New Variants of Perfect Non-crossing Matchings

Ioannis Mantas[1(✉)], Marko Savić[2(✉)], and Hendrik Schrezenmaier[3(✉)]

[1] Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland
ioannis.mantas@usi.ch

[2] Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia
marko.savic@dmi.uns.ac.rs

[3] Institut für Mathematik, Technische Universität Berlin, Berlin, Germany
schrezen@math.tu-berlin.de

Abstract. Given a set of points in the plane, we are interested in matching them with straight line segments. We focus on perfect (all points are matched) non-crossing (no two edges intersect) matchings. Apart from the well known MinMax variant, where the length of the longest edge is minimized, we extend work by looking into different optimization variants such as MaxMin, MinMin and MaxMax. We consider both the monochromatic and bichromatic versions of these problems and by employing diverse techniques we provide efficient algorithms for various input point configurations.

Keywords: Perfect · Non-crossing · Matchings · Monochromatic · Bichromatic · Bottleneck · MinMax · MaxMin · MaxMax · MinMin

1 Introduction

In the *matching problem*, given is a set of objects, the goal is to partition the set into pairs such that no object belongs in two pairs. This simple problem is a classic in graph theory, which has received a lot of attention, both in an abstract and in a geometric setting. There are plenty of variants of the problem and there is a great plethora of results.

In this paper, we consider the geometric setting where, given a set P of $2n$ points in the plane, the goal is to match points of P with straight line segments, in the sense that each pair of points induces an *edge* of the matching. A matching is *perfect* if it consists of exactly n pairs. A matching is *non-crossing* if all edges induced by the matching are pairwise disjoint. When there are no restrictions on which pairs of points can be matched, the problem is called *monochromatic*. In the *bichromatic* variant, P is partitioned into two sets B and R of blue and red points, respectively, and only points of different colors are allowed to be matched. When $|B| = |R| = n$, the bichromatic point set P is called *balanced*.

A. Mudgal and C. R. Subramanian (Eds.): CALDAM 2021, LNCS 12601, pp. 151–164, 2021.
https://doi.org/10.1007/978-3-030-67899-9_12

1.1 Related Work

Geometric matchings find applications in many diverse fields, with the most famous perhaps being operations research, where it is known as the *assignment problem*. They are useful in the field of shape matching, when shapes are represented by finite point sets, see e.g., [30], and it is a fundamental problem in pattern recognition. Among others, geometric matchings appear in VLSI design problems, see e.g., [11], in computational biology, see e.g., [10], and are used for map construction or comparison algorithms, see e.g., [16].

Requiring the matching to be non-crossing or perfect is rather natural. Given a monochromatic or balanced bichromatic point set, a perfect non-crossing matching always exists and it can be found in $O(n \log n)$ time by recursively computing *ham-sandwich cuts* [20] or by using the algorithm of Hershberger and Suri [18]. Many times though, not any perfect non-crossing matching is sufficient and the interest lies in finding a matching with respect to some optimization criterion.

A well-studied optimization criterion is minimizing the sum of lengths of all edges, which we call the MinSum variant. It is also known as the *Euclidean assignment* or *Euclidean matching* problem. It is interesting, and not difficult to show, that such a matching is always non-crossing. For monochromatic point sets, an $O(n^{1.5} \log n)$-time algorithm was given by Varadarajan [29]. For bichromatic point sets, Kaplan et al. [19] recently presented an $O(n^2 \log^9 n \lambda_6(\log n))$-time algorithm, outperforming previous results [3,28]. When points are in convex position, Marcotte and Suri [22] solved the problem in $O(n \log n)$ time for both the monochromatic and bichromatic settings.

Another popular goal is to minimize the length of the longest edge, which we call the MinMax variant and is also known as the *bottleneck matching*. Given monochromatic points, Abu-Affash et al. [1] showed that finding such a matching is $\mathcal{NP}$-hard. This was accompanied by an $O(n^3)$-time algorithm for points in convex position. Recently this was improved to $O(n^2)$ time by Savić and Stojaković [24]. For bichromatic points, Carlsson et al. [8] proved that finding a MinMax matching is $\mathcal{NP}$-hard. Biniaz et al. [7] gave algorithms with $O(n^3)$-time for points in convex position and $O(n \log n)$-time for points on a circle. These were improved to $O(n^2)$ and $O(n)$, respectively, by Savić and Stojaković [25].

Several other optimzation goals have been studied. In a *fair matching* the goal is to minimize the length difference between the longest and the shortest edge, and in a *minimum deviation matching*, the difference between the length of the shortest edge and the average edge length should be minimized, see [14,15]. Alon et al. [5] studied the MaxSum variant, where the goal is to maximize the sum of edge lengths. They conjectured that the problem is $\mathcal{NP}$-hard, and gave an approximation algorithm, later improved by Dumitrescu and Toth [12].

