Connecting the Dots (with Minimum Crossings)

Akanksha Agrawal
Hungarian Academy of Sciences (MTA SZTAKI) Budapest, Hungary, akanksha@sztaki.mta.hu

Grzegorz Guśpiel
Jagiellonian University Krakow, Poland, guspiel@tcs.uj.edu.pl

Jayakrishnan Madathil
The Institute of Mathematical Sciences, HBNI, Chennai, India, jayakrishnanm@imsc.res.in

Saket Saurabh
The Institute of Mathematical Sciences, HBNI, Chennai, India, saket@imsc.res.in

Meirav Zehavi
Ben-Gurion University, Beer-Sheva, Israel, meiravze@bgu.ac.il

Abstract

We study a prototype Crossing Minimization problem, defined as follows. Let $\mathcal{F}$ be an infinite family of (possibly vertex-labeled) graphs. Then, given a set P of (possibly labeled) n points in the Euclidean plane, a collection $L \subseteq \mathsf{Lines}(P) = \{\ell : \ell$ is a line segment with both endpoints in $P\}$, and a non-negative integer k, decide if there is a subcollection $L' \subseteq L$ such that the graph $G = (P, L')$ is isomorphic to a graph in $\mathcal{F}$ and L' has at most k crossings. By $G = (P, L')$, we refer to the graph on vertex set P, where two vertices are adjacent if and only if there is a line segment that connects them in L'. Intuitively, in Crossing Minimization, we have a set of locations of interest, and we want to build/draw/exhibit connections between them (where L indicates where it is feasible to have these connections) so that we obtain a structure in $\mathcal{F}$. Natural choices for $\mathcal{F}$ are the collections of perfect matchings, Hamiltonian paths, and graphs that contain an (s,t)-path (a path whose endpoints are labeled). While the objective of seeking a solution with few crossings is of interest from a theoretical point of view, it is also well motivated by a wide range of practical considerations. For example, links/roads (such as highways) may be cheaper to build and faster to traverse, and signals/moving objects would collide/interrupt each other less often. Further, graphs with fewer crossings are preferred for graphic user interfaces.

As a starting point for a systematic study, we consider a special case of Crossing Minimization. Already for this case, we obtain NP-hardness and W[1]-hardness results, and ETH-based lower bounds. Specifically, suppose that the input also contains a collection D of d non-crossing line segments such that each point in P belongs to exactly one line in D, and L does not contain line segments between points on the same line in D. Clearly, Crossing Minimization is the case where $d = n$—then, P is in general position. The case of $d = 2$ is of interest not only because it is the most restricted non-trivial case, but also since it corresponds to a class of graphs that has been well studied—specifically, it is Crossing Minimization where $G = (P, L)$ is a (bipartite) graph with a so called *two-layer drawing*. For $d = 2$, we consider three basic choices of $\mathcal{F}$. For perfect matchings, we show *(i)* NP-hardness with an ETH-based lower bound, *(ii)* solvability in subexponential parameterized time, and *(iii)* existence of an $\mathcal{O}(k^2)$-vertex kernel. Second, for Hamiltonian paths, we show *(i)* solvability in subexponential parameterized time, and *(ii)* existence of an $\mathcal{O}(k^2)$-vertex kernel. Lastly, for graphs that contain an (s,t)-path, we show *(i)* NP-hardness and W[1]-hardness, and *(ii)* membership in XP.

2012 ACM Subject Classification F.2.2 Geometrical problems and computations

Keywords and phrases crossing, parameterized complexity, FPT algorithm, W[1]-hardness

Lines 1924

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:56
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Let $\mathcal{F}$ be an infinite family of (possibly vertex-labeled) graphs. Suppose that given a graph F, the membership of F in $\mathcal{F}$ is testable in time polynomial in the size of F. For the family $\mathcal{F}$, we define a prototype Crossing Minimization problem as follows (see Fig. 1). Given a set P of (possibly labeled) n points in the two-dimensional Euclidean plane, a collection $L \subseteq \mathsf{Lines}(P) = \{\ell : \ell \text{ is a line segment with both endpoints in } P\}$, and a non-negative integer k, decide if there exists a subcollection $L' \subseteq L$ such that the graph $G = (P, L')$ is isomorphic[1] to a graph in $\mathcal{F}$ and L' has at most k crossings. The notation $G = (P, L')$ refers to the graph on vertex set P, where two vertices are adjacent if and only if there is a line segment that connects them in L'. Moreover, the number of crossings of L' is the number of pairs of line segments in L' that intersect each other at a point other than their possible common endpoint. The Crossing Minimization problem is a general model for a wide range of scenarios where we have a set of points of interest that correspond to geographical areas or fixed objects such as cities, manufacturing machinery or immobile equipment, attractions and mailboxes, and we want to build, draw or exhibit connections between them (where L indicates where it is feasible to have these connections) in order to obtain a structure in $\mathcal{F}$.

While the objective of seeking a solution with few crossings is of interest from a theoretical viewpoint, it is also well motivated by practical considerations. For example, public tracks (such as roads, highways or even paths in amusement parks) with fewer crossings require the construction of less bridges, elevated tracks, traffic lights and roundabouts, and therefore they are likely to be cheaper to build [48], easier and faster to traverse [12], and cause less accidents [23]. Moreover, signals and moving objects would interrupt each other less often. This property may be crucial as frequent collision between signals can distort or weaken them [4]. Furthermore, for moving objects such as robots (cleaning robots, autonomous agents and self-driving cars) that cannot physically be present in an intersection point simultaneously, encountering a large number of crossings may require the development of more complex navigation and sensory systems [41]. Lastly, graphs with fewer crossings are easier to view and analyze—in graphic user interfaces, for example, visual clarity is a major issue [15].

Keeping the above applications in mind, three natural choices for the family $\mathcal{F}$ are the family of (Hamiltonian) paths, the family of graphs that contain an (s,t)-path (identification of s and t is modeled by vertex labels), and the family of (possibly vertex-labeled) perfect matchings. Indeed, these families model the most basic scenarios where all points must be connected by a path (e.g., to plan tracks for sightseeing trains or maintenance equipment such as cleaning robots or lawn mowers), only a specific pair of points must be connected by a path (e.g., to transport goods between two destinations), or the points are to be matched with one another (e.g., to pair up robots and charging ports). Furthermore, the computational problems that correspond to these families—Hamiltonian Path, (s,t)-Path and Perfect Matching, respectively—are among the most classical problems in computer science [24, 31, 21, 13].

As a starting point for a systematic study, we consider a special case of Crossing Minimization. Already for this case, we obtain NP-hardness and W[1]-hardness results, and ETH-based lower bounds, alongside positive results. Specifically, suppose that the input also contains a collection D of d non-crossing line segments such that each point in P belongs to exactly one line in D, and L does not contain line segments between points on the same line

[1] With respect to vertex-labeled graphs, isomorphism also preserves the labeling of vertices rather than only their adjacency relationships—that is, a vertex labeled i can only be mapped to a vertex labeled i.

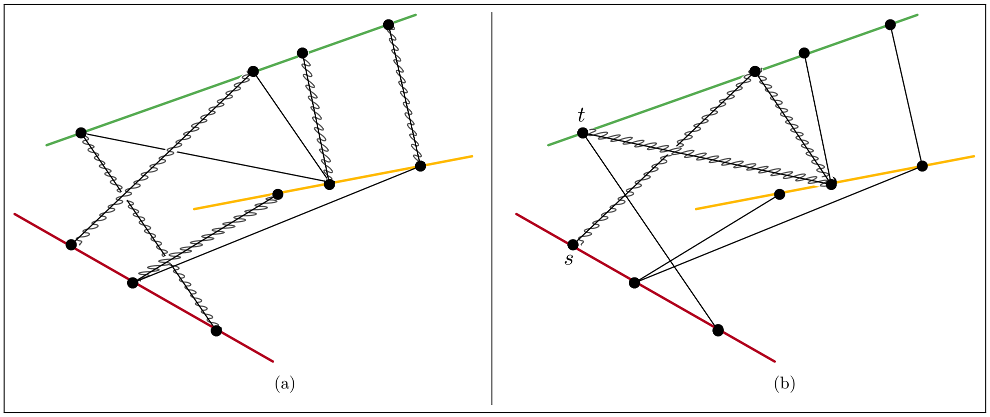


Figure 1 An instance of CROSSING MINIMIZATION (in black) where $\mathcal{F}$ is the family of **(a)** perfect matchings, and **(b)** graphs that have an (s,t)-path. Solution edges are marked by squiggly lines—the number of crossings is 2 in **(a)** and 1 in **(b)**. The $d = 3$ colorful line segments display D.

in D (see Fig. 1).[2] Clearly, CROSSING MINIMIZATION is the case where $d = n$—then, the set P can be in general position. The case of $d = 2$ is of interest not only because it is the most restricted non-trivial case, but also since it corresponds to a class of graphs that has been well studied in the literature—specifically, this case is precisely CROSSING MINIMIZATION where $G = (P, L)$ is a (bipartite) graph with a so called *two-layer drawing*. Clearly, our hardness results carry over to any generalization of the case where $d = 2$. For this case, we consider the aforementioned three basic choices of $\mathcal{F}$, and obtain a comprehensive picture of their complexity. In what follows, we discuss our contribution, and then review related literature.

1.1 Our Contribution

Our study focuses on the class of two-layered graphs. Formally, a *two-layered graph* is a bipartite graph G with vertex bipartition $V(G) = X \cup Y$ that has a *two-layer drawing*—that is, a placement of the vertices of X on distinct points on a straight line segment L_1, and the vertices of Y on distinct points on a different (non-intersecting) straight line segment L_2. The relative positions of the vertices in X and Y on L_1 and L_2, respectively, are given by permutations σ_X and σ_Y. Each edge is drawn using a straight line segment connecting the points of its end-vertices. We refer to (σ_X, σ_Y) as the *two-layered embedding/drawing* of G. Note that (σ_X, σ_Y) uniquely determines which edges intersect. The crossing minimization problem that corresponds to PERFECT MATCHING on two-layered graphs is defined as follows.

> CROSSING-MINIMIZING PERFECT MATCHING (CM-PM) **Parameter:** k
> **Input:** A two-layered graph G (i.e., a bipartite graph G with bipartition $V(G) = X \cup Y$, and orderings σ_X and σ_Y of X and Y, respectively), and a non-negative integer k.
> **Question:** Does G have a perfect matching with at most k crossings?

Similarly, we define the crossing minimization variants of HAMILTONIAN PATH (the existence of a path that visits all vertices)[3] and (s,t)-PATH (the existence of a path between

[2] Having lines segments between points on the same line in D only makes the problem more general.
[3] We remark that our results for HAMILTONIAN PATH extend to HAMILTONIAN CYCLE.

two designated vertices). We refer to these problems as Crossing-Minimizing Hamiltonian Path (CM-HP) and Crossing-Minimization (s,t)-Path (CM-Path), respectively.

Our Results. In this paper, we present a comprehensive picture of both the classical and parameterized computational complexities of these three problems as follows. (Definitions of standard notions in Parameterized Complexity can be found in Section 2.)

CM-PM.
- **Negative.** NP-complete *even on graphs of maximum degree* 2. Moreover, unless the ETH fails, it can be solved neither in time $2^{o(n+m)}$ nor in time $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ on these graphs.
- **Positive.** Admits a kernel with $\mathcal{O}(k^2)$ vertices. Moreover, it admits a subexponential parameterized algorithm with running time $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$. In light of the negative result above, the running time of this algorithm is *optimal*.

We briefly remark that the proof of NP-completeness of CM-PM resolves an open question related to a problem called Token Swapping (see Section 1.2), introduced in 2014 by Yamanaka et al. [54, 1]. Two generalizations of Token Swapping were introduced by Yamanaka et al. [54, 1] and Bonnet et al. [8], both known to be NP-complete due to Miltzow et al. [44]. One of the results of Bonnet et al. [8] is the analysis of the complexity of all three token swapping problems on simple graph classes, including trees, cliques, stars and paths. Subset Token Swapping was shown to be NP-complete on the first three classes, but the status of the problem for paths was unknown. Since Subset Token Swapping restricted to paths is equivalent to our CM-PM (noted by Miltzow [43]), we derive that Subset Token Swapping restricted to paths is NP-complete as well.

CM-HP.
- **Negative.** NP-complete even on graphs that admit a Hamiltonian path. Moreover, unless the ETH fails, it can be solved neither in time $2^{o(n+m)}$ nor in time $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ on these graphs.
- **Positive.** Admits a kernel with $\mathcal{O}(k^2)$ vertices. Moreover, it admits a subexponential parameterized algorithm with running time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$. In light of the negative result above, the running time of this algorithm is *almost optimal*.

