# A bottleneck matching problem with edge-crossing constraints

John Gunnar Carlsson*, Benjamin Armbruster†, Haritha Bellam‡, and Rahul Saladi§

July 23, 2015

**Abstract**

Motivated by a crane assignment problem, we consider a Euclidean bipartite matching problem with edge-crossing constraints. Specifically, given $n$ red points and $n$ blue points in the plane, we want to construct a perfect matching between red and blue points that minimizes the length of the longest edge, while imposing a constraint that no two edges may cross each other. We show that the problem cannot be approximately solved within a factor less than 1.277 in polynomial time unless P = NP. We give simple dynamic programming algorithms that solve our problem in two special cases, namely (1) the case where the red and blue points form the vertices of a convex polygon and (2) the case where the red points are collinear and the blue points lie to one side of the line through the red points.

## 1 Introduction

Our problem can be motivated by a problem in port operations. Consider an equal number of cranes and containers in the plane. Each crane $i$ must be assigned to a container $j$. However, crane $i$ can only be assigned to containers within a distance $R$ from it and no two cranes may cross each other. This is a Euclidean bipartite matching problem: given a collection of $n$ red points $\{r_1, \ldots, r_n\}$ (the cranes) and a collection of $n$ blue points $\{b_1, \ldots, b_n\}$ (the containers), our goal is to find a crossing-free matching between red-blue pairs. We call such a matching a *Euclidean non-crossing bipartite matching* (ENCBM) and denote the *size* of the matching as the length of the longest edge, i.e. the bottleneck size. Thus our decision problem is to determine whether an ENCBM of size at most $R$ exists, and our optimization problem is to find an ENCBM of minimal size. This optimization problem always has a feasible solution because a minimum-weight Euclidean matching (i.e., a matching that minimizes the total length of the edges) is always crossing-free [1]. In fact, it is easy to see that the minimum Euclidean matching is a factor $n$ approximation algorithm to our problem, as shown in Figure 1.

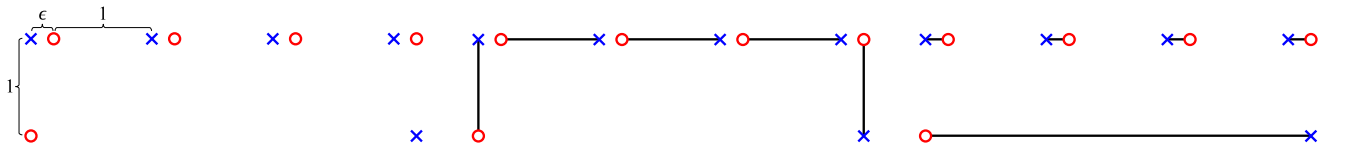
**Previous work** The problem of finding a crossing-free configuration of a graph in the plane is well-studied. The paper [5] considers the problem of determining whether a crossing-free spanning tree, perfect matching, or two-factor exists in a planar embedding of a graph under various restrictions on the segments that connect points. Most closely related to our result is their proof that the problem of finding a (non-bipartite) crossing-free perfect matching in a set of points on the lattice $\mathbb{Z} \times \mathbb{Z}$ is NP-hard when we restrict ourselves to only edges of constant length $d \in \mathbb{Z}$. The paper [7] shows that reconstructing a set of $n$ orthogonal line segments in the plane from their set of endpoints can be done in $\mathcal{O}(n \log n)$ time if the segments are allowed to cross, and if the segments are not allowed to cross, then the problem is NP-hard. The paper [3] shows that, if we are given $r$ red points in the plane and $b$ blue points in the

*Department of Industrial and Systems Engineering, University of Southern California. J. G. C. gratefully acknowledges DARPA Young Faculty Award N66001-12-1-4218, NSF grant CMMI-1234585, and ONR grant N000141210719.

†Department of Industrial Engineering and Management Science, Northwestern University.

‡Department of Computer Science & Engineering, University of Minnesota.

§Department of Computer Science & Engineering, University of Minnesota.




(a) Arrangement of points (b) ENCBM of minimal size (c) Minimum-weight Euclidean matching

Figure 1: A tight example showing the factor $n$ approximation of our problem given by the minimum-weight Euclidean matching: 1b has a maximum edge length of 1 while 1c has a maximum edge length of $(n-2)(1+\epsilon)$.

plane, then there exists a crossing-free matching of points *of the same color* that matches at least 83.33% of the points and can be found in $\mathcal{O}((r+b)\log(r+b))$ time. Very recently, the paper [2] gives a factor $2\sqrt{10}$ approximation algorithm for the *non-bipartite* version of our problem. Unfortunately, as explained in Remark 2 and Figure 5 thereof, it does not appear that this approach lends itself to our bipartite problem.

Detection of crossing-free subgraphs is a useful problem in several domains. The paper [9] considers the problem of *scheduling* a set of cranes to cover a set of jobs when a non-crossing constraint (among other spatial considerations) is imposed. The authors assume that the cranes and containers are located in two parallel columns. The paper [10] extends this; proves that the scheduling problem is NP-hard; and gives a branch-and-bound algorithm for solving it. Crossing-free configurations are relevant in VLSI applications where the number of crossings in a matching corresponds to the number of layers required in the layout of the circuit design [4].

**New results** This paper presents three new results about the problem of finding a minimum-cost ENCBM: first, we show that it is NP-hard and cannot be approximately solved within a factor less than 1.277 in polynomial time unless P = NP. Second, we give an algorithm that finds an optimal ENCBM in $\mathcal{O}(n^3)$ time for the special case where the the points $\{r_1,\ldots,r_n\}$ and $\{b_1,\ldots,b_n\}$ form the vertices of a convex polygon. Third, we give an algorithm that finds an optimal ENCBM in $\mathcal{O}(n^4)$ time for the special case where the points $\{r_1,\ldots,r_n\}$ lie on a line and the points $\{b_1,\ldots,b_n\}$ all lie to one side of that line.

## 2 Hardness

In this section we show that the decision problem "does there exist a non-crossing perfect matching of a given red-blue configuration with all edge lengths at most $R$?" reduces to the planar 3-satisfiability problem (planar 3-SAT), which is known to be NP-hard [6]. The paper [7] also uses a reduction to planar 3-SAT on a problem dealing with non-crossing *orthogonal* edges linking points and no constraints on distances. Our proof uses a similar reduction although we require a few new ideas to tie everything together. Our main result here is given in Theorem 1, which we will state after defining the relevant structures in the transformation.

Let us define an instance of planar 3-SAT with variables $X := \{x_1,\ldots,x_m\}$, and clauses $C := \{c_1,\ldots,c_n\}$. We denote negation with a tilde, e.g., $\tilde{x}_1$. As in normal 3-SAT, each clause is a logical disjunction of three variables or their negations, e.g., $c_1 := x_1 \vee x_5 \vee \tilde{x}_7$. We now define a graph whose vertices are the variables and clauses, $X \cup C$, and whose edges connect any variable and the clauses it appears in, $E := \{(x_i, c_j) | x_i \in c_j \text{ or } \tilde{x}_i \in c_j\}$. In planar 3-SAT, this graph, $(X \cup C, E)$ is planar. (The definition of planar 3-SAT in [6] has some additional edges between the vertices in $X$. That this larger graph is planar obviously implies that our smaller graph, $(X \cup C, E)$, is planar.)