1.2 Problem Variants Considered and Our Contribution

In this work, we continue exploring similar optimization variants in different settings and give efficient algorithms for constructing optimal matchings. We only deal with perfect non-crossing matchings, so these properties will always be

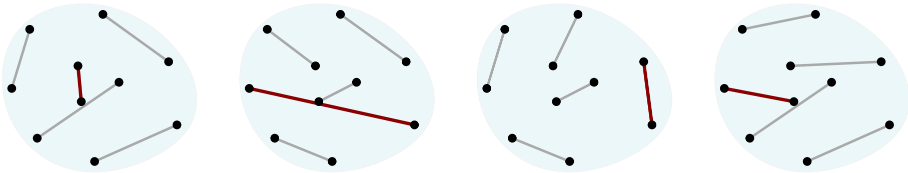

Fig. 1. Optimal MinMin1, MaxMax1, MinMax1, and MaxMin1 matchings of monochromatic points. The edges realizing the values of the matchings are highlighted.

assumed from now on, without further mention. We consider four optimization variants: MinMin where the length of the shortest edge is minimized, MaxMax where the length of the longest edge is maximized, MaxMin where the length of the shortest edge is maximized, and MinMax where the length of the longest edge is minimized. See Fig. 1 for an illustration of these four variants.

To the best of our knowledge, except for MinMax, the other three variants have not been considered before. Studying the MinMin and MaxMax variants is motivated by the analysis of worst-case scenarios for problems where very short or long edges are undesirable, but the selection of edges is not something that we can control. More generally, the values of MinMin and MaxMax serve as lower and upper bounds on the length of any feasible edge and can be helpful in estimating the quality of a matching, with respect to some objective function. The MaxMin variant, similar to MinMax, resembles fair matchings in the sense that all edges have similar lengths, analogously to the variants studied in [14].

We study both the monochromatic and bichromatic versions of these variants in different point configurations. For bichromatic points, we assume that P is balanced. We denote the monochromatic problems with the index 1, e.g., MinMin1, and the bichromatic with the index 2, e.g., MinMin2. In Sect. 2, we consider monochromatic points in general position. In Sect. 3, points are in convex position. In Sect. 4, points lie on a circle. In Sect. 5, we consider *doubly*

Table 1. Summary of results on the optimization of perfect non-crossing matchings. The value of the matching can be obtained in the time not indicated with (*). The time marked with (*) represents the extra time needed to also return a matching. h denotes the size of the convex hull. ε denotes an arbitrarily small positive constant. Results without reference are given in this paper.

Monochromatic	MinMin1	MaxMax1	MinMax1	MaxMin1
General position	$O(nh + n\log n)$, $O(n^{1+\epsilon}+n^{2/3}h^{4/3}\log^3 n)$	$O(nh) + O(n\log n)^*$, $O(n^{1+\epsilon}+n^{2/3}h^{4/3}\log^3 n)$	$\mathcal{NP}$-hard [1]	?
Convex position	$O(n)$	$O(n)$	$O(n^2)$ [24]	$O(n^3)$
Points on circle	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Bichromatic	**MinMin2**	**MaxMax2**	**MinMax2**	**MaxMin2**
General position	?	?	$\mathcal{NP}$-hard [8]	?
Convex position	$O(n)$	$O(n)$	$O(n^2)$ [25]	$O(n^3)$
Points on circle	$O(n)$	$O(n)$	$O(n)$ [25]	$O(n^3)$
Doubly collinear	$O(n)$	$O(1) + O(n)^*$	$O(n^4 \log n)$	?

collinear bichromatic points, where the blue points lie on one line and the red points on another line.

Table 1 summarizes the best-known running times for different matching variants including the contributions of this paper. For each variant, we study their structural properties and combine diverse techniques with existing results in order to tackle as many configurations as possible. The various open questions that arise throughout the paper, pave the way for further research in this family of problems. The proofs which are omitted due to lack of space, together with extra details and suggestions for future work, can be found in our full paper [21].

2 Monochromatic Points in General Position

In this section, P is a monochromatic set of points in general position, where we assume that no three points are collinear. We denote by $\mathrm{CH}(P)$ the boundary of the convex hull of P, by h the number of vertices of $\mathrm{CH}(P)$, by $q_1, \ldots, q_h$ the counterclockwise ordering of the vertices along $\mathrm{CH}(P)$, and by $d(v, w)$ the Euclidean distance between two points v and w. We call an edge (v, w) *feasible*, if there exists a matching which contains (v, w), and *infeasible* otherwise.

The following lemma gives us a feasibility criterion for an edge (v, w).

Lemma 1. *An edge (v, w) is infeasible if and only if (1) $v, w \in \mathrm{CH}(P)$ and (2) there is an odd number of points on each side of (v, w).*

Proof. Let l be the line through the points v, w and let A, B be the subdivision of $P \setminus \{v, w\}$ induced by l. For the if-part, let $v, w \in \mathrm{CH}(P)$. Then each edge (a, b) with $a \in A$, $b \in B$ intersects (v, w) and thus cannot be in a matching with (v, w). If further A, B have an odd number of points each, at least one point from each set will not be matched if (v, w) is in the matching. So, (v, w) is infeasible.

