets P_u with $u \in S(i)$ are separated by vertical lines; see Figure 8.7. Therefore, M_i is a perfect plane matching in P ; which proves the claim.

CLAIM 2. For all $M_i, M_j \in \mathcal{M}$, where $0 \leq i, j < L$ and $i \neq j$, $M_i \cap M_j = \emptyset$.

In order to prove that M_i and M_j are edge-disjoint, we show that for each pair of distinct internal nodes u and v , $M_u \cap M_v = \emptyset$. If u and v are in the same level, then P_u and P_v are separated by $\ell(u)$, thus, M_u and M_v do not share any edge. Thus, assume that $u \in S(i)$ and $v \in S(j)$ such that $0 \leq i, j < L$, $i \neq j$, and w.l.o.g. assume that $i < j$. If $v \notin T_u$, then P_u and P_v are separated by line $\ell(w)$, where w is the lowest common ancestor of u and v ; this implies that M_u and M_v do not share any edge. Therefore, assume that $v \in T_u$, and w.l.o.g. assume that v is in the left subtree of T_u . Thus, P_v —and consequently M_v —is to the left of $\ell(u)$. The case where v is in the right subtree of T_u is symmetric. Let $m \geq 4$ be the number stored at u . We differentiate between three cases:

- If m is divisible by 4, then all the edges in M_u cross $\ell(u)$, while the edges in M_v are to the left of $\ell(u)$. This implies that M_u and M_v are disjoint.
- If m is not divisible by 4 and u is the root or a left child, then all the edges of M_u cross $\ell(u)$, except the rightmost edge ab which is to the right of $\ell(u)$. Since M_v is to the left of $\ell(u)$, it follows that M_u and M_v are disjoint.
- If m is not divisible by 4 and u is a right child, then all the edges of M_u cross $\ell(u)$, except the leftmost edge ab . If $a, b \notin P_v$, then $ab \notin M_v$, and hence M_u and M_v are disjoint. If $a, b \in P_v$, then v is the left child of its parent and all the edges in M_v cross $\ell(v)$ (possibly except one edge which is to the right of $\ell(v)$), while ab is to the left of $\ell(v)$. Therefore M_u and M_v do not share any edge. This completes the proof of the claim.

CLAIM 3. For every two nodes u and v in the same level of T which store m and m' , respectively, $|m - m'| \leq 2$.

We prove the claim inductively for each level i of T . For the base case, where $i = 1$: (a) if n is divisible by four, then both u and v store $\frac{n}{2}$ and the claim holds, (b) if n is not divisible by four then u stores $2\lfloor \frac{n}{4} \rfloor$ and v stores $n - 2\lfloor \frac{n}{4} \rfloor$; as $0 \leq n - 2\lfloor \frac{n}{4} \rfloor - 2\lfloor \frac{n}{4} \rfloor \leq 2$, the claim holds for $i = 1$. Now we show that if the claim is true for the i th level of T , then the claim is true for the $(i + 1)$ th level of T . Let u and v , storing m and m' , respectively, be in the i th level of T . By the induction hypothesis, the claim holds for the i th level, i.e., $|m - m'| \leq 2$. We prove that the claim holds for the $(i + 1)$ th level of T , i.e., for the children of u and v . Since m and m' are even numbers, $|m - m'| \in \{0, 2\}$. If $|m - m'| = 0$, then $m = m'$, and by a similar argument as in the base case, the claim holds for the children of u and v . If $|m - m'| = 2$, then w.l.o.g. assume that $m' = m + 2$. Let α be the smallest number and β be the largest number stored at the children of u and v (which are at the $(i + 1)$ th level). We show that $\beta - \alpha \leq 2$. It is obvious that $\alpha = 2\lfloor \frac{m}{4} \rfloor$ and $\beta = m' - 2\lfloor \frac{m'}{4} \rfloor$. Thus,

$$\beta - \alpha = m' - 2\left\lfloor \frac{m'}{4} \right\rfloor - 2\left\lfloor \frac{m}{4} \right\rfloor = m + 2 - 2\left\lfloor \frac{m+2}{4} \right\rfloor - 2\left\lfloor \frac{m}{4} \right\rfloor \quad (8.1)$$