Our reduction will produce a set of red and blue points $V$ such that an ENCBM of size at most $R$ exists if and only if our planar 3-SAT instance, $(X \cup C, E)$ has a solution. For any set of red and blue

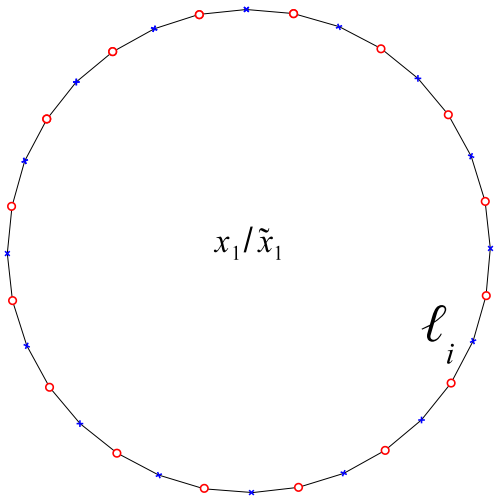



Figure 2: A collection of red and blue points $V_1$ corresponding to the 3-SAT variable $x_1$.

points, $P$, we define $G(P)$ as the graph formed by constructing all edges between red and blue points with length at most $R$. Thus an ENCBM of size at most $R$ exists on a set $P$ if and only if $G(P)$ has a perfect matching. We will replace any variable $x_i$ with a large collection of red and blue points $V_i$ arranged in a loop $\ell_i := G(V_i)$ with the points a distance $R$ away from each other and alternating in color as in Figure 2. (For now, we omit the discussion of exactly how many points are needed.) This configuration clearly has two possible perfect matchings: one with red points matched to blue points in the clockwise direction and one in the counterclockwise direction. We decide arbitrarily that matching red points to blue in the clockwise direction corresponds to setting variable $x_i$ true. We treat each edge in the planar 3-SAT between a variable, $x_i$, and a clause $c_j$, $(x_i, c_j) \in E$, like a variable whose value is $x_i$ or $\tilde{x}_i$ if $x_i$ or $\tilde{x}_i$ appears in $c_j$, respectively. As with a variable, we thus represent an edge in the planar 3-SAT by a collection of red and blue points $V_{ij}$ arranged in a loop $\ell_{ij} := G(V_{ij})$ with the points a distance $R$ away from each other and alternating in color. Again, the orientation of the perfect matching on $\ell_{ij}$ corresponds to the truth-value of the edge. To enforce this connection we intersect the loops $\ell_i$ and $\ell_{ij}$ as in Figure 3.

We now describe how loop $\ell_{ij}$ will intersect the loop $\ell_i$ in such a way that the matching on $\ell_i$ will determine the matching on $\ell_{ij}$. At each intersection of $\ell_{ij}$ and $\ell_i$, $G(V_{ij} \cup V_i)$ has two additional edges connecting red and blue points on $\ell_i$ to points of the opposite color on $\ell_{ij}$ as shown in Figure 4. This is problematic because the matching on $\ell_i$ does not affect the matching on $\ell_{ij}$ as shown in Figure 5. Fortunately, we can resolve this issue by inserting two additional red points and two additional blue points at each intersection (see Figure 6), and therefore we can assume from now on that the orientations of all loops $\ell_{ij}$ are imposed by the orientation of $\ell_i$. The key property that we exploit here is that the two intersections occur at edges with *opposite* orientation; this allows us to force the orientation of $\ell_{ij}$ to match the orientation of $\ell_i$. Figure 7 shows a single loop $\ell_1$, corresponding to a variable $x_1$, and its intersection with four loops $\ell_{11}, \ldots, \ell_{14}$, each of which in turn intersects one of four clauses $c_1, c_2, c_3, c_4$.

Let us examine clause $c_j$ and assume without loss of generality that $c_j$ is connected by three edges to $x_1$, $x_2$, and $x_3$ in the instance of planar 3-SAT, i.e. that $c_j = x_1 \vee x_2 \vee x_3$ (or, with only a slight abuse of notation, $c_j = \ell_{1j} \vee \ell_{2j} \vee \ell_{3j}$). Such an assumption is made without loss of generality because we can easily handle the case where some of the entries $x_i$ are negated by positioning the loops relative to the loop $\ell_i$ in an appropriate way as shown in Figure 7, in which we represent each clause $c_j$ as a rectangle $\square_j$ that intersects $\ell_{ij}$. The construction of $\square_j$ is shown in Figure 8. Each loop $\ell_{1j}$, $\ell_{2j}$, $\ell_{3j}$ enters $\square_j$ on the bottom side. Let $\Sigma_j$ denote a union of three alternating red-blue loops in $\square_j$ that cross each $\ell_{ij}$. Finally, let $\Pi_j$ denote an alternating red-blue loop in $\square_j$ that intersects each of the three loops in $\Sigma_j$. Whenever two loops intersect in this construction, we "fix" the intersection using two additional red points and two

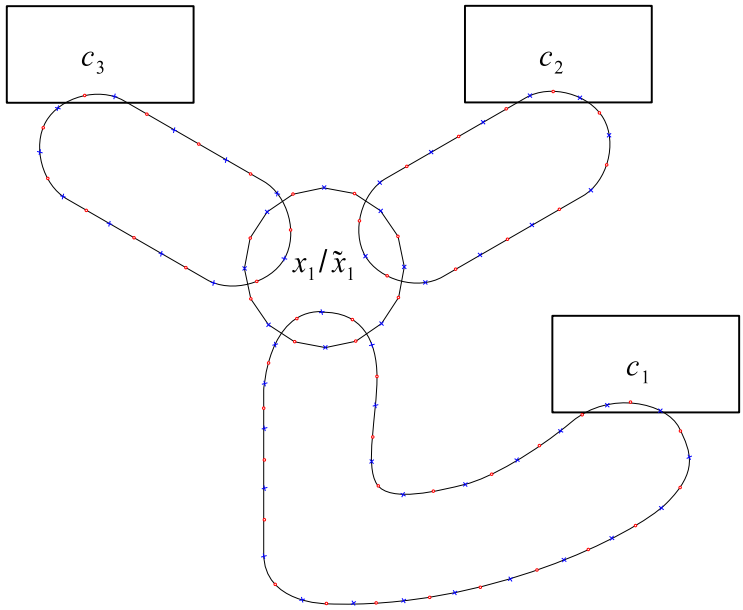



Figure 3: A collection of loops representing the connections between variable $x_1$ and its associated clauses $c_1$, $c_2$, and $c_3$.

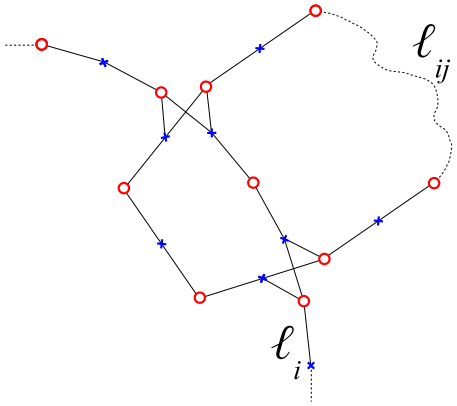
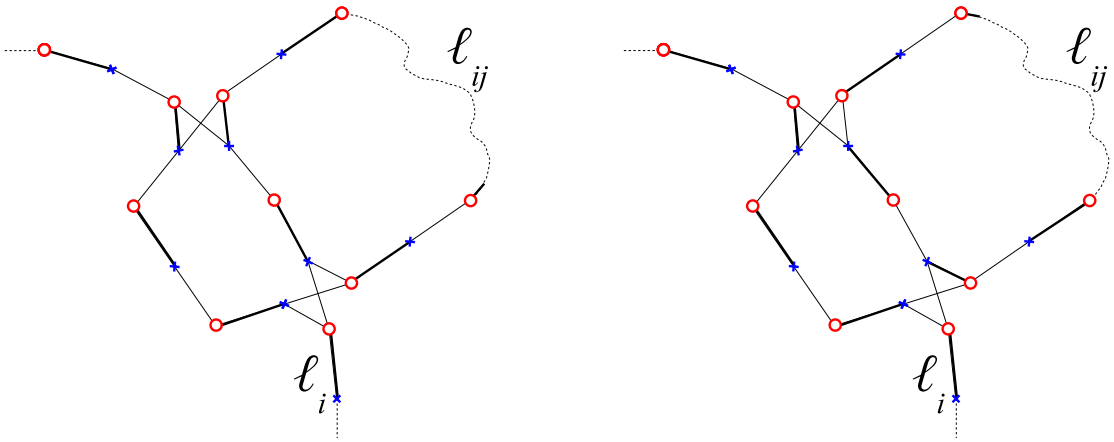



Figure 4: Each intersection of $\ell_{ij}$ and $\ell_i$, creates two additional edges in $G(V_{ij} \cup V_i)$ between the two loops.




(a) (b)

Figure 5: While fixing the same matching on $\ell_i$, we can construct two distinct perfect matchings of the loop $\ell_{ij}$. This is undesirable because we want the matching on $\ell_{ij}$ to be induced by the matching on $\ell_i$, and we show how to ameliorate this issue in Figure 6 by inserting four more points into the configuration.

additional blue points as described above.