For the only-if-part, let (v, w) be infeasible and suppose that (1) or (2) is not fulfilled. If (2) is not fulfilled, then A, B have an even number of points each. Thus, we can find matchings of A and B independently without intersecting (v, w). Hence, (v, w) is a feasible edge, a contradiction. If (1) is not fulfilled, then not both of v, w are in $\mathrm{CH}(P)$. So, l crosses at least one edge (x, y) of $\mathrm{CH}(P)$, with $x \in A, y \in B$, see Fig. 2a. But then, both $A \setminus \{x\}$ and $B \setminus \{y\}$ contain an even number of points. Thus, there exist matchings of $A \setminus \{x\}$ and $B \setminus \{y\}$, which together with (v, w) and (x, y) form a matching of P, a contradiction. □

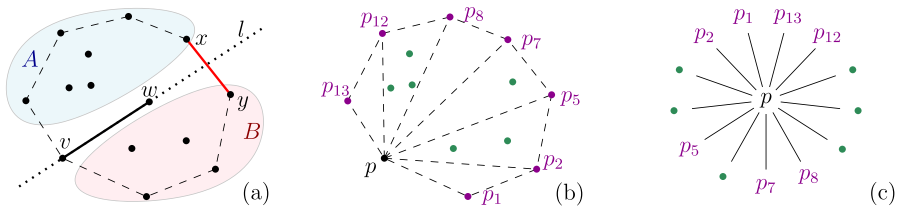

Fig. 2. (a) Illustration for the proof of Lemma 1. (b),(c) The weak radial ordering of p with the points of $P \setminus \mathrm{CH}(P)$ considered as unlabeled points.

2.1 MinMin1 and MaxMax1 Matchings in General Position

The problems MinMin and MaxMax are equivalent to finding the *extremal*, shortest or longest, feasible pair. A main challenge is to check the feasibility of an edge according to Lemma 1. We propose two different approaches.

Using Radial Orderings. The *radial ordering* of a point $p \in P$ is the counterclockwise circular ordering of the points in $P \setminus p$ by angle around p. The radial orderings of all $p \in P$ can be computed in $O(n^2)$ total time using the dual line arrangement of P, see e.g., [4,6].

Given a subset $A \subseteq P$, we define the *A-weak radial ordering* of a point $p \in P$ as the radial ordering of p where the points from A that occur between two points from $\overline{A} := P \setminus A$ are given as an unordered set, see Figs. 2b and 2c. We are interested in the $\overline{\text{CH}(P)}$-weak radial orderings of the points in $\text{CH}(P)$. These are of interest, as they allow us to check the feasibility of all pairs (q_i, q_j) of points $q_i, q_j \in \text{CH}(P)$ in $O(nh)$ total time using Lemma 1.

Lemma 2. *Given a point set P and a subset $A \subseteq P$ with $|P| = n$ and $|A| = k$, the $\overline{A}$-weak radial orderings of all points in A can be computed in $O(nk)$ time.*

Proof. First, we use a point-line duality and compute the dual line arrangement $\mathcal{L}_A$ of A in $O(k^2)$ time [4,6]. We denote the dual line of a point p by l_p. For each edge e of $\mathcal{L}_A$, we initialize a set $X_e := \emptyset$, also in $O(k^2)$ total time. Then, for each point $p \in P \setminus A$, we find the set E_p of edges of $\mathcal{L}_A$ that are intersected by l_p and add p to all sets X_e with $e \in E_p$. Due to the zone theorem [4] this takes $O(k)$ time for each p. Finally, we can read off the weak radial ordering of a point $q \in A$ from $\mathcal{L}_A$ and the sets X_e in the following way: Let $p_1, \ldots, p_{k-1}$ be the ordering of the points in $A \setminus q$ corresponding to the order of intersections of l_q with the other lines in $\mathcal{L}_A$. Further, let e_i be the edge of $\mathcal{L}_A$ between the intersections of l_q with l_{p_i} and $l_{p_{i+1}}$ (with indices understood modulo $k-1$). Then the weak radial ordering of q is $p_1, X_{e_1}, p_2, X_{e_2}, \ldots, X_{e_{k-1}}$. □

We use the feasibility criterion of Lemma 1 and the concept of weak radial orderings to provide algorithms for MinMin1 and MaxMax1.

Theorem 1. *If P is in general position,* MinMin1 *can be solved in $O(nh + n \log n)$ time.*

Proof. We initially construct $\text{CH}(P)$ in $O(n \log h)$ time [9]. Then, we compute the $\overline{\text{CH}(P)}$-weak radial orderings of the points in $\text{CH}(P)$ in $O(nh)$ total time using Lemma 2. Now we look for the shortest feasible edge.

We first consider edges (v, w) with $v \notin \text{CH}(P)$ and we want to find $m_1 := \min(\{\, d(v,w) \colon v \in P \setminus \text{CH}(P), w \in P \,\})$. By Lemma 1, such edges are always feasible. We can find m_1 in $O(n \log n)$ time using a standard algorithm via a Voronoi diagram. Now we consider edges (v, w) with both $v, w \in \text{CH}(P)$ and we want to find $m_2 := \min(\{\, d(v,w) \colon v, w \in \text{CH}(P) \,\})$. By Lemma 1, an edge (q_i, q_{i+1}) is always feasible and an edge (q_i, q_{j+1}) is feasible if and only if (i) (q_i, q_j) is feasible and there is an odd number of points between q_j, q_{j+1} in the radial ordering of q_i or (ii) (q_i, q_j) is infeasible and there is an even number of points between q_j, q_{j+1} in the radial ordering of q_i. Thus, we can find m_2 in $O(nh)$ time, using weak radial orderings. So, we can find the overall minimum $m_{\text{sol}} = \min(m_1, m_2)$, in $O(n \log n + nh)$ time. □

Observe that using the same algorithm but considering the maximum feasible values for m_1, m_2 and m_{sol}, also solves MaxMax1 in $O(nh+n \log n)$ time. Using the following lemma we further improve the time complexity to $O(nh)$.