Now, we differentiate between two cases, where $m = 4k$ or $m = 4k + 2$. If $m = 4k$, then by Equation 8.1,

$$\beta - \alpha = 4k + 2 - 2\left\lfloor \frac{4k+2}{4} \right\rfloor - 2\left\lfloor \frac{4k}{4} \right\rfloor = 4k + 2 - 2k - 2k = 2.$$

If $m = 4k + 2$, then by Equation 8.1,

$$\beta - \alpha = 4k + 4 - 2\left\lfloor \frac{4k+4}{4} \right\rfloor - 2\left\lfloor \frac{4k+2}{4} \right\rfloor = 4k + 4 - 2(k+1) - 2k = 2$$

which completes the proof of the claim.

CLAIM 4. $L \geq \lfloor \log_2 n \rfloor - 1$.

It follows from Claim 3 that all the leaves of T are in the last two levels. Since T has $\frac{n}{2}$ leaves, T has $n - 1$ nodes. Recall that L is the number of edges in a shortest path from the root to any leaf in T . Thus, $L \geq h - 1$, where h is the height of T . To give a lower bound on h , one may assume that the last level of T is also full, thus,

$$n - 1 \leq 2^0 + 2^1 + 2^2 + \dots + 2^h \leq 2^{h+1} - 1$$

and hence, $h \geq \log_2 n - 1$. Therefore, $L \geq h - 1 \geq \log_2 n - 2$. Since L is an integer and n is not a power of two, $L \geq \lceil \log_2 n \rceil - 2 = \lfloor \log_2 n \rfloor - 1$; which proves the claim.

Claim 1 and Claim 2 imply that \mathcal{M} contains L edge-disjoint plane perfect matchings. Claim 4 implies that $L \geq \lfloor \log_2 n \rfloor - 1$, which proves the statement of the theorem. \square

8.3.4 Non-crossing Plane Matchings

In this section we consider the problem of packing plane matchings into $K(P)$ such that any two different matchings in the packing are non-crossing. Two edge-disjoint plane matchings M_1 and M_2 are *non-crossing* (or *compatible*), if no edge in M_1 crosses any edge in M_2 . For a set P of n points in general position in the plane, with n even, we give tight lower and upper bounds on the number of pairwise non-crossing plane perfect matchings that can be packed into $K(P)$. The union of k disjoint perfect matchings in $K(P)$ has $\frac{kn}{2}$ edges. By Euler’s formula for planar graphs we know that the number of edges of a planar graph is at most $3n - 6$. This implies the following lemma.

LEMMA 8.13. *For a set P of n points in general position in the plane, with n even, at most five pairwise non-crossing plane matchings can be packed into $K(P)$.*

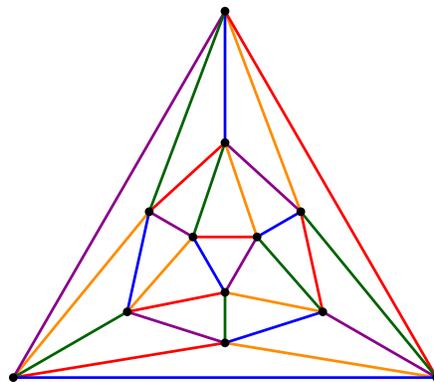


Figure 8.8: A point set with five non-crossing edge-disjoint plane perfect matchings.

Figure 8.8 shows a 5-regular geometric graph on a set of 12 points in the plane which contains five non-crossing edge-disjoint plane matchings. In [26], the authors showed how to generate an infinite family of 5-regular planar graphs using the graph in Figure 8.8. By an extension of the five matchings shown in Figure 8.8, five non-crossing matchings for this family of graphs is obtained. Thus, the bound provided by Lemma 8.13 is tight.

It is obvious that if P contains at least four points, the minimum length Hamiltonian cycle in $K(P)$ contains two non-crossing edge-disjoint plane matchings. Alternatively, in [29] the following result is proved. Given a plane perfect matching M_1 in $K(P)$, we can always find another plane perfect matching M_2 that is disjoint from M_1 and for which $M_1 \cup M_2$ is plane. In the following lemma we show that there exist point sets which contain at most two non-crossing edge-disjoint plane matchings.