It is now easy to verify that a non-crossing matching of the points in $\square_j$ exists if and only if one of $\ell_{1j}$, $\ell_{2j}$, or $\ell_{3j}$ is true, as shown in Figures 8, 9, and 10. We have now almost completed the proof of our main theorem:

**Theorem 1.** *Let $\{r_1, \ldots, r_n\}$ denote a set of $n$ red points in the plane, let $\{b_1, \ldots, b_n\}$ denote a set of $n$ blue points in the plane, and let $R > 0$ be a given positive number. Unless $P = NP$, the decision problem*

> *"Does there exist a non-crossing perfect bipartite matching between red and blue points, all of whose edges have length at most $R$?"*

*cannot be solved in running time polynomial in $n$.*

*Proof.* To complete the proof, we merely have to show that, given an instance of planar 3-SAT, the the number of red and blue points $n$ that are required for our reduction depends on the number of variables $v$ and the number of clauses $c$ in a polynomial fashion. A well-known theorem of Schnyder [8] says that any planar graph with $k$ vertices has an embedding on the $k-2 \times k-2$ grid. We thus embed our instance of planar 3-SAT on the $(c+v-2) \times (c+v-2)$ grid. We then place the loops $\ell_i$ and $\ell_{ij}$ and boxes, $\square_j$, in the rough locations of the corresponding vertices and edges of the 3-SAT graph on the grid. Each $\square_j$ contains a constant number of red and blue points and thus has a diameter $\mathcal{O}(R)$. Since the edges in the grid have length less than $(c+v-2)\sqrt{2}$ the loops $\ell_{ij}$ have $\mathcal{O}((c+v)/R)$ nodes. The loops $\ell_i$ corresponding to variables are connected to at most $v$ edges and thus contain $\mathcal{O}(v)$ red and blue points and has diameter $\mathcal{O}(vR)$. To ensure that loops $\ell_i$ and boxes $\square_j$ have diameter (say) $1/2$, we choose $R$ of $\mathcal{O}(1/v)$. Thus a loop such as $\ell_{ij}$, of which there are $3n$, has $\mathcal{O}(v(c+v))$ nodes. Therefore, the total number of red and blue points is $\mathcal{O}((c+v)cv)$. □

The following two corollaries are also evident:

**Corollary 2.** *Let $\{r_1, \ldots, r_m\}$ denote a set of $m$ red points in the plane, let $\{b_1, \ldots, b_n\}$ denote a set of $n$ blue points in the plane, and let $R > 0$ be a given positive number. Then the problem of finding a* maximum *non-crossing matching between red and blue points such that all edges have length at most $R$ cannot be solved in running time polynomial in $n$.*

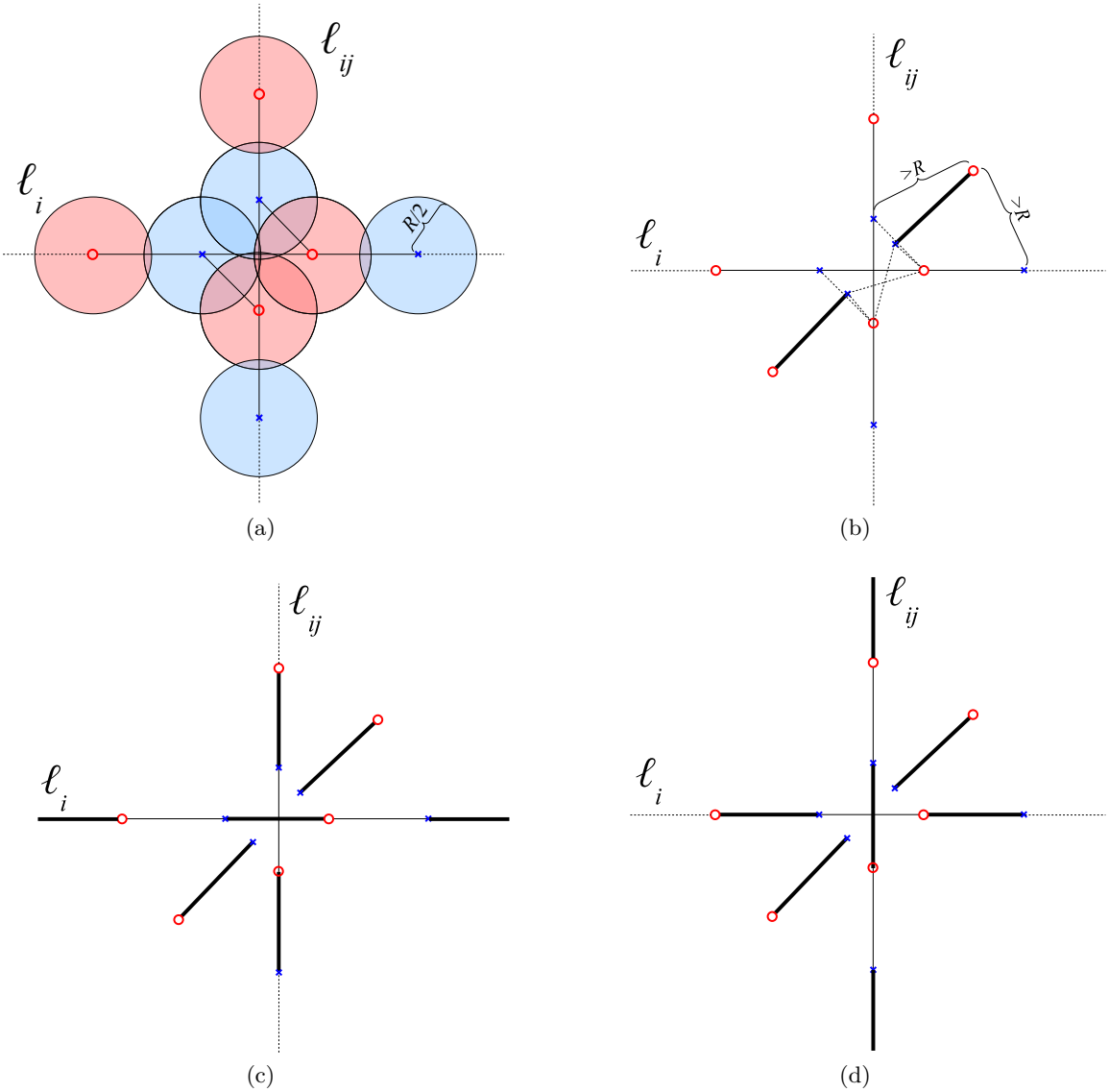



Figure 6: In (a), we show an intersection between two loops $\ell_i$ and $\ell_{ij}$. In (b), we insert two new red points and two new blue points strategically so as to restrict the possible matchings on $\ell_i$ and $\ell_{ij}$. Since the two newly inserted red points have only one permissible (blue) neighbor each, they must be connected to that neighbor in a perfect matching. The non-crossing constraint eliminates the dotted edges, which in turn implies that – as shown in (c) and (d) – the matching on $\ell_i$ will always force a particular matching on $\ell_{ij}$.