Lemma 3. *If (v, w) is a longest feasible edge, then one of $v, w \in \text{CH}(P)$.*

Theorem 2. *If P is in general position,* MaxMax1 *can be solved in $O(nh)$ time.*

Proof. The algorithm is similar to the MinMin1, described in Theorem 1, with two changes: The minimizations of m_1, m_2, m_{sol} are replaced by maximizations and, to find m_1, we only consider edges (v, w) with $v \in P \setminus \text{CH}(P)$ and $w \in \text{CH}(P)$. This is sufficient, due to Lemma 3, and reduces the time for finding m_1 to $O((n-h)h)$, by simply comparing all $(n-h)h$ edges. Hence, the overall running time is reduced to $O((n-h)h + nh) = O(nh)$. □

Using Halfplane Range Queries. Now we take another approach to decide the feasibility of a pair of points from CH(P). The task of determining the number of points of a given point set lying on one side of a given straight line is known as *halfplane range query* and has been studied extensively over the last decades, see e.g., [2]. Using these results to check the criterion of Lemma 1, we obtain the following algorithms that are more efficient than those of Theorems 1 and 2, when $h = \Omega(n^c)$ for some constant $c > 0$.

Theorem 3. *Let P be in general position. Then* MinMin1 *and* MaxMax1 *can be solved in $O(n^{1+\epsilon} + n^{2/3}h^{4/3}\log^3 n)$ time where $\epsilon > 0$ is an arbitrary constant.*

Proof. We show that the feasibility of all pairs of points of CH(P) can be decided in the claimed running times. Then, with the aforementioned algorithm and an additional effort of $O(n \log n)$ time, MinMin1 and MaxMax1 can be solved.

We distinguish two classes of values of h. Let $h \leq n^{1/4}$. According to [23], halfplane range queries can be answered in $O(n^{1/2})$ time after a preprocessing step, costing $O(n^{1+\epsilon})$ time. We have to do $\binom{h}{2} = O(h^2)$ queries, so the time needed for the queries is $O(h^2 n^{1/2}) = O(n)$. Therefore the preprocessing step dominates the overall time needed, resulting in $O(n^{1+\epsilon})$ total time.

Now let $h \geq n^{1/4}$. We set $m = n^{2/3}h^{4/3}$. Then we have $n \leq m \leq n^2$, which is required by [23] for the following to hold: Halfplane range queries can be answered in $O(\frac{n}{m^{1/2}} \log^3 \frac{m}{n})$ time after a preprocessing step costing $O(n^{1+\epsilon} + m \log^{\epsilon} n)$ time. Thus the time needed for the $O(h^2)$ queries is $O(n^{2/3}h^{4/3}\log^3 n)$ and for the preprocessing is $O(n^{1+\epsilon} + n^{2/3}h^{4/3}\log^{\epsilon} n)$, so $O(n^{1+\epsilon} + n^{2/3}h^{4/3}\log^3 n)$ time overall. Combining the two cases for h, the claim follows. □

3 Points in Convex Position

In this section, we assume that the points in P are in convex position and their counterclockwise ordering, $p_0, \ldots, p_{2n-1}$, is given. For simplicity, we address points by their indices, i.e., we refer to p_i as i. Arithmetic operations with indices are done modulo $2n$. We call edges of the form $(i, i+1)$ *boundary edges* and we call the remaining edges *diagonals*.

We remark that all four optimization variants, for both monochromatic and bichromatic point sets, can be solved in $O(n^3)$ time by a dynamic programming approach. This approach has also been used in [1,7,8] for MinMax problems. We present more efficient algorithms for MinMin and MaxMax.

3.1 MinMin1 and MaxMax1 Matchings in Convex Position

We make use of the following two algorithms. Given two convex polygons P and Q, Toussaint's algorithm [27] finds in $O(|P| + |Q|)$ time the vertices that realize the minimum distance between P and Q. Analogously, Edelsbrunner's algorithm [13] finds in $O(|P| + |Q|)$ time the vertices that realize the maximum distance between P and Q.

Theorem 4. *If P is in convex position,* MinMin1 *and* MaxMax1 *can be solved in $O(n)$ time.*

Proof. A pair (i, j) is feasible if and only if i and j are of different parity. This suggests that we can split P into two sets, P_{even} and P_{odd}, one containing the even and the other containing the odd indices. Then, any edge (v, w) with $v \in P_{\text{even}}$ and $w \in P_{\text{odd}}$ is feasible. Considering P_{odd} and P_{even} as convex polygons, we apply Toussaint's algorithm [27] for MinMin1 or Edelsbrunner's algorithm [13] for MaxMax1. All steps can be done in $O(n)$ time. □

3.2 MinMin2 and MaxMax2 Matchings in Convex Position

We now combine the monochromatic algorithms with the theory of *orbits* [25], a concept which captures well the nature of bichromatic matchings in convex position. More specifically, P is partitioned into orbits, which are balanced sets