LEMMA 8.14. For a set P of $n \geq 4$ points in convex position in the plane, with n even, at most two pairwise non-crossing plane matchings can be packed into $K(P)$.

Proof. The proof is by contradiction. Consider three pairwise non-crossing plane matchings M_1, M_2, M_3 in $K(P)$. Let G be the induced subgraph of $K(P)$ by $M_1 \cup M_2 \cup M_3$. Note that G is a 3-regular plane graph. Moreover, G is an outerplanar graph. It is well known that every outerplanar graph has a vertex of degree at most 2. This contradicts that every vertex in G has degree 3. \square

We conclude this section with the following theorem.

THEOREM 8.15. For a set P of $n \geq 4$ points in general position in the plane, with n even, at least two and at most five pairwise non-crossing plane matchings can be packed into $K(P)$. These bounds are tight.

8.4 MATCHING REMOVAL PERSISTENCY

In this section we define the matching persistency of a graph. A graph G is *matching persistent* if by removing any perfect matching M from G , the resulting graph, $G - M$, has a perfect matching. We define the *matching persistency* of G , denoted by $\text{mp}(G)$, as the size of the smallest set \mathcal{M} of edge-disjoint perfect matchings that can be removed from G such that $G - \mathcal{M}$ does not have any perfect matching. In other words, if $\text{mp}(G) = k$, then

1. by removing an arbitrary set of $k - 1$ edge-disjoint perfect matchings from G , the resulting graph still contains a perfect matching, and
2. there exists a set of k edge-disjoint perfect matchings such that by removing these matchings from G , the resulting graph does not have any perfect matching.

In particular, G is matching persistent iff $\text{mp}(G) \geq 2$.

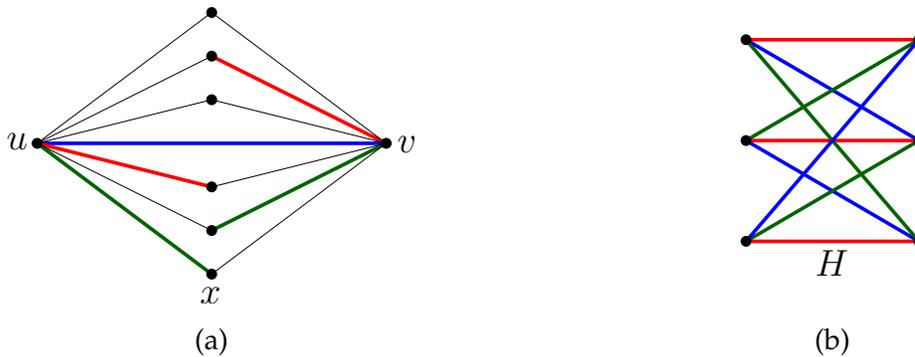


Figure 8.9: (a) By removing any matching (red, green, or blue) from K_n , at most two paths between u and v disappear. (b) The edges of H are partitioned to $\frac{n}{2}$ perfect matchings, thus, $K_n - H$ does not have any perfect matching.

LEMMA 8.16. Let K_n be a complete graph with n vertices, where n is even, and let \mathcal{M} be a set of k edge-disjoint perfect matchings in K_n . Then, $K_n - \mathcal{M}$ is an $(n - 1 - k)$ -regular graph which is $(n - 1 - 2k)$ -connected.

Proof. The regularity is trivial, because K_n is $(n - 1)$ -regular and every vertex has degree k in \mathcal{M} , thus, $K_n - \mathcal{M}$ is an $(n - 1 - k)$ -regular graph. Now we prove the connectivity. Consider two

vertices u and v in $V(K_n)$. There are $n - 1$ many edge-disjoint paths between u and v in K_n : $n - 2$ many paths of length two of the form (u, x, v) , where $x \in V(K_n) - \{u, v\}$ and a path (u, v) of length one; see Figure 8.9(a). By removing any matching in \mathcal{M} from K_n , at most two paths disappear, because u and v have degree one in each matching. Thus in $G - \mathcal{M}$, there are $(n - 1 - 2k)$ many edge-disjoint paths between u and v , which implies that $K_n - \mathcal{M}$ is $(n - 1 - 2k)$ -connected. \square