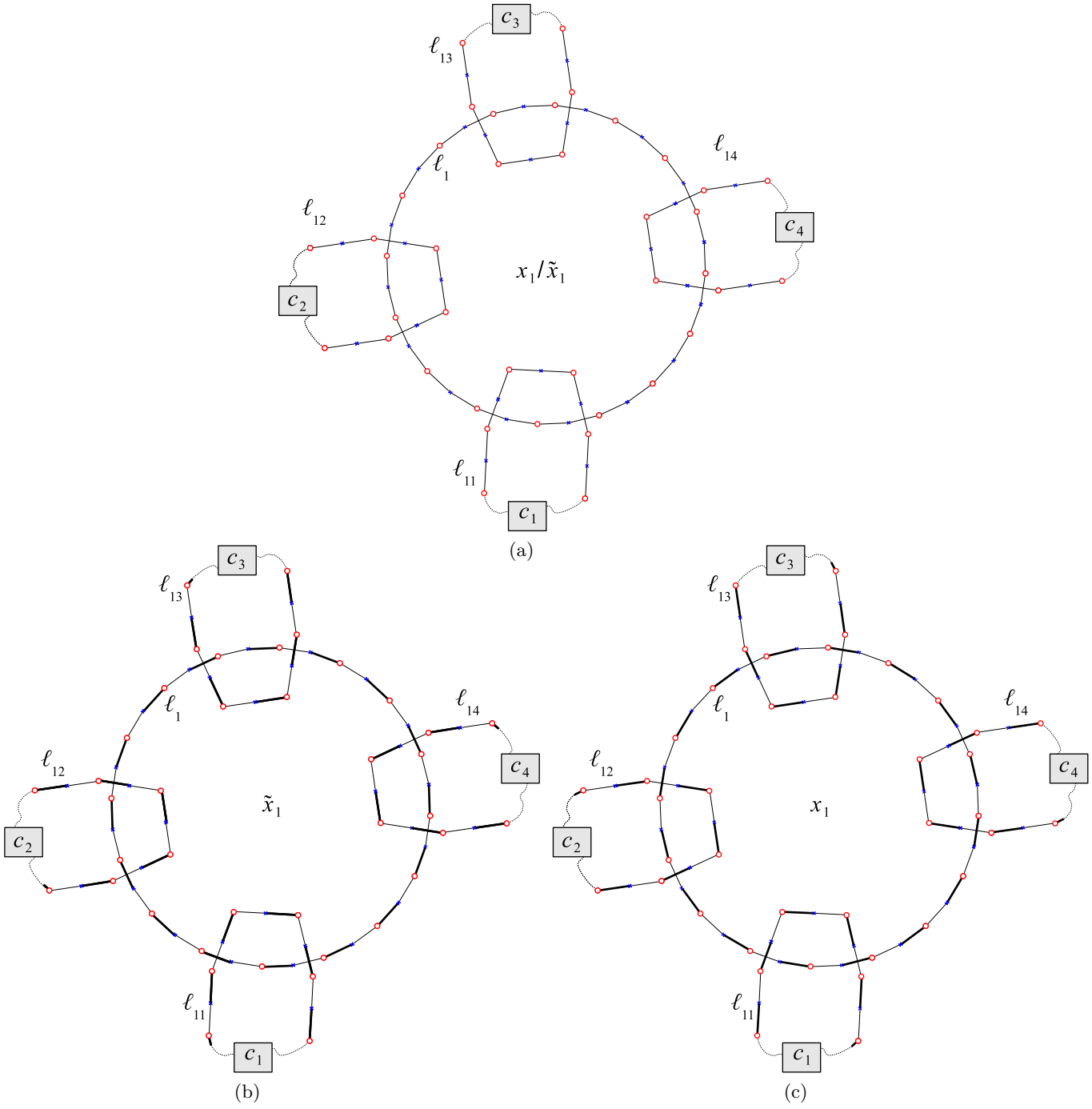



Figure 7: An arrangement of loops corresponding to a variable $x_1$ and four clauses $c_1, c_2, c_3, c_4$ containing it. Notice that $\ell_{11}$ must have the same matching orientation as $\ell_1$ but $c_2$, $c_3$, and $c_4$ have opposing orientations to $\ell_1$. This corresponds to the case where $c_1$ uses $x_1$ and $c_2$, $c_3$, and $c_4$ use $\tilde{x}_1$.

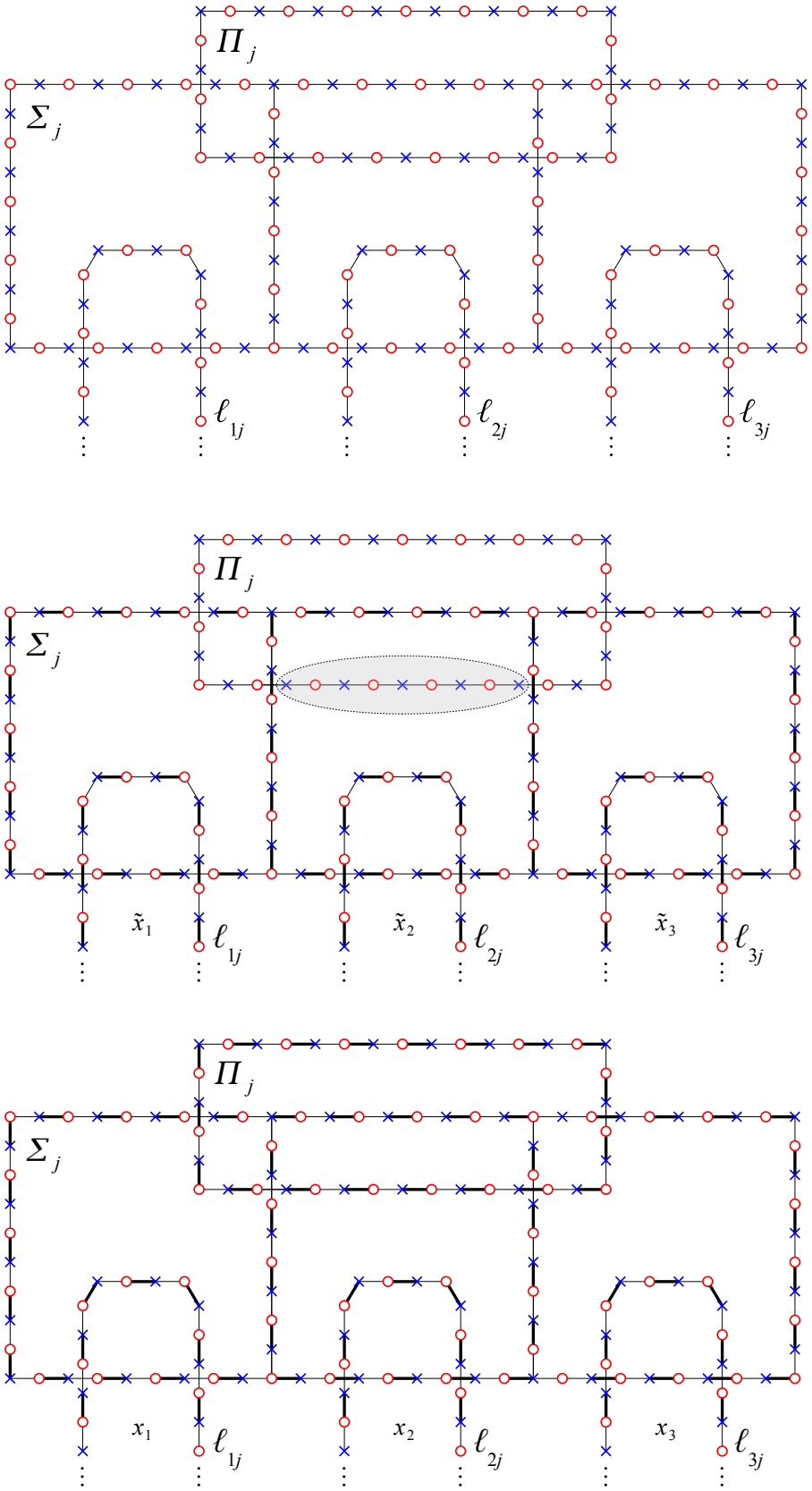



Figure 8: The contents of rectangle $\square_j$, which corresponds to clause $c_j = \ell_{1j} \vee \ell_{2j} \vee \ell_{3j}$. As shown in the middle diagram, setting $\ell_{1j}$, $\ell_{2j}$, and $\ell_{3j}$ false ensures that no perfect matching can exist in loop $\Pi_j$. This is because the required matching on $\Sigma_j$ prevents a perfect matching from occurring in the shaded ellipse shown. On the other hand, setting the three variables true allows a perfect matching to exist as shown in the third diagram.