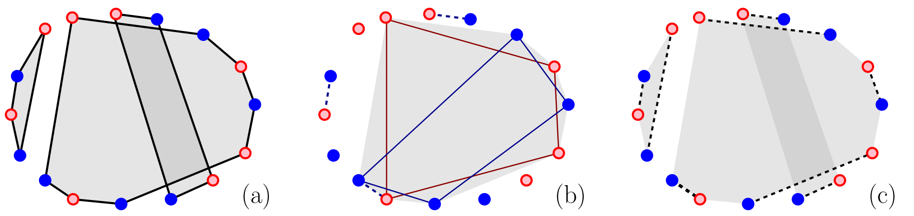

Fig. 3. MinMin2 for P in convex position. (a) Find orbits. (b) Find the shortest edge between the blue and red polygon of an orbit. (c) Extend to a perfect matching. (Color figure online)

of points, and the colors of the points along the boundary of the orbit are alternating, see Fig. 3a. An important property is that a bichromatic edge (b, r) is feasible if and only if b and r are in the same orbit [25].

Theorem 5. *If P is in convex position,* MinMin2 *and* MaxMax2 *can be solved in $O(n)$ time.*

Proof. We first compute all orbits in $O(n)$ time [25]. Due to the alternation of red and blue points along the boundary of the orbits, a single orbit can be considered as a set of points in the monochromatic setting, with respect to the feasibility of the edges. Thus, for MinMin2, we can select for each orbit the shortest edge in $O(n)$ time using Theorem 4. The shortest edge out of all these selected edges is the shortest overall feasible edge of P, see Fig. 3b. Selecting the maximum edges instead solves MaxMax2. □

To construct, in $O(n)$ time, optimal matchings from Theorems 4 and 5 after finding an extremal feasible edge (i, j), we can apply the following lemma to the sets $\{i+1, \ldots, j-1\}$ and $\{j+1, \ldots, i-1\}$, see Fig. 3c. The idea is to pair consecutive edges along the boundary of the convex hull or each orbit.

Lemma 4. *If P is in convex position, we can construct an arbitrary matching in $O(n)$ time, both in the monochromatic and bichromatic case.*

4 Points on a Circle

In this section, we assume that all points lie on a circle. Obviously, the points are in convex position, so all the results from Sect. 3 also apply here. We present algorithms with a better time complexity. We use the notation from Sect. 3.

In addition to convexity, the results in this section rely on a property of points lying on a circle, which we call the *decreasing chords property.* A point set has this property if, for any edge (i, j), in at least one of its sides, an edge between any two points on that side is not longer than (i, j) itself, see Fig. 4a.

Due to the decreasing chords property, we can easily infer the following.

Lemma 5. *Any shortest edge of a matching on P is a boundary edge.*

4.1 MaxMin1 Matching on a Circle

Lemma 5 suggests an approach for MaxMin by *forbidding* short boundary edges and checking whether we can find a matching without them. Let some boundary edges be *forbidden* and the remaining be *allowed.* A *forbidden chain* is a maximal sequence of consecutive forbidden edges. A forbidden chain has endpoints i, j if the edges $(i, i+1), \ldots, (j-1, j)$ are forbidden and the edges $(i-1, i)$ and $(j, j+1)$ are allowed. Refer to Fig. 4b for an illustration.

Lemma 6. *Given a set of forbidden edges, there exists a matching without forbidden edges if and only the length of a longest forbidden chain is less than n.*

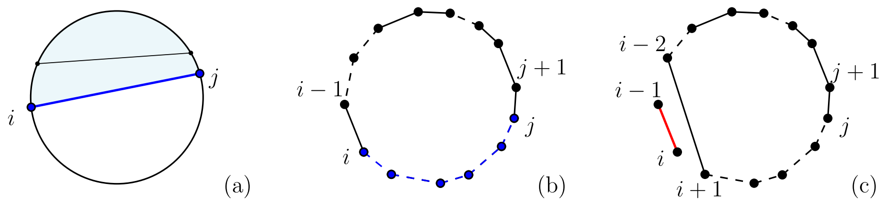

Fig. 4. (a) The decreasing chords property. (b) A forbidden chain with endpoints i, j. (c) Proof of Lemma 6. (Forbidden edges are dashed and allowed edges are solid.)

Proof. If the boundary edges are either all forbidden or all allowed, then the statement trivially holds. So, let us assume that there exists at least one forbidden and at least one allowed boundary edge.

Consider a forbidden chain of length l which has endpoints i and j. First, we assume that $l \geq n$. Then, at least one matched pair (a, b) has both endpoints in $\{i, \ldots, j\}$. Thus, either (a, b) is a forbidden boundary edge or it splits P in a way that all points on one side of the line through (a, b) lie completely in $\{i, \ldots, j\}$. So, there exists a matched boundary edge inside $\{i, \ldots, j\}$ and, thus, a matching without forbidden edges does not exist.

Now let us assume that $l < n$. We construct a matching without forbidden edges using a recursive approach. We match the pair $(i-1, i)$ and consider the set $P' = P \setminus \{i-1, i\}$, see Fig. 4c. In P', $(i-2, i+1)$ is an allowed boundary edge since it is a diagonal in P. We show that P' can be matched by showing that the condition of the lemma holds for P'.

