



HAL
open science

Complexity results on untangling red-blue matchings

Arun Kumar Das, Sandip Das, Guilherme da Fonseca, Yan Gerard, Bastien
Rivier

► **To cite this version:**

Arun Kumar Das, Sandip Das, Guilherme da Fonseca, Yan Gerard, Bastien Rivier. Complexity results on untangling red-blue matchings. *Computational Geometry*, 2023, 111, pp.101974. 10.1016/j.comgeo.2022.101974 . hal-04072822

HAL Id: hal-04072822

<https://hal.science/hal-04072822>

Submitted on 13 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Complexity Results on Untangling Red-Blue Matchings*

Arun Kumar Das[†] Sandip Das[†] Guilherme D. da Fonseca[‡] Yan Gerard[§]
Bastien Rivier[§]

Abstract

Given a matching between n red points and n blue points by line segments in the plane, we consider the problem of obtaining a crossing-free matching through flip operations that replace two crossing segments by two non-crossing ones. We first show that (i) it is NP-hard to α -approximate the shortest flip sequence, for any constant α . Second, we show that when the red points are collinear, (ii) given a matching, a flip sequence of length at most $\binom{n}{2}$ always exists, and (iii) the number of flips in any sequence never exceeds $\binom{n}{2} \frac{n+4}{6}$. Finally, we present (iv) a lower bounding flip sequence with roughly $1.5 \binom{n}{2}$ flips, which shows that the $\binom{n}{2}$ flips attained in the convex case are not the maximum, and (v) a convex matching from which any flip sequence has roughly $1.5n$ flips. The last four results, based on novel analyses, improve the constants of state-of-the-art bounds.

1 Introduction

We consider the problem of untangling a perfect red-blue matching drawn in the plane with straight line segments. We are given a set of $2n$ points in the plane, partitioned into a set R of n red points, and a set B of n blue points, in general position (no three collinear points, unless they have the same color).

In combinatorial reconfiguration, a flip is an operation changing a configuration into another [8, 19]. In our case, a *configuration* is a set of n line segments where each point of R is matched to exactly one point of B , i.e., a perfect straight-line red-blue matching (a *matching* for short), and a *flip* replaces two crossing segments by two non-crossing ones (Figure 1).

The *reconfiguration graph* is the directed simple graph whose vertices \mathcal{V} are the configurations, and such that there is a directed edge from a configuration M_1 to another one M_2 whenever a flip transforms M_1 into M_2 . Note that, in our case, since a flip strictly shortens the total length of the segments (triangle inequality in the two triangles of Figure 1), the reconfiguration graph is acyclic [7]. Let $\mathcal{S} \subseteq \mathcal{V}$ be the set of sinks, which corresponds to the crossing-free matchings. Given two configurations $u, v \in \mathcal{V}$, let $\mathcal{P}(u, v)$ be the set of directed paths from u to v . Given a path P , let the *length* of P , denoted $|P|$, be the number of edges in P . We are interested in two parameters of this reconfiguration graph:

$$\mathbf{d}(R, B) = \max_{u \in \mathcal{V}} \min_{v \in \mathcal{S}} \min_{P \in \mathcal{P}(u, v)} |P| \quad \text{and} \quad \mathbf{D}(R, B) = \max_{u \in \mathcal{V}} \max_{v \in \mathcal{S}} \max_{P \in \mathcal{P}(u, v)} |P|.$$

*Preliminary versions of this paper appeared on the 38th European Workshop on Computational Geometry (EuroCG'22) and the 15th Latin American Theoretical Informatics Symposium (LATIN'22). This work is partially supported by the IFCAM project Applications of Graph Homomorphisms (MA/IFCAM/18/39), and by the French ANR PRC grant ADDS (ANR-19-CE48-0005).

[†]Indian Statistical Institute, Kolkata, India. arund426@gmail.com, sandipdas@isical.ac.in

[‡]Aix-Marseille Université and LIS, France. guilherme.fonseca@lis-lab.fr

[§]Université Clermont Auvergne and LIMOS, France. {yan.gerard, bastien.rivier}@uca.fr

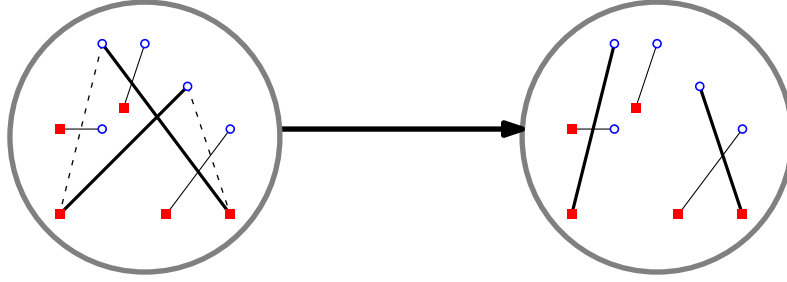


Figure 1: Matchings before and after a flip. Solid squares are red points and hollow circles are blue points.

This leads to the definitions of $\mathbf{d}(n)$ and $\mathbf{D}(n)$ respectively as the maximum of $\mathbf{d}(R, B)$ and $\mathbf{D}(R, B)$ over all sets R, B with $|R| = |B| = n$. An *untangle sequence* is a path in the reconfiguration graph ending in \mathcal{S} . Intuitively, \mathbf{d} corresponds to the minimum length of an untangle sequence in the worst case, while \mathbf{D} corresponds to the longest untangle sequence.

We also consider a more specific version of the problem where the red points are collinear [5], say, on the x -axis. As the flips on each half-plane defined by the x -axis are independent, we additionally suppose all blue points to lie on the upper half-plane without loss of generality. The matchings in this case are called *red-on-a-line* matchings.

Related Work. The parameters \mathbf{d} and \mathbf{D} have been studied in several different contexts with similar definitions of a flip, but considering other configurations.

In 1981, an $O(n^3)$ upper bound on $\mathbf{D}(n)$ was stated in the context of optimizing a TSP tour [26] (the configurations are polygons). This upper bound should be compared to the exponential lower bound on $\mathbf{D}(n)$ when the flips are not restricted to crossing segments, as long as they decrease the Euclidean length of the tour [12]. The convex case (i.e., the case where the points are in convex position) has been studied in [23, 28].

In the non-bipartite version of the straight-line perfect matching problem, there are two possible pairs of segments to replace a crossing pair. This additional choice yields an $n^2/2$ upper bound on $\mathbf{d}(n)$ [7].

It is also possible to relax the flip definition to all operations that replace two segments by two others with the same four endpoints, whether they cross or not, and generalize the configurations to multigraphs with the same degree sequence [14, 15, 19]. In this context, finding the shortest path from a given configuration to another in the reconfiguration graph is NP-hard, yet 1.5-approximable [3, 4, 13, 27]. If we additionally require the configurations to be connected graphs, the same problem is NP-hard and 2.5-approximable [9].

Reconfiguration problems in the context of triangulations are widely studied [22]. A flip consists of removing one edge and adding another one while preserving a triangulation. It is known that $\Theta(n^2)$ flips are sufficient and sometimes necessary to obtain a Delaunay triangulation [17, 20]. Determining the flip distance between two triangulations of a point set [21, 24] and between two triangulations of a simple polygon [1] are both NP-hard.

Considering perfect matchings of an arbitrary graph (instead of the complete bipartite graph on R, B), a flip amounts to exchanging the edges in an alternating cycle of length four. It is then PSPACE-complete to decide whether there exists a path from a configuration to another [6]. There is, actually, a wide variety of reconfiguration contexts derived from NP-complete problems where this same accessibility problem is PSPACE-complete [18]. Many other reconfiguration problems are presented in [25].

Getting back to our context of straight-line red-blue matchings, the values of \mathbf{d} and \mathbf{D} have been determined almost exactly in the convex case (see Table 1). Notice that the $n - 1$ lower bound on $\mathbf{d}(n)$ carries to both the general and red-on-a-line cases [7]. It is notable that the

Table 1: Lower and upper bounds on $\mathbf{d}(n)$ and $\mathbf{D}(n)$ for red-blue matchings.

| | $\mathbf{d}(n)$ bounds | | $\mathbf{D}(n)$ bounds | |
|---------------|--------------------------------|-------------------------------|---|--|
| | lower | upper | lower | upper |
| general | $\frac{3}{2}n - 2$, Thm. 5.2* | $\binom{n}{2}(n-1)$, [7, 26] | $\frac{3}{2}\binom{n}{2} - \frac{n}{4}$, Thm. 5.1* | $\binom{n}{2}(n-1)$, [7, 26] |
| convex | $\frac{3}{2}n - 2$, Thm. 5.2* | $2n - 2$, [5] | $\binom{n}{2}$, [7] | $\binom{n}{2}$, [5] |
| red-on-a-line | $n - 1$, [7] | $\binom{n}{2}$, Thm. 3.1 | $\frac{3}{2}\binom{n}{2} - \frac{n}{4}$, Thm. 5.1* | $\binom{n}{2}\frac{n+4}{6}$, Thm. 4.1 |

* For even n .

upper bound on $\mathbf{D}(n)$ is also the best known bound on $\mathbf{d}(n)$ and has not been improved since 1981 [26].

As a final side note, given a red-blue point set, a crossing-free red-blue matching can be computed in $O(n \log n)$ time [16]. The algorithm is based on semi-dynamic convex hull data structures and does not use flips. The problem has also been considered in higher dimensions [2].

Contributions. We show in Section 2 that it is NP-hard to α -approximate the shortest untangle sequence starting at a given matching, for any fixed $\alpha \geq 1$.

The following results are summarized in Table 1. An improved lower bound on $\mathbf{d}(n)$ in the convex case is presented in Section 5.2. The remainder of the paper considers the red-on-a-line case. In Section 3, we slightly improve the former $2\binom{n}{2}$ upper bound on $\mathbf{d}(n)$ [5], using a simpler algorithm and a novel analysis. In Section 4, we asymptotically divide by 6 the historical $\binom{n}{2}(n-1)$ upper bound on $\mathbf{D}(n)$ [7, 26], using a different potential argument.

In Section 5.1, we present a counterexample to the intuition that the longest untangle sequence is attained in the convex case (where the number of crossings is maximal). We take advantage of points that are not in convex position to increase the lower bound by a factor of $\frac{3}{2}$. This red-on-a-line lower bound on $\mathbf{d}(n)$ carries over to the general case (and even to non-bipartite perfect matchings). The conjecture that $\mathbf{D}(n)$ is quadratic [7] remains open, though.

2 NP-Hardness

In this section, we prove the NP-hardness of the following problem. Let $d(M)$ denote the minimum path length from a matching M to \mathcal{S} , the set of crossing-free matchings, in the reconfiguration graph.

Problem 1. *Let $\alpha \geq 1$ be a constant.*

Input: M , a red-blue matching with rational coordinates.

Output: An untangle sequence starting at M of length at most α times $d(M)$.

We have the following theorem.

Theorem 2.1. *Problem 1 is NP-hard for all $\alpha \geq 1$.*

Reduction Strategy. De Berg and Khosravi [11] showed that the rectilinear planar monotone 3-SAT problem (*RPM 3-SAT*) is NP-hard. The RPM 3-SAT problem is a special case of the classic 3-SAT problem in which the clauses consist only of either all positive or all negative literals and the layout is planar (Figure 2). We reduce RPM 3-SAT to Problem 1. The key elements of the reduction are described next.

Given a planar embedding of an RPM 3-CNF formula Φ (Figure 2), we construct a matching M_Φ of polynomial size. The property of this matching M_Φ is that its shortest untangle sequence has a length below a certain constant if Φ is satisfiable and above α times this constant otherwise. Figure 3 shows the matching M_Φ corresponding to the formula $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_4 \vee x_5) \wedge (x_3 \vee x_5 \vee x_6) \wedge (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)$ from Figure 2.

The aforesaid matching M_Φ is built using two types of gadgets. The variable rectangles are replaced by variable gadgets (Figure 4). The clause rectangles together with the corresponding edges are replaced with padded clause gadgets. A padded clause gadget is represented in Figure 8 with plain segments. Throughout all the figures in Section 2, the dashed segments represent all the possibly created segments after any sequence of flips.

A *variable gadget* is a three-segment matching with two crossings. It allows for two possible flips, either of which produces a crossing-free matching, as shown in Figure 4. The flip generating the topmost segment stands for *false* ($x = 0$ in Figure 4), while the flip generating the bottom segment stands for *true* ($x = 1$).

A *clause gadget* is an OR gate with three inputs (Figure 9). The RPM 3-CNF clauses are either positive or negative. We describe the gadget for a positive clause, but the gadget for a negative clause can be defined analogously (by a vertical reflection). Three variable gadgets are the inputs of a clause gadget. In the crossing-free matching obtained for the clause gadget, the presence of the topmost segment (r_4b_7 in Figures 6, 7, 8, and 9) stands for a false output.

A *padding gadget* is a gadget that serves to force an arbitrarily large number k of flips if a clause is false. It consists of a series of k non-crossing segments (the plain segments in Figure 7, r_4b_7 aside). A *padded clause gadget* is a clause gadget coupled with a padding gadget in such a way that the presence of the output segment triggers k extra flips (Figure 8).

Let c be the number of clauses and v be the number of variables of the formula Φ . If Φ is *satisfiable*, then the shortest untangle sequence of M_Φ has at most 5 flips per clause plus 1 flip per variable. In this case, we have $d(M_\Phi) \leq 5c + v$. We choose the size of the padding gadget so that a non-satisfied clause triggers $k = \alpha(5c + v) + 1$ flips. If the formula Φ is *not satisfiable*, then at least one of the padding gadgets is triggered and $d(M_\Phi) > \alpha(5c + v)$.