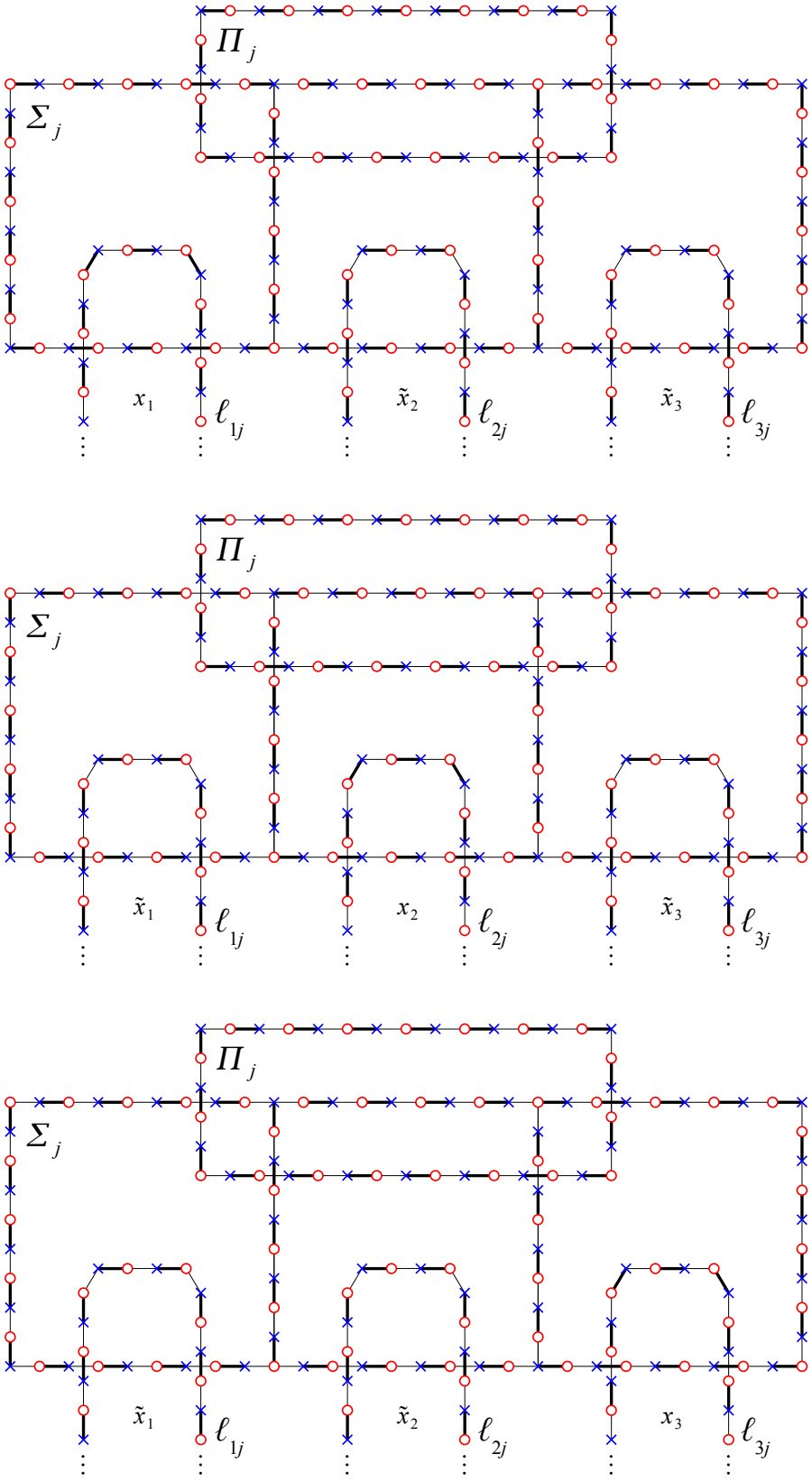



Figure 9: The perfect matchings in $\square_j$ corresponding to the cases where one of $\ell_{1j}$, $\ell_{2j}$, and $\ell_{3j}$ is true.

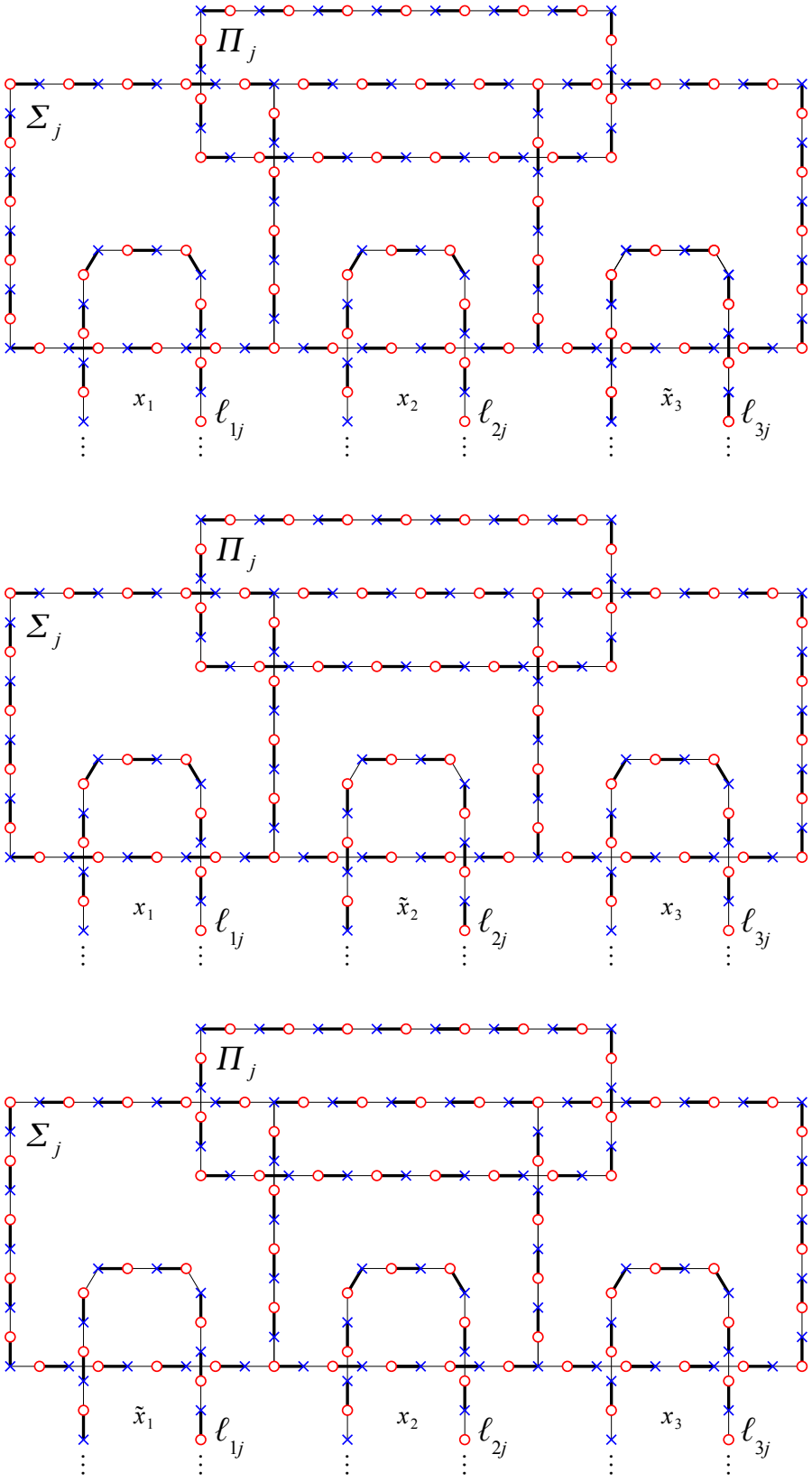



Figure 10: The perfect matchings in $\square_j$ corresponding to the cases where two of $\ell_{1j}$, $\ell_{2j}$, and $\ell_{3j}$ are true.

*Proof.* The maximum matching problem is a generalization of the perfect matching problem that we have already discussed and is therefore at least as difficult. □

**Corollary 3.** *Let $\{r_1, \ldots, r_n\}$ and $\{b_1, \ldots, b_n\}$ be as before, and consider the problem of finding the* minimal *$R$ such that there exists a non-crossing perfect bipartite matching between red and blue points consisting of edges whose length is at most $R$. Then unless $P = NP$, the optimization problem of minimizing $R$ cannot be solved approximately within a factor less than* 1.277 *in polynomial running time.*

*Proof.* We do the following: given an instance of planar 3-SAT, we will build an instance of ENCBM such that a matching of size $R$ exists if and only if a matching of size less than $1.277R$ exists. In our previous construction, this is already the case for nodes that are not near a loop intersection (since their next nearest neighbors are approximately a distance $3R$ away already). Thus, it will suffice to describe a construction of an intersection such that our condition holds.