Let i' and j' be the endpoints of a longest forbidden chain in P', going counterclockwise from i' to j', and let l' be its length. If $l' < n-1$, a matching of P' without forbidden edges can be computed recursively. Otherwise, if $l' \geq n-1$, from $l' \leq l < n$ we infer that $l' = l = n-1$. Since $l = l'$, the new longest forbidden chain is disjoint from $\{i+1, \ldots, j\}$, so it is contained in $\{j+1, \ldots, i-2\}$, see Fig. 4c. But since $|P'| = 2n-2$ and $|\{i+1, \ldots, j\}| = n-1$, we have $|\{j+1, \ldots, i-2\}| = n-1$ and thus $l' < n-1$, a contradiction. □

MaxMin is equivalent to finding the largest value μ such that there exists a matching with all edges of length at least μ. By Lemma 5, it suffices to search for μ among the lengths of the boundary edges. By Lemma 6, this means that we need to find the maximal length μ of a boundary edge such that there are no n consecutive boundary edges all shorter than μ. An obvious way to find μ is to employ binary search over the boundary edge lengths and check at each step whether the condition is satisfied or not, which yields an $O(n \log n)$-time algorithm. A faster approach to find μ is as follows. Consider all $2n$ sets of n consecutive boundary edges and associate to each set the longest edge in it. Then, out of the $2n$ longest edges, search for the shortest one. This can be done in $O(n)$ time using a data structure for *range maximum query*, see e.g., [17]. However, our approach fits under the more restricted *sliding window maximum problem*, for which several simple optimal algorithms are known, see e.g., [26].

Theorem 6. *If P lies on a circle,* MaxMin1 *can be solved in $O(n)$ time.*

We can also construct an optimal matching within the same time complexity, as the following lemma states.

Lemma 7. *Given a value $\mu > 0$, a matching consisting of edges of length at least μ can be constructed in $O(n)$ time if it exists.*

4.2 Other Matchings on a Circle

Theorem 7. *If P lies on a circle,* MinMax1 *can be solved in $O(n)$ time.*

Proof. We show that there exists a MinMax1 matching using only boundary edges. Suppose we have a MinMax1 matching M containing a diagonal (i, j). Assume, without loss of generality, that all edges with endpoints in $\{i, \ldots, j\}$ are at most as long as (i, j). We construct a new matching M' by taking all matched pairs in M that are outside of $\{i, \ldots, j\}$ together with edges $(i, i+1), (i+2, i+3), \ldots, (j-1, j)$. The longest edge of M' is not longer than the longest edge of M, proving our claim. There are only two matchings consisting only of boundary edges and in $O(n)$ time we choose the one with the shorter longest edge. □

Points on a circle are in convex position, so, both MinMin1 and MinMin2 can be found in $O(n)$ time using Theorems 4 and 5. Instead, we can do it much simpler by finding the shortest feasible boundary edge. By Lemma 5, the shortest edge of a matching is a boundary edge in both settings. This can then be extended to a perfect matching using Lemma 4.

5 Doubly Collinear Points

In this section, we consider a *doubly collinear* setting. A bichromatic point set P is *doubly collinear* if the blue points lie on a line l_B and the red points lie on a line l_R. We assume that l_B and l_R are not parallel and that the ordering of the points along each line is given. Let $x = l_B \cap l_R$ and assume, for simplicity, that $x \notin P$. Lines l_B and l_R are split at x into two *half-lines*, and the plane is subdivided into four *sectors*. We call a sector *small*, if its angle is acute.

Let $l \in \{l_B, l_R\}$. Then, for two points a, b on l, we denote by (a, b) the open line segment connecting a and b. Further, if $a \neq x$, we denote by $(a_{\rightarrow x \rightarrow})$, ${}_{x\cdots}(a_{\rightarrow}) \subset l$ the open half-lines starting at a that contain x and do not contain x, respectively. If we use square brackets, e.g., in ${}_{x\cdots}[a_{\rightarrow})$, $(a, b]$, or $[a, b]$, the corresponding endpoint is contained in the set.

The following lemma gives us a feasibility criterion for an edge (r, b), see Fig. 5a and 5b, which can be checked in $O(1)$ time. We give a constructive proof that also indicates an algorithm which, given a feasible edge (r, b), returns a matching containing (r, b) in $O(n)$ time.

Lemma 8. *An edge (r, b) is feasible if and only if $|(r, x) \cap P| \leq |(b_{\rightarrow x \rightarrow}) \cap P|$ and $|(b, x) \cap P| \leq |(r_{\rightarrow x \rightarrow}) \cap P|$.*

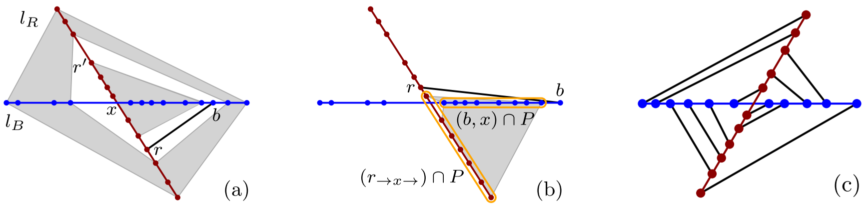

Fig. 5. (a) A feasible edge (r, b). (b) An infeasible edge (r, b). (c) An optimal MinMax2 matching of the type described in Lemma 10.