The Problem to Be Reduced. In *RPM 3-SAT*, the *graph* of a conjunctive normal form (CNF) formula is the bipartite graph with the variables and clauses as vertices, and where there is an edge between a variable and a clause if and only if the clause contains the variable. A clause is said to be *positive* if it contains only positive variables; it is said to be *negative* if it contains only negative variables. A CNF formula is *monotone* if each clause is either positive or negative.

A *rectilinear planar monotone 3-CNF (RPM 3-CNF)* formula is a monotone formula with 3-variables per clause whose graph can be drawn with the following conventions (Figure 2). (i) The variables and the clauses are represented by axis-parallel non-overlapping closed rectangles. (ii) The variable rectangle centroids lie on the x -axis. (iii) The positive clause rectangles are above the x -axis, the negative ones, below. (iv) The edges connecting a variable to a clause are vertical line segments and do not cross any other rectangle. We call such a drawing a *planar embedding* of Φ .

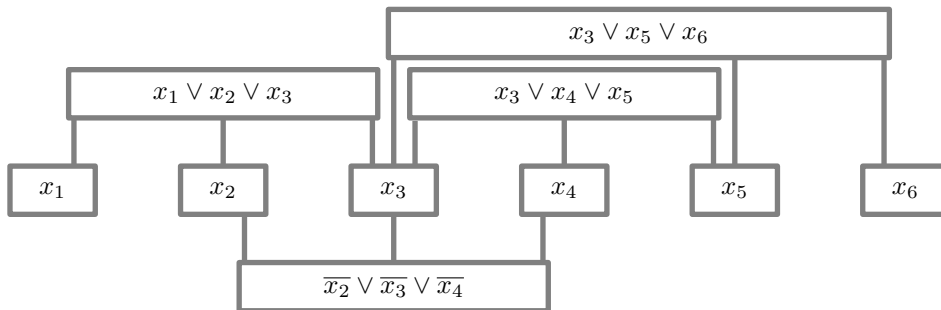


Figure 2: A planar embedding of an RPM 3-CNF formula Φ .

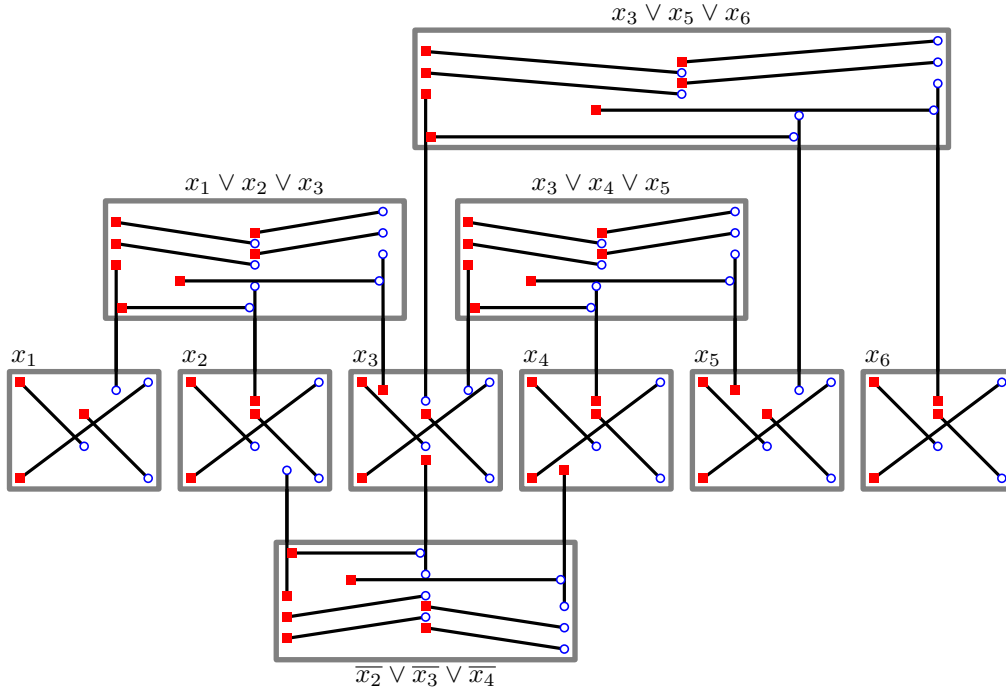


Figure 3: The matching M_Φ of the formula Φ from Figure 2.

Variable Gadgets. A *variable gadget* is a three-segment matching built on the four endpoints of an axis-parallel rectangle as follows (Figure 4). The two leftmost endpoints of the rectangle are colored red, the two rightmost ones are colored blue. One of the segments of the matching is the diagonal joining the bottom left red point to the top right blue point. We add one red point on the vertical line splitting the rectangle in two symmetric halves, just above the diagonal, in the inside of the rectangle. This red point is connected to the bottom right blue point. Similarly, we add one blue point on the same vertical, just below the diagonal. This blue point is connected to the top left red point.

We will refer to the triangle consisting of the three topmost points of a variable gadget as the *top triangle* of the variable gadget.

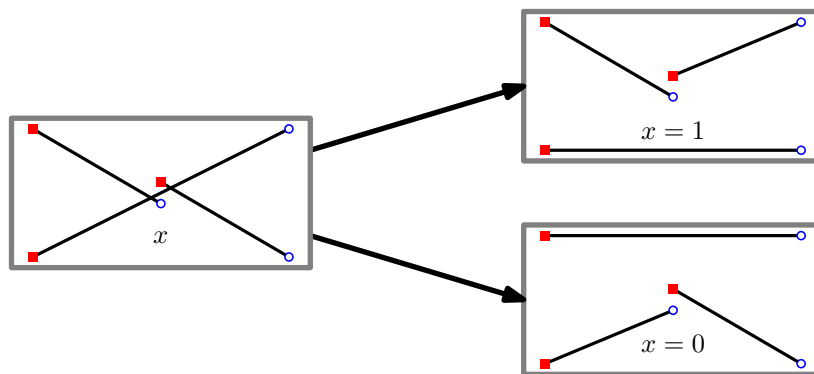


Figure 4: A variable gadget and its two untangle sequences.

Lemma 2.2. A *variable gadget* is the starting matching of exactly two untangle sequences of length 1 ending in distinct matchings.

Proof. It is straightforward to check the two possible cases. □

We can therefore represent each variable x of a propositional formula by a variable gadget.

Assigning x to a truth value amounts to choosing one of the two possible untangle sequences, with the convention that the lower edge of the rectangle is present in the final matching if $x = 1$ (i.e., x is “true”), and that the upper edge of the rectangle is present if $x = 0$ (Figure 4).

OR Gadgets. An *OR gadget* consists of four three-segment matchings built on a common point set, say $\{r_1, r_2, r'_2, r_3\}$ for the red points, and $\{b_1, b'_1, b_2, b_3\}$ for the blue points, as follows (see the first matching in each of Figures 5(a), 5(b), 5(c), and 5(d), ignoring the dashed segments). The $0 \vee 0$ matching consists of the segments $r_1b'_1, r'_2b_2, r_3b_3$, and only the first two are not crossing. The $0 \vee 1$ matching consists of the segments $r_1b'_1, r_3b_3, r_2b_2$, and only the first two are crossing. The $1 \vee 0$ matching consists of the segments r'_2b_2, r_3b_3, r_1b_1 , and only the first two are crossing. The $1 \vee 1$ matching consists of the segments r_1b_1, r_2b_2, r_3b_3 , and is crossing-free. In addition to these constraints, the point set also satisfies the following ones. The following three matchings are crossing-free: $\{r_1b_2, r'_2b_3, r_3b'_1\}$, $\{r_1b_3, r_2b_2, r_3b'_1\}$, and $\{r_1b_1, r'_2b_3, r_3b_2\}$. In each of the following two matchings, only the first two segments are crossing: $\{r_1b_3, r'_2b_2, r_3b'_1\}$, and $\{r_1b'_1, r_3b_2, r'_2b_3\}$.

Note that, in any of the four matchings of an OR gadget, there is one unused blue point and one unused red point. If the unused blue point is b_1 (respectively b'_1), we say that the *left input* of the OR gadget is 0 (respectively 1). Similarly, if the unused red point is r_2 (respectively r'_2), we say that the *right input* of the OR gadget is 0 (respectively 1). To complete the similarity with a logical gate, we also define the *output* of the OR gadget as 0 if the segment r_1b_2 is present in all the final matchings of any untangle sequence starting at the OR gadget and as 1 if the segment r_1b_2 is absent of all the same final matchings. The output is undefined otherwise. The following lemma states that the truth table of the logical gate associated with an OR gadget is indeed the one of an OR gate.

We will refer to the smallest of the triangles consisting of the segment r_1b_2 and induced by all the other segments we have mentioned in the definition of an OR gadget as the *top triangle* of the OR gadget. It is the shaded triangle in Figure 5(d).

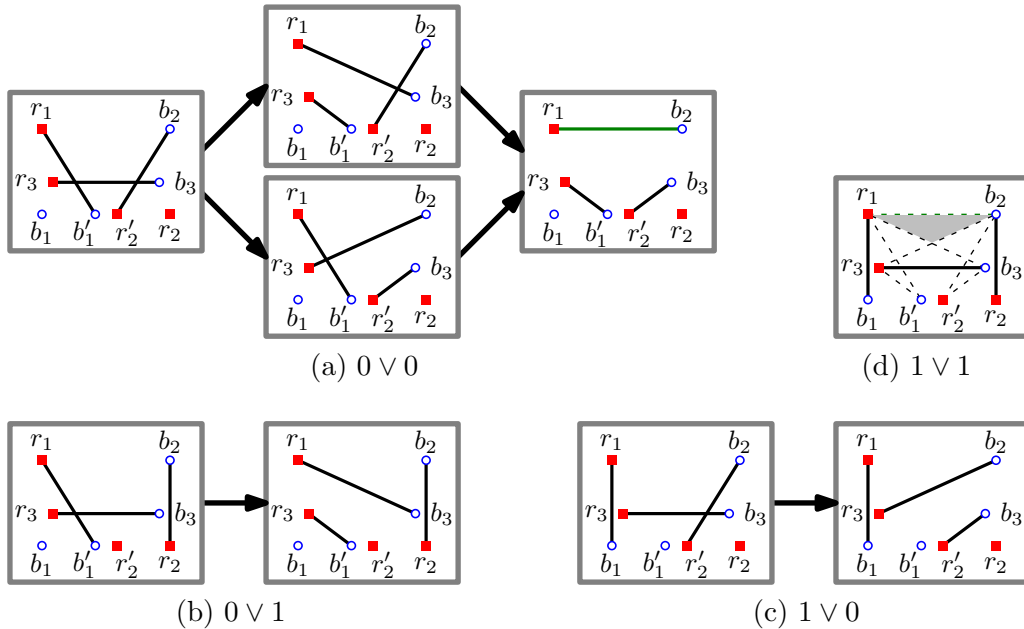


Figure 5: The four matchings of an OR gadget, with their untangle sequences.

Lemma 2.3. *The output of an OR gadget is always well defined, and is 0 if and only if the two inputs of the OR gadget are both 0. More precisely, we have the following.*

1. *The $0 \vee 0$ matching is the starting matching of exactly two untangle sequences, each of length 2, and ending at the same matching containing the upper segment r_1b_2 (Figure 5(a)).*

2. The $0 \vee 1$ matching is the starting matching of a unique untangle sequence of length 1 ending at a matching excluding the upper segment r_1b_2 (Figure 5(b)).
3. The $1 \vee 0$ matching is the starting matching of a unique untangle sequence of length 1 ending at a matching excluding the upper segment r_1b_2 (Figure 5(c)).
4. The $1 \vee 1$ matching is already crossing free. It excludes the upper segment r_1b_2 (Figure 5(d)).

Proof. For each of the four $x \vee y$ matchings with $x, y \in \{0, 1\}$, we enumerate all the possible untangling sequences. These sequences are all shown in Figure 5. Lemma 2.3 then follows. \square

Clause Gadgets. A *clause gadget* consists of two OR gadgets, the output of the first one being “connected” to the left input of the second one (Figure 6). More precisely, a clause gadget is built on seven red points, say $r_4, r_5, r_6, r_7, r_8, r_{10}, r_{11}$, and six blue points, say $b_4, b_5, b_6, b_7, b_8, b_9$ such that the following maps correspond to two OR gadgets (using the OR gadget previous notations), and such that r_8 lie in the inside of the top triangle of the first OR gadget and is the only overlap between the two OR gadgets.

First OR gadget: $(r_4, b_4, r_5, b_5, r_6, b_6, b_9, r_{10}) \mapsto (r_1, b_1, r_2, b_2, r_3, b_3, b'_1, r'_2)$.

Second OR gadget: $(r_4, b_6, r_7, b_7, r_8, b_8, b_5, r_{11}) \mapsto (r_1, b_1, r_2, b_2, r_3, b_3, b'_1, r'_2)$,

with the exception that the segment r_6b_5 may also play the role of r_1b_1 .

Similarly to an OR gadget, a clause gadget consists of 2^3 matchings, namely the $x \vee y \vee z$ matchings with $x, y, z \in \{0, 1\}$. We define the *left input*, *middle*, and *right input* of a clause gadget as the left input of the first OR gadget, the right input of the first OR gadget, and the right input of the second OR gadget. We define the *output* of a clause gadget as the output of the second OR gadget.