Precisely this construction is shown in Figure 11, which completes the proof. □

## 3 An algorithm for the convex case

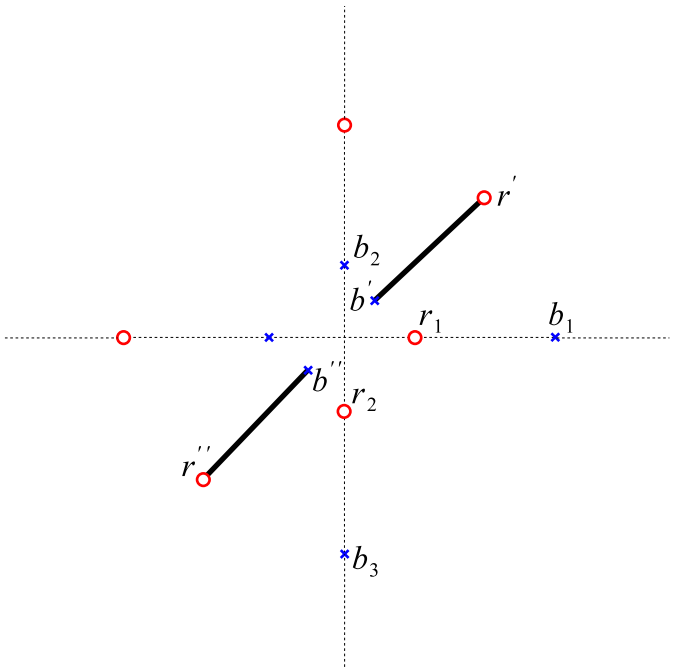
In this section we consider the special case when all points $\{r_1, \ldots, r_n\}$ and $\{b_1, \ldots, b_n\}$ form the vertices of a convex polygon. We give an algorithm that finds an ENCBM of minimal size in $\mathcal{O}(n^3)$ time. Our algorithm actually generalizes to certain other cases as well, which we describe at the end of this section. Let $\{x_1, \ldots, x_{2n}\}$ denote the red and blue vertices of this polygon in counter-clockwise order with an arbitrary first vertex $x_1$. Connect all pairs of vertices with an edge $(x_i, x_j)$ if $x_i$ and $x_j$ are different colors, and assign this edge a weight $w_{ij} = \|x_i - x_j\|$. Two edges $(x_i, x_j)$ and $(x_s, x_t)$ with $i < j$ and $s < t$ are non-crossing if and only if one interval (i.e. $[i, i+1, \ldots, , j]$ or $[s, s+1, \ldots, t]$) is contained within the other, or if the intervals are entirely disjoint; equivalently, the edges are crossing if and only if $i \leq s \leq j \leq t$ or $s \leq i \leq t \leq j$.

For each pair $x_s, x_t$ with $s < t$, we define the *cost* $V(s,t)$ to be the size (i.e. the length of the longest edge) of the ENCBM of points $x_s, \ldots, x_t$, with $V(s,t) = \infty$ if such a matching does not exist. It is obvious that $V(s,t) < \infty$ if and only if the set $\{x_s, \ldots, x_t\}$ contains the same number of red points and blue points. Assuming that this is the case, the bottleneck ENCBM of $\{x_s, \ldots, x_t\}$ either uses the edge $(x_s, x_t)$, or it does not (note that $(x_s, x_t)$ may not even be a valid edge because $x_s$ and $x_t$ could be the same color). If the bottleneck ENCBM uses edge $(x_s, x_t)$, then its cost is the maximum of $w_{st}$ and $V(s+1, t-1)$. If edge $(x_s, x_t)$ is not used, then there must be some index $i \in \{s+1, \ldots, t-2\}$ such that $x_s$ is paired with $x_i$ and $x_{i+1}$ is paired with $x_t$ (with all remaining interior vertices $x_j$ paired off with one another in some valid way). In other words, we have

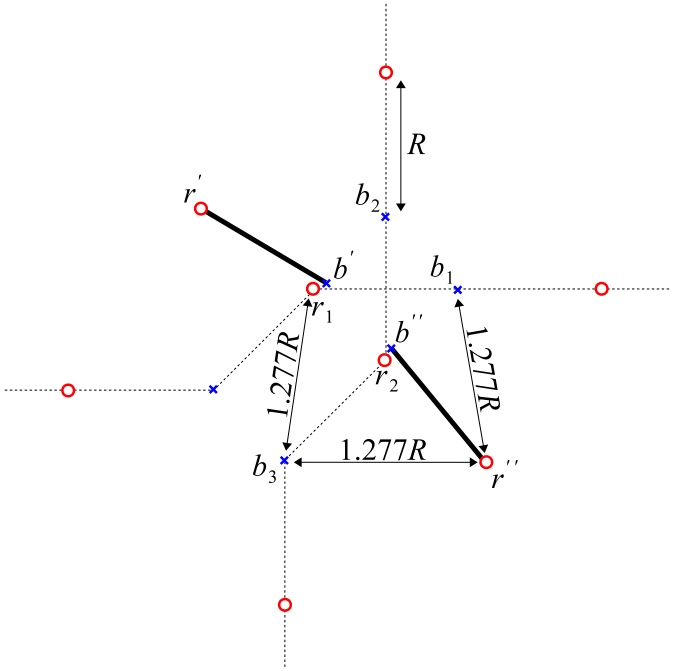
$$V(s,t) = \min \left\{ \max\{w_{st}, V(s+1, t-1)\}, \min_{i \in \{s+1, \ldots, t-2\}} \max\{V(s,i), V(i+1,t)\} \right\}.$$

We initialize $V(\cdot, \cdot)$ by setting $V(i, i+1) = w_{i,i+1}$ for all indices $i$ such that $x_i$ and $x_{i+1}$ are different colors. In order to compute arbitrary $V(s,t)$, we must know the values $V(s+1, t-1)$ and the values $V(s,i)$ and $V(i+1,t)$ for all $i \in \{s+1, \ldots, t-2\}$. Thus, we can build all values of $V(\cdot, \cdot)$ by enumerating through the pairs $(s,t)$ in order of their *width*, $t - s$. Our goal is of course to compute $V(1, 2n)$. Each calculation of $V(s,t)$ can be computed in $\mathcal{O}(n)$ running time due to the enumeration over all $i$, and therefore we can compute all possible $V(s,t)$ – and therefore a minimum-cost ENCBM – in $\mathcal{O}(n^3)$ running time.

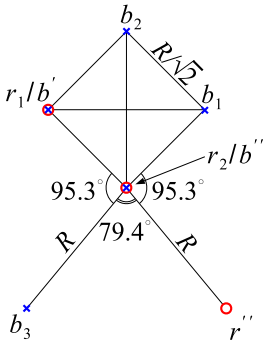
In the algorithm above, we do not explicitly use the fact that edge weights are Euclidean; the procedure as described can be used to find a non-crossing perfect matching of *any* graph whose vertices form the vertices of a convex polygon (provided such a matching exists). For example, we might instead constrain ourselves to consider only edges that subtend a particular angle. The same result holds for points on the boundary of a simply connected polygon, if we impose the constraint that edges may not leave the polygon.




(a) Original configuration from figure 6, labelled.




(b) Revised configuration, with corresponding labels




(c)

Figure 11: In (11a), we show the original configuration for intersections as described previously in this section. In (11b) we show an alternative layout, so that a matching of size $R$ exists if and only if a matching of size $1.277R$ also exists. The exact positioning of the various points is shown in detail in (11c). The main idea is to position $r_1$ and $b'$ (and $r_2$ and $b''$) as close as possible while still maintaining the property that edge $b_1 - r_2$ and edge $b_2 - r_1$ are prevented.

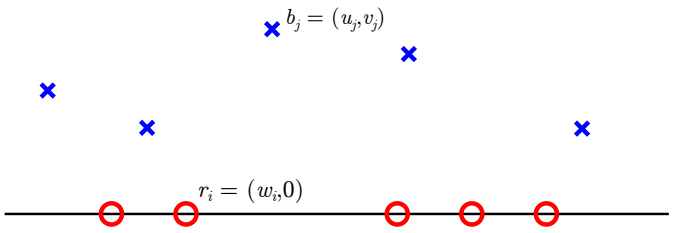



Figure 12: The setup for the semi-collinear case of our problem.

## 4 An algorithm for the semi-collinear case

In this section we consider the special case when the points $\{r_1, \ldots, r_n\}$ lie on a line and the blue points $\{b_1, \ldots, b_n\}$ all lie on one side of that line. We assume without loss of generality that the line in question is the horizontal axis so that each point $r_i$ takes the form $r_i = (w_i, 0)$, with $w_i < w_{i+1}$ for all $i$, and that each blue point takes the form $b_j = (u_j, v_j)$ with $v_j > 0$; this setup is shown in Figure 12. We give an algorithm that finds an ENCBM of minimal size in running time $\mathcal{O}(n^4)$. For expositional purposes, we will make a technical assumption that the vertical coordinates $v_i$ are all distinct and that all points are in general position, with the obvious exception of the collinear red points.