While Hamiltonian Path is a classical NP-complete problem [24], we prove that in the case of CM-HP, the hardness holds even if we know of a Hamiltonian path in the input graph (in which case Hamiltonian Path is trivial). We also comment that in the case of CM-HP (and also CM-Path), unlike the case of CM-PM, the problem becomes trivially solvable in polynomial time on graphs of maximum degree 2. Indeed, graphs of maximum degree 2 are collections of paths and cycles, and hence admit only linearly in n many Hamiltonian paths that can be easily enumerated in polynomial time. Then, CM-HP is solved by testing whether at least one of these Hamiltonian paths has at most k crossings. In fact, most natural NP-complete graph problems become solvable in polynomial time on graphs of maximum degree 2, therefore we find the hardness of CM-PM on these graphs quite surprising.

CM-Path.

- **Negative.** NP-complete and W[1]-hard. Specifically, unless W[1] = FPT, it admits neither an algorithm with running time $f(k)n^{\mathcal{O}(1)}$ nor a kernel of size $f(k)$, *for any computable function f of k.*
- **Positive.** Member in XP. Specifically, it is solvable in time $n^{\mathcal{O}(k)}$.

In light of our first two sets of results, we find our third set of results quite surprising: (s,t)-Path is the easiest to solve among itself, Perfect Matching and Hamiltonian Path,[4] yet when crossing minimization is involved, (s,t)-Path is substantially more difficult than the other two problems—indeed, CM-PM is not even FPT (unless W[1] = FPT).

Our Methods. In what follows, we give a brief overview of our methods.

CM-PM. We prove that CM-PM on graphs of maximum degree 2 is NP-hard by a reduction from Vertex Cover. The same reduction shows that CM-PM does not admit any $2^{o(n+m)}$-time (or $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$-time) algorithm unless the ETH fails.

For our algorithm and kernel, consider an instance (G,k) of CM-PM, where $V(G) = X \cup Y$ is the vertex bipartition with $|X| = |Y| = n$. For $i \in [n]$, let x_i and y_i denote the i^{th} vertices of X and Y, respectively, in the given two-layered embedding of G. It is not difficult to see that the only perfect matching with no crossings, if such a matching exists, is $\{x_iy_i \mid i \in [n]\}$. Therefore, if M is a perfect matching and $x_iy_j \in M$ with $i \neq j$, then the edge x_iy_i must intersect another edge in M, which yields a crossing. In fact, x_iy_j must intersect at least $|j-i|$ edges. Therefore, no feasible solution to CM-PM can contain an edge x_iy_j with $|j-i| > k$. This observation plays a key role in both our algorithm and kernel designs. Our algorithm is based on dynamic programming, and its analysis is based on Hardy-Ramanujan numbers [28]. (By considering these numbers, we are able to derive a running time bound of $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$.) Very briefly, at stage i we consider the graph G_i, the subgraph of G induced by $X_i \cup Y_i = \{x_j, y_j \mid j \leq i\}$. Our algorithm "guesses" which subsets of $V(G_i)$ are going to be matched to "future vertices", i.e., vertices in $V(G) \setminus V(G_i)$ in an optimal solution, and solves the problem optimally on the graph induced by the remaining vertices. For the kernel, we show that either (G,k) is a no-instance or the number of "bad pairs", i.e., $\{x_i, y_i\}$ where $x_iy_i \notin E(G)$, cannot exceed $2k$. We then bound the number of pairs $\{x_i, y_i\}$ between two consecutive bad pairs by $\mathcal{O}(k)$ again, which gives a kernel with $\mathcal{O}(k^2)$ vertices.

CM-HP. By a reduction from a variant of Hamiltonian Path on bipartite graphs, we show that CM-HP is NP-hard even if the input graph is assumed to have a Hamiltonian path. For our FPT algorithm and kernel, we adopt a strategy similar to the one we employed for CM-PM. For the algorithm, we guess which subsets of G_i have a neighbor in the future, and proceed accordingly. As for the kernel, we identify a set of bad structures—namely, configurations of vertices and edges that result in crossings in any Hamiltonian path in G. We show that both the number and the size of bad structures cannot exceed $\mathcal{O}(k)$. Then we bound the number of vertices between two consecutive bad structures by $\mathcal{O}(k)$ as well, which gives a kernel with $\mathcal{O}(k^2)$ vertices.

CM-PATH. We prove the W[1]-hardness of CM-Path by giving an appropriate reduction from Multi-Colored Clique, which is known to be W[1]-hard [22]. Given an instance $(G, V_1, V_2, \ldots, V_k)$ of Multi-Colored Clique (G is a k-partite graph, and the problem

[4] In particular, (s,t)-Path can be directly solved in linear time via BFS [13], while Perfect Matching is only known to be solvable by more complex (non-linear time) algorithms such as Edmonds algorithm [21], and the status of Hamiltonian Path is even worse given that it is NP-complete [24].

is to check whether G contains a clique with exactly one vertex from each V_i), we create an equivalent instance (G', X, Y, s, t, k') of CM-Path, where G' is a two-layered graph, as follows. We create an s-t path in G' that "selects" a vertex from each V_i and an edge for each (distinct) pair (V_i, V_j). To this end, for each V_i, we have a vertex selection gadget $\mathcal{V}_i$, and for (distinct) V_i, V_j, we have an edge selection gadget $\mathcal{E}_{ij}$. The vertex and edge selection gadgets are arranged in a linear fashion to create an $s - t$ path in G'. In the construction, we add a pair of non-adjacent vertices in $\mathcal{E}_{ij}$ for each edge between V_i and V_j. We also add a path between the pair of (non-adjacent) vertices whose edges cross the gadgets $\mathcal{V}_i$ and $\mathcal{V}_j$, which enforces compatibility between vertices and edges that are selected. Finally, by setting k' appropriately, we get the desired reduction.

As for the XP algorithm for CM-Path, we guess which edges of G are going to be involved in crossings in a feasible solution. The problem then reduces to connecting these guessed edges using crossing-free subpaths, which can be done in polynomial time.

1.2 Related Works

The Crossing Number Problem. The *crossing number* of a graph G is the minimum number of crossings in a plane drawing of G. The notion of a crossing number originally arose in 1940 by Turán [52] for bipartite graphs in the context of the minimization of the number of crossings between tracks connecting brick kilns to storage sites. Computationally, the input of the Crossing Number problem is a graph G and a non-negative integer k, and the task is to decide whether the crossing number of G is at most k. This problem is among the most classical and fundamental graph layout problems in computer science. It was shown to be NP-complete by Garey and Johnson in 1983 [25]. Not only is the problem NP-complete on graphs of maximum degree 3 [29], but also it is surprisingly NP-complete even on graphs that can be made planar and hence crossing-free by the removal of just a single edge [9]. Nevertheless, Crossing Number was shown to be FPT by Grohe already in 2001 [26], who developed an algorithm that runs in time $f(k)n^2$ where f is at least double exponential.[5] A further development was achieved by Kawarabayashi and Reed [36], who showed that the problem is solvable in time $f(k)n$. On the negative side, Hlinený and Dernár [30] proved that Crossing Number does not admit a polynomial kernel unless NP $\subseteq$coNP/poly.

Variants of Crossing Number where the vertices can be placed only on prespecified curves are extensively studied. Closely related to our work is the well-known Two-Layer Crossing Minimization problem: given a bipartite graph G with vertex bipartition $V(G) = X \cup Y$, and a non-negative integer k, the task is to decide whether G admits a two-layered drawing where the number of crossings is at most k. This problem originated in VLSI design [50]. A solution to the Two-Layer Crossing Minimization problem is also useful in solving the rank aggregation problem, which has applications in meta-search and spam reduction on the Web [7]. We refer the reader to [55] and references therein for other applications. The Two-Layer Crossing Minimization problem is long known to be NP-complete, even in its one sided version where we are allowed to permute vertices only from one (fixed) side [19, 20]. Further, the membership of Two-Layer Crossing Minimization in FPT has already been proven close to two decades ago by Dujmovic et al. [18]. Noteworthy is also the well-studied variant of Crossing Number that restricts the vertices to be placed only on a prespecified circle and edges are drawn as straight line

[5] We find the contrast between this result and our result on CM-Path somewhat surprising. At first glance, our CM-Path problem seems computationally simpler than Crossing Number (where the embedding is computed from scratch), yet our problem is W[1]-hard while Crossing Number is FPT.

segments. Both of these variants as well as their various versions are subject to an active line of research [37]. Further, aesthetic display of these layouts are of importance in biology [40], and included in standard graph layout software [35] such as yFiles, Graphviz, or OGDF. For more information on Crossing Number and its variants, we refer to surveys such as [49].

Problems on Fixed Point Sets. Settings where we are given a set of points P in the plane that represent vertices, and edges are to be drawn as straight lines between them, are intensively studied since the early 80s. A large body of work has been devoted to the establishment of combinatorial bounds on the number of *crossing-free* graphs on P, where particular attention is given to crossing-free triangulations, perfect matchings and Hamiltonian paths and cycles. Originally, the study of these bounds was initiated Newborn and Moser in 1980 [47] for crossing-free Hamiltonian cycles. For more information, we refer to the excellent Introduction of Sharir and Welz [51] and the references therein. Computationally, the problem of *counting* the number of such crossing-free graphs (faster than the time required to enumerate them) is of great interest (see, e.g., [53, 5, 42]). Furthermore, the computation of a single crossing-free graph on P (such as a perfect matching), possibly with a special property of being "short" [3, 2, 11], has already been studied since 1993 [34]. To the best of our knowledge, the minimization of the number of crossings (rather than the detection of a crossing-free graph) has received only little attention, mostly in an ad-hoc fashion. An exception to this is the work of Halldórsson et al. [27] with respect to spanning trees. We remark that they study the problem in its full generality, where the computation of even a crossing-free spanning tree is already NP-complete [38, 34].

Related to our study is also the Metro Line Crossing Minimization problem, introduced by Benkert et al. [6] in 2007. Given an embedded graph G on P, as well as k pairs of vertices (called terminals), a solution to this problem is a set of paths that connect their respective pairs of terminals, and which has minimum number of "crossings" under a definition different than ours. Specifically, paths are thought of as being drawn in the plane "alongside" the edges of G rather than on the edges themselves. Such a formulation allows to reuse a single edge a large number of times. Therefore, the avoidance of crossings might come at the cost of congesting the same tracks by buses and trains (or building many parallel tracks). Finally, we mention the Token Swapping problem, where we are given a graph with a token placed on each vertex, and each token has a unique target vertex. The objective is to move the tokens with minimum number of swaps so that each token is placed on its target vertex. We remind that this problem was discussed in Section 1.1. Although it seems unrelated to our study, recall that a variant of it is equivalent to CM-PM [43].

2 Preliminaries

Sets and functions. We use $\mathbb{N}$ to denote the set $\{0, 1, 2, \ldots\}$. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2, 3, \ldots, n\}$, and $[n]_0 = [n] \cup \{0\}$. We define $[0] = \emptyset$. For a set A, 2^A denotes the power set of A. For sets A, B, $A' \subseteq A$ and a function $f : A \to B$, $f|_{A'}$ denotes the restriction of f to A'. That is, $f|_{A'}$ is the function from A' to B, defined as $f|_{A'}(x) = f(x)$ for every $x \in A'$.

Graphs. All graphs in this paper are simple and undirected. For a graph G, $V(G)$ and $E(G)$, respectively, denote the vertex set and edge set of G. For an edge $e = uv$, the vertices u and v are called the endpoints of e. For a set $E' \subseteq E(G)$, $V(E')$ denotes the set of endpoints of edges in E'. A set of edges $M \subseteq E(G)$ is said to be a *matching* in G if for every pair of distinct edges $e, e' \in M$, $V(\{e\}) \cap V(\{e'\}) = \emptyset$. A matching $M \subseteq E(G)$ is said to *saturate* a vertex $v \in V(G)$ if $v \in V(M)$. Moreover, M is said to saturate a set of vertices $V' \subseteq V(G)$ if

$V' \subseteq V(M)$. A matching M in G is said to be a *perfect matching* if M saturates the entire vertex set $V(G)$. An ordered sequence P of distinct vertices $v_1 v_2 \dots v_r$ is said to be a *path* in G if $v_i v_{i+1} \in E(G)$ for every $i \in [r-1]$. We refer to vertices v_1 and v_r as the end vertices or terminal vertices of the path P, and vertices $v_2, v_3, \dots, v_{r-1}$ as the internal vertices of the path P. For every $i \in [r]$, we say that the path P visits (or passes through) the vertex v_i. A path in G is called a *Hamiltonian path* if it visits every vertex of G. Terminology and notation not defined here can be found in the book of Diestel [16].