Note that the middle input segment, i.e., the vertical segment lying in between the two other vertical segments (r_5b_5 in Figure 6), need not be evenly placed between the left input segment and the right input segment. This feature is used to build clause gadgets with non-consecutive variables, such as the topmost clause gadget in Figure 3.

The idea is to have a $0 \vee 0 \vee 0$ clause gadget in M_Φ for each clause in Φ . As we will see next, in the beginning of an untangling sequence starting at M_Φ , each input may be set to 1 or may be kept as 0, changing the $0 \vee 0 \vee 0$ clause gadget into one of the $x \vee y \vee z$ matchings with $x, y, z \in \{0, 1\}$.

The following lemma states that the truth table of the logical gate associated with a clause gadget is indeed the expected one.

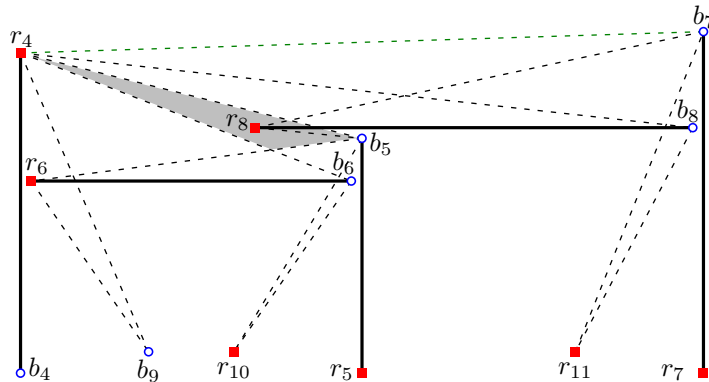


Figure 6: A clause gadget. The $0 \vee 0 \vee 0$ matching is drawn with plain segments.

Lemma 2.4. *The output of a clause gadget is always well defined, and is 0 if and only if the three inputs of the clause gadget are all 0. More precisely, we have the following.*

1. All the untangle sequences starting at the $0 \vee 0 \vee 0$ matching are of length 4, and they end at the same matching containing the upper segment $r_4 b_7$.
2. All the untangle sequences starting at each of the $x \vee y \vee z$ matchings, where exactly one of x, y , or z is 1, are of length 2, and they end at matchings excluding the upper segment $r_4 b_7$.
3. The unique untangle sequence starting at each of the $x \vee y \vee z$ matchings, where exactly two of x, y , and z are 1, is of length 1, and it ends at a matching excluding the upper segment $r_4 b_7$.
4. The $1 \vee 1 \vee 1$ matching is already crossing free, and it excludes the upper segment $r_4 b_7$.

Proof. It is a consequence of Lemmas 2 and of the fact that the OR gadgets are connected so as to not interfere. Indeed, by construction, r_8 lies in the inside of the top triangle of the first OR gadget, and is the only overlap between the two OR gadgets. This ensures that all untangle sequences never give rise to an extra crossing that does not already belong to one of the two OR gadgets. In Figure 6, we have drawn with dashed line segments all the possible created segments during any possible untangle sequence. \square

Padding Gadgets. Let k be a non-negative integer. A k -padding gadget triggered by the segment s consists of two matchings built by induction as follows.

The first matching, denoted M_k , contains s ($s = r_4 b_7$ in Figure 7) and is called the *triggered matching* of the padding gadget (s creates a crossing). The second matching is called the *non-triggered matching*, and is deduced from the triggered one by removing s (it is crossing free).

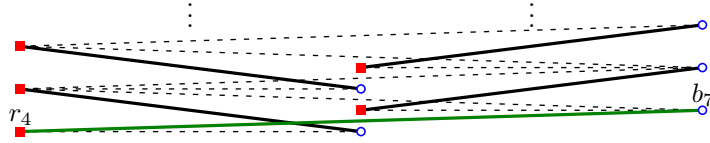


Figure 7: The triggered matching of a padding gadget.

If $k = 0$, then the triggered matching of a k -padding gadget consists of only the segment s . If $k \geq 1$, then the triggered matching of a k -padding gadget consists of M_{k-1} , the triggered matching of a $(k - 1)$ -padding gadget, to which we add one new segment crossing only the last created segment of the only untangle sequence starting at M_{k-1} (Figure 7, the dashed segments are all the possible created segments in the unique untangle sequence).

Lemma 2.5. *Let k be a non-negative integer. There is a unique untangle sequence starting at the triggered matching of a padding gadget, and it is of length k . The non-triggered matching of a padding gadget is already crossing free.*

Proof. The definition of a k -padding gadget yields Lemma 2.5. \square

We complete each clause gadget with a padding gadget in order to penalize a non-satisfied clause by an arbitrary long untangle sequence (Figure 8). Notice that a padded clause gadget can be arbitrarily scaled and that the position of a clause rectangle is only constrained by the planar embedding of Φ .

Matching Computation. We now describe, given a planar embedding of Φ , the construction steps of the matching M_Φ . Without loss of generality, we only specify the construction of the positive clauses, the construction of the negative clauses being similar.

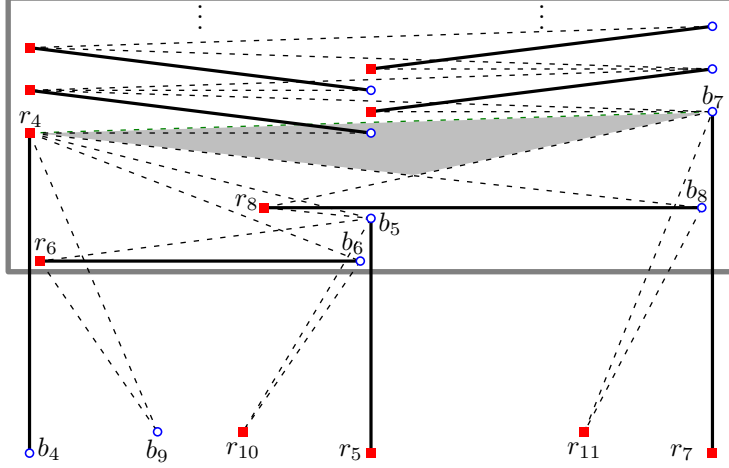


Figure 8: A clause gadget connected to a padding gadget.

We need the following definitions for the description. The *top vertical points* of a positive clause gadget are the topmost endpoints of the vertical segments (e.g. $b_{14}, r_4, r_{12}, b_5, b_7$ in Figure 9). Similarly, the *bottom vertical points* are the bottom endpoints of the same vertical segments (e.g. $r_{14}, b_4, b_{12}, r_5, r_7$ in Figure 9). Let \bar{p} be a top vertical point. The *horizontal segment of \bar{p}* is the horizontal segment lying below \bar{p} which is the closest to \bar{p} . Finally, we define the *substitute point of \bar{p}* as the endpoint of the horizontal segment of \bar{p} which is the closest to \bar{p} (e.g. r_6 is the substitute point of r_4 in Figure 9).

The construction steps of the matching M_Φ are the following.

1. Place a clause gadget connected to a k -padding gadget in each clause rectangle, and a variable gadget in each variable rectangle, with appropriate scaling.
2. Connect each clause gadget to its corresponding three variable gadgets with the three vertical segments of the clause gadget aligned with the corresponding vertical edges of the planar embedding of Φ .
3. Adjust the x -coordinates of the vertical segments of each variable gadget to have the top vertical points and the two topmost points of the variable gadget, all in convex position (e.g. in Figure 9, r_{12} is on the right of the segment r_4b_9).
4. Adjust the y -coordinates of the bottom vertical points in the top triangle of each variable gadget so as to place them and the two topmost points of the variable gadget in convex position.
5. Let \bar{p} be a top vertical point which is not the highest of a variable gadget (e.g. $\bar{p} = r_{12}$ in Figure 9). Let \underline{p} be the corresponding bottom vertical point (e.g. $\underline{p} = b_{12}$). Let \bar{q} be the top vertical point immediately above \bar{p} (e.g. $\bar{q} = r_4$). Let \underline{q} be the point immediately above \underline{p} , taken among the bottom vertical points together with the two topmost points of the variable gadget (e.g. $\underline{q} = b_9$). Adjust the x -coordinate of \bar{p} , the substitute point of \bar{p} (e.g. $\tilde{p} = r_{13}$), so that \tilde{p} lies in the triangle $\bar{p}\bar{q}\underline{q}$ (e.g. a shaded triangle in Figure 9; segment $r_{13}b_{13}$ must not cross r_4b_9 , but it has to cross $r_{12}b_9$).

We have the following lemma.

Lemma 2.6. *Let Φ be an instance of RPM 3-SAT with c clauses and v variables. Let k be a non-negative integer, polynomial in c and v . The matching M_Φ with k -padding gadgets is computed in polynomial time in c and v .*

Proof. The number of operations in any execution of these construction steps is linear in c and v . The coordinates of the points of M_Φ are rational numbers with $O(\log n)$ bits. \square

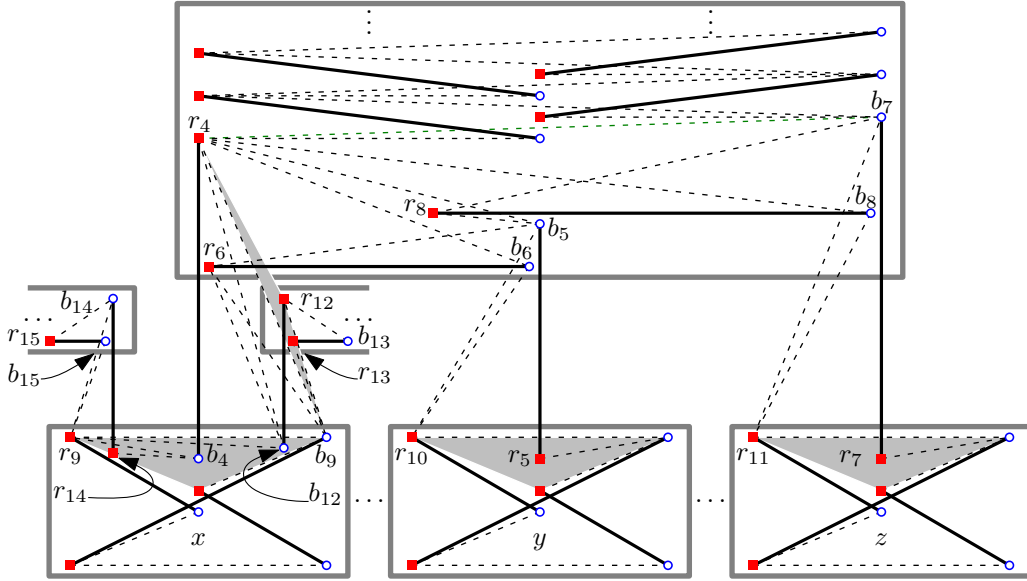


Figure 9: A padded clause gadget connected to x, y, z , with branching on x .

Branching. The following lemma ensures that the connection of multiple vertical segments to a same variable gadget always triggers all the corresponding clause gadgets. We start with some definitions.

The set consisting of the top segment of a variable gadget set to true, together with the vertical segments crossing it, and their horizontal segment is called a *branching matching* (such as drawn in Figure 10(b) with plain segments). The bottom vertical points of a branching matching, listed from left to right, always consist of a certain number, say a , of red points followed by a certain number, say b , of blue points. We say that such a branching matching has parameters a, b . These matchings have the following property.

Lemma 2.7. *All the untangle sequences starting at a branching matching with parameters a, b have length $2(a + b)$ and end at the same crossing-free matching (e.g. the segments $r_{15}b_{14}, r_9b_{15}, r_{14}b_4, r_4b_6, r_6b_{12}, r_{12}b_{13}, r_{13}b_9$ in Figure 10(b)).*

Proof. First note that a simplified version of this result has been proven in [7]. This simplified version amounts to forget all the horizontal segments, except the top segment of the variable gadget (Figure 10(c)).

It is useful to start by proving this simplified version before Lemma 2.7. We do an induction on $a + b$. The base case is trivial, but it provides the possible positions of created segments in the untangle sequences (the dashed segments in Figure 10(c)). The inductive case relies on the fact that the points are in convex position. Indeed, after any flip, the two created segments play the role of the initial horizontal segment because convex position ensures that any of the non-flipped vertical segments will cross exactly one of the two created segments, and that no extra crossing is created. The induction hypothesis then applies on both right and left submatching whose convex hulls are now disjoint.