LEMMA 8.17. *Let n to be an even number. Then, $mp(K_n) \geq \frac{n}{2}$.*

Proof. We show that by removing any set \mathcal{M} of k edge-disjoint perfect matchings from K_n , where $0 \leq k < \frac{n}{2}$, the resulting graph still has a perfect matching. By Lemma 8.16, the graph $K_n - \mathcal{M}$ is an $(n - 1 - k)$ -regular graph which is $(n - 1 - 2k)$ -connected. Since $k < \frac{n}{2}$, $K_n - \mathcal{M}$ is a connected graph and the degree of each vertex is at least $\frac{n}{2}$. Thus, by a result of Dirac [18], $K_n - \mathcal{M}$ has a Hamiltonian cycle and consequently a perfect matching. Therefore, by removing k arbitrary perfect matchings from K_n , where $k < \frac{n}{2}$, the resulting graph still has a perfect matching, which proves the claim. \square

LEMMA 8.18. *If $n \equiv 0 \pmod 4$, then $mp(K_n) \geq \frac{n}{2} + 1$.*

Proof. By Lemma 8.17, $mp(K_n) \geq \frac{n}{2}$. Let \mathcal{M} be any set of $\frac{n}{2}$ edge-disjoint perfect matchings in K_n . We will show that $K_n - \mathcal{M}$ contains a perfect matching. If $K_n - \mathcal{M}$ contains a Hamiltonian cycle then it has a perfect matching and we are done. Assume $K_n - \mathcal{M}$ does not contain any Hamiltonian cycle, while it is a $(\frac{n}{2} - 1)$ -regular graph. A result of Cranston and O [17] implies that $K_n - \mathcal{M}$ is disconnected. In order for $K_n - \mathcal{M}$ to be $(\frac{n}{2} - 1)$ -regular each component has to have at least $\frac{n}{2}$ vertices. Thus, $K_n - \mathcal{M}$ consists of two disjoint copies of $K_{\frac{n}{2}}$. Each of these components has a Hamiltonian cycle, and hence a perfect matching. Therefore, the union of these two components has a perfect matching. \square

THEOREM 8.19. *If $n \equiv 2 \pmod 4$, then $mp(K_n) = \frac{n}{2}$.*

Proof. By Lemma 8.17, $mp(K_n) \geq \frac{n}{2}$. In order to complete the proof, we show that $mp(K_n) \leq \frac{n}{2}$. Let $H = K_{\frac{n}{2}, \frac{n}{2}}$ be a complete bipartite subgraph of K_n . Note that $\frac{n}{2}$ is an odd number and H is an $\frac{n}{2}$ -regular graph. According to Hall's marriage theorem [24], for $k \geq 1$, every k -regular bipartite graph contains a perfect matching [25]. Since by the iterative removal of perfect matchings from H the resulting graph is still regular, the edges of H can be partitioned into $\frac{n}{2}$ perfect matchings; see Figure 8.9(a). It is obvious that $K_n - H$ consists of two connected components of odd size. Thus, by removing the $\frac{n}{2}$ matchings in H , the resulting graph, $K_n - H$, does not have any perfect matching. This proves the claim. \square

THEOREM 8.20. *If $n \equiv 0 \pmod 4$, then $mp(K_n) = \frac{n}{2} + 1$.*

Proof. By Lemma 8.18, $mp(K_n) \geq \frac{n}{2} + 1$. In order to complete the proof, we show that $mp(K_n) \leq \frac{n}{2} + 1$. Assume $n = 4k$. Let $A = \{a_1, \dots, a_{2k-1}\}$ and $B = \{b_1, \dots, b_{2k+1}\}$ be a partition of vertices of K_n . Let M_i be a matching consisting of the edges $b_i b_{i+1}$ and $a_j b_{j+i+1}$, where $+$ is modulo $2k + 1$ and j runs from 1 to $2k - 1$. It is easy to see that M_1, \dots, M_{2k+1} are disjoint perfect matchings, and after removing them we have a complete graph on A and a graph on B , which are disjoint. Since each of A and B has an odd number of points, there is no more perfect matching. This proves the claim. \square

In the rest of this section we consider plane matching removal from geometric graphs.