Two-layered graphs. Consider a two-layered graph G. Whenever the context is clear, we denote the vertex bipartition of G (given by the two-layer drawing) by X and Y. We use n_X and n_Y to denote $|X|$ and $|Y|$, respectively. For $i \in [n_X]$, we let x_i be the ith vertex of X and for $j \in [n_Y]$, we let y_j be the jth vertex of Y, in the two-layered drawing of G. Also, we say that i is the index of the vertex x_i and j is the index of the vertex y_j. We write $\texttt{index}(x_i) = i$ and $\texttt{index}(y_j) = j$. Similarly, we let X_i denote the set $\{x_r \mid 1 \leq r \leq i\}$, and we let Y_j denote the set $\{y_r \mid 1 \leq r \leq j\}$. For $i, j \in [n_X]$, where $i \leq j$, the set $X_{i,j}$ denotes the set $\{x_p \mid i \leq p \leq j\}$. Moreover, if $i < j$, then the set $X_{j,i} = \emptyset$. (Note that $X_{i,j}$ is not the same as $X_{j,i}$, unless $i = j$.) The set $Y_{i,j}$ is defined analogously for $i, j \in [n_Y]$. A *crossing* in G is a pair of edges that intersect at a point other than their possible common endpoints. Note that two edges $x_i y_j$ and $x_r y_s$, where $i, r \in [n_X]$ and $j, s \in [n_Y]$, form a crossing (or, cross each other) if and only if $i \leq j, r > i, j > s$ or $r \leq s, i > r, s > j$. We say that an edge $e \in E(G)$ participates in a crossing if there is another edge $e' \in E(G)$ such that e and e' cross each other. Similarly, we say that a vertex $v \in V(G)$ participates in a crossing if v is an endpoint of an edge that participates in a crossing. For a subgraph H of G, $\texttt{cr}(H)$ denotes the number of crossings in H. Similarly, for a set of edges $E' \subseteq E(G)$, $\texttt{cr}(E')$ denotes the number of crossings in the subgraph induced by E'.

Parameterized Complexity. In the framework of parameterized complexity, each problem instance is associated with a non-negative integer k, called a *parameter*. A problem is said to be *fixed-parameter tractable* (FPT) if it admits an algorithm with running time $f(k)n^{\mathcal{O}(1)}$ time for some computable function f, where n is the input size. Moreover, if the problem is solvable in time $n^{g(k)}$, then it is said to admit an XP algorithm. A companion notion of fixed-parameter tractability is that of *kernelization*. A *kernelization algorithm* is a polynomial-time algorithm that transforms an arbitrary instance of the problem to an equivalent instance of the same problem whose size is bounded by some computable function g of the parameter of the original instance. The resulting instance is called a *kernel*, and if g is a polynomial function, then it is called a *polynomial kernel*, and we say that the problem admits a polynomial kernel. Parameterized complexity provides a theory of intractability as well, which enables us to show that certain problems are unlikely to be fixed-parameter tractable. This is done by giving an appropriate reduction from a so called W-hard problem.

To obtain (essentially) tight conditional lower bounds for the running time of FPT or XP algorithms, we rely on the well-known *Exponential-Time Hypothesis (*ETH*)* [32, 33, 10]. To formalize the statement of ETH, recall that given a formula φ in conjunctive normal form (CNF) with n variables and m clauses, the task of CNF-SAT is to decide whether there is a truth assignment to the variables that satisfies φ. In the p-CNF-SAT problem, each clause is restricted to have at most p literals. ETH states that 3-CNF-SAT cannot be solved in time $2^{o(n)}$. Additional details on parameterized complexity and ETH can be found in [14, 17].

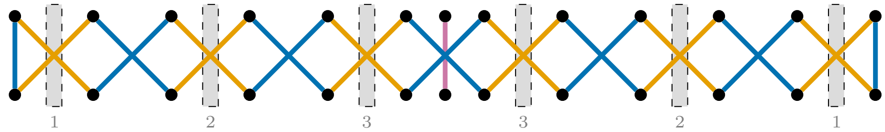


Figure 2 The vertex gadget of size 3.

3 NP-hardness, FPT Algorithm and Polynomial Kernel for Crossing-Minimizing Perfect Matching

In this section, we show that CM-PM is NP-hard, but can be solved in time $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ using an algorithm based on dynamic programming. We also design an $\mathcal{O}(k^2)$ vertex kernel for CM-PM. The problem is formally defined as follows.

> Crossing-Minimizing Perfect Matching (CM-PM) **Parameter:** k
> **Input:** A two-layered graph G and a non-negative integer k.
> **Question:** Does G have a perfect matching with at most k crossings?

3.1 NP-hardness for CM-PM

We show that CM-PM is NP-hard, even if the maximum degree of the input graph is 2. Our proof of NP-hardness is a polynomial-time reduction from Vertex Cover. The problem Vertex Cover takes as input a graph G and an integer k, and the objective is to check if there is $S \subseteq V(G)$ of size k, such that $G - S$ has no edges (in other words, S is a vertex cover in G). Vertex Cover is known to be NP-hard from [39].

▶ **Theorem 1.** CM-PM *is NP-hard, even if the maximum degree of the input graph is 2.*

Proof. We give a reduction from the Vertex Cover problem. Let (G, k) be an instance of Vertex Cover. In polynomial time, we will create an (equivalent) instance (H, m) of CM-PM. Our construction will be based on two gadgets. The first one is created for every vertex of G. For every integer $s \geqslant 1$, the *vertex gadget of size s* is a cycle on $8s$ vertices together with a path on 2 vertices, positioned as shown in Figure 2. We distinguish special areas in the vertex gadget, in which we put other elements of our construction. These areas are called *slots* and are marked with gray rectangles. We also number them as in the figure. The ones to the left of the purple edge are called *left slots* and the ones to the right are called *right slots*. The vertex gadget of size s has s left slots and s right slots. Furthermore, observe that there are only two ways to choose a perfect matching in this gadget: either take the blue edges and the purple edge in the middle, or the yellow edges and the purple one. Choosing the blue (the yellow) matching is interpreted as selecting (not selecting) the vertex in the vertex cover and we say that the gadget is 'selected' ('not selected').

The second gadget, the *edge gadget*, is created for every edge of G. It is shown in Figure 3.

The construction proceeds as follows. First, for every $v \in V(G)$, we create a copy of the vertex gadget of size $2d(v)$. We place them on the two horizontal lines in such a way that each gadget occupies a separate range of the x axis, in any order. Now, for every edge $uv \in E(G)$, where the gadget of u is to the left of the gadget of v, we select two consecutive right slots in the gadget of u and two consecutive left slots in the gadget of v, create a copy of the edge gadget and place its vertices as follows:

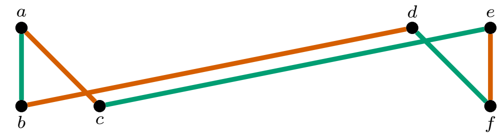


Figure 3 The edge gadget.

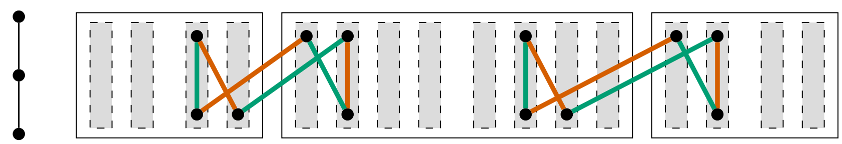


Figure 4 A graph G and a possible bipartite graph obtained by passing G to the reduction algorithm, vertex gadgets presented schematically.

- vertices a and b in the left selected slot of the gadget of u,
- vertex c in the right selected slot of the gadget of u,
- vertex d in the left selected slot of the gadget of v,
- verticed e and f in the right selected slot of the gadget of v.

Such a selection of consecutive slots for each edge is of course possible, as we set the size of the vertex gadget to be $2d(v)$. See Figure 4 for a complete example. The edge gadget admits exactly two perfect matchings as well and just like previously, we give interpretations to these matchings. If the red (green) matching is selected, we say that the edge gadget is 'covered' at the right (left) side and 'not covered' at the left (right) side. Our naming convention may be confusing, as in the case of vertex covers, an edge may be covered at both sides, and our edge gadgets are always 'not covered' at one side. The property that we want to enforce is as follows: in every optimal solution, when the edge gadget is 'covered' at one side, the corresponding vertex gadget must be 'selected', and when the edge gadget is 'not covered' at this side, the vertex gadget may be either 'selected' or 'not selected'.

Now, we assume that the positions of all the gadgets are fixed and count the number of crossing edges. In our analysis, we are only interested in how this number changes when a different matching is chosen, and for this reason we introduce constants $c_1, c_2, \ldots$ that are dependent on the way the gadgets were assembled on the two horizontal lines, but not on the choice of matching. First, we count such crossings, where an edge of the vertex gadget crosses another edge of the same vertex gadget. As the vertex gadget of size s admits $2s+1$ crossings if 'selected' and $2s$ otherwise, this number is equal to:

$$\#\mathsf{s} + \sum_{v \in V(G)} 2 \cdot 2d(v) = \#\mathsf{s} + c_1,$$

where $\#\mathsf{s}$ is the number of 'selected' vertex gadgets.

The number of crossings between edges of edge gadgets turns out to be independent of the matching chosen and we denote it by c_2. To see this, first observe that the number of crossings inside the edge gadget is always 1. Second, note that for two different copies of the edge gadget, the number of crossings between them is either 0, 1, 2 or 3, but in all cases it is independent of the choice of the matching.

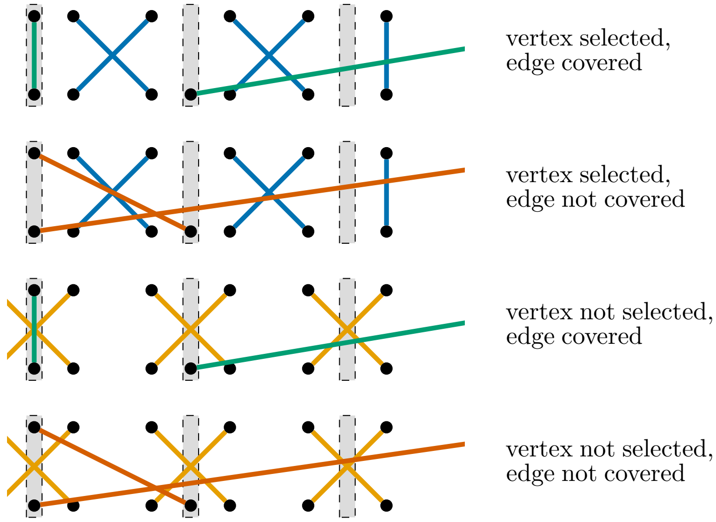


Figure 5 The 4 possible configurations of a crossing of the right part of the vertex gadget and the left part of the edge gadget.

It remains to count the number of crossings such that one edge belongs to the vertex gadget and the other to the edge gadget. Fix $v \in V(G)$ and $e \in E(G)$. We count crossings between edges of the vertex gadget of v and edges of the edge gadget of e. If $v \notin \{u', v'\}$, where $e = u'v'$, then this number is independent of the choice of the matching. Hence, we denote the number of such crossings between every vertex gadget and every edge gadget by c_3. Now assume that $v \in \{u', v'\}$. As the vertex gadget and the edge gadget admit 2 possible perfect matchings each, we have 4 possibilities, as listed in Figure 5. The figure does not lose generality: in the figure, we are considering the right part of the vertex gadget and the left part of the edge gadget, but the analysis is the same in the opposite case. Let $s, s+1$ be the numbers of the two slots in the vertex gadget of v occupied by vertices of the edge gadget of e. The number of crossings between edges of the gadget of v and edges of the gadget of e is equal to:

- $2(s-1) + 1 = 1 + c_4$ in the 'vertex selected, edge covered' case,
- $2(s-1) + 5 = 5 + c_4$ in the 'vertex selected, edge not covered' case,
- $2(s-1) + 3 = 3 + c_4$ in the 'vertex not selected, edge covered' case,
- $2(s-1) + 5 = 5 + c_4$ in the 'vertex not selected, edge not covered' case.