5.1 MinMin2 and MaxMax2 Matchings on Doubly Collinear Points

Let l'_R and l'_B be a red and a blue half-line, respectively. The following lemma is a consequence of Lemma 8. It allows us to find, for each point in $l'_R \cap P$, the closest point in $l'_B \cap P$ it induces a feasible edge with, in $O(n)$ total time.

Lemma 9. *Let $r \in R$ and $r' \in {}_{x\cdots}(r_{\rightarrow}) \cap P$. Let $b, b' \in l'_B \cap P$ be closest to r and r', respectively, such that (r, b) and (r', b') are feasible. Then $b' \in {}_{x\cdots}[b_{\rightarrow}) \cap P$.*

Theorem 8. *If P is doubly collinear,* MinMin2 *can be solved in $O(n)$ time.*

We call a point $p \in P$ an *extremal* point if $|{}_{x\cdots}(p_{\rightarrow}) \cap P| = 0$. We show that the longest edge between points in R and B is realized by extremal points, so there are $O(1)$ candidate edges. Hence, we can find the longest feasible edge in $O(1)$ time. This can later be extended in $O(n)$ time to an optimal matching, using the algorithm which appears in the constructive proof of Lemma 8.

Theorem 9. *If P is doubly collinear,* MaxMax2 *can be solved in $O(1)$ time.*

5.2 MinMax2 and MaxMin2 Matchings on Doubly Collinear Points

We start by considering the *one-sided* doubly collinear case, where all red points are on the same side of l_B. Then, we turn to the general (two-sided) case. Finally, for the general case, we present improved results for some special cases.

One-Sided Doubly Collinear. Observe that, in this case, the extremal red point must be matched with one of the two extremal blue points. Thus, using dynamic programming, all four optimization variants can be solved in $O(n^2)$ time.

Theorem 10. *If P is one-sided doubly collinear,* MinMax2 *and* MaxMin2 *can be solved in $O(n^2)$ time.*

For MinMax2, we show that there exists (also in the two-sided case) an optimal matching of a special form, described in the following lemma, allowing us to design a faster algorithm. It can be obtained from an arbitrary optimal matching by applying local changes that do not change the objective value.

Lemma 10. *There exists an optimal matching for* MinMax2 *of the following form. For each half-line l', the points of $l' \cap P$ that are matched in the small incident sector are consecutive points, see Fig. 5c.*

Theorem 11. *If P is* one-sided doubly collinear*,* MinMax2 *can be solved in $O(n \log n)$ time.*

General Doubly Collinear. We return to the two-sided case and look at the MinMax2 variant. By only considering matchings of the form described in Lemma 10, enumerating all possible choices for the decision which blue point is matched through which sector, and applying Theorem 11 for the two resulting one-sided subproblems, we obtain the following.

Theorem 12. *If P is doubly collinear,* MinMax2 *can be solved in $O(n^4 \log n)$ time.*

Special Angles of Intersection. Let α be the angle of intersection l_B and l_R, with $\alpha \in (0, \frac{\pi}{2}]$. We prove the existence of optimal matchings having a special form, and we then use these to derive improved algorithms for these cases as follows.

Theorem 13. *If $\alpha = \frac{\pi}{2}$,* MinMax2 *and* MaxMin2 *can be solved in $O(n)$ time.*

Theorem 14. *If $\alpha \leq \frac{\pi}{4}$,* MinMax2 *can be solved in $O(n)$ time.*

6 Conclusions and Future Work

We considered new variants for perfect non-crossing matchings. In most MinMin and MaxMax variants, we came up with optimal algorithms by exploiting structural properties of the point sets, combined with existing techniques from diverse problems. On the contrary, the MaxMin variant exhibits a significant difficulty. Designing efficient algorithms even for simple configurations, as cocircular or doubly collinear, is not at all obvious and thus quite interesting on its own. Throughout the paper many open questions have arisen. For instance, regarding convex bichromatic point sets, can orbits help to improve the MaxMin algorithms? Regarding arbitrary point sets, is there a polynomial time feasibility check for a bichromatic edge? Are the MaxMin variants $\mathcal{NP}$-hard as their MinMax counterparts? It would be interesting to see how Table 1 can be filled with improved algorithms or hardness results.

Acknowledgements. M. S. was partially supported by the Ministry of Education, Science and Technological Development, Republic of Serbia, project 174019, and H. S. by the German Research Foundation, DFG grant FE-340/11-1.

Initial discussions took place at the Intensive Research Program in Discrete, Combinatorial and Computational Geometry which took place in Barcelona in 2018. We are grateful to CRM, UAB for hosting the event and to the organizers for providing the platform to meet and collaborate. We would like to thank Carlos Alegría, Carlos Hidalgo Toscano, Oscar Iglesias Valiño, and Leonardo Martínez Sandoval for preliminary discussions, and Carlos Seara for raising a question that motivated this work. Finally, we would like to thank an anonymous reviewer for bringing to our attention the halfplane range queries.