Let P be a set of n points in general position in the plane, with n even. Given a geometric graph G on P , we say that G is *plane matching persistent* if by removing any plane perfect matching M from G , the resulting graph, $G - M$, has a plane perfect matching. We define the *plane matching persistency* of G , denoted by $\text{pmp}(G)$ as the size of the smallest set \mathcal{M} of edge-disjoint plane perfect matchings that can be removed from G such that $G - \mathcal{M}$ does not have any plane perfect matching. In particular, G is plane matching persistent iff $\text{pmp}(G) \geq 2$.

Aichholzer et al. [4] and Perles (see [30]) showed that by removing any set of at most $\frac{n}{2} - 1$ edges from $K(P)$, the resulting graph has a plane perfect matching. This bound is tight [4]; that is, there exists a point set P such that by removing a set H of $\frac{n}{2}$ edges from $K(P)$ the resulting graph does not have any plane perfect matching. In the examples provided by [4], the $\frac{n}{2}$ edges in H form a connected component which has $\frac{n}{2} + 1$ vertices.

Thus, one may think if the removed edges are disjoint, it may be possible to remove more than $\frac{n}{2} - 1$ edges while the resulting graph has a plane perfect matching. In the following lemma we show that by removing any plane perfect matching, i.e., a set of $\frac{n}{2}$ disjoint edges, from $K(P)$, the resulting graph still has a perfect matching.

LEMMA 8.21. *Let P be a set of n points in general position in the plane with n even, then $\text{pmp}(K(P)) \geq 2$.*

Proof. Let M be any plane perfect matching in $K(P)$. Assign $\frac{n}{2}$ distinct colors to the points in P such that both endpoints of every edge in M have the same color. By Theorem 8.2, P has a plane colored matching, say M' . Since both endpoints of every edge in M have the same color while the endpoints of every edge in M' have distinct colors, M and M' are edge-disjoint. Therefore, by removing any plane perfect matching from $K(P)$, the resulting graph still has a plane perfect matching, which implies that $\text{pmp}(K(P)) \geq 2$. \square

THEOREM 8.22. *For a set P of $n \geq 4$ points in convex position in the plane with n even, $\text{pmp}(K(P)) = 2$.*

Proof. By Lemma 8.21, $\text{pmp}(K(P)) \geq 2$. In order to prove the theorem, we need to show that $\text{pmp}(K(P)) \leq 2$. Let M_1 and M_2 be two edge-disjoint plane matchings obtained from $CH(P)$. By Lemma 8.3, any plane perfect matching in $K(P)$ contains at least two edges of $CH(P)$, while $K(P) - \{M_1 \cup M_2\}$ does not have convex hull edges, and hence does not have any plane perfect matching. Therefore, $\text{pmp}(K(P)) \leq 2$. \square

OBSERVATION 8.23. *The union of two edge-disjoint perfect matchings in any graph is a set of even cycles.*

LEMMA 8.24. *There exists a point set P in general position such that $\text{pmp}(K(P)) \geq 3$.*

Proof. We prove this lemma by providing an example. Figure 8.10(a) shows a set $P = \{a_1, \dots, a_n, b_1, \dots, b_n, c_1, \dots, c_n\}$ of $3n$ points in general position, where n is an even number. In order to prove that $\text{pmp}(K(P)) \geq 3$, we show that by removing any two edge-disjoint plane matchings from $K(P)$, the resulting graph still has a plane perfect matching. Let M_1 and M_2 be any two plane perfect matchings in $K(P)$. Let G be the subgraph of $K(P)$ induced by the edges in $M_1 \cup M_2$. Note that G is a 2-regular graph and by Observation 8.23 does not contain any odd cycle. For each $1 \leq i \leq n$, let t_i be the triangle which is defined by the three points a_i, b_i , and c_i . Let \mathcal{T} be the set of these n (nested) triangles. Since G does not have any odd cycle, for each $t_i \in \mathcal{T}$, at least one edge of t_i is not in G . Let M_3 be the matching containing an edge e_i from each $t_i \in \mathcal{T}$