We now address the issue where each vertical segment is paired with its horizontal segment. Recall that at step 5 of the construction of M_Φ , we have adjusted the x -coordinate of \tilde{p} , the substitute point of each top vertical point \bar{p} , so that \tilde{p} lies in the triangle $\bar{p}\underline{q}\bar{q}$. This ensures that

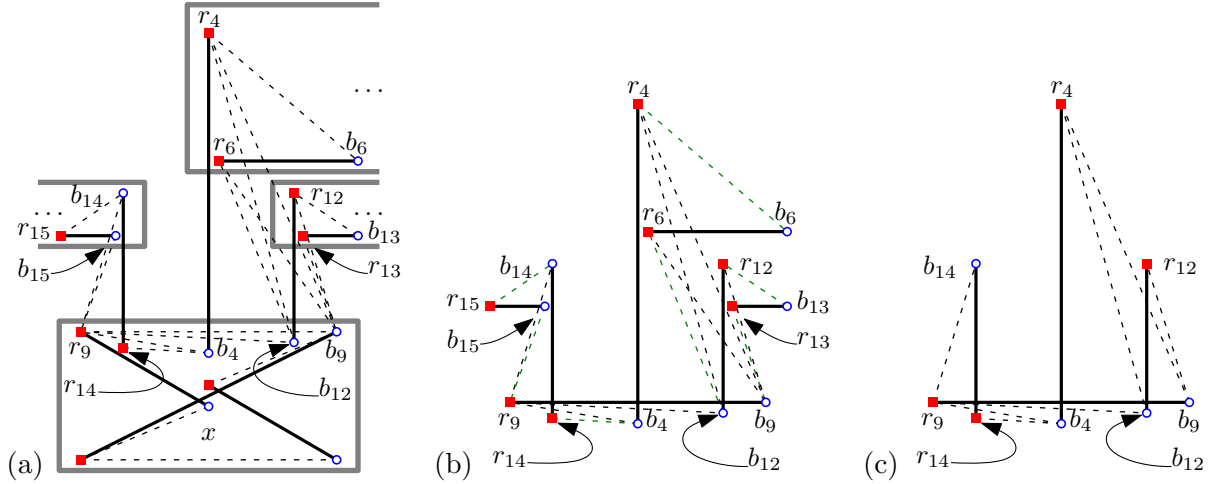


Figure 10: Three views of a branching matching.

each substitute point can play the role of its corresponding top vertical point from whenever the corresponding horizontal segment has been flipped in an untangle sequence. \square

Result. The RPM 3-SAT instance being encoded in the matching M_Φ , we have the property that the shortest untangling sequence of M_Φ is short if the instance Φ is satisfiable, and long otherwise.

Lemma 2.8. *We have the following case distinction.*

- Φ is satisfiable if and only if there exist untangle sequences starting at M_Φ which do not trigger any padding gadget, in which case $d(M_\Phi)$ is at most $v + 5c$.
- Φ is not satisfiable if and only if all untangle sequences starting at M_Φ trigger at least one padding gadget, in which case $d(M_\Phi)$ is at least $v + 7 + k$ where k is arbitrarily large.

Proof. It is consequence of Lemmas 2.2, 2.4, 2.5 and 2.7, as we examine the longest possible untangle sequences of M_Φ which do not trigger any padding gadget, and the shortest possible untangle sequences of M_Φ which trigger at least one padding gadget. In any case, v flips will be performed, one per variable (Lemma 2.2).

In the case where no padding gadget is triggered, the length of the longest possible untangle sequences starting at a clause gadget connected to three variable gadgets is 5, and is obtained by adding 3, the length of the untangle sequences of a $0 \vee 0 \vee 1$ matching, and 2, for the two connections to the negative variables. Counting 5 flips per clause yields $v + 5c$.

If at least one padding gadget is triggered, this very padding gadget generates k flips. In this case, the length of the shortest possible untangle sequences starting at a clause gadget connected to three variable gadgets and which is known to trigger its padding gadget is 7, and is obtained by adding 4, the length of the untangle sequences of a $0 \vee 0 \vee 0$ matching, and 3, for the three connections to the negative variables. All the other clause gadgets may be set to their $1 \vee 1 \vee 1$ matching, adding no flip to the shortest untangle sequence, the length of which is thus $v + 7 + k$. \square

We now prove Theorem 2.1, reducing RPM 3-SAT to Problem 1. Let Φ be an instance of RPM 3-SAT with c clauses and v variables. We build the matching M_Φ , which serves as an instance of Problem 1, choosing $k = \alpha(v + 5c) + 1$. As k is polynomial in the size of the input (α is a constant), the computation of the matching M_Φ is polynomial (Lemma 2.6).

By hypothesis, we compute an untangle sequence starting at M_Φ of length ℓ at most $\alpha d(M_\Phi)$. We decide that Φ is satisfiable if $\ell \leq \alpha(v + 5c)$, and that Φ is not satisfiable if $\ell > \alpha(v + 5c)$.

Indeed, Lemma 2.8 ensures the following. If Φ is satisfiable, then the length of the shortest untangle sequence of M_Φ is at most $v + 5c$. Otherwise the length of the shortest sequence is at least $v + 7 + k \geq k = \alpha(v + 5c) + 1$. This ends the reduction, and proves Theorem 2.1.

3 Upper Bound on $d(n)$

In this section, we prove the following upper bound.

Theorem 3.1. *In the red-on-a-line case, $d(n) \leq \binom{n}{2}$.*

The proof consists of the analysis of the number of flips performed by the recursive algorithm described next. This analysis is based on a novel approach called *state tracking*. State tracking is in fact not specific to the red-on-a-line case, which is why Lemma 3.4 is stated and proven in the non-bipartite setting. Lemma 3.4 is then used in the red-on-a-line case to prove Lemma 3.5, which in turn is used to prove Theorem 3.1. Lemma 3.4 also provides an alternative proof of the well-known Theorem 3.6 [5], which we present at the end of this section.

Throughout, we assume general position (no two blue points with the same y -coordinate). Let the *top* segment of a red-on-a-line matching be the segment with the topmost blue endpoint (Figure 11(a)).

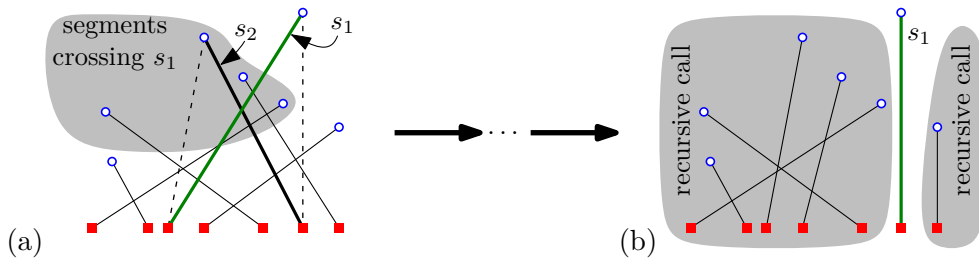


Figure 11: (a) A red-on-a-line matching with s_1 as the top segment. (b) The matching just before the first recursive calls of the algorithm, where s_1 is free.

Algorithm. While the top segment s_1 of the matching crosses another segment s_2 , we flip s_1 and s_2 . If multiple segments cross s_1 , then we choose s_2 as the top segment among the segments crossing s_1 .

The previous loop stops when the top segment s_1 has no crossings. At this point, we have that s_1 splits the matching into at most two non-empty submatchings, one to each side of s_1 . We recursively call the algorithm on these submatchings (Figure 11(b)).

Correctness. The next two lemmas prove the correctness of the algorithm.

Lemma 3.2 ([7]). *If a matching admits a partition of submatchings whose convex hulls are all disjoint, then, any sequence of flips in one of the submatchings never affects the other submatchings (Figure 12).*

Proof. This result can be found in [7]. Its proof amounts to the observation that the flip operation leaves the convex hull unchanged (in Figure 12, the dashed segments are the results of possible flip sequences). \square

We say that a segment s is *free* if the matching admits a partition of submatchings whose convex hulls are all disjoint, and one of the submatchings consists of the segment s alone. In Figure 12, the segment s is the only free segment.

Lemma 3.3. *The algorithm always makes the top segment free before recursive calls.*

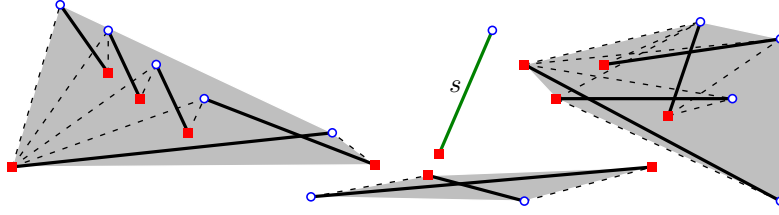


Figure 12: A partition of 4 submatchings whose convex hulls are all disjoint. The segment s is the only free segment.

Proof. The algorithm repeats the flip step until the top segment is free. As any sequence of flips is finite, this eventually happens. The recursive calls of the algorithm happen when and only when the top segment is free. \square

The correctness of the algorithm follows from Lemma 3.2 and 3.3.

Flip Complexity. The analysis of the number of flips performed by the algorithm stems from the following observations. We define three possible *states* for a pair of segments (Figure 13).

- State **X**: the segments are crossing.
- State **H**: the segments are not crossing and their endpoints are in convex position.
- State **T**: the endpoints are not in convex position.

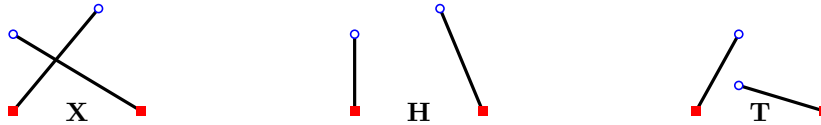


Figure 13: The three different states of pairs of segments.

In the convex case, there are no **T**-states and a flip increases the number of **H**-pairs by at least 1 unit, and decreases the number of **X**-pairs as well. Hence, counting either **X** or **H**-pairs yields the $\binom{n}{2}$ upper bound on $\mathbf{D}(n)$ (this upper bound is in [5] and the alternative proof we mentioned is made precise in Theorem 3.6). However, when the points are not in convex position, counting **H** and **X**-pairs is fundamentally different. We will see that counting **H**-pairs is more useful to prove the desired bounds.

When the points are not in convex position, a flip may decrease the number of **H**-pairs. Figure 14 shows two such situations where flipping s_1, s_2 does not increase the number of **H**-pairs. There is one **H**-pair involving segment s before the flip, and none after the flip. Notice that, if we added multiple segments close to s , the number of **H**-pairs would actually decrease. However, the algorithm avoids these situations by choosing to flip top segments. The full proof involves state tracking, a novel approach to analyze flip sequences, which is described next.

State Tracking. We have $\binom{n}{2}$ pairs of segments before and after a flip. Each pair has an associated state. However, since two segments change in the matchings, there is no clear correspondence between the state of each pair before and after the flip. State tracking establishes this correspondence by making choices of which pair of segments in the initial matching corresponds to which pair of segments in the resulting matching. These choices are performed deliberately to obtain certain state transitions instead of others and prove the desired bounds.

The following notations will be used throughout the rest of this section and are summarized in Figure 15. Let r_1, r_2 be two red points and b_1, b_2 be two blue points. Let s_1, s_2, s'_1, s'_2 be the

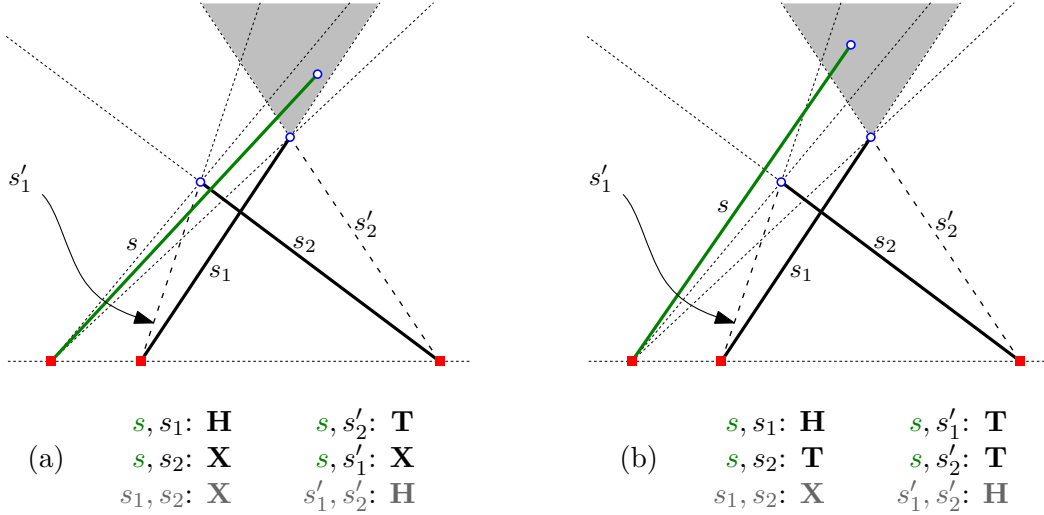


Figure 14: Two cases where flipping s_1, s_2 does not increase the number of \mathbf{H} -pairs. The upper cone of s_1, s'_2 is shaded.

following four segments respectively: $r_1b_1, r_2b_2, r_1b_2, r_2b_1$. We consider a flip that replaces the pair of segments s_1, s_2 by s'_1, s'_2 . Let M denote the matching before the flip and M' denote the resulting matching after the flip.

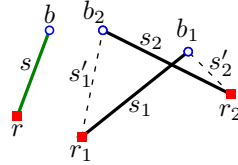


Figure 15: Notations for a generic flip and for a variable segment s .

We order the $\binom{n}{2}$ pairs of segments of M in a column vector. There are three types of pairs of segments in M with respect to the flip: the *unaffected* pairs (involving neither s_1 nor s_2), the *flipping* pair s_1, s_2 , and the *affected* pairs (involving exactly one of s_1 or s_2). We choose the new order of the $\binom{n}{2}$ pairs of segments of M' in a way that satisfies the following properties with respect to the previous vector. The unaffected pairs keep the same indices. The pair s'_1, s'_2 gets the index of s_1, s_2 . Next, we describe the remaining indices.

Let s be a segment of M distinct from s_1 and s_2 . Let r and b be the red and blue endpoints of s . Let i_1 and i_2 be the indices of s, s_1 and s, s_2 , and let \mathbf{S}_1 and \mathbf{S}_2 be their respective states. Let \mathbf{S}'_1 and \mathbf{S}'_2 be the respective states of s, s'_1 and s, s'_2 . We restrict our choice to the following two options:

- index s, s'_1 with i_1 , and s, s'_2 with i_2 , or
- index s, s'_1 with i_2 , and s, s'_2 with i_1 .