We begin by constructing the graph $G = (\{r_1, \ldots, r_n\} \cup \{b_1, \ldots, b_n\}, E)$ in which we have an edge between every pair $r_i, b_j$ with weight $w_{ij} = \|r_i - b_j\|$. Consider any triple $(r_i, r_j, b_k)$, with $i < j$. We can construct a "rightward slab" $\mathscr{R}_{ij}^k$ by intersecting the two open half-spaces $\{(x_1, x_2) \in \mathbb{R}^2 : x_2 > 0\}$ and $\{(x_1, x_2) \in \mathbb{R}^2 : x_2 < v_k\}$ with the open region lying strictly to the right of the oriented line $\overrightarrow{r_i b_k}$, as shown in Figure 13. We can uniquely define an index $p^{\text{right}}(i, j, k)$ to be the index of the blue point $b^*$ contained in $\mathscr{R}_{ij}^k$ such that the region obtained by intersecting $\mathscr{R}_{ij}^k$ with the open region lying strictly to the left of the oriented line $\overrightarrow{r_j b^*}$ contains exactly $j - i - 1$ blue points (of course, such a point is not guaranteed to exist, but if one does, it must be unique). This now defines an open "rightward trapezoid" $T_{ijk}^{\text{right}}$, which is also shown in Figure 13. Note that we can also perform an analogous procedure starting with a "leftward slab" $\mathscr{L}_{ij}^k$ that is obtained by intersecting the same two open half-spaces $\{(x_1, x_2) \in \mathbb{R}^2 : x_2 > 0\}$ and $\{(x_1, x_2) \in \mathbb{R}^2 : x_2 < v_k\}$ with the open region lying strictly to the *left* of the oriented line $\overrightarrow{r_j b_k}$. This would then also define an index $p^{\text{left}}(i, j, k)$ to be the index of the blue point $b^*$ such that the region obtained by intersecting $\mathscr{L}_{ij}^k$ with the open region lying strictly to the *right* of the oriented line $\overrightarrow{r_i b^*}$ contains exactly $j - i - 1$ blue points, thereby specifying a "leftward trapezoid" $T_{ijk}^{\text{left}}$.

We have thus shown that, given three points $(r_i, r_j, b_k)$ with $i < j$, we can construct a unique rightward trapezoid whose closure contains $j - i + 1$ points of each color (if such a trapezoid exists). The triple $(i, j, k)$ uniquely defines the index $p^{\text{right}}(i, j, k)$ as well as another index, which we will write as $q^{\text{right}}(i, j, k)$, which is the index of the highest blue point in the interior of $T_{ijk}^{\text{right}}$ (which must lie below $b_k$ by our construction). Let $V(i, j, k)$ denote the size of the optimal ENCBM of all of the points in $T_{ijk}^{\text{right}}$, subject to the constraint that $r_i$ and $b_k$ are matched together, as are $r_j$ and $b_p$, where we write $p = p^{\text{right}}(i, j, k)$ for brevity. Define $V(i, j, k) = \infty$ if no such trapezoid $T_{ijk}^{\text{right}}$ exists. If $T_{ijk}^{\text{right}}$ does exist, then $V(i, j, k)$ is the maximum of $w_{ik}$, $w_{jp}$, and the size of the optimal ENCBM of the remaining points, namely $\{r_{i+1}, \ldots, r_{j-1}\}$ and $\{b_1, \ldots, b_n\} \cap T_{ijk}^{\text{right}}$.

Let $W(i, j, k)$ denote the analogous size of the optimal ENCBM for $T_{ijk}^{\text{left}}$ (in which case we would constrain that $r_j$ and $b_k$ be paired together, and so on). We now show that $V(i, j, k)$ and $W(i, j, k)$ can be described in a recursive fashion: assuming that $T_{ijk}^{\text{right}}$ does indeed exist, let $\mathcal{M}$ denote its optimal ENCBM (with the additional constraint that $r_i$ and $b_k$ be matched together, as well as $r_j$ and $b_p$), and consider the highest blue point, whose index is $q^{\text{right}}(i, j, k)$, which we abbreviate as $q$. Let $\ell \in \{i + 1, \ldots, j - 1\}$ denote the index of its (red) partner under $\mathcal{M}$. The following statements must be true by inspection (we will continue to abbreviate $p = p^{\text{right}}(i, j, k)$ as before):

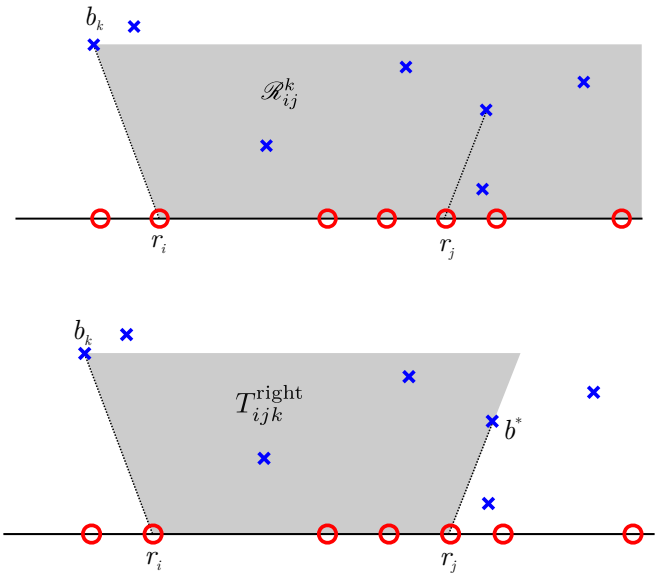



Figure 13: The "rightward slab" $\mathscr{R}_{ij}^k$ and the "rightward trapezoid" $T_{ijk}^{\text{right}}$.

- If $b_q$ is above $b_p$, then the optimal ENCBM for $T_{ijk}^{\text{right}}$ (with $r_i$ matched with $b_k$ and $r_j$ matched with $b_p$) is equivalent to the optimal ENCBM for $T_{i\ell k}^{\text{right}}$ (with $r_i$ matched with $b_k$ and $r_\ell$ matched with $b_q$) together with the optimal ENCBM for $T_{\ell jq}^{\text{right}}$ (with $r_\ell$ matched with $b_q$ and $r_j$ matched with $b_p$). Note that both of these two matchings have $r_\ell$ and $b_q$ matched together, by construction.
- If $b_p$ is above $b_q$, then the optimal ENCBM for $T_{ijk}^{\text{right}}$ (with $r_i$ matched with $b_k$ and $r_j$ matched with $b_p$) is equivalent to the optimal ENCBM for $T_{i\ell k}^{\text{right}}$ (with $r_i$ matched with $b_k$ and $r_\ell$ matched with $b_q$) together with the optimal ENCBM for $T_{\ell jp}^{\text{left}}$ (with $r_\ell$ matched with $b_q$ and $r_j$ matched with $b_p$). These two matchings also both have $r_\ell$ and $b_q$ matched together, by construction.

These two scenarios are shown in Figure 14. Note that, for fixed $(i,j,k)$, we can construct an index set $\mathcal{I}(i,j,k) \subset \{i+1,\ldots,j-1\}$ of all red indices $\ell$ that could possibly be partnered with $b_q$; this is simply the set of all $\ell \in \{i+1,\ldots,j-1\}$ such that that $p^{\text{right}}(i,\ell,k) = q$ (of course, if $T_{ijk}^{\text{right}}$ does not exist, then $\mathcal{I}(i,j,k) = \emptyset$ and $V(i,j,k) = \infty$). Expressed algebraically, the preceding statements are equivalent to writing

$$V(i,j,k) = \begin{cases} \infty & \text{if } T_{ijk}^{\text{right}} \text{ does not exist} \\ \min_{\ell \in \mathcal{I}(i,j,k)} \max\{V(i,\ell,k)\,,\, V(\ell,j,q^{\text{right}}(i,j,k))\} & \text{if } b_{q^{\text{right}}(i,j,k)} \text{ is above } b_{p^{\text{right}}(i,j,k)} \\ \min_{\ell \in \mathcal{I}(i,j,k)} \max\{V(i,\ell,k)\,,\, W(\ell,j,p^{\text{right}}(i,j,k))\} & \text{if } b_{q^{\text{right}}(i,j,k)} \text{ is below } b_{p^{\text{right}}(i,j,k)} \end{cases}$$

and equivalently

$$W(i,j,k) = \begin{cases} \infty & \text{if } T_{ijk}^{\text{left}} \text{ does not exist} \\ \min_{\ell \in \mathcal{J}(i,j,k)} \max\{W(\ell,j,k)\,,\, W(i,\ell,q^{\text{left}}(i,j,k))\} & \text{if } b_{q^{\text{left}}(i,j,k)} \text{ is above } b_{p^{\text{left}}(i,j,k)} \\ \min_{\ell \in \mathcal{J}(i,j,k)} \max\{W(\ell,j,k)\,,\, V(i,\ell,p^{\text{left}}(i,j,k))\} & \text{if } b_{q^{\text{left}}(i,j,k)} \text{ is below } b_{p^{\text{left}}(i,j,k)} \end{cases}$$

where $\mathcal{J}(i,j,k)$ is the analogous index set to $\mathcal{I}(i,j,k)$. We initialize $V(\cdot,\cdot,\cdot)$ by computing $V(i,i+1,k)$ for all $i$ and $k$ (and similarly for $W(\cdot,\cdot,\cdot)$); this is straightforward because $V(i,i+1,k) = \max\{w_{ik}, w_{i+1,p}\}$, with $p = p^{\text{right}}(i,i+1,k)$, and with $V(i,i+1,k) = \infty$ if no such $p$ (and therefore no such trapezoid) exists. This clearly suffices to compute all values of $V(\cdot,\cdot,\cdot)$ because any $V(i,j,k)$ is expressible in terms of strictly smaller intervals $[i,i+1,\ldots,\ell]$ and $[\ell,\ell+1,\ldots,j]$. For any triple $(i,j,k)$, we can compute