such that $e_i \notin G$. See Figure 8.10(b). Now we describe how to complete M_3 , i.e., complete it to a perfect matching. Partition the triangles in \mathcal{T} into $\frac{n}{2}$ pairs of consecutive triangles. For each pair (t_i, t_{i+1}) of consecutive triangles we complete M_3 locally—on $a_i, b_i, c_i, a_{i+1}, b_{i+1}, c_{i+1}$ —in the following way. Let $t_i = (a_i, b_i, c_i)$ and $t_{i+1} = (a_{i+1}, b_{i+1}, c_{i+1})$. See Figure 8.10(c). W.l.o.g. assume that M_3 contains $a_i b_i$ and $a_{i+1} c_{i+1}$, that is $a_i b_i \notin G$ and $a_{i+1} c_{i+1} \notin G$. If $c_i b_{i+1} \notin G$, then we complete M_3 by adding $c_i b_{i+1}$. If $c_i b_{i+1} \in G$, then $a_{i+1} b_{i+1} \notin G$ or $c_{i+1} b_{i+1} \notin G$ because b_{i+1} has degree two in G . W.l.o.g. assume that $a_{i+1} b_{i+1} \notin G$. Then we modify M_3 by removing $a_{i+1} c_{i+1}$ and adding $a_{i+1} b_{i+1}$. Now, if $c_i c_{i+1} \notin G$, then we complete M_3 by adding $c_i c_{i+1}$. If $c_i c_{i+1} \in G$, then by Observation 8.23, $b_{i+1} c_{i+1} \notin G$. We modify M_3 by removing $a_{i+1} b_{i+1}$ and adding $b_{i+1} c_{i+1}$. At this point, since $c_i b_{i+1}$ and $c_i c_{i+1}$ are in G , $c_i a_{i+1} \notin G$ and we complete M_3 by adding $c_i a_{i+1}$. \square

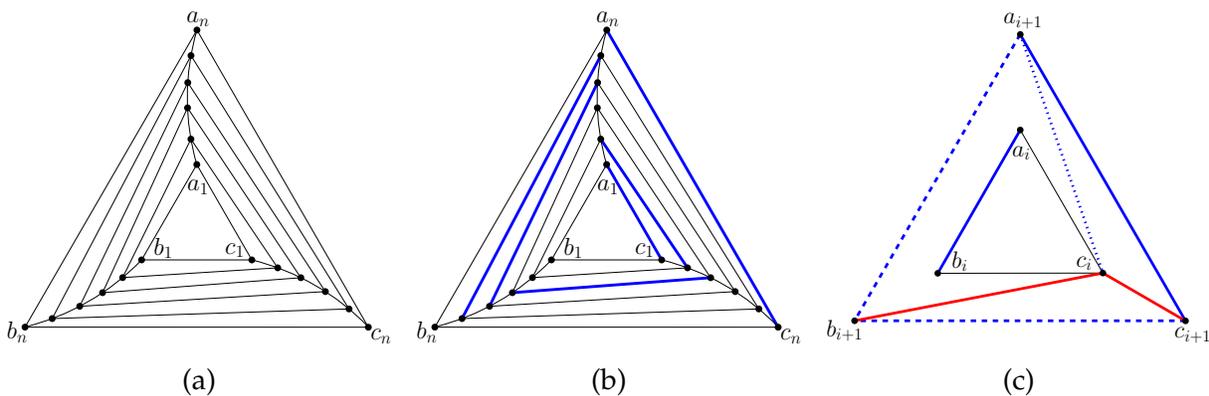


Figure 8.10: (a) Set P of $3n$ points in general position. (b) M_3 contains one edge from each triangle. (c) Locally converting M_3 to a perfect matching, for t_i and t_{i+1} .