Let the variables $\#\mathsf{sc}, \#\mathsf{snc}, \#\mathsf{nsc}, \#\mathsf{nsnc}$ count occurences of each of the four cases above in the entire graph H, respectively. The total number of crossing edges (allowed) is equal to:

$$\#\mathsf{s} + c_1 + c_2 + c_3 + (1 + c_4)\#\mathsf{sc} + (3 + c_4)\#\mathsf{nsc} + (5 + c_4)\#\mathsf{snc} + (5 + c_4)\#\mathsf{nsnc}.$$

However, as every edge gadget must be 'covered' at one side and must 'not be covered' at the other, we have $\#\mathsf{sc} + \#\mathsf{nsc} = \#\mathsf{snc} + \#\mathsf{nsnc} = |E(G)|$ and hence the calculation simplifies to

$$\#\mathsf{s} + c_1 + c_2 + c_3 + 2 \cdot \#\mathsf{nsc} + (1 + c_4)|E(G)| + (5 + c_4)|E(G)| = \#\mathsf{s} + 2 \cdot \#\mathsf{nsc} + c_5.$$

To complete the description of the reduction algorithm, we set $m = k + c_5$.

It is straightforward to implement the reduction algorithm in polynomial time. It remains to prove that G admits a vertex cover of size k if and only if H admits a perfect matching with at most m crossings.

First suppose that G admits a vertex cover C of size at most k. Then one can choose the 'selected' perfect matching for vertex gadgets of every vertex in C and the 'not selected' perfect matching for every other vertex. Moreover, as every edge of G is covered, one can choose perfect matchings in edge gadgets so that their 'covered' side is in a 'selected' vertex gadget. Then $\#\mathsf{nsc} = 0$ and the number of intersecting edges is equal to $\#\mathsf{s} + c_5 \leqslant k + c_5 = m$, so H admits a perfect matching with at most m crossings.

For the second implication, assume that H admits a perfect matching with at most m crossings. Let M be any matching with minimal number of crossings. Observe that $\#\mathsf{nsc} = 0$, as if there exists a vertex gadget that is 'not selected' and intersects a 'covered' edge gadget, one can choose the vertex gadget to be 'selected' instead, and achieve a perfect matching with fewer crossings, which contradicts the minimality of M. Now we construct a vertex cover of G: we select exactly the vertices whose vertex gadgets were 'selected'. To see that this is a vertex cover, fix an edge of G. At the 'covered' side of its edge gadget, the vertex gadget is 'selected', because $\#\mathsf{nsc} = 0$. Thus, the corresponding vertex is selected to the cover. Finally, as the number of crossings in our construction is equal to $\#\mathsf{s} + c_5$ and is at most m, the size of the vertex cover, equal to $\#\mathsf{s}$, is at most $m - c_5 = k$. ◀

Observe that in the proof above, the size of the CM-PM instance is linear in the size of the VERTEX COVER instance. Indeed, for every vertex $v \in V(G)$ we produce $16d(v) + 2$ vertices of H, and for every edge of G, six vertices are produced. Hence, the number of vertices in the graph H, outputted by the reduction algorithm, is bounded by $\mathcal{O}(|V(G)| + |E(G)|)$. As the vertices in H are of degree at most 2, we have $|E(H)| \in \mathcal{O}(|V(G)| + |E(G)|)$. We note that VERTEX COVER does not admit an algorithm running in time $2^{o(|V(G)|+|E(G)|)}$ (assuming the Exponential Time Hypothesis), Theorem 14.6 in [14]. From the above discussions, we can conclude that CM-PM does not admit an algorithm running in time $2^{o(|V(H)|+|E(H)|)}$.

3.2 FPT Algorithm for CM-PM

Let (G, k) be an instance of CM-PM, with vertex bipartition X and Y, where $|X| = |Y| = n$. (Here, we note that if $|X| \neq |Y|$ then (G, k) is a no-instance as it does not admit a perfect matching.) We will design an FPT algorithm for CM-PM running in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$. Our algorithm will be a dynamic programming algorithm which processes the graph from left to right. That is to say, for each $i = 1, 2, \ldots, n$, at stage i, we consider the graph $G_i = G[X_i \cup Y_i]$, the graph induced by $\{x_1, \ldots, x_i, y_1, \ldots, y_i\}$, and solve a family of subproblems, the solution to one of which will lead to an optimal solution of the entire graph G. We will bound the number of sub-instances that we need to solve at each stage i, for $i \in [n]$, by $2^{\mathcal{O}(\sqrt{k})}$. To achieve the above, we will use the well known bound on partitions of an integer (and in, particular, partitions where all numbers are distinct). (For the integer 6, a partition of it is $1 + 2 + 3$.) We will rely on the fact that for a number t, we can compute all its partitions in time bounded by $2^{\mathcal{O}(\sqrt{t})}$. The above bound will be crucial to achieve the running time of our algorithm.

We first explain the intuition behind our algorithm. Suppose (G, k) is a yes-instance and let M be a perfect matching of G with $\mathtt{cr}(M) \leqslant k$. Fix $i \in [n]$. Consider how M saturates the "future vertices," i.e., vertices in $X_{i+1,n} \cup Y_{i+1,n}$. Consider a future vertex, say x_j for some $j > i$. Using the fact that $\mathtt{cr}(M) \leqslant k$, we will show that M cannot match x_j to a vertex in Y_{i-k}. Therefore, the only vertices in $X_i \cup Y_i$ that can possibly be matched to vertices in the future belong to $X_{i-k+1} \cup Y_{i-k+1}$. In other words, while doing a dynamic programming from left to right, by the time we get to stage i, the intersection of the potential solution with $X_{i-k} \cup Y_{i-k}$ is completely determined. This observation suggests the most obvious strategy: at stage i, "guess" how the solution matches (and saturates) the vertices in $X_{i-k+1,i} \cup Y_{i-k+1,i}$. But this strategy will only lead to an algorithm running in time $k^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$. Observe that since we are only interested in a matching with the least possible number of crossings, we need not look at all possible matchings in $G[X_{i-k+1,i} \cup Y_{i-k+1,i}]$. We only need to look at which subsets of $X_{i-k+1,i}$ and $Y_{i-k+1,i}$ are saturated by M. Thus, from each collection of matchings that saturate the same subset of $X_{i-k+1,i} \cup Y_{i-k+1,i}$, we remember the matching that incurs the least number of crossings. This observation can be used to obtain an algorithm running in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$. To further improve this running time, we show that the number of subsets of $X_{i-k+1,i} \cup Y_{i-k+1,i}$ that are not saturated by the intersection of any potential solution with $X_i \cup Y_i$ cannot exceed $2^{\mathcal{O}(\sqrt{k})}$. (This is where we will use the bound that the number of partitions of an integer t is bounded by $2^{\mathcal{O}(\sqrt{t})}$.) This will lead us to an algorithm with the claimed running time for the problem.

We start by giving some notations and preliminary results that will be helpful in designing our algorithm.

Notations and Preliminary Results

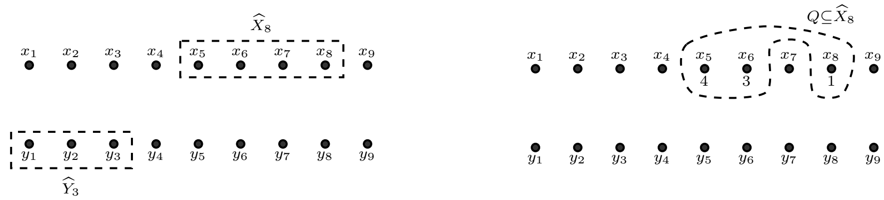
A matching M of G is said to saturate a vertex $v \in V(G)$ if M contains an edge incident on v. Moreover, M is said to saturate a set of vertices $V' \subseteq V(G)$ if M saturates every vertex in V'. We let $\mathsf{Sat}(M) = \{u, v \mid uv \in M\}$. That is, $\mathsf{Sat}(M)$ is the set of vertices saturated by M in G. The analysis of our algorithm requires an important result pertaining to the partitions of an integer. We introduce it below.

Partitions of an integer. For a positive integer α, a partition of α refers to writing α as a sum of positive integers (greater than zero), where the order of the summands is immaterial. And each summand in such a sum is called a *part* of α. For example, take $\alpha = 16$; then $16 = 1 + 4 + 4 + 7$ is a partition of 16. Note that here two of the parts (the two 4s) are the same. We, however, are interested in only those partitions of α in which the parts are all distinct. Let us call such partitions *distinct-part partitions*. As the numbers appearing in a distinct-part partitions of a number are all distinct, we can use the set notation instead. For example, $\{1, 2, 6, 7\}$ is a distinct-part partition of 16. We use letters $\mathtt{P}, \mathtt{P}_1, \mathtt{P}_2$ etc. to denote a partition (in set form) of a number.

▶ **Lemma 2** ([28]). *There exists a constant $c > 0$ such that the number of partitions, and hence the number of distinct-part partitions of any positive integer k, is at most $2^{c\sqrt{k}}$. Moreover, given any positive integer k as input, we can generate all partitions, and hence all distinct-part partitions, of all integers α, where $\alpha \leqslant k$, in time $2^{\mathcal{O}(\sqrt{k})}$.*

Some important sets for the algorithm. For $i \in [n]$, we let $\widehat{X}_i = \{x_{i-k+\ell} \mid \ell \in [k] \text{ and } i - k + \ell \geqslant 1\}$ and $\widehat{Y}_i = \{y_{i-k+\ell} \mid \ell \in [k] \text{ and } i - k + \ell \geqslant 1\}$. We will argue that in any perfect matching M in G with $\mathtt{cr}(M) \leqslant k$, the vertices from X_i which are matched to a vertex y_s, with $s \geqslant i+1$, belong to the set $\widehat{X}_i$. Similarly, we can argue that $\widehat{Y}_i$ is the set of vertices from Y_i which can possibly be matched to vertices x_s, with $s \geqslant i+1$.

We will now associate costs to vertices (and subsets) of $\widehat{X}_i$ (resp. $\widehat{Y}_i$), which will be helpful in obtaining lower bounds on the number of crossings, when vertices from $\widehat{X}_i$ (resp. $\widehat{Y}_i$) are matched to vertices y_s (resp. x_s), where $s \geqslant i+1$. To this end, consider $i \in [n]$ and a vertex



(a) The dashed rectangles show $\widehat{Y}_3$ and $\widehat{X}_8$, when $k = 4$.

(b) $k = 4$. $Q = \{x_5, x_6, x_8\} \subseteq \widehat{X}_8$. The number below each $x_j \in Q$ shows $\mathsf{cst}_8(x_j)$. $\mathsf{CstSet}_8(Q) = \{4, 3, 1\}$ and $\mathsf{cst}_8(Q) = 8$.

Figure 6 An example of $\widehat{X}_i, \widehat{Y}_i, Q \subseteq \widehat{X}_i, \mathsf{CstSet}_i(Q)$ and $\mathsf{cst}_i(Q)$.

$x_r \in \widehat{X}_i$. We let $\mathsf{cst}_i(x_r) = i + 1 - r$. Since $x_r \in \widehat{X}_i$, we have $r \leqslant i$, and thus, $\mathsf{cst}_i(x_r) \geqslant 1$. For a subset $Q \subseteq \widehat{X}_i$, we let $\mathsf{CstSet}_i(Q) = \{\mathsf{cst}_i(x) \mid x \in Q\}$ and $\mathsf{cst}_i(Q) = \sum_{x \in Q} \mathsf{cst}_i(x)$. Similarly, for $i \in [n]$ and a vertex $y_r \in \widehat{Y}_i$, we let $\mathsf{cst}_i(y_r) = i + 1 - r \geqslant 1$. Moreover, for a subset $Q \subseteq \widehat{Y}_i$, we let $\mathsf{CstSet}_i(Q) = \{\mathsf{cst}_i(y) \mid y \in Q\}$ and $\mathsf{cst}_i(Q) = \sum_{y \in Q} \mathsf{cst}_i(y)$. We note that, for each $i \in [n]$, we have $\mathsf{cst}_i(\emptyset) = 0$. In order to understand the intuition behind these definitions, look at the ith stage in our dynamic programming algorithm. At stage i, we consider the graph $G[X_i \cup Y_i]$. Consider the vertices in $\widehat{X}_i$ that are matched to vertices in the future (i.e., vertices y_s where $s > i$). Note that if x_i gets matched to a future vertex, then x_i participates in at least one crossing (in the final solution), and if x_{i-1} gets matched to a future vertex, then x_{i-1} participates in at least two crossings and so on. In particular, $x_r \in \widehat{X}_i$, if matched to a future vertex participates in at least $i + 1 - r$ crossings. So, $\mathsf{cst}_i(x_r)$ is a lower bound on the number of crossings in which x_r participates (or cost incurred by x_r) if it gets matched to a future vertex. For a set $Q \subseteq \widehat{X}_i$, $\mathsf{CstSet}_i(Q)$ is the set of minimum costs incurred by each element of Q. Moreover, $\mathsf{cst}_i(Q)$ is the cost incurred by Q if all its elements get matched to future vertices. Now using the notion of distinct-part partitions of an integer, we introduce some "special" sets of subsets of X and Y, respectively. These sets will be crucially used while creating the sub-instances in our dynamic programming algorithm. For $\alpha \in [k]$, let $\mathcal{P}_\alpha$ be the set of all distinct-part partitions of α. Furthermore, let $\mathcal{P}_{\leqslant k} = \cup_{\alpha \in [k]} \mathcal{P}_\alpha$. From Lemma 2, we have $|\mathcal{P}_{\leqslant k}| = 2^{\mathcal{O}(\sqrt{k})}$. Consider $i \in [n]$, $\alpha \in [k]$, and $\mathtt{P} \in \mathcal{P}_{\leqslant \alpha}$. We let $S^i_X(\mathtt{P}) = \{x_{i+1-\beta} \mid \beta \in \mathtt{P} \text{ and } i + 1 - \beta \geqslant 1\}$. (For example, for $\mathtt{P} = \{1, 2, 6, 7, 8\}$ and $i = 6$, we have $S^i_X(\mathtt{P}) = \{x_6, x_5, x_1\}$.) Note that $S^i_X(\mathtt{P}) \subseteq \widehat{X}_i$, $\mathsf{CstSet}_i(S^i_X(\mathtt{P})) = \mathtt{P}$, and $\mathsf{cst}_i(S^i_X(\mathtt{P})) = \alpha$, where $\mathtt{P}$ is a partition of $\alpha \in [k]$. Similarly, we define $S^i_Y(\mathtt{P}) = \{y_{i+1-\beta} \mid \beta \in \mathtt{P} \text{ and } i + 1 - \beta \geqslant 1\} \subseteq \widehat{Y}_i$. Again, note that $\mathsf{CstSet}_i(S^i_X(\mathtt{P})) = \mathtt{P}$ and $\mathsf{cst}_i(S^i_X(\mathtt{P})) = \alpha$.