We call such a choice a *tracking choice*. We say that a pair of segments in M *turns into* a pair in M' when they have the same index. We denote $\mathbf{S} \rightarrow \mathbf{S}'$ to specify that the pairs of segments with a given index go from the state \mathbf{S} to the state \mathbf{S}' . In the following, we use $\mathbf{S}_1\mathbf{S}_2 \rightarrow \mathbf{S}'_1\mathbf{S}'_2$ as a shorthand notation to say that we have the two following tracking choices: either $\mathbf{S}_1 \rightarrow \mathbf{S}'_1$ and $\mathbf{S}_2 \rightarrow \mathbf{S}'_2$ or $\mathbf{S}_1 \rightarrow \mathbf{S}'_2$ and $\mathbf{S}_2 \rightarrow \mathbf{S}'_1$.

There are 3^2 possible such *transitions* $\mathbf{S} \rightarrow \mathbf{S}'$. Yet, the next two lemmas ensure that some transitions can be ruled out by tracking choices. Lemma 3.4 actually holds for any (possibly non-bipartite) matching, while Lemma 3.5 is specific to the red-on-a-line case. Both lemmas are

proved analyzing the tracking choices of each possible position of a segment s relative to the flipping pair.

Lemma 3.4. *There always exists a tracking choice avoiding the $\mathbf{H} \rightarrow \mathbf{X}$ transition.*

Proof. There clearly exists a tracking choice avoiding the $\mathbf{H} \rightarrow \mathbf{X}$ transition unless we have either a transition (i) $\mathbf{HH} \rightarrow \mathbf{XS}$ or (ii) $\mathbf{HS} \rightarrow \mathbf{XX}$, where $\mathbf{S} \in \{\mathbf{X}, \mathbf{H}, \mathbf{T}\}$. We show that these two cases are not possible.

(i) $\mathbf{HH} \rightarrow \mathbf{XS}$: If both the pairs s, s_1 and s, s_2 are \mathbf{H} while at least one of the two pairs s, s'_1 and s, s'_2 is \mathbf{X} , then the final \mathbf{X} state implies that s crosses s_1 or s_2 , which contradicts the two initial \mathbf{H} states.

(ii) $\mathbf{HS} \rightarrow \mathbf{XX}$: If one of the two pairs s, s_1 and s, s_2 is \mathbf{H} while both pairs s, s'_1 and s, s'_2 are \mathbf{X} , then the two final \mathbf{X} states imply that s crosses s'_1 and s'_2 . It follows that s also crosses s_1 and s_2 , which is again a contradiction. \square

State Tracking in the Red-on-a-Line Case. Figure 16 summarizes the notations for a generic red-on-a-line flip and an variable segment s . Figures 17, 18, 19, and 20 then provide “maps” of essentially all the possible situations of tracking choices in the red-on-a-line case. These figures are used to prove the next lemma.

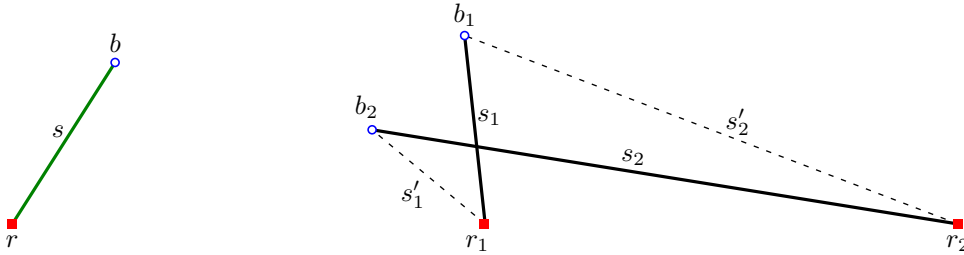


Figure 16: Notations used in Figures 17, 18, 19, and 20 for a generic red-on-a-line flip and an variable segment s .

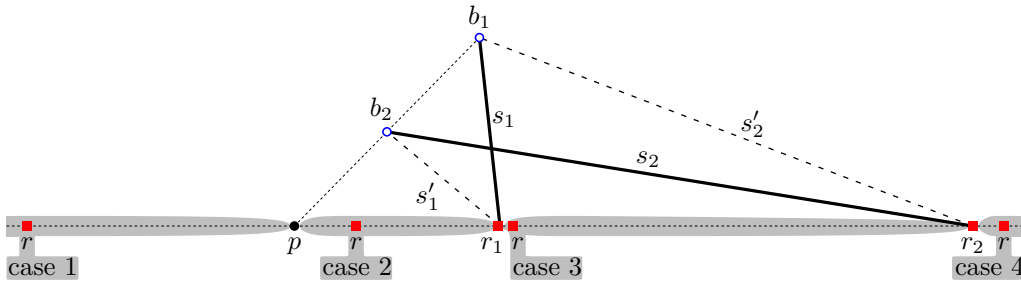


Figure 17: The four possible cases for the position of r .

Figures 18, 19, and 20 are generated by a brute force computation of the states $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}'_1, \mathbf{S}'_2$ of the four pairs $s, s_1, s, s_2, s, s'_1, s, s'_2$ for each position case for r (Figure 17) and for each position case for b (in Figures 18, 19, and 20, each cell of the arrangement of lines corresponds to a position case for b). In the following, we make sure that no case is forgotten.

We assume, without loss of generality, that r_1 is on the left of r_2 , and that b_1 is higher than b_2 . Let p be the intersection between the line b_1b_2 and the red-point line. There are, indeed, four possible open intervals for the position of r on the red-point line: $]-\infty, p[$, $]p, r_1[$, $]r_1, r_2[$, and $]r_2, \infty[$ (Figure 17). This yields four cases, respectively. We do not explicitly describe case 4 as it is similar to case 2. Indeed, case 2 and case 4 map to each other by exchanging the labels of r_1 and r_2 , as well as b_1 and b_2 . The fact that the point p is still on the left of r_1 and r_2 is not

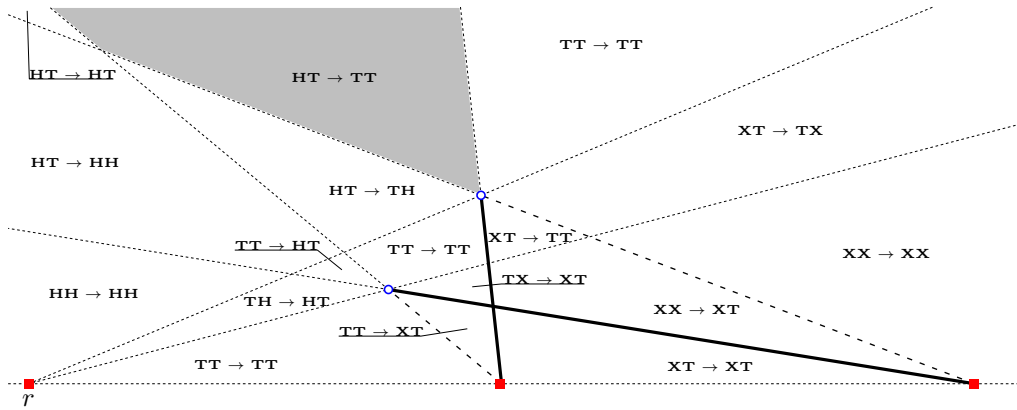


Figure 18: The case 1 “map” of all the possible red-on-a-line tracking choices. Tracking choices cannot avoid the transition $\mathbf{H} \rightarrow \mathbf{T}$ in the shaded region.

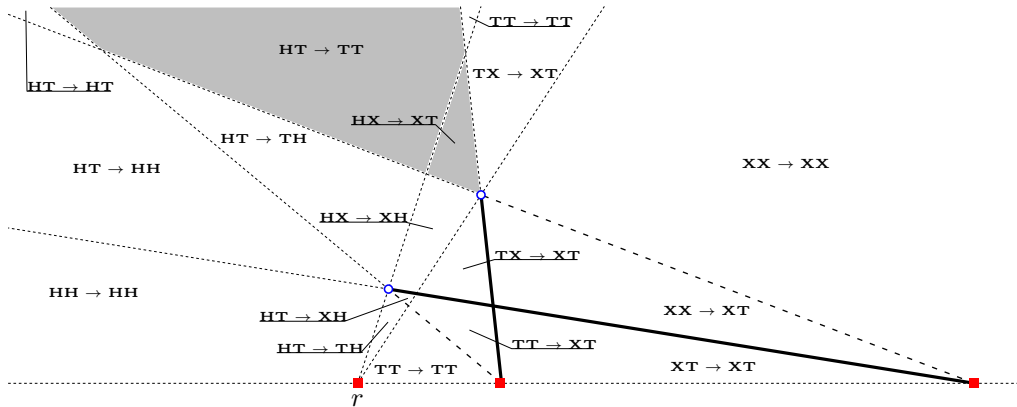


Figure 19: The case 2 “map” of all the possible red-on-a-line tracking choices. Tracking choices cannot avoid the transition $\mathbf{H} \rightarrow \mathbf{T}$ in the two shaded regions.

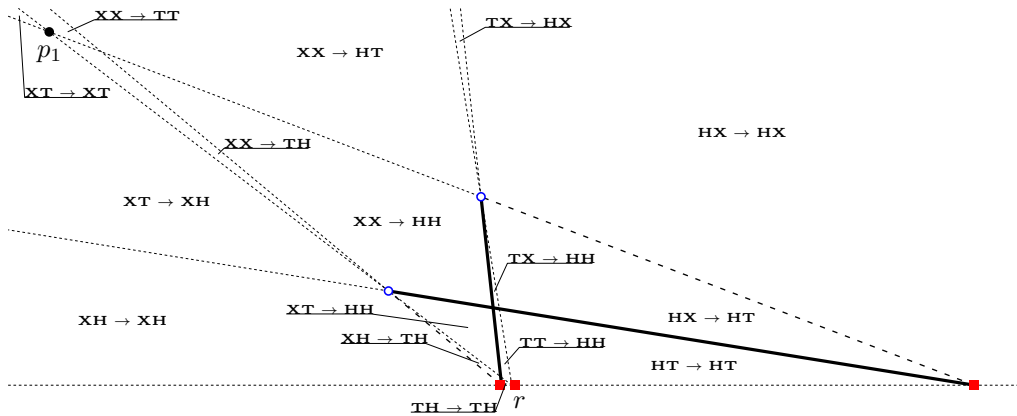


Figure 20: The case 3.1 “map” of all the possible red-on-a-line tracking choices.

a problem since we are studying incidence proprieties. Another way to see it, is to consider the projective plane.

As we have assumed the blue points to lie in the upper half-plane, these four cases split further into sub-cases. However, no-loss-of-generality assumptions and symmetries simplify the analysis. Without loss of generality, we first assume that the lines r_1b_2 and r_2b_1 intersect in the upper half-plane, as it will only generate more cells to the upper part of the arrangement of lines.

Second, we examine case 3. Let p_1 be the intersection of the lines rb_2 (see Figure 20) and r_2b_1 , and p_2 be the intersection of the lines rb_1 and r_1b_2 . Case 3 decomposes into:

- case 3.1 where p_1 lies in the upper half-plane and p_2 in the lower,
- case 3.2 where both p_1 and p_2 lie in the upper half-plane,
- case 3.3 where p_1 lies in the lower half-plane and p_2 in the upper, and
- case 3.4 where both p_1 and p_2 lie in the lower half-plane.

Cases 3.1 and 3.3 are similar, while case 3.2 is just a superposition of both of them. More precisely, when compared to case 3.4, the extra cell of the arrangement generated by case 3.1 (the cell in the top left corner of Figure 20) corresponds to the possible tracking choices summarized by the notation $\mathbf{XT} \rightarrow \mathbf{XT}$. Similarly, the extra cell generated by case 3.3 corresponds to $\mathbf{TX} \rightarrow \mathbf{TX}$. The two extra cells generated by case 3.2 are the same as the two previous ones. We thus assume case 3.1 (as it is easier to draw in our setting) without loss of generality. All these assumptions made, the remaining cases now corresponds to Figures 18, 19, and 20.

The next lemma is similar to Lemma 3.4, but specific to red-on-a-line matchings. We will use it to additionally avoid the $\mathbf{H} \rightarrow \mathbf{T}$ transition. To state Lemma 3.5, we define the *upper cone* of two segments r_3b_3, r_4b_3 as the locus of the points that are separated from the horizontal line r_3r_4 by the two lines r_3b_3 and r_4b_3 (Figure 21(a)). We also define the *upper ray* of a segment as the open ray with the blue point as its origin, the segment as its direction, and going upwards (Figure 21(b)).



Figure 21: (a) The upper cone of r_3b_3 and r_4b_3 is shaded. (b) The upper ray of the segment is dotted.

Lemma 3.5. *In the red-on-a-line case, if the blue point b of s is not in any of the two upper cones of s_1, s'_2 and s_2, s'_1 , then there always exists a tracking choice that avoids $\mathbf{H} \rightarrow \mathbf{T}$ for the pairs s, s_1 and s, s_2 while still avoiding $\mathbf{H} \rightarrow \mathbf{X}$.*

Proof. First, we check that there are only two possible upper cones defined by two segments of s_1, s_2, s'_1, s'_2 . Indeed, only two pairs among them have a common blue point.