8.5 CONCLUSIONS

We considered the problem of packing edge-disjoint plane perfect matchings into a set P of n points in the plane. If P is in general position, we showed how to pack $\lfloor \log_2 n \rfloor - 1$ matchings. We also looked at some special cases and variants of this problem. We believe that the number of such matchings is linear. A natural open problem is to improve either the provided lower bound or the trivial upper bound of $n - 1$, where $n > 6$. Another problem is to provide point sets with large plane matching persistency.

BIBLIOGRAPHY

- [1] The 1979 Putnam exam. In G. L. Alexanderson, L. F. Klosinski, and L. C. Larson, editors, *The William Lowell Putnam Mathematical Competition Problems and Solutions: 1965-1984*. Mathematical Association of America, USA, 1985.
- [2] O. Aichholzer, A. Asinowski, and T. Miltzow. Disjoint compatibility graph of non-crossing matchings of points in convex position. *The Electronic Journal of Combinatorics*, 22(1):P1.65, 2015.

- [3] O. Aichholzer, S. Bereg, A. Dumitrescu, A. G. Olaverri, C. Huemer, F. Hurtado, M. Kano, A. Márquez, D. Rappaport, S. Smorodinsky, D. L. Souvaine, J. Urrutia, and D. R. Wood. Compatible geometric matchings. *Computational Geometry: Theory and Applications*, 42(6-7):617–626, 2009.
- [4] O. Aichholzer, S. Cabello, R. F. Monroy, D. Flores-Peñaloza, T. Hackl, C. Huemer, F. Hurtado, and D. R. Wood. Edge-removal and non-crossing configurations in geometric graphs. *Disc. Math. & Theo. Comp. Sci.*, 12(1):75–86, 2010.
- [5] O. Aichholzer, T. Hackl, M. Korman, M. van Kreveld, M. Löffler, A. Pilz, B. Speckmann, and E. Welzl. Packing plane spanning trees and paths in complete geometric graphs. In *Proceedings of the 26th Annual Canadian Conference on Computational Geometry, CCCG'14*, pages 233–238, 2014.
- [6] M. Ajtai, V. Chvátal, M. M. Newborn, and E. Szemerédi. Crossing-free subgraphs. *Ann. Discrete Math.*, 12:9–12, 1982.
- [7] J. Akiyama and N. Alon. Disjoint simplices and geometric hypergraphs. *Annals of the New York Academy of Sciences*, 555(1):1–3, 1989.
- [8] N. Alon and P. Erdős. Disjoint edges in geometric graphs. *Discrete & Computational Geometry*, 4:287–290, 1989.
- [9] G. Aloupis, J. Cardinal, S. Collette, E. D. Demaine, M. L. Demaine, M. Dulieu, R. F. Monroy, V. Hart, F. Hurtado, S. Langerman, M. Saumell, C. Seara, and P. Taslakian. Non-crossing matchings of points with geometric objects. *Computational Geometry: Theory and Applications*, 46(1):78–92, 2013.
- [10] S. Avital and H. Hanani. Graphs (in Hebrew). *Gilyonot Lematematika*, 3:2–8, 1966.
- [11] F. Bernhart and P. C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979.
- [12] A. Biniarz, P. Bose, A. Maheshwari, and M. H. M. Smid. Packing plane perfect matchings into a point set. *Discrete Mathematics & Theoretical Computer Science*, 17(2):119–142, 2015.
- [13] P. Bose, F. Hurtado, E. Rivera-Campo, and D. R. Wood. Partitions of complete geometric graphs into plane trees. *Computational Geometry: Theory and Applications*, 34(2):116–125, 2006.
- [14] D. Callan. A combinatorial survey of identities for the double factorial. arXiv: 0906.1317, 2009.
- [15] J. Černý. Geometric graphs with no three disjoint edges. *Discrete & Computational Geometry*, 34(4):679–695, 2005.
- [16] J. Černý, Z. Dvořák, V. Jelínek, and J. Kára. Noncrossing Hamiltonian paths in geometric graphs. *Discrete Applied Mathematics*, 155(9):1096–1105, 2007.
- [17] D. W. Cranston and S. O. Hamiltonicity in connected regular graphs. *Information Processing Letters*, 113(22-24):858–860, 2013.
- [18] G. A. Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, 2:69–81, 1952.