We let $\mathcal{S}^i_X = \{S^i_X(\mathtt{P}) \mid \mathtt{P} \in \mathcal{P}_{\leqslant k}\} \cup \{\emptyset\} \subseteq 2^{\widehat{X}_i}$ and $\mathcal{S}^i_Y = \{S^i_Y(\mathtt{P}) \mid \mathtt{P} \in \mathcal{P}_{\leqslant k}\} \cup \{\emptyset\} \subseteq 2^{\widehat{Y}_i}$. Here we add the empty set to the collections to simplify some of our arguments in the later parts of the section.

From Lemma 2, we obtain the following result.

▶ **Lemma 3.** *The families $\mathcal{S}^i_X$ and $\mathcal{S}^i_Y$ contain at most $|\mathcal{P}_{\leqslant k}| + 1 = 2^{\mathcal{O}(\sqrt{k})}$ sets each. Moreover, for each $i \in [n]$, the families $\mathcal{S}^i_X$ and $\mathcal{S}^i_Y$ can be generated in $2^{\mathcal{O}(\sqrt{k})}$ time.*

We will now associate a set of integers to every pair $(S, S') \in \mathcal{S}^i_X \times \mathcal{S}^i_Y$, for each $i \in [n]$. Intuitively speaking, these sets will give the "allowed" number of crossings for a matching in the graph G_i. Consider $i \in [n]$, $S \in \mathcal{S}^i_X$, and $S' \in \mathcal{S}^i_Y$. We let $\mathsf{Alw}_i(S, S') = \{\ell \in [k]_0 \mid \ell \leqslant$

$k - \max\{\mathsf{cst}_i(S), \mathsf{cst}_i(S')\}\}$.

In what follows, we make a few observations regarding the sets we defined. These observations will be useful in establishing the correctness of our algorithm.

▶ **Observation 4.** *Consider $i \in [n] \setminus \{1\}$. For $S \in \mathcal{S}_X^i$ and $Q \subseteq S \setminus \{x_i\}$, we have $Q \in \mathcal{S}_X^{i-1}$. Similarly, for $S' \in \mathcal{S}_Y^i$ and $Q' \subseteq S' \setminus \{y_i\}$, we have $Q' \in \mathcal{S}_Y^{i-1}$.*

Proof. We will only prove the first statement. (The second statement can be proved by identical arguments.) Note that if $Q = \emptyset$, then by definition, we have $Q \in \mathcal{S}_X^{i-1}$. Otherwise, $Q \subseteq \widehat{X}_{i-1}$ and $Q \neq \emptyset$. Let $I = \mathsf{CstSet}_i(S)$. Note that $|I| = |S| > 0$. As $S \neq \emptyset$ and $S \in \mathcal{S}_X^i$, there is an integer $\alpha \in [k]$ and $\mathtt{P} \in \mathcal{P}_\alpha$, such that $S = S_X^i(\mathtt{P})$. Notice that $I = \mathtt{P}$. Let $I' = \mathsf{CstSet}_i(Q)$. Note that $\emptyset \subset I' \subseteq I$. Thus, there is an integer $1 \leqslant \alpha' \leqslant \alpha$, and a partition $\mathtt{P}' \in \mathcal{P}_\alpha$, such that $I' = \mathtt{P}'$. As $x_i = x_{i+1-1}$ and $x_i \notin Q$, we have $1 \notin I'$. That is, for each $\beta \in I'$, we have $2 \leqslant \beta \leqslant \alpha'$. Let $I' = \{\beta_1, \beta_2, \ldots, \beta_\ell\}$, where $\ell = |I'|$. Furthermore, let $\widehat{I} = \{\beta_1 - 1, \beta_2 - 1, \ldots, \beta_\ell - 1\}$. Note that for $\widehat{\beta} \in \widehat{I}$, we have $1 \leqslant \widehat{\beta} \leqslant \alpha' - 1 \leqslant k$. Thus, $\widehat{I} \in \mathcal{P}_{\leqslant k}$. From the above we have that $Q = S_X^{i-1}(\widehat{I}) = S_X^i(I')$. Thus, we can conclude that $Q \in \mathcal{S}_X^{i-1}$. ◀

▶ **Observation 5.** *Consider $i \in [n] \setminus \{1\}$. For $S \in \mathcal{S}_X^i$ and $Q \subseteq S \setminus \{x_i\}$, we have $\mathsf{cst}_{i-1}(Q) \leqslant \mathsf{cst}_i(S) - |S|$. Similarly, for $S' \in \mathcal{S}_Y^i$ and $Q' \subseteq S' \setminus \{x_i\}$, we have $\mathsf{cst}_{i-1}(Q') \leqslant \mathsf{cst}_i(S') - |S'|$.*

Proof. We will prove the first statement. The proof of the second statement is symmetric. Note that for each $x \in S$, we have $\mathsf{cst}_i(x) \geqslant 1$. From Observation 4, we have $Q \in \mathcal{S}_X^{i-1}$, and thus, $Q \subseteq \widehat{X}_{i-1}$. For a vertex $x_j \in \widehat{X}_{i-1} \cap \widehat{X}_i$, $\mathsf{cst}_i(x_j) = i + 1 - j$ and $\mathsf{cst}_{i-1}(x_j) = i - j$. That is, $\mathsf{cst}_{i-1}(x_j) = \mathsf{cst}_i(x_j) - 1$. Thus, $\mathsf{cst}_{i-1}(Q) = \sum_{x_j \in Q} \mathsf{cst}_{i-1}(x_j) = \sum_{x_j \in Q}(\mathsf{cst}_i(x_j) - 1) = \sum_{x_j \in Q} \mathsf{cst}_i(x_j) - |Q| = \sum_{x_j \in S} \mathsf{cst}_i(x_j) - \sum_{x_j \in S \setminus Q} \mathsf{cst}_i(x_j) - |Q|$. Hence, $\mathsf{cst}_{i-1}(Q) \leqslant \mathsf{cst}_i(S) - |S|$. ◀

▶ **Observation 6.** *Consider $i \in [n]$ and $Q \subseteq \widehat{X}_i$. If $\mathsf{cst}_i(Q) \leqslant k$, then $Q \in \mathcal{S}_X^i$.*

Proof. If $Q = \emptyset$, the by definition, we have $Q \in \mathcal{S}_X^i$. Thus, we assume that $Q \neq \emptyset$. Recall that $\mathsf{cst}_i(Q) = \sum_{x \in Q} \mathsf{cst}_i(x) \leqslant k$ and $\mathsf{cst}_i(x_j) = i + 1 - j \geqslant 1$. Note that $\mathsf{cst}_i(x) \neq \mathsf{cst}_i(x')$ for distinct vertices $x, x' \in Q$. Hence, $\mathsf{CstSet}_i(Q)$ is a distinct-part partition of an integer α, where $\alpha \in [k]$. Therefore, by the definition of $\mathcal{S}_X^i$, $Q \in \mathcal{S}_X^i$. ◀

Next, we prove a few observations regarding matchings in G_i. To this end, we first define the notion of a "compatible matching." Consider $i \in [n]$, $S \in \mathcal{S}_X^i$, and $S' \in \mathcal{S}_Y^i$. We say that a matching M in G_i is (i, S, S')-*compatible* if $S = \widehat{X}_i \setminus \mathsf{Sat}(M)$, $S' = \widehat{Y}_i \setminus \mathsf{Sat}(M)$, and $\mathtt{cr}(M) \leqslant k - \max\{\mathsf{cst}_i(S), \mathsf{cst}_i(S')\}$. Compatible matchings will be helpful in establishing the correctness of our algorithm, in which we will be considering matchings of G_i that saturate exactly $(X_i \cup Y_i) \setminus (S \cup S')$, while incurring at most a certain allowed number of crossings. Suppose at the ith stage of our algorithm, we consider a matching, say M_i, of G_i that does not saturate S. We would like to extend M_i to a matching of G with at most k crossings. That is, at stage i, M_i matches S to future vertices. Therefore, while extending M_i to a matching of the entire graph G, we will incur at least $\mathsf{cst}_i(S)$ more crossings (in addition to $\mathtt{cr}(M_i)$). Therefore, in order to be able to extend M_i to matching of G with at most k crossings, $\mathtt{cr}(M_i)$ cannot exceed $k - \mathsf{cst}_i(S)$. (Note that this is only a necessary condition for extending M_i.) Identical reasoning holds for the set S'. This is the intuition behind compatible matchings.

▶ **Observation 7.** *Consider $i \in [n] \setminus \{1\}$, $S \in \mathcal{S}_X^i$, and $S' \in \mathcal{S}_Y^i$. Let M be an (i, S, S')-compatible matching in G_i. Then, the following holds.*

1. *If $y_i x_j \in M$, where $j < i$, then $x_j \in \widehat{X}_{i-1}$.*
2. *Similarly, if $x_i y_j \in M$, where $j < i$, then $y_j \in \widehat{Y}_{i-1}$.*

Proof. We only prove the first statement, as the proof of the second statement is symmetric. Note first that since M is an (i, S, S')-compatible matching, we have $\mathtt{cr}(M) \leqslant k - \max\{\mathsf{cst}_i(S), \mathsf{cst}_i(S')\}$. In particular, $\mathtt{cr}(M) \leqslant k - \mathsf{cst}_i(S)$.