Then, we note that, for s, s_1 or s, s_2 to be in state \mathbf{H} , the red point r of s cannot be between r_1 and r_2 , the red points of s_1 and s_2 . Without loss of generality, we assume r to lie on the left side of r_1 and r_2 .

For s, s'_1 or s, s'_2 to be in state \mathbf{T} , s has to cross at least one of the upper rays of s'_1 or s'_2 .

The only two combinations of states for $\{s, s_1, s, s_2\}$ and $\{s, s'_1, s, s'_2\}$ which do not leave us the choice to avoid the $\mathbf{H} \rightarrow \mathbf{T}$ transition are $\{\mathbf{H}, \mathbf{T}\}$ and $\{\mathbf{T}, \mathbf{T}\}$, and $\{\mathbf{H}, \mathbf{X}\}$ and $\{\mathbf{T}, \mathbf{X}\}$. In any case, b must be in the right most of the two upper cones of segments s_1, s_2, s'_1, s'_2 . More precisely, b lies in one of the three shaded regions of Figures 18 and 19. These three shaded regions also correspond to Figure 14 where case 1 is omitted but similar. The other cases are either not feasible geometrically, or with a possibility to make tracking choices so as to avoid transition $\mathbf{H} \rightarrow \mathbf{T}$. \square

Proof of Theorem 3.1. We are now ready to prove Theorem 3.1.

Proof. Let $f(M)$ be the total number of flips performed by the algorithm on an n -segment input matching M and let $g(M)$ be the number of flips performed by the algorithm before the recursive calls. Let M_r denote the matching before the recursive calls. The recursive calls take two submatchings of M_r that we call M_1 and M_2 , yielding the following recurrence relation.

$$f(M) = f(M_1) + f(M_2) + g(M)$$

Let $\bar{h}(M)$ be the number of \mathbf{X} -pairs plus the number of \mathbf{T} -pairs in a matching M , that is, the number of pairs that are not \mathbf{H} -pairs. Lemma 3.5 ensures that

$$g(M) \leq \bar{h}(M) - \bar{h}(M_r) \leq \bar{h}(M) - \bar{h}(M_1) - \bar{h}(M_2).$$

Clearly, $f(\emptyset) = 0$. We suppose that, for all M' with less than n segments, we have $f(M') \leq \bar{h}(M')$. Then by induction we get

$$f(M) \leq \bar{h}(M_1) + \bar{h}(M_2) + \bar{h}(M) - \bar{h}(M_1) - \bar{h}(M_2) = \bar{h}(M).$$

Theorem 3.1 follows since $\bar{h}(M) \leq \binom{n}{2}$. \square

State Tracking in the Convex Case. State tracking also applies to the widely studied convex case, providing a more conceptual proof of the following theorem from [5]. Even though we will not use this well-known result, we may as well state it. This theorem actually holds for any straight-line non-bipartite perfect matching.

Theorem 3.6 (Theorem 5 of [5]). *In the convex case, any untangle sequence is of length at most $\binom{n}{2}$.*

Proof. In the convex case, the \mathbf{T} -state does not exist. Lemma 3.4 thus ensures that the number of \mathbf{H} -pairs increases of at least 1 unit at each flip. \square

4 Upper Bound on $\mathbf{D}(n)$

In this section we prove the following theorem.

Theorem 4.1. *In the red-on-a-line case, $\mathbf{D}(n) \leq \binom{n}{2} \frac{n+4}{6}$.*

To prove Theorem 4.1, we define a potential function Φ that maps a red-on-a-line matching to an integer from 0 to $\binom{n}{2} \frac{n+4}{3}$. Since Φ decreases by at least 2 units at each flip, the theorem follows. We first give the definitions needed to present Φ . Then, we prove four lemmas yielding Theorem 4.1.

Let M be a red-on-a-line matching. Let r_1, \dots, r_n be the red points, from left to right. Let ℓ be a line, parallel to the line of the red points and above all the points. For each k in $\{1, \dots, n\}$, we project the blue points onto ℓ , using r_k as a focal point. More precisely, each blue point b maps to a point $t_k(b)$, the intersection between the ray $r_k b$ and the line ℓ (Figure 22(a)). We also define the function t_k of a red-blue segment rb as the segment $t_k(rb) = r t_k(b)$ (Figure 22(b)).

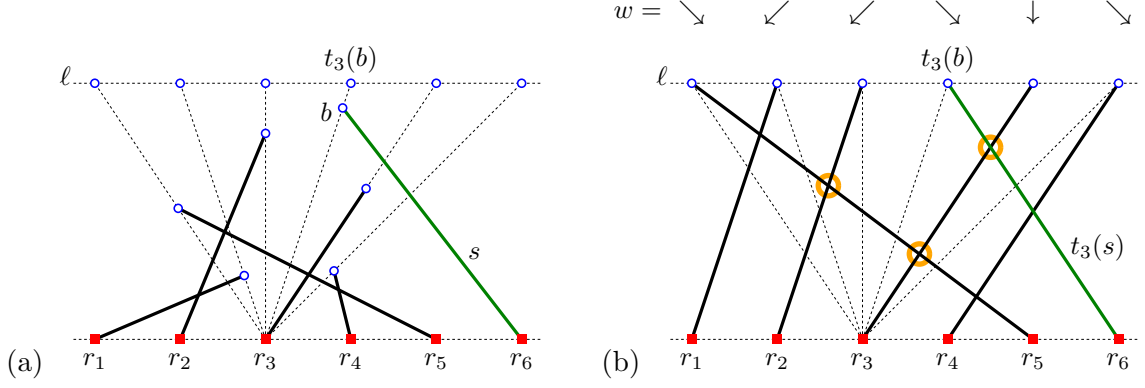


Figure 22: (a) The projection t_k for $k=3$. (b) The segments $t_3(\cdot)$. The three 3-observed crossing 3-pairs are circled.

We may abbreviate a pair of segments $r_i b, r_j b'$ as $\langle i, j \rangle$ when the points b and b' can be deduced from the underlying matching. Let k be an integer in $\{1, \dots, n\}$. We say that two segments are k -observed crossing if the extended projection $t_k(\cdot)$ maps them to crossing segments (Figure 22(b)). A pair of segments $\langle i, j \rangle$ is a k -pair if $i \leq k \leq j$. A k -flip is then a flip of a k -pair. We have the following lemma.

Lemma 4.2. *A crossing k -pair is necessarily k -observed crossing.*

Proof. Let $r_i b, r_j b'$ be a crossing k -pair. We suppose, without loss of generality, that $i < j$ (e.g. $i=2, k=3$, and $j=5$ in Figure 22).

The fact that the k -pair $r_i b, r_j b'$ is crossing means that the four points are in convex position, and that they appear as r_i, r_j, b, b' on their convex hull in counter-clockwise order. Since $i \leq k \leq j$, the point r_k is also on the boundary of the convex hull of the four points. Therefore, the projection $t_k(\cdot)$ will not change the convex-hull order and the segments $r_i t_k(b)$ and $r_j t_k(b')$ will cross. \square

We define $\Phi_k(M)$, the k -th potential of M , as the number of k -observed crossing k -pairs (Figure 22(b)). Lemma 4.3 shows that the k -th potential Φ_k is at most $(k-1)(n-k) + n - 1$. Lemma 4.4 shows that Φ_k never increases, and decreases by at least 1 unit at each k -flip.

Lemma 4.3. *The k -th potential Φ_k takes integer values from 0 to $k(n+1) - k^2 - 1$.*

Proof. The k -th potential $\Phi_k(M)$ is at most the number of k -pairs in M , crossing or not. There are exactly $(k-1)(n-k)$ k -pairs of the form $\langle i, j \rangle$ with $i < k < j$. There are exactly $k-1$ k -pairs of the form $\langle i, k \rangle$ with $i < k$. There are exactly $n-k$ k -pairs of the form $\langle k, j \rangle$ with $k < j$. In total, there are $k(n+1) - k^2 - 1$ k -pairs in M . \square

Lemma 4.4. *The k -th potential Φ_k never increases, and decreases by at least 1 unit at each k -flip.*

Proof. We order the projected blue points on ℓ from left to right. We then map each projected blue point $t_k(b)$ to an element in $\{\swarrow, \downarrow, \searrow\}$:

- $t_k(b)$ is mapped to \swarrow if b is matched to a red point on the left of r_k ,
- $t_k(b)$ is mapped to \downarrow if b is matched to r_k ,
- $t_k(b)$ is mapped to \searrow if b is matched to a red point on the right of r_k .

Let $w = w_1 \dots w_n$ be the word on the alphabet $\{\swarrow, \downarrow, \searrow\}$ induced by the order of the projected blue points and the map. For instance, in Figure 22 with $k = 3$, $w = \searrow\swarrow\swarrow\swarrow\downarrow\searrow$.

Let the total order of the symbols be $\swarrow \prec \downarrow \prec \searrow$. An *inversion* in w is a pair w_i, w_j with $i < j$ and $w_j \prec w_i$. The inversions in w are in bijection with the k -observed crossing k -pairs in M . Thus, by definition, $\Phi_k(M)$ is the number of inversions in w . Lemma 4.4 follows from the following two observations.

(i) Any flip which is not a k -flip swaps two \swarrow or two \searrow in w , resulting in word w' identical to w .

(ii) Lemma 4.2 ensures that a crossing k -pair corresponds to an inversion in w . Thus, a k -flip exchanges the two symbols of an inversion in w , resulting in word w' with at least one inversion less than in w . \square

We now define $\Phi(M)$, the *potential* of M , as the sum of $\Phi_k(M)$, for k in $\{1, \dots, n\}$. The following lemma presents the key properties of Φ .

Lemma 4.5. *The potential Φ takes integer values from 0 to $\binom{n}{2} \frac{n+4}{3}$, and decreases by at least 2 units at each flip.*

Proof. We know that Φ takes non-negative integer values by definition and, by Lemma 4.3, an upper bound on Φ is

$$\begin{aligned} \sum_{k=1}^n (k(n+1) - k^2 - 1) &= (n+1) \sum_{k=1}^n k - \sum_{k=1}^n k^2 - n \\ &= (n+1) \frac{n(n+1)}{2} - \frac{n(n+1)(2n+1)}{6} - n \\ &= \frac{n}{6} (n^2 + 3n - 4) \\ &= \binom{n}{2} \frac{n+4}{3}. \end{aligned}$$

Finally, Lemma 4.4 ensures that Φ decreases by at least 2 units at each flip. Indeed, a flip of a pair $\langle i, j \rangle$ is counted at least twice: once in Φ_i as an i -flip, and once in Φ_j as a j -flip. \square

Theorem 4.1 follows from Lemma 4.5.

5 Lower Bounds

In this section, we prove the following two lower bounds.

Theorem 5.1. *In the red-on-a-line case, for even n , $\mathbf{D}(n) \geq \frac{3}{2} \binom{n}{2} - \frac{n}{4}$.*

Theorem 5.2. *In the convex case, for even n , $\mathbf{d}(n) \geq \frac{3n}{2} - 2$.*

To prove Theorem 5.1, it suffices to present a long untangle sequence. The initial matching of the sequence is represented in Figure 23(a). To prove Theorem 5.2, we need to show that every untangle sequence starting at a given configuration (represented in Figure 23(b)) is long enough. We do so by showing that every flip reduces the number of crossings by exactly one unit.

5.1 Lower Bound on $D(n)$

We provide a $2m$ -segment red-on-a-line matching which we call an *m -butterfly*. There exists an untangle sequence starting at an m -butterfly of length $\frac{3}{2} \binom{2m}{2} - \frac{m}{2}$. Next, we give the precise definition of an m -butterfly and some of its important properties. Then, we give some intuition of how to come up with an untangle sequence longer than the number of pairs of segments. Finally, we prove that there exists an untangle sequence starting at an m -butterfly of length $\frac{3}{2} \binom{2m}{2} - \frac{m}{2}$ with two lemmas.

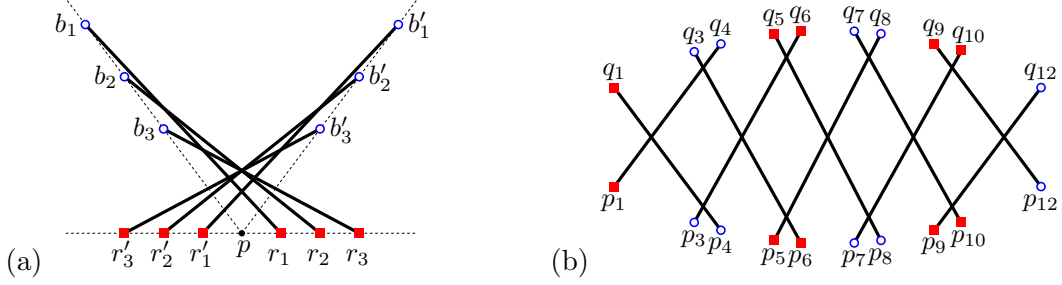


Figure 23: (a) A 3-butterfly to lower bound $\mathbf{D}(6)$. (b) A 5-fence to lower bound $\mathbf{d}(10)$.

Butterfly. For an integer m , we define an m -butterfly as the following matching with $n = 2m$ segments. For i from 1 to m we have red points $r_i = (i/(m+1), 0)$ and $r'_i = (-i/(m+1), 0)$ as well as blue points $b_i = (i - (m+1), (m+1) - i)$ and $b'_i = ((m+1) - i, (m+1) - i)$. We match r_i to b_i and r'_i to b'_i . Next, we discuss important properties of an m -butterfly.