- [19] A. Dumitrescu. On two lower bound constructions. In *Proceedings of the 11th Canadian Conference on Computational Geometry, UBC, Vancouver, British Columbia, Canada, August 15-18, 1999*, 1999.
- [20] P. Erdős. On the set of distances of n points. *The American Mathematical Monthly*, 53(5):248–250, 1946.
- [21] S. Felsner. *Geometric Graphs and Arrangements: Some Chapters from Combinatorial Geometry*. Vieweg, Wiesbaden, 2004.
- [22] A. García, M. Noy, and J. Tejel. Lower bounds on the number of crossing-free subgraphs of K_N . *Computational Geometry: Theory and Applications*, 16(4):211–221, 2000.
- [23] W. Goddard, M. Katchalski, and D. J. Kleitman. Forcing disjoint segments in the plane. *European Journal of Combinatorics*, 17(4):391–395, 1996.
- [24] P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935.
- [25] F. Harary. *Graph theory*. Addison-Wesley, 1991.
- [26] M. Hasheminezhad, B. D. McKay, and T. Reeves. Recursive generation of simple planar 5-regular graphs and pentangulations. *Journal of Graph Algorithms and Applications*, 15(3):417–436, 2011.
- [27] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32(2):249–267, 1992.
- [28] H. Hopf and E. Pannwitz. Aufgabe no. 167. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 43:114, 1934.
- [29] M. Ishaque, D. L. Souvaine, and C. D. Tóth. Disjoint compatible geometric matchings. *Discrete & Computational Geometry*, 49(1):89–131, 2013.
- [30] C. Keller and M. A. Perles. On the smallest sets blocking simple perfect matchings in a convex geometric graph. *Israel Journal of Mathematics*, 187(1):465–484, 2012.
- [31] S. Kundu. Bounds on the number of disjoint spanning trees. *Journal of Combinatorial Theory, Series B*, 17(2):199–203, 1974.
- [32] Y. Kupitz. Extremal problems in combinatorial geometry. *Aarhus University Lecture Notes Series*, 53, 1979.
- [33] C. Lo, J. Matousek, and W. L. Steiger. Algorithms for ham-sandwich cuts. *Discrete & Computational Geometry*, 11:433–452, 1994.
- [34] T. Motzkin. Relations between hypersurface cross ratios, and a combinatorial formula for partitions of a polygon, for permanent preponderance, and for non-associative products. *Bull. Amer. Math. Soc.*, 54(4):352–360, 1948.
- [35] C. St. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 36(1):445–450, 1961.

- [36] M. M. Newborn and W. O. J. Moser. Optimal crossing-free Hamiltonian circuit drawings of K_n . *Journal of Combinatorial Theory, Series B*, 29(1):13–26, 1980.
- [37] J. Pach and J. Törőcsik. Some geometric applications of Dilworth’s theorem. *Discrete & Computational Geometry*, 12:1–7, 1994.
- [38] M. Sharir and A. Sheffer. Counting triangulations of planar point sets. *The Electronic Journal of Combinatorics*, 18(1), 2011.
- [39] M. Sharir and E. Welzl. On the number of crossing-free matchings, cycles, and partitions. *SIAM Journal on Computing*, 36(3):695–720, 2006.
- [40] D. Sitton. Maximum matchings in complete multipartite graphs. *Furman University Electronic Journal of Undergraduate Mathematics*, 2:6–16, 1996.
- [41] A. Soifer. *The Mathematical Coloring Book: Mathematics of Coloring and the Colorful Life of its Creators*. New York: Springer, 2009.
- [42] J. W. Sutherland. Lösung der aufgabe 167. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 45:33–35, 1935.
- [43] G. Tóth. Note on geometric graphs. *Journal of Combinatorial Theory, Series A*, 89(1):126–132, 2000.
- [44] G. Tóth and P. Valtr. Geometric graphs with few disjoint edges. *Discrete & Computational Geometry*, 22(4):633–642, 1999.
- [45] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 36(1):221–230, 1961.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and \LyX :

<http://code.google.com/p/classicthesis/>

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

<http://postcards.miede.de/>

Final Version as of October 18, 2016 (`classicthesis` version 4.1).