Towards a contradiction, we assume that $x_j \notin \widehat{X}_{i-1}$, i.e. $j \leqslant i - k - 1$ (recall that $j < i$). Now, consider the set $\widehat{X}_{i-1} \cup \{x_i\}$. Note that $|\widehat{X}_{i-1} \cup \{x_i\}| = k + 1$. Of the $k + 1$ vertices of $\widehat{X}_{i-1} \cup \{x_i\}$, all but $|S|$ many are saturated by M, as $S = \widehat{X}_i \setminus \mathsf{Sat}(M)$. That is, for each vertex $x_r \in (\widehat{X}_{i-1} \cup \{x_i\}) \setminus S$, M matches x_r to some y_s, where $s < i$. This means that M contains $(k + 1) - |S|$ edges of the form $x_r y_s$, where $i - k \leqslant r \leqslant i$ and $s < i$. Each of these $(k + 1) - |S|$ edges crosses the edge $y_i x_j$. Thus, $\mathtt{cr}(M) \geqslant (k + 1) - |S| \geqslant (k + 1) - \mathsf{cst}_i(S)$ (as $\mathsf{cst}_i(S) \geqslant |S|$), which contradicts the fact that $\mathtt{cr}(M) \leqslant k - \mathsf{cst}_i(S)$. ◀

▶ **Observation 8.** *Consider $i \in [n] \setminus \{1\}$, $S \in \mathcal{S}^i_X$, and $S' \in \mathcal{S}^i_Y$. Let M be an (i, S, S')-compatible matching in G_i. If $x_j y_i \in M$, then $x_j y_i$ crosses exactly $|X_{j+1,i} \setminus S|$ edges in M. Similarly, if $x_i y_{j'} \in M$, then $x_i y_{j'}$ crosses exactly $|Y_{j'+1,i} \setminus S'|$ edges in M.*

Proof. We will prove the fist statement. The proof of the second statement is symmetric. Note that since M saturates all vertices of $X_i \setminus S$, every vertex $x_r \in X_{j+1,i} \setminus S$ is matched to some vertex $y_s \in Y_{i-1}$. Each such edge $x_r y_s \in M$ crosses the edge $x_j y_i$. Also, note that no other edge in M crosses $x_j y_i$. Thus, $x_j y_i$ crosses exactly $|X_{j+1,i} \setminus S|$ edges in M. ◀

▶ **Observation 9.** *Consider $i \in [n] \setminus \{1\}$, $S \in \mathcal{S}^i_X$, and $S' \in \mathcal{S}^i_Y$. Let M be an (i, S, S')-compatible matching in G_i. Then, the following holds.*

1. *If $y_i x_j \in M$, where $j < i$, then $(S \setminus \{x_i\}) \cup \{x_j\} \in \mathcal{S}^{i-1}_X$.*
2. *Similarly, if $x_i y_j \in M$, where $j < i$, then $(S' \setminus \{y_i\}) \cup \{y_j\} \in \mathcal{S}^{i-1}_Y$.*

Proof. Let $Q_j = (S \setminus \{x_i\}) \cup \{x_j\}$. From Observation 7, we have $x_j \in \widehat{X}_{i-1}$, i.e. $i - k \leqslant j < i$. Thus, $Q_j \subseteq \widehat{X}_{i-1}$. Consider the case when $Q_j \setminus \{x_j\} = \emptyset$. Notice that $\mathsf{cst}_{i-1}(\{x_j\}) = i - j \leqslant k$. Thus, from Observation 6, we can conclude that $Q_j = \{x_j\} \in \mathcal{S}^{i-1}_X$. Now consider the case when $Q_j \setminus \{x_j\} \neq \emptyset$, and let $Q' = Q_j \setminus \{x_j\}$. We first show that $\mathsf{cst}_{i-1}(Q_j) \leqslant k$. From Observation 5, we have $\mathsf{cst}_{i-1}(Q') \leqslant \mathsf{cst}_i(S) - |S| \leqslant k - |S|$. As M is (i, S, S')-compatible we have $\mathtt{cr}(M) \leqslant k - \mathsf{cst}_i(S)$. Furthermore, as $y_i x_j \in M$, where $j < i$, from Observation 8, we have $|X_{j+1,i} \setminus S| \leqslant \mathtt{cr}(M)$. Thus, we obtain that $\mathsf{cst}_i(S) + |X_{j+1,i} \setminus S| \leqslant k$. Note that $\mathsf{cst}_{i-1}(Q_j) = \mathsf{cst}_{i-1}(Q') + \mathsf{cst}_{i-1}(x_j) \leqslant \mathsf{cst}_i(S) - |S| + i - j = \mathsf{cst}_i(S) + (|X_{j+1,i}| - |S|) \leqslant \mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|$. As $\mathsf{cst}_i(S) + |X_{j+1,i} \setminus S| \leqslant k$, we obtain that $\mathsf{cst}_{i-1}(Q_j) \leqslant k$. The above statement together with Observation 6 implies that $Q_j \in \mathcal{S}^{i-1}_X$. ◀

Dynamic Programming Algorithm for CM-PM

We are now ready to define the states of our dynamic programming table. For each $i \in [n]$, $S \in \mathcal{S}^i_X$ and $S' \in \mathcal{S}^i_Y$ with $|S| = |S'|$, and an integer $\ell \in \mathsf{Alw}_i(S, S') = \{\ell \in [k]_0 \mid \ell \leqslant k - \max\{\mathsf{cst}_i(S), \mathsf{cst}_i(S')\}\}$, we define

$$T[i, S, S', \ell] = \begin{cases} 1, & \text{if there is a matching } M \text{ in } G_i \text{, such that } \mathtt{cr}(M) = \ell \text{ and} \\ & \mathsf{Sat}(M) = (X_i \setminus S) \cup (Y_i \setminus S'), \\ 0, & \text{otherwise.} \end{cases}$$

Observe that (G,k) is a yes-instance of CM-PM if and only if there is $\ell \in [k]_0$, such that $T[n,\emptyset,\emptyset,\ell]=1$. A matching M in G_i is said to *realize* $T[i,S,S',\ell]$, if $\mathtt{cr}(M)=\ell$ and M is (i,S,S')-compatible. In the above we note that $\ell \leqslant k-\max\{\mathsf{cst}_i(S),\mathsf{cst}_i(S')\}$, as $\ell \in \mathsf{Alw}_i(S,S')$. Let us now see how $T[i,S,S',\ell]$ can be computed.

Base Case: We are at our base case when $i=1$. Consider the entry $T[1,S,S',\ell]$. Note that G_1 has $\mathtt{cr}(G_1)=0$. Thus, if $\ell>0$, we have $T[1,S,S',\ell]=0$. Now we consider the case when $\ell=0$. Recall that by definition, we have $|S|=|S'|$. If $S=\{x_1\}$ and $S'=\{y_1\}$, then we should not match any vertex. Thus, we have a matching (which is the empty set) with 0 crossings, and thus, $T[1,S,S',\ell]=1$. Otherwise, we have $S=S'=\emptyset$. Note that the only possible matching in the graph $G[\{x_1,y_1\}]$ is $\{x_1y_1\}$. So, if $x_1y_1 \in E(G)$, then $\{x_1y_1\}$ is a matching with 0 crossings, and hence $T[1,S,S',\ell]=0$. Otherwise, we have $x_1y_1 \notin E(G)$, and hence $T[1,S,S',\ell]=0$.

We now move to our recursive formulae for the computation of the entries of our DP table. We set the value of $T[i,S,S',\ell]$ (recursively) based on the following cases, where $i>1$.

Case 1: $x_i \in S$ and $y_i \in S'$. From Observation 4, we have that $S\setminus\{x_i\} \in \mathcal{S}_X^{i-1}$ and $S'\setminus\{y_i\} \in \mathcal{S}_Y^{i-1}$. Also, from Observation 5 it follows that $\ell \in \mathsf{Alw}_{i-1}(S\setminus\{x_i\},S'\setminus\{y_i\})$. We set $T[i,S,S',\ell]=T[i-1,S\setminus\{x_i\},S'\setminus\{y_i\},\ell]$. In the following lemma, we prove the correctness of computation for Case 1.

▶ **Lemma 10.** *The computation of $T[i,S,S']$ at Case 1 is correct.*

Proof. To establish the proof, we will show that $T[i,S,S',\ell]=1$ if and only if $T[i-1,S\setminus\{x_i\},S'\setminus\{y_i\},\ell]=1$. For the proof of one direction, assume that $T[i,S,S',\ell]=1$. Let M be a matching in G_i that realizes $T[i,S,S',\ell]$. Note that $\mathsf{Sat}(M)=(X_i\setminus S)\cup(Y_i\setminus S')$. As $x_i \in S$ and $y_i \in S'$, we have $x_i,y_i \notin \mathsf{Sat}(M)$. Thus, M is also a matching in G_{i-i}, with $\mathsf{Sat}(M)=X_{i-1}\setminus(S\setminus\{x_i\})\cup Y_{i-1}\setminus(S'\setminus\{y_i\})$ and $\mathtt{cr}(M)=\ell$. Thus, M realizes $T[i-1,S\setminus\{x_i\},S'\setminus\{y_i\},\ell]$.

For the other direction, assume that $T[i-1,S\setminus\{x_i\},S'\setminus\{y_i\},\ell]=1$. Consider a matching M in G_{i-1} that realizes $T[i-1,S\setminus\{x_i\},S'\setminus\{y_i\}]$. Note that M is a matching in G_i with $\mathsf{Sat}(M)=(X_i\setminus S)\cup(Y_i\setminus S')$ and $\mathtt{cr}(M)=\ell$. Thus, M realizes $T[i,S,S',\ell]$, and hence $T[i,S,S',\ell]=1$. ◀

Case 2: $x_i \in S$ and $y_i \notin S'$, or $x_i \notin S$ and $y_i \in S'$. We will only argue for the case when $x_i \in S$ and $y_i \notin S'$. (The other case can be handled symmetrically.) Thus, hereafter we assume that $x_i \in S$ and $y_i \notin S'$. In this case, a matching, say M, which realizes $T[i,S,S',\ell]$, must saturate the vertex y_i and must not saturate the vertex x_i. Thus, M must have an edge x_jy_i, where $j<i$ (here we rely on the fact that y_i cannot be matched to x_i, as $x_i \in S$). As M must satisfy the constraint $\mathtt{cr}(M)=\ell \leqslant k$, we must have $i-k \leqslant j < i$ (see Lemma 18). That is, the vertex to which y_i is matched, must belong to the set $\widehat{X}_{i-1}$. We will construct a set $\mathcal{Q} \subseteq \mathcal{S}_X^{i-1} \subseteq 2^{\widehat{X}_{i-1}}$. This set will be used for creating sub-instances whose values are needed for the computation of $T[i,S,S',\ell]$. Intuitively speaking, each sets in $\mathcal{Q}$ will determine a vertex to which y_i is matched, in the matching that we are seeking for. Note that as y_i must be saturated by any matching that realizes (or complies) with $T[i,S,S',\ell]$, the edge, say $\widehat{x}y_i$ in the matching might intersect other edges of the matching. Therefore, we will have to account for this extra overhead in the number of crossing edges. To count these extra crossings incurred, we will define an "overhead" function.

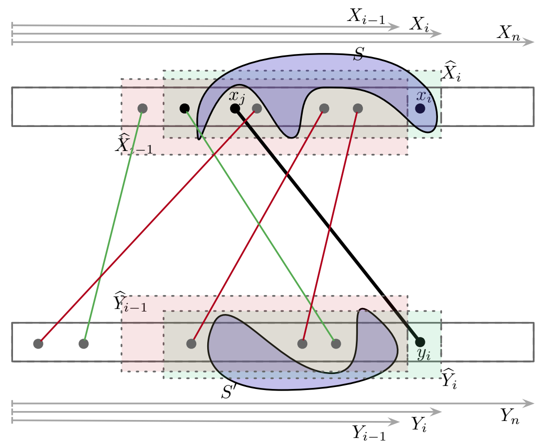


Figure 7 An illustration of the edges intersecting x_jy_i, where $x_j \in \widehat{X}_{i-1} \setminus S$. Here, the red edges intersect x_jy_i and the green edges do not intersect x_jy_i.

To construct $\mathcal{Q}$, we first construct two sets $\widehat{\mathcal{Q}}, \widetilde{\mathcal{Q}} \subseteq 2^{\widehat{X}_{i-1}}$ (each of size at most $\mathcal{O}(k)$). We will obtain $\widehat{\mathcal{Q}} \supseteq \widetilde{\mathcal{Q}} \supseteq \mathcal{Q}$ (in that order, by removing some "bad sets"). For a vertex $x_j \in (N(y_i) \cap \widehat{X}_{i-1}) \setminus S$, let $Q_j = (S \setminus \{x_i\}) \cup \{x_j\}$. Intuitively, the vertex y_i will be matched to x_j, when Q_j is under consideration. Note that $Q_j \subseteq \widehat{X}_{i-1}$. We let $\widehat{\mathcal{Q}} = \{Q_j \mid x_j \in (N(y_i) \cap \widehat{X}_{i-1}) \setminus S\}$. In the above definition, we only consider the neighbors of y_i from $\widehat{X}_{i-1} \setminus S$, because we require that the desired matching must not saturate a vertex from S. We let $\widetilde{\mathcal{Q}} = \widehat{\mathcal{Q}} \cap \mathcal{S}_X^{i-1}$. We now define a function $\mathsf{ovh} : \widetilde{\mathcal{Q}} \to \mathbb{N}$ (see Figure 7 for an intuitive illustration). For $Q_j \in \widetilde{\mathcal{Q}}$, we set $\mathsf{ovh}(Q_j) = |X_{j+1,i} \setminus S|$. To obtain $\mathcal{Q}$, we will delete those sets from $\widetilde{\mathcal{Q}}$ which will incur an "overhead" of crossings more than the "allowed" budget. Before constructing $\mathcal{Q}$, we first recall the following facts. By the definition of $\widetilde{\mathcal{Q}}$, we have $Q \in \mathcal{S}_X^{i-1}$. Moreover, from Observation 4 it follows that $S' \in \mathcal{S}_Y^{i-1}$ (as $y_i \notin S'$). We set $\mathcal{Q} = \{Q \in \widetilde{\mathcal{Q}} \mid \ell - \mathsf{ovh}(Q) \in \mathsf{Alw}_{i-1}(Q, S')\}$.