We call a red-on-a-line *convex* matching an n -star if all the $\binom{n}{2}$ pairs of segments cross. We say that an n -star *looks* at a point p if the blue points are all on a common line, and if p is the intersection of this line with the line of the red points. We also say that two red-blue point sets R, B and R', B' are *fully crossing* if all the pairs of segments of the form $\{rb, r'b'\}$ cross, where $(r, b, r', b') \in R \times B \times R' \times B'$. Two matchings are fully crossing if their underlying red-blue point sets are fully crossing. An m -butterfly is a red-on-a-line matching consisting of two fully crossing m -stars both looking at the same point $p = (0, 0)$ (Figure 23(a) represents these properties but it is not drawn to scale).

Intuition. In the following, we use the state tracking framework from Section 3 to describe how to come up with an untangle sequence starting at an m -butterfly with more than $\binom{2m}{2}$ flips. We consider a sequence of tracking choices with no $\mathbf{H} \rightarrow \mathbf{X}$ transition (Lemma 3.4) for the long untangle sequence we build. We take advantage of the non-convex position of the blue points to create flip situations such as in Fig. 14(a), where an \mathbf{H} -pair is turned into a \mathbf{T} -pair.

For instance, let us consider an \mathbf{X} -pair of one of the m -stars composing the m -butterfly. At some point of the untangle sequence, we flip this \mathbf{X} -pair, turning it into an \mathbf{H} -pair. Later on, we turn this \mathbf{H} -pair into a \mathbf{T} -pair, as in Fig. 14(a). Still later on, we turn this \mathbf{T} -pair into an \mathbf{X} -pair again, similarly to the pairs involving the horizontal segment in Fig. 1. This \mathbf{X} -pair will be flipped again.

We manage to carry out this whole process to flip twice all the $2\binom{m}{2}$ pairs of the two m -stars composing the m -butterfly while still having one flip for every other pair. In total, we reach $\frac{3}{2}\binom{2m}{2} - \frac{m}{2}$ flips.

Proof of Theorem 5.1. We prove Theorem 5.1 with two lemmas, showing that there exists an untangle sequence of length $\frac{3}{2}\binom{2m}{2} - \frac{m}{2}$, starting at an m -butterfly.

Lemma 5.3 ([7]). *There exists an untangle sequence starting at any n -star of length $\binom{n}{2}$.*

Proof. This result has been shown in [7]. We present a short proof for the sake of completeness.

Provided we number from 1 to n the red points in their convex hull counter-clockwise order, and do the same for the blue points but clock-wise, then a red-on-a-line convex matching can be seen as a way to draw a permutation of n elements. An inversion, then, corresponds to a crossing. A bubble sort, thus, corresponds to an untangle sequence starting at such a matching.

The case of an n -star leads to $\binom{n}{2}$ inversion swaps, or, in other words, flips. \square

Lemma 5.4. *There exists an untangle sequence starting at any m -butterfly of length $\frac{3}{2}\binom{2m}{2} - \frac{m}{2}$ (Figure 24 and Figure 25).*

Proof. The untangle sequence can be divided into two phases.

The first phase consists of (i) $\binom{m}{2}$ flips applied to the m -star submatching defined by the m leftmost red points (see Lemma 5.3, and Figure 24, steps 0 to 3), and of (ii) $\binom{m}{2}$ more flips applied to the m rightmost red points (Figure 24, steps 3 to 6). At this point, we have two sets of m crossing-free segments, each set fully crossing the other.

The second phase repeats m times the following routine.

1. Flip the segments defined by the innermost red points r_1 and r'_1 (Figure 24 and 25, steps 6 to 7, 11 to 12, and 16 to 17). After this flip, the submatching defined by the m leftmost red points and their matched points consists of $m - 1$ crossing-free segments intersected by the segment from r'_1 . A similar statement holds for the submatching defined by the m rightmost red points.
2. Untangle the submatching defined by the m leftmost red points with $m - 1$ flips in the following manner (Figure 24 and 25, steps 7 to 9, 12 to 14, and 17 to 19). Flip of the two crossing segments with the rightmost red points, say r'_k and r'_{k+1} with $k \in \{1, \dots, m - 1\}$, and repeat. Such a flip produces a segment from r'_{k+1} crossing the segments whose red points are on the left of r_{k+1} , and another segment from r'_k crossing none of segments of the submatching. The number of crossings in the submatching decreases by 1 unit at each flip.
3. Similarly, untangle the m rightmost red points with $m - 1$ more flips (Figure 24 and 25, steps 9 to 11, 14 to 16, and 19 to 21).

Each loop decreases the number of “long” segments (i.e., segments joining one of the leftmost red points to one of the rightmost blue points, or vice-versa) by 2. At the end of the process, the left submatching is crossing-free; so is the right one; and the two of them do not intersect anymore.

Summing up, the total number of flips is $2\binom{m}{2} + 2m(m - 1) + m$. Simple calculation yields the lemma. \square

Theorem 5.1 follows from Lemma 5.4.

5.2 Lower Bound on $d(n)$

We provide a convex red-blue matching which we call an m -fence, with $2m$ segments and $3m - 2$ crossings (Figure 23(b)). Next, we give the precise definition of an m -fence, together with some useful terminology. Then, we prove Theorem 5.2 with three lemmas inferring that all untangle sequences starting at an m -fence have length $3m - 2$, that is, each flip reduces the number of crossings by exactly one unit.

Fence. Let $q_{2m+2}, q_{2m}, q_{2m-1}, \dots, q_4, q_3, q_1, p_1, p_3, p_4, \dots, p_{2m-1}, p_{2m}, p_{2m+2}$ be $4m$ points in convex position, ordered counter-clockwise, and with colors alternating every two points (Figure 23(b)). More precisely, points p_i, q_i are red if $i \equiv 1, 2 \pmod{4}$ and blue otherwise. We deliberately avoid using the indices 2 and $2m + 1$ to simplify the description. The segments of an m -fence are the $p_i q_{i+3}$ and the $q_i p_{i+3}$ where i is odd and varies between 1 and $2m - 1$.

For $1 \leq k \leq m + 1$, the k -th column consists of the at most 4 points with indices $2k - 1$ and $2k$. We say that a convex red-blue matching with the same point set as an m -fence is a *derived m -fence* if, for all $k \in \{2, \dots, m\}$, for all $w \in \{p, q\}$, one of the following statements holds:

1. w_{2k-1} is matched to a point of the $(k - 1)$ -th column, and w_{2k} is matched to a point of the $(k + 1)$ -th column, or
2. w_{2k-1} is matched to a point of the $(k + 1)$ -th column, and w_{2k} is matched to a point of the $(k - 1)$ -th column.

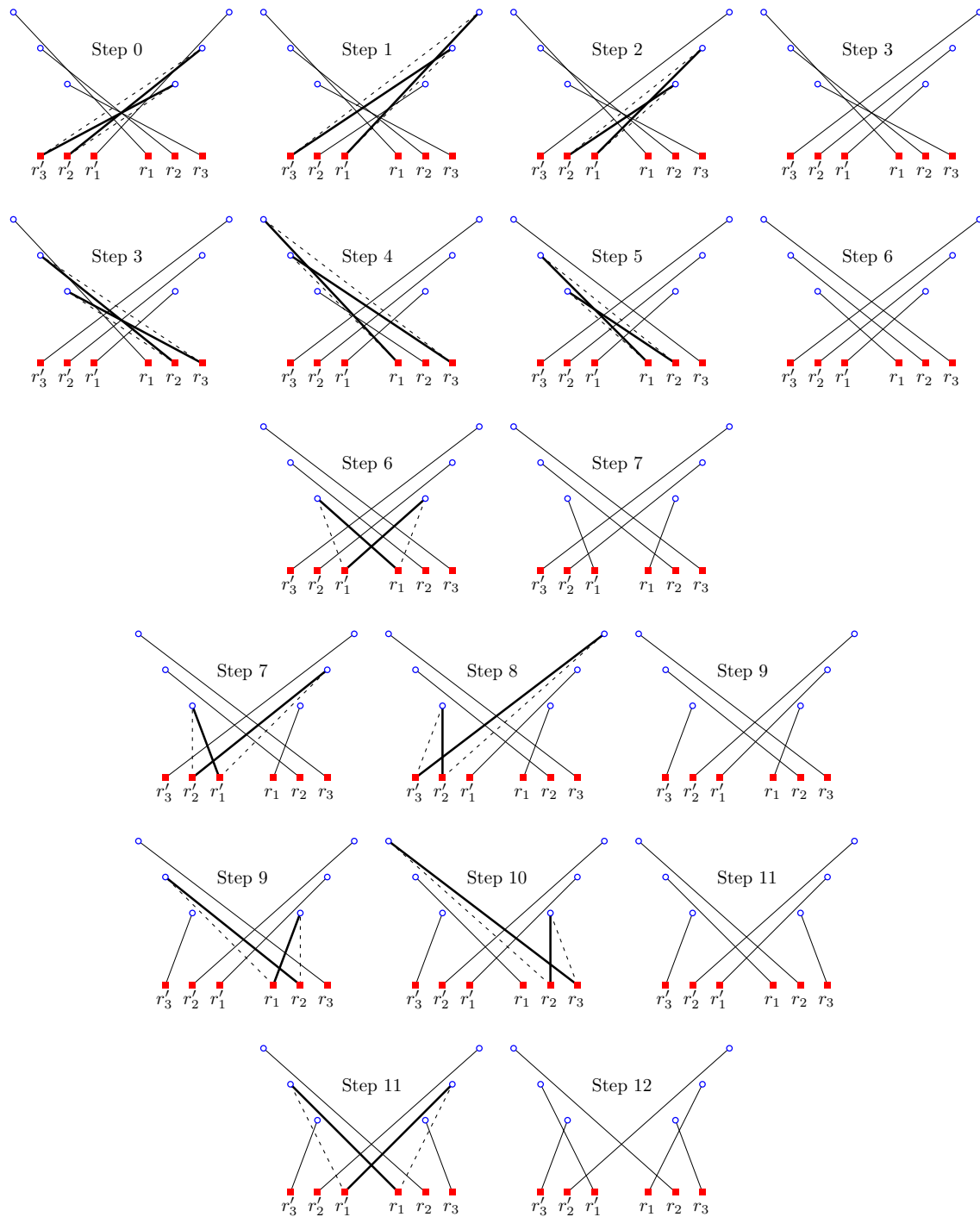


Figure 24: First part of an untangle sequence starting at a 3-butterfly of length 21 illustrating Lemma 5.4 and its proof. Each line corresponds to a portion of the proof, with repetitions added for clarity.

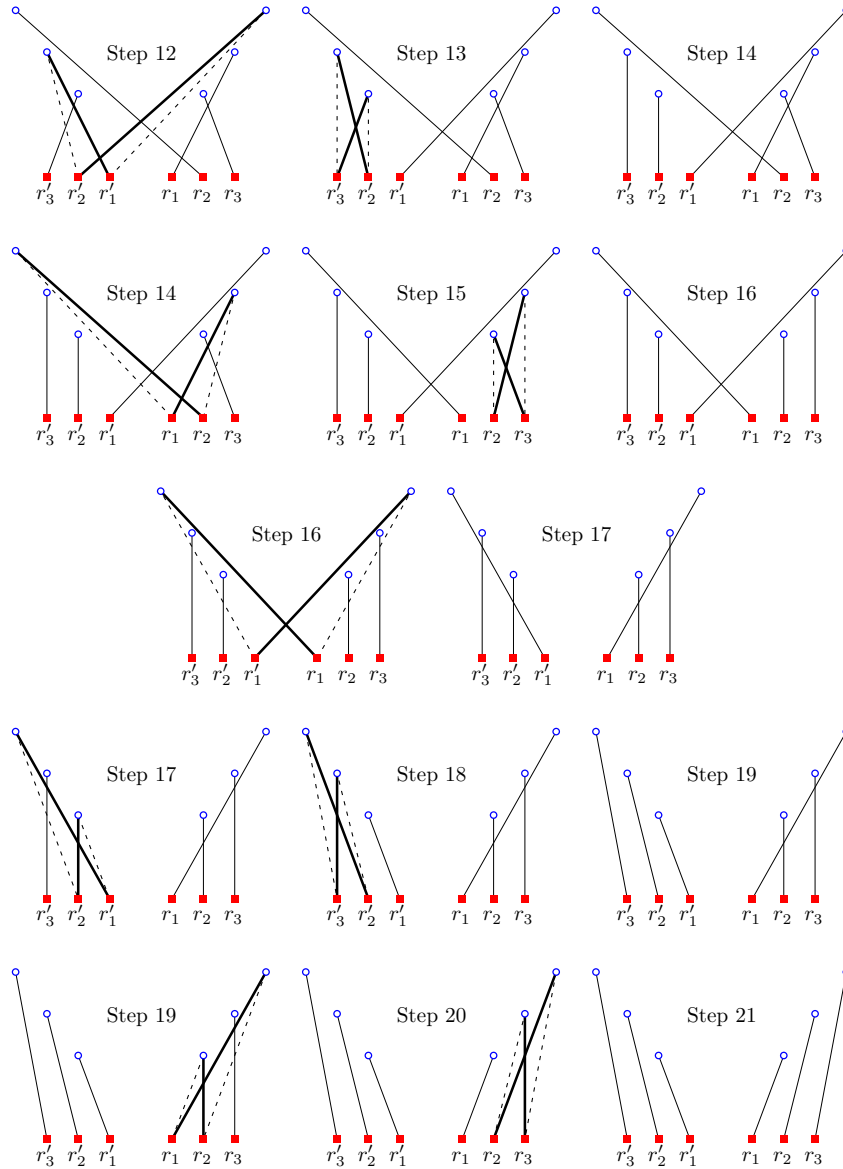


Figure 25: Second part of an untangle sequence starting at a 3-butterfly of length 21 illustrating Lemma 5.4 and its proof. Each line corresponds to a portion of the proof, with repetitions added for clarity.

Five examples of derived m -fences are presented in Figure 26. Note that an m -fence is in particular a derived m -fence.

When statement 2 holds, the two segments cross. We call such a crossing an *end* crossing. Similarly, a *middle* crossing is a crossing of the form $\{p_i q_j, q_{i'} p_{j'}\}$, where i and i' are of the same column, and j and j' are of the same column.

Proof of Theorem 5.2. To prove Theorem 5.2, we first show with two lemmas that a flip changes a derived m -fence into another derived m -fence. Finally, we show that a flip of a derived m -fence reduces its number of crossings by exactly one unit.

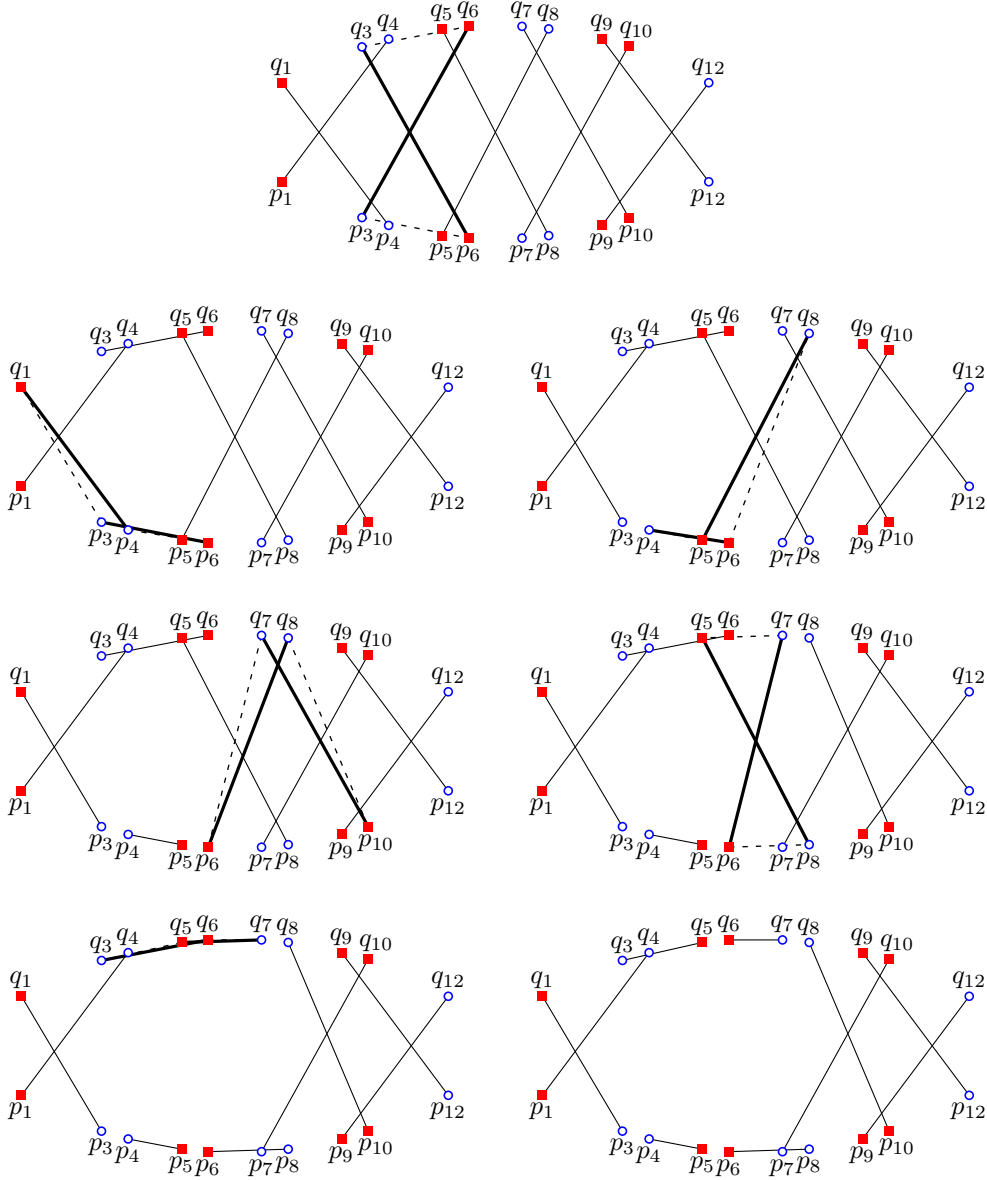


Figure 26: The beginning of an untangle sequence starting at a 5-fence. It is composed of derived 5-fences.

Lemma 5.5. *A crossing in a derived m -fence is either an end crossing or a middle crossing.*

Proof. The definition of a derived m -fence implies that a crossing must involve two or three consecutive columns. If exactly three columns are involved, the same definition excludes any crossing aside from the end crossings. If exactly two columns are involved, the definition again excludes any crossing aside from the middle crossings. \square

Lemma 5.6. *A flip changes a derived m -fence into another derived m -fence.*

Proof. Lemma 5.5 ensures that we only have the following two cases. (i) The flip of an end crossing on the w_0 side ($w_0 \in \{p, q\}$) of the k_0 -th column only changes statement 2 of the definition of a derived m -fence into statement 1 for $k, r = k_0, r_0$. The statements for the other k, r are unchanged. (ii) The flip of a middle crossing simply leaves unchanged the statements for all k, r .

Figure 26 is actually a sequence of flips starting at an m -fence and it contains essentially all the possible cases (symmetries aside). \square

Lemma 5.7. *A flip of a derived m -fence reduces its number of crossings by exactly one unit.*

Proof. Let M be a derived m -fence. Let s_1 and s_2 be two crossing segments of M . Let s be any other segment of M . Let s'_1 and s'_2 be the two segments replacing s_1 and s_2 after they have been flipped, changing M into M' . We show that the number of crossings between s and s_1, s_2 is the same as between s and s'_1, s'_2 , ensuring that M' has exactly 1 crossing less than M .

Let us recall that, as for any convex matching, the number of crossings cannot increase [5]. The proof of this result consists of the analysis of the five possible typical convex matchings (symmetries aside) of the three segments s_1, s_2, s (Figure 27). It is notable that only one of these five matchings, the one where each endpoint of s lies in between two endpoints of s_1, s_2 of the same color, corresponds to an actual decrease in the number of crossings involving s .

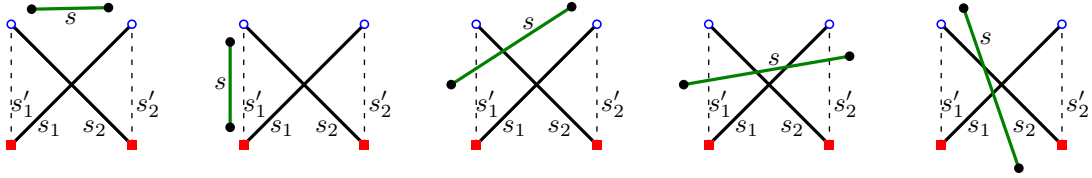


Figure 27: The five convex positions of s with respect to the flipping pair s_1, s_2 . This is used in the proof of Lemma 5.7.

This crossing-destructive case cannot occur if two endpoints of s_1, s_2 of the same color are adjacent on the convex hull. Thus, Lemma 5.7 holds for flips of end crossings.

If s_1, s_2 is a middle crossing, then, by definition of a derived m -fence, no segment s intersects both s_1 and s_2 . Thus, the crossing-destructive case cannot occur, and Lemma 5.7 holds for flips of middle crossings. \square

Theorem 5.2 follows from Lemma 5.6 and Lemma 5.7.

6 Concluding Remarks

Untangle sequences of TSP tours have been investigated since the 80s, when a cubic upper bound on $\mathbf{D}(n)$ has been discovered [26]. This bound also holds for matchings (even non-bipartite ones) and has not been improved ever since. Except for the convex case, there are big gaps between the lower and upper bounds, as can be seen in Table 1. Experiments on tours and matchings have shown that, in all cases tested, the cubic upper bound is not tight and the lower bounds seem to be asymptotically tight.

Untangle sequences have many unexpected properties which make the problem harder than it seems at first sight. The following questions remain open.

1. If we add a new segment to a crossing-free matching, what is the maximum length of an untangle sequence? Notice that an $o(n^2)$ bound would lead to an $o(n^3)$ bound for $\mathbf{d}(n)$.

2. Is it always possible to find an untangle sequence that does not flip the same pair of segments twice? Using a balancing argument, we can show that the number of *distinct* flips in any untangle sequence is $O(n^{8/3})$ [10].
3. What is the maximum number of flips involving a given point? The cubic potential provides a quadratic bound which leads again to an $O(n^3)$ bound for $\mathbf{D}(n)$.

We proved the NP-hardness of computing the shortest untangle sequence for a red-blue matching. What is the complexity of computing the shortest untangle sequence for a TSP tour, for a red-on-a-line matching, or even for a convex instance? What about the longest untangle sequence?

References

- [1] Oswin Aichholzer, Wolfgang Mulzer, and Alexander Pilz. Flip distance between triangulations of a simple polygon is NP-complete. *Discrete & Computational Geometry*, 54(2):368–389, 2015.
- [2] Jin Akiyama and Noga Alon. Disjoint simplices and geometric hypergraphs. In *Third international conference on Combinatorial mathematics*, pages 1–3, 1989.
- [3] Sergey Bereg and Hiro Ito. Transforming graphs with the same degree sequence. In *Computational Geometry and Graph Theory*, pages 25–32, 2008.
- [4] Sergey Bereg and Hiro Ito. Transforming graphs with the same graphic sequence. *Journal of Information Processing*, 25:627–633, 2017.
- [5] Ahmad Biniiaz, Anil Maheshwari, and Michiel Smid. Flip distance to some plane configurations. *Computational Geometry*, 81:12–21, 2019.
- [6] Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenhaller, and Kunihiro Wasa. The perfect matching reconfiguration problem. In *44th International Symposium on Mathematical Foundations of Computer Science*, volume 138 of *LIPICs*, pages 80:1–80:14, 2019.
- [7] Édouard Bonnet and Tillmann Miltzow. Flip distance to a non-crossing perfect matching. *Computing Research Repository*, abs/1601.05989, 2016.
- [8] Prosenjit Bose and Ferran Hurtado. Flips in planar graphs. *Computational Geometry*, 42(1):60–80, 2009.
- [9] Nicolas Bousquet and Alice Joffard. Approximating shortest connected graph transformation for trees. In *Theory and Practice of Computer Science*, pages 76–87, 2020.
- [10] Guilherme D. da Fonseca, Yan Gerard, and Bastien Rivier. On the longest flip sequence to untangle segments in the plane, 2022. URL: <https://arxiv.org/abs/2210.12036>, doi:10.48550/ARXIV.2210.12036.
- [11] Mark De Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry & Applications*, 22(03):187–205, 2012.
- [12] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2014.

- [13] Péter L Erdős, Zoltán Király, and István Miklós. On the swap-distances of different realizations of a graphical degree sequence. *Combinatorics, Probability and Computing*, 22(3):366–383, 2013.
- [14] Seifollah Louis Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. I. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.
- [15] Seifollah Louis Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph II. uniqueness. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):135–147, 1963.
- [16] John Hershberger and Subhash Suri. Applications of a semi-dynamic convex hull algorithm. *BIT Numerical Mathematics*, 32(2):249–267, 1992.
- [17] Ferran Hurtado, Marc Noy, and Jorge Urrutia. Flipping edges in triangulations. *Discrete & Computational Geometry*, 22(3):333–346, 1999.
- [18] Takehiro Ito, Erik D. Demaine, Nicholas J.A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12):1054–1065, 2011.
- [19] Alice Joffard. *Graph domination and reconfiguration problems*. PhD thesis, Université Claude Bernard Lyon 1, 2020.
- [20] Charles L Lawson. Transforming triangulations. *Discrete Mathematics*, 3(4):365–372, 1972.
- [21] Anna Lubiw and Vinayak Pathak. Flip distance between two triangulations of a point set is NP-complete. *Computational Geometry*, 49:17–23, 2015.
- [22] Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4), 2018.
- [23] Yoshiaki Oda and Mamoru Watanabe. The number of flips required to obtain non-crossing convex cycles. In *Kyoto International Conference on Computational Geometry and Graph Theory*, pages 155–165, 2007.
- [24] Alexander Pilz. Flip distance between triangulations of a planar point set is apx-hard. *Computational Geometry*, 47(5):589–604, 2014.
- [25] Jan van den Heuvel. The complexity of change. *Surveys in Combinatorics*, 409:127–160, 2013.
- [26] Jan van Leeuwen. Untangling a traveling salesman tour in the plane. In *7th Workshop on Graph-Theoretic Concepts in Computer Science*, 1981.
- [27] Todd G Will. Switching distance between graphs with the same degrees. *SIAM Journal on Discrete Mathematics*, 12(3):298–306, 1999.
- [28] Ro-Yu Wu, Jou-Ming Chang, and Jia-Huei Lin. On the maximum switching number to obtain non-crossing convex cycles. In *26th Workshop on Combinatorial Mathematics and Computation Theory*, pages 266–273, 2009.