References

1. Abu-Affash, A.K., Carmi, P., Katz, M.J., Trabelsi, Y.: Bottleneck non-crossing matching in the plane. Comput. Geom. **47**(3A), 447–457 (2014)
2. Agarwal, P.K.: Simplex range searching and its variants: a review. In: Loebl, M., Nešetřil, J., Thomas, R. (eds.) A Journey Through Discrete Mathematics, pp. 1–30. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-44479-6_1
3. Agarwal, P.K., Efrat, A., Sharir, M.: Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. SIAM J. Comput. **29**(3), 912–953 (2000)
4. Agarwal, P.K., Sharir, M.: Arrangements and their applications. In: Handbook of Computational Geometry, chap. 2, pp. 49–119. North-Holland (2000)
5. Alon, N., Rajagopalan, S., Suri, S.: Long non-crossing configurations in the plane. In: Proceedings of the 9th Annual Symposium on Computational Geometry, pp. 257–263 (1993)
6. Asano, T., Ghosh, S.K., Shermer, T.C.: Visibility in the plane. In: Handbook of Computational Geometry, chap. 19, pp. 829–876. North-Holland (2000)
7. Biniaz, A., Maheshwari, A., Smid, M.H.: Bottleneck bichromatic plane matching of points. In: Proceedings of the 26th Canadian Conference on Computational Geometry, pp. 431–435 (2014)
8. Carlsson, J.G., Armbruster, B., Rahul, S., Bellam, H.: A bottleneck matching problem with edge-crossing constraints. Int. J. Comput. Geom. Appl. **25**(4), 245–261 (2015)
9. Chan, T.M.: Optimal output-sensitive convex hull algorithms in two and three dimensions. Discret. Comput. Geom. **16**(4), 361–368 (1996). https://doi.org/10.1007/BF02712873
10. Colannino, J., et al.: An O(n log n)-time algorithm for the restriction scaffold assignment problem. J. Comput. Biol. **13**(4), 979–989 (2006)
11. Cong, J., Kahng, A.B., Robins, G.: Matching-based methods for high-performance clock routing. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **12**(8), 1157–1169 (1993)
12. Dumitrescu, A., Tóth, C.D.: Long non-crossing configurations in the plane. Discret. Comput. Geom. **44**, 727–752 (2010)
13. Edelsbrunner, H.: Computing the extreme distances between two convex polygons. J. Algorithms **6**(2), 213–224 (1985)
14. Efrat, A., Itai, A., Katz, M.J.: Geometry helps in bottleneck matching and related problems. Algorithmica **31**(1), 1–28 (2001)
15. Efrat, A., Katz, M.J.: Computing fair and bottleneck matchings in geometric graphs. In: Asano, T., Igarashi, Y., Nagamochi, H., Miyano, S., Suri, S. (eds.) ISAAC 1996. LNCS, vol. 1178, pp. 115–125. Springer, Heidelberg (1996). https://doi.org/10.1007/BFb0009487
16. Eppstein, D., van Kreveld, M., Speckmann, B., Staals, F.: Improved grid map layout by point set matching. Int. J. Comput. Geom. Appl. **25**(02), 101–122 (2015)
17. Fischer, J., Heun, V.: Space-efficient preprocessing schemes for range minimum queries on static arrays. SIAM J. Comput. **40**(2), 465–492 (2011)
18. Hershberger, J., Suri, S.: Applications of a semi-dynamic convex hull algorithm. BIT Numer. Math. **32**(2), 249–267 (1992)
19. Kaplan, H., Mulzer, W., Roditty, L., Seiferth, P., Sharir, M.: Dynamic planar Voronoi diagrams for general distance functions and their algorithmic applications. Discret. Comput. Geom. **64**(3), 838–904 (2020)

20. Lo, C.-Y., Matoušek, J., Steiger, W.: Algorithms for ham-sandwich cuts. Discret. Comput. Geom. **11**(4), 433–452 (1994). https://doi.org/10.1007/BF02574017
21. Mantas, I., Savić, M., Schrezenmaier, H.: New variants of perfect non-crossing matchings. arXiv preprint arXiv:2001.03252 (2020)
22. Marcotte, O., Suri, S.: Fast matching algorithms for points on a polygon. SIAM J. Comput. **20**(3), 405–422 (1991)
23. Matoušek, J.: Range searching with efficient hierarchical cuttings. Discret. Comput. Geom. **10**(2), 157–182 (1993)
24. Savić, M., Stojaković, M.: Faster bottleneck non-crossing matchings of points in convex position. Comput. Geom. **65**, 27–34 (2017)
25. Savić, M., Stojaković, M.: Bottleneck bichromatic non-crossing matchings using orbits (2018). arxiv.org/abs/1802.06301
26. Tangwongsan, K., Hirzel, M., Schneider, S.: Low-latency sliding-window aggregation in worst-case constant time. In: Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems, pp. 66–77 (2017)
27. Toussaint, G.T.: An optimal algorithm for computing the minimum vertex distance between two crossing convex polygons. Computing **32**(4), 357–364 (1984)
28. Vaidya, P.M.: Geometry helps in matching. SIAM J. Comput. **18**(6), 1201–1225 (1989)
29. Varadarajan, K.R.: A divide-and-conquer algorithm for min-cost perfect matching in the plane. In: Proceedings of the 39th Symposium on Foundations of Computer Science, pp. 320–329 (1998)
30. Veltkamp, R.C., Hagedoorn, M.: State of the art in shape matching. In: Lew, M.S. (ed.) Principles of Visual Information Retrieval. ACVPR, pp. 87–119. Springer, London (2001). https://doi.org/10.1007/978-1-4471-3702-3_4