Now we set $T[i, S, S', \ell]$ as follows.

$$T[i, S, S', \ell] = \begin{cases} 0, \text{ if } \mathcal{Q} = \emptyset, \\ \bigvee_{Q \in \mathcal{Q}} T[i-1, Q, S', \ell - \mathsf{ovh}(Q)], \text{ otherwise.} \end{cases}$$

In the following two lemmata (Lemma 11 and 12), we prove the correctness of our computation for Case 2.

▶ **Lemma 11.** *If $T[i, S, S', \ell] = 1$, then there is $Q \in \mathcal{Q}$, such that $T[i-1, Q, S', \ell - \mathsf{ovh}(Q)] = 1$.*

Proof. Assume that $T[i, S, S', \ell] = 1$. Let M be a matching in G_i that realizes $T[i, S, S', \ell]$. By definition, M is (i, S, S')-compatible. Note that $y_i \in \mathsf{Sat}(M)$ and $x_i \notin \mathsf{Sat}(M)$. Let $x_jy_i \in M$. From Observation 7, we have $x_j \in \widehat{X}_{i-1}$. Thus, we can conclude that $i - k \leqslant j < i$. Recall that $Q_j = (S \setminus \{x_i\}) \cup \{x_j\}$. Now from Observation 9, it follows that $Q_j \in \mathcal{S}_X^{i-1}$. As $y_i \notin S'$, from Observation 4 it follows that $S' \in \mathcal{S}_Y^{i-1}$.

Next, we will show that $\ell - \mathsf{ovh}(Q_j) \in \mathsf{Alw}_{i-1}(Q_j, S')$. Let $\widetilde{\ell} = \mathsf{ovh}(Q_j) = |X_{j+1,i} \setminus S|$. From Observation 8 it follows that the edge x_jy_i intersects exactly $|X_{j+1,i} \setminus S|$ many edges from M. Thus, $|X_{j+1,i} \setminus S| \leqslant \ell$, and $0 \leqslant \widetilde{\ell} \leqslant \ell \leqslant k$. Recall that $\mathsf{Alw}_{i-1}(Q_j, S') = \{p \in [k]_0 \mid p \leqslant k - \max\{\mathsf{cst}_{i-1}(Q_j), \mathsf{cst}_{i-1}(S')\}\}$. To show that $\ell - \widetilde{\ell} \in \mathsf{Alw}_{i-1}(Q_j, S')$, it is enough to show that $\ell - \widetilde{\ell} \leqslant k - \max\{\mathsf{cst}_{i-1}(Q_j), \mathsf{cst}_{i-1}(S')\}$. Note that $\ell \leqslant k - \max\{\mathsf{cst}_i(S), \mathsf{cst}_i(S')\}$ as $\ell \in \mathsf{Alw}_i(S,S')$. Using Observation 5, we obtain that $\mathsf{cst}_{i-1}(S') \leqslant \mathsf{cst}_i(S')$. Thus, $\ell - \widetilde{\ell} \leqslant \ell \leqslant k - \mathsf{cst}_i(S') \leqslant k - \mathsf{cst}_{i-1}(S')$. Now we will argue that $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(Q_j)$. We start by arguing that $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) + \widetilde{\ell}$. As $Q_j \setminus \{x_j\} = S \setminus \{x_i\}$, using Observation 5, we obtain that $\mathsf{cst}_{i-1}(Q_j \setminus \{x_j\}) \leqslant \mathsf{cst}_i(S) - |S|$. Note that $\mathsf{cst}_{i-1}(Q_j) = \mathsf{cst}_{i-1}(Q_j \setminus \{x_j\}) + \mathsf{cst}_{i-1}(x_j)$. Recall that $\mathsf{cst}_{i-1}(x_j) = i-j$. Thus, $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) - |S| + i - j \leqslant \mathsf{cst}_i(S) - |S \cap X_{j+1,i}| + i - j$. Note that $|X_{j+1,i}| = i-j$. Thus, $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) - |S \cap X_{j+1,i}| + |X_{j+1,i}| = \mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|$. Hence, $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) + \widetilde{\ell}$. We will use the above statement to argue that $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(Q_j)$. As $\ell \in \mathsf{Alw}_i(S,S')$, we have $\ell + \mathsf{cst}_i(S) \leqslant k$. Thus, $\ell + \mathsf{cst}_i(S) = \ell - \widetilde{\ell} + \mathsf{cst}_i(S) + \widetilde{\ell} = \ell - \widetilde{\ell} + \mathsf{cst}_{i-1}(Q_j) \leqslant k$. Hence, $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(Q_j)$. From the above discussions, we can conclude that $\ell - \mathsf{ovh}(Q_j) \in \mathsf{Alw}_{i-1}(Q_j, S')$.

We have obtained that $T[i-1, Q_j, S', \ell - \mathsf{ovh}(Q_j)]$ exists. Note that M' is a matching which realizes $T[i-1, Q_j, S', \ell - \mathsf{ovh}(Q_j)]$. This concludes the proof. ◀

▶ **Lemma 12.** *If there is $Q \in \mathcal{Q}$, such that $T[i-1, Q, S', \ell - \mathsf{ovh}(Q)] = 1$, then $T[i, S, S', \ell] = 1$.*

Proof. Assume that $T[i-1, Q_j, S', \ell - \mathsf{ovh}(Q_j)] = 1$. Let M' be a matching in G_{i-1} that realizes $T[i, S, S', \ell]$. Note that $x_j \notin \mathsf{Sat}(M')$. Let $M = M' \cup \{x_j y_i\}$. Observe that $\mathsf{Sat}(M) = (X_i \setminus S) \cup (Y_i \setminus S')$. From Observation 8, the edge $x_j y_i$ intersects exactly $|X_{j+1,i} \setminus S| = \mathsf{ovh}(Q_j)$ edges from M'. This together with the fact that $\mathtt{cr}(M') = \ell - \mathsf{ovh}(Q_j)$, implies that $\mathtt{cr}(M) = \ell$. Thus, we can conclude that M realizes $T[i, S, S']$, and hence $T[i, S, S'] = 1$. ◀

Case 3: $x_i \notin S$ and $y_i \notin S'$. In this case, a matching, say M, which realizes $T[i, S, S', \ell]$, must saturate both the vertices x_i and y_i. Thus, M must have edges $x_j y_i$ and $x_i y_{j'}$, where $j \leqslant i$ and $j' \leqslant i$. (Assuming x_i is adjacent to y_i in G, it can be the case that $j = j' = i$, in which case $x_i y_i \in M$.) We will thus have $T[i, S, S', \ell] = T_1[i, S, S', \ell] \vee T_2[i, S, S', \ell]$, where $T_1[i, S, S', \ell]$ and $T_2[i, S, S', \ell]$ are boolean variables that correspond respectively to the cases $j = j' = i$ and $j \neq i$ (and $j' \neq i$). We now define $T_1[i, S, S', \ell]$ and $T_2[i, S, S', \ell]$, formally.

Defining $T_1[i, S, S', \ell]$. Since $x_i \notin S$, we have $S \subseteq \widehat{X}_{i-1}$. Since $y_i \notin S'$, we have $S' \subseteq \widehat{Y}_{i-1}$. By Observation 4, $S \in \mathcal{S}_X^{i-1}$ and $S' \in \mathcal{S}_Y^{i-1}$. Note that if a matching M that realizes $T[i, S, S', \ell]$ contains the edge $x_i y_i$ (assuming $x_i y_i$ is indeed an edge in the graph G), then $\mathtt{cr}(M) = \mathtt{cr}(M \setminus \{x_i y_i\})$. That is, no additional crossing is incurred by adding the edge $x_i y_i$ to the matching $M \setminus \{x_i y_i\}$. Also, note that $\ell \in \mathsf{Alw}_{i-1}(S, S')$. With these observations, we define $T_1[i, S, S', \ell]$ as follows.

$$T_1[i, S, S', \ell] = \begin{cases} 0, & \text{if } x_i y_i \notin E(G), \\ T[i-1, S, S', \ell], & \text{otherwise.} \end{cases}$$

Defining $T_2[i, S, S', \ell]$. Now, to define $T_2[i, S, S', \ell]$, we proceed as in Case 2. We will rely on the fact that the matching we are seeking for does not contain the edge $x_i y_i$. Since we need both x_i and y_i to be matched here, we will construct a set $\mathcal{Q} \subseteq \mathcal{S}_X^{i-1} \subseteq 2^{\widehat{X}_{i-1}}$ and a set $\mathcal{R} \subseteq \mathcal{S}_Y^{i-1} \subseteq 2^{\widehat{Y}_{i-1}}$ (each of size $\mathcal{O}(k)$). We define $\mathcal{Q}$ (almost) the same way as we did in Case 2. We also define $\mathcal{R}$, the Y-counterpart of $\mathcal{Q}$, analogously.

For a vertex $x_j \in (N(y_i) \cap \widehat{X}_{i-1}) \setminus S$, let $Q_j = S \cup \{x_j\}$. We let $\widehat{\mathcal{Q}} = \{Q_j \mid x_j \in (N(y_i) \cap \widehat{X}_{i-1}) \setminus S\}$, and $\mathcal{Q} = \widehat{\mathcal{Q}} \cap \mathcal{S}_X^{i-1}$. Similarly, for a vertex $y_{j'} \in (N(x_i) \cap \widehat{Y}_{i-1}) \setminus S'$, let $R_{j'} = S' \cup \{y_{j'}\}$. We let $\widehat{\mathcal{R}} = \{R_{j'} \mid y_{j'} \in (N(x_i) \cap \widehat{Y}_{i-1}) \setminus S'\}$, and $\mathcal{R} = \widehat{\mathcal{R}} \cap \mathcal{S}_Y^{i-1}$. We will now construct a set of "crucial pairs" from $\mathcal{Q} \times \mathcal{R}$, for the computation of $T_2[i, S, S', \ell]$. Towards this, we define a function $\mathsf{ovh} : \mathcal{Q} \times \mathcal{R} \to \mathbb{N}$. We set $\mathsf{ovh}(Q_j, R_{j'}) = |X_{j+1,i} \setminus S| + |Y_{j'+1,i} \setminus S'| - 1$, for

$Q_j \in \mathcal{Q}$ and $R_{j'} \in \mathcal{R}$. Finally, we let $\mathcal{C} = \{(Q, R) \in \mathcal{Q} \times \mathcal{R} \mid \ell - \mathsf{ovh}(Q, R) \in \mathsf{Alw}_{i-1}(Q, R)\}$. Now we set $T_2[i, S, S', \ell]$ as follows.

$$T_2[i, S, S', \ell] = \begin{cases} 0, \text{ if } \mathcal{C} = \emptyset, \\ \bigvee_{(Q,R)\in\mathcal{C}} T[i-1, Q, R, \ell - \mathsf{ovh}(Q, R)], \text{ otherwise.} \end{cases}$$

We set $T = T_1[i, S, S', \ell] \vee T_2[i, S, S', \ell]$. Using the following four lemmata (Lemma 13 to 16), we establish that the computation at Case 3 is correct. The proofs of Lemma 14 and 16 will use arguments similar to the ones used for the proof of Lemma 11 and 12, respectively.

▶ **Lemma 13.** *Let $T[i, S, S', \ell] = 1$ and M be a matching realizing $T[i, S, S', \ell]$, such that $x_iy_i \in M$ (i.e., $T_1[i, S, S', \ell] = 1$). Then, $T[i-1, S, S', \ell] = 1$.*

Proof. Let $M' = M \setminus \{x_iy_i\}$. Note that $\mathtt{cr}(M) = \mathtt{cr}(M) = \ell$, as the edge x_iy_i does not intersect any edge in M. Moreover, $\mathsf{Sat}M' = (X_{i-1} \setminus S) \cup (Y_{i-1} \setminus S)$. Thus, we conclude that $T[i-1, S, S', \ell] = 1$. ◀

▶ **Lemma 14.** *Let $T[i, S, S', \ell] = 1$ and M be a matching realizing $T[i, S, S', \ell]$, such that $x_iy_i \notin M$ (i.e., $T_2[i, S, S', \ell] = 1$). Then, there is $(Q, R) \in \mathcal{C}$, such that $T[i-1, Q, R, \ell - \mathsf{ovh}(Q, R)] = 1$.*

Proof. Note that $x_i, y_i \in \mathsf{Sat}(M)$, as $x_i \notin S$ and $y_i \notin S'$. Let $x_jy_i, x_iy_{j'} \in M$. By the premise of the lemma, we have $j \neq j' \neq i$. Note that the edges x_jy_i and $x_iy_{j'}$, intersect each other. From Observation 7, we have $x_j \in \widehat{X}_{i-1}$ and $y_{j'} \in \widehat{Y}_{i-1}$. Thus, we can conclude that $i - k \leqslant j, j' < i$. Recall that $Q_j = S \cup \{x_j\}$ and $R_{j'} = S' \cup \{y_{j'}\}$. Now from Observation 9, it follows that $Q_j \in \mathcal{S}_X^{i-1}$ (as $x_i \notin S$) and $R_{j'} \in \mathcal{S}_Y^{i-1}$ (as $y_i \notin S'$). Since $x_j \in N(y_i)$ and $y_{j'} \in N(x_i)$, we have $Q_j \in \mathcal{Q}$ and $R_{j'} \in \mathcal{R}$.

Next, we will show that $\ell - \mathsf{ovh}(Q_j, R_{j'}) \in \mathsf{Alw}_{i-1}(Q_j, R_{j'})$ (and thus, $(Q_j, R_{j'}) \in \mathcal{C}$). Let $\widetilde{\ell} = \mathsf{ovh}(Q_j, R_{j'}) = |X_{j+1,i} \setminus S| + |Y_{j'+1,i} \setminus S'| - 1$. From Observation 8 it follows that the edges x_jy_i and $x_iy_{j'}$ intersects exactly $|X_{j+1,i} \setminus S|$ and $|Y_{j'+1,i} \setminus S'|$ many edges from M, respectively. (Note that x_jy_i and $x_iy_{j'}$ intersect each other.) Thus, $|X_{j+1,i} \setminus S| + |Y_{j'+1,i} \setminus S'| - 1 \leqslant \ell$. In the sum on the left hand side of this inequality, the term -1 ensures that the intersection of the edges x_jy_i and $x_iy_{j'}$ is counted exactly once. Recall that $\widetilde{\ell} = |X_{j+1,i} \setminus S| + |Y_{j'+1,i} \setminus S'| - 1$. Thus, $0 \leqslant \ell - \widetilde{\ell} \leqslant \ell \leqslant k$. Recall that $\mathsf{Alw}_{i-1}(Q_j, R_{j'}) = \{p \in [k]_0 \mid p \leqslant k - \max\{\mathsf{cst}_{i-1}(Q_j), \mathsf{cst}_{i-1}(R_{j'})\}\}$. To show that $\ell - \widetilde{\ell} \in \mathsf{Alw}_{i-1}(Q_j, R_{j'})$, it is enough to show that $\ell - \widetilde{\ell} \leqslant k - \max\{\mathsf{cst}_{i-1}(Q_j), \mathsf{cst}_{i-1}(R_{j'})\}$. Note that $\ell \leqslant k - \max\{\mathsf{cst}_i(S), \mathsf{cst}_i(S')\}$ as $\ell \in \mathsf{Alw}_i(S, S')$. We will argue that $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(Q_j)$. (We can obtain that $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(R_{j'})$, by following similar arguments.) We start by arguing that $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|$. As $Q_j \setminus \{x_j\} = S$, using Observation 5, we obtain that $\mathsf{cst}_{i-1}(Q_j \setminus \{x_j\}) \leqslant \mathsf{cst}_i(S) - |S|$. Note that $\mathsf{cst}_{i-1}(Q_j) = \mathsf{cst}_{i-1}(Q_j \setminus \{x_j\}) + \mathsf{cst}_{i-1}(x_j)$. Recall that $\mathsf{cst}_{i-1}(x_j) = i - j$. Thus, $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) - |S| + i - j \leqslant \mathsf{cst}_i(S) - |S \cap X_{j+1,i}| + i - j$. Note that $|X_{j+1,i}| = i - j$. Thus, $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) - |S \cap X_{j+1,i}| + |X_{j+1,i}| = \mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|$. Hence, $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|$. We will use the above statement to argue that $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(Q_j)$. As $\ell \in \mathsf{Alw}_i(S, S')$, we have $\ell + \mathsf{cst}_i(S) \leqslant k$. Thus, $\ell + \mathsf{cst}_i(S) = \ell - \widetilde{\ell} + (\mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|) + (|Y_{j'+1,i} \setminus S'| - 1) \leqslant k$. As $\mathsf{cst}_{i-1}(Q_j) \leqslant \mathsf{cst}_i(S) + |X_{j+1,i} \setminus S|$, we have $\ell - \widetilde{\ell} + \mathsf{cst}_{i-1}(Q_j) + (|Y_{j'+1,i} \setminus S'| - 1) \leqslant k$. Note that $Y_{j'+1,i} \setminus S' \neq \emptyset$, as $y_i \notin S'$, and therefore, $|Y_{j'+1,i} \setminus S'| - 1 \geqslant 0$. From the above discussions, we can obtain that $\ell - \widetilde{\ell} \leqslant k - \mathsf{cst}_{i-1}(Q_j)$. Thus, we can conclude that $\ell - \mathsf{ovh}(Q_j, R_{j'}) \in \mathsf{Alw}_{i-1}(Q_j, R_{j'})$, and hence $(Q_j, R_{j'}) \in \mathcal{C}$.

We have obtained that $T[i-1, Q_j, R_{j'}, \ell - \mathsf{ovh}(Q_j, R_{j'})]$ exists. Note that M' is a matching which realizes $T[i-1, Q_j, R_{j'}, \ell - \mathsf{ovh}(Q_j, R_{j'})]$. This concludes the proof. ◀

▶ **Lemma 15.** *If $T[i-1, S, S', \ell] = 1$, then $T[i, S, S', \ell] = 1$ (in particular, $T_1[i, S, S', \ell] = 1$).*

Proof. Consider a matching M' realizing $T[i-1, S, S', \ell] = 1$, and let $M = M' \cup \{x_i y_i\}$. Note that $\mathtt{cr}(M) = \mathtt{cr}(M) = \ell$, as the edge $x_i y_i$ does not intersect any edge in M'. Moreover, $\mathsf{Sat} M = (X_i \setminus S) \cup (Y_i \setminus S)$, as $x_i \notin S$ and $y_i \notin S'$. Thus, we conclude that $T[i, S, S', \ell] = 1$. ◀

▶ **Lemma 16.** *If there is $(Q, R) \in \mathcal{C}$, such that $T[i-1, Q, R, \ell - \mathsf{ovh}(Q, R)] = 1$, then $T[i, S, S', \ell] = 1$ (in particular, $T_2[i, S, S', \ell] = 1$).*

Proof. Assume that $T[i-1, Q_j, R_{j'}, \ell - \mathsf{ovh}(Q_j, R_{j'})] = 1$, and let M' be a matching in G_{i-1} realizing it. Note that $x_j, y_{j'} \notin \mathsf{Sat}(M')$. Let $M = M' \cup \{x_j y_i, x_i y_{j'}\}$. Observe that $\mathsf{Sat}(M) = (X_i \setminus S) \cup (Y_i \setminus S')$. From Observation 8, the edges $x_j y_i$ and $x_i y_{j'}$ intersect exactly $|X_{j+1,i} \setminus S|$ and $|Y_{j'+1,i} \setminus S'|$ many edges in M, respectively. Moreover, $x_j y_i$ and $x_i y_{j'}$ intersect each other. Recall that $\mathsf{ovh}(Q_j, R_{j'}) = |X_{j+1,i} \setminus S| + |Y_{j'+1,i} \setminus S'| - 1$. From the above discussions and the fact that $\mathtt{cr}(M') = \ell - \mathsf{ovh}(Q_j, R_{j'})$, we can conclude that $\mathtt{cr}(M) = \ell$. Thus, M realizes $T[i, S, S']$, and hence $T[i, S, S'] = 1$. ◀

As observed earlier, (G, k) is a yes-instance of CM-PM if and only if there is $\ell \in [k]_0$, such that $T[n, \emptyset, \emptyset, \ell] = 1$. Note that for each $i \in [n]$, $S \in \mathcal{S}^i_X$, $S' \in \mathcal{S}^i_Y$, and $\ell \in \mathsf{Alw}_i(S, S')$, we can compute the entry $T[i, S, S', \ell]$ in time bounded by $n^{\mathcal{O}(1)}$. Moreover, the number of entries in our table is bounded by $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ (see Lemma 3). Thus, the running time of the algorithm is bounded by $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$. The correctness of the algorithm follows from the correctness of base case and recursive formulae (Lemma 10 to 16). The above discussions lead us to the following theorem.

▶ **Theorem 17.** Crossing-Minimizing Perfect Matching *admits an algorithm running in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$, where n is the number of vertices in the input graph.*

3.3 Polynomial kernel for CM-PM

In this section, we design a kernel with $\mathcal{O}(k^2)$ vertices for CM-PM. Let (G, k) be an instance of CM-PM. To obtain our kernel we first bound the number of pairs (x_i, y_i) (called a *bad pair*), where $x_i y_i$ is not an edge, by $\mathcal{O}(k)$. This bound is obtained by arguing that bad pairs contribute to edge crossings. Next, we argue that not all the vertices between two consecutive bad pairs is necessary, for preserving the answer. In fact, we argue that keeping $\mathcal{O}(k)$ vertices between each consecutive bad pairs is enough. This strategy leads us to a kernel with $\mathcal{O}(k^2)$ vertices.

Before moving to the formal description of our algorithm, we start by introducing some notations which will be useful later. Let (G, k) be an instance of CM-PM. For each $i \in [n]$, if $x_i y_i \in E(G)$, then we call the pair (x_i, y_i) a *good pair*, otherwise we call (x_i, y_i) a *bad pair*. A perfect matching M of G is said to be an *optimal perfect matching* of G if $\mathtt{cr}(M) \leqslant \mathtt{cr}(M')$ for every perfect matching M' of G. If $i = j$, then we call $x_i y_j$ a *vertical edge*, if $i < j$, then we call $x_i y_j$ a *left-leaning* edge, and if $i > j$ then we call $x_i y_j$ a *right-leaning* edge.

We first prove two lemmata that will be crucial for the correctness of our kernelization algorithm. The first lemma shows that every left- or right-leaning edge in a perfect matching of G participates in at least one crossing. Moreover, the second lemma provides a lower bound on the number of crossings in a perfect matching of G.

▶ **Lemma 18.** *Let (G,k) be an instance of* CM-PM. *Let $M \subseteq E(G)$ be a perfect matching of G such that $x_iy_j \in M$. Then $\mathtt{cr}(M) \geqslant |j-i|$. In particular, if x_iy_j is a left-leaning edge, then it intersects at least $j-i$ edges $x_ry_s \in M$ with $r > i$ and $s < j$; and if x_iy_j is a right-leaning edge, then it intersects at least $i-j$ edges $x_ry_s \in M$ with $r < i$ and $s > j$.*

Proof. If $i = j$, then there is nothing to prove. Assume $i < j$. Consider the $j-1$ vertices $y_1, y_2, \ldots, y_{j-1}$. In M, at most $i-1$ of them are matched to $\{x_r \mid r < i\}$. Therefore, at least $(j-1)-(i-1) = j-i$ of them are matched to $\{x_r \mid r > i\}$. That is, M contains at least $j-i$ edges x_ry_s, where $r > i$ and $s < j$. Moreover, each of these edges crosses x_iy_j. The case when $i > j$ is symmetric. ◀

▶ **Lemma 19.** *Let (G,k) be an instance of* CM-PM *and $M \subseteq E(G)$ a perfect matching of G. Let $M_L \subseteq M$ be the set of left-leaning edges in M and $M_R \subseteq M$ the set of right-leaning edges in M. Then,*

$$\mathtt{cr}(M) \geqslant \max\left\{\sum_{x_iy_j \in M_L}(j-i), \sum_{x_iy_j \in M_R}(i-j)\right\}.$$

Proof. As shown in the proof of Lemma 18, each edge $x_iy_j \in M_L$ intersects at least $(j-i)$ edges x_ry_s with $r > i$ and $s < j$. Moreover, because $r > i$ and $s < j$, these $(j-i)$ crossings are counted exactly once. Summing over all edges $x_iy_j \in M$, we get $\mathtt{cr}(M) \geqslant \sum_{x_iy_j \in M_L}(j-i)$. Using symmetric arguments, we can also show that $\mathtt{cr}(M) \geqslant \sum_{x_iy_j \in M_R}(i-j)$. ◀

We are now ready to present our kernelization algorithm. In the following we prove a lemma which bounds the number of bad pairs in the input