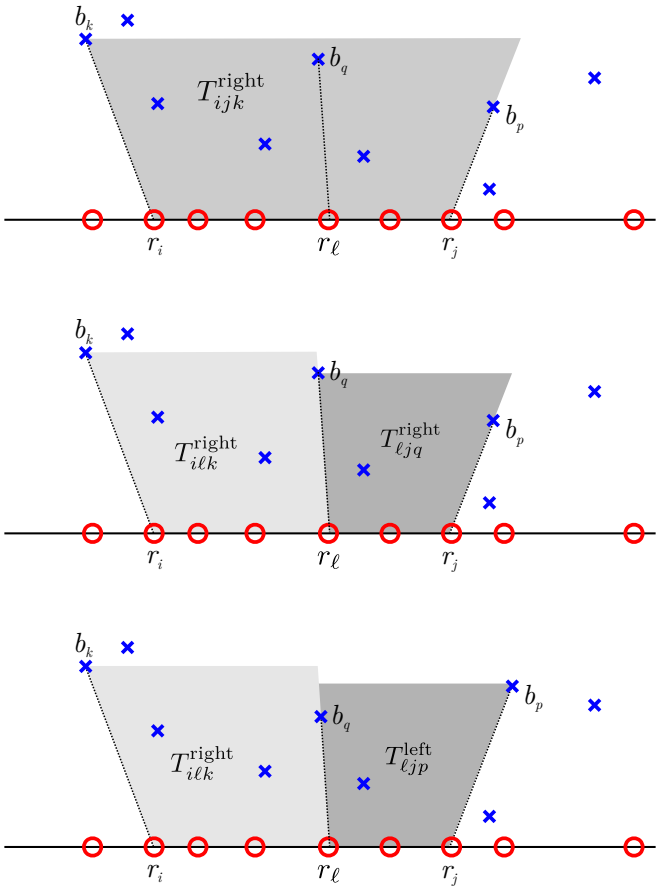
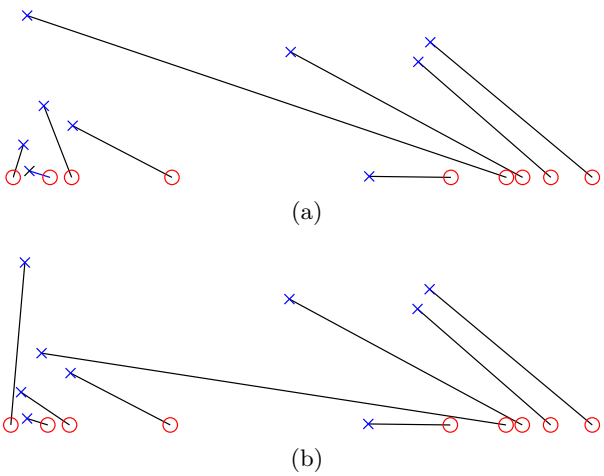



Figure 14: A rightward trapezoid $T_{ijk}^{\text{right}}$ and the two scenarios relating $b_p$ and $b_q$.



(a)

(b)

Figure 15: A minimum-weight Euclidean matching (15a) and a bottleneck ENCBM (15b).

$p^{\text{right}}(i,j,k)$ (and equivalently $p^{\text{left}}(i,j,k)$) or determine that no such point exists in $\mathcal{O}(n)$ running time, and thereby determine if $T_{ijk}^{\text{right}}$ or $T_{ijk}^{\text{left}}$ exists: to see this, we note that it is necessary to search through blue points based on the angle of the line that joints them to red point $r_j$. Thus, as a pre-processing step, for each red point $r_j$, we must sort the blue points $b_i$ based on the angle of line $\overrightarrow{r_j b_i}$ (which would require running time $\mathcal{O}(n^2 \log n)$ in total). For each triple $(i,j,k)$, we can identify the blue points that lie in the "rightward slab" $\mathscr{R}_{ij}^k$ in $\mathcal{O}(n)$ running time, and then find $p^{\text{right}}(i,j,k)$ by selecting the $(j-i)$-th element of these points (sorted by the angle they make with $r_j$). It is obvious that we can also compute $q^{\text{right}}(i,j,k)$ in $\mathcal{O}(n)$ running time because it is simply the highest point in the trapezoid $T_{ijk}^{\text{right}}$. Thus, we conclude that each term $V(i,j,k)$ and $W(i,j,k)$ can be computed in $\mathcal{O}(n)$ running time.

In order to solve our problem, we find it necessary to augment the graph $G$ of allowable edges by inserting points $r_0 = (-M, 0)$ and $r_{n+1} = (M, 0)$ and points $b_0 = (-M, R)$ and $b_{n+1} = (M, R)$, where $M \gg \max_i |w_i|$ is a large number and $R = \max_j\{v_j\} + \epsilon$ is slightly larger than the maximum height of our blue points. It is obvious that the minimum-cost ENCBM is the same as the minimum-cost ENCBM of the trapezoid $T_{0,n+1,0}^{\text{right}}$, subject to the constraint that $r_0$ is paired with $b_0$ and that $r_{n+1}$ is paired with $b_{n+1}$, precisely as defined earlier. We require $\mathcal{O}(n) \cdot \mathcal{O}(n^3) = \mathcal{O}(n^4)$ iterations in total to compute $V(0, n+1, 0)$, and therefore to find the ENCBM of minimum size. See Figure 15 for an example of such a matching. As in Section 3, the algorithm we have just described does not explicitly use the fact that edge weights are Euclidean; the procedure as described can be used to find a non-crossing perfect matching of *any* graph whose vertices are configured as described herein.

## References


[1] Putnam, 1979. Problem A4.

[2] A.Karim Abu-Affash, Paz Carmi, MatthewJ. Katz, and Yohai Trabelsi. Bottleneck non-crossing matching in the plane. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA 2012*, volume 7501 of *Lecture Notes in Computer Science*, pages 36–47. Springer Berlin Heidelberg, 2012.

[3] Adrian Dumitrescu and William Steiger. On a matching problem in the plane. In *Workshop on Algorithms and Data Structures, 1999 (WADS ́99), and in Discrete Mathematics*, pages 1–3, 2000.

[4] Guy Even, Sudipto Guha, and Baruch Schieber. Improved approximations of crossings in graph drawings and VLSI layout areas. *SIAM J. Comput.*, 32(1):231–252, 2003.

[5] Klaus Jansen and Gerhard J. Woeginger. The complexity of detecting crossingfree configurations in the plane. *BIT Numerical Mathematics*, 33:580–595, 1993. 10.1007/BF01990536.

[6] David Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*

[7] Franz Rendl and Gerhard Woeginger. Reconstructing sets of orthogonal line segments in the plane. *Discrete Mathematics*, 119(1-3):167 – 174, 1993.

[8] Walter Schnyder. Embedding planar graphs on the grid. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, SODA '90, pages 138–148, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[9] Yi Zhu and Andrew Lim. Crane scheduling with spatial constraints: Mathematical model and solving approaches. *NAVAL RESEARCH LOGISTICS*, 51:386–406, 2004.

[10] Yi Zhu and Andrew Lim. Crane scheduling with non-crossing constraint. *Journal of the Operational Research Society*, 57:1464–1471(8), December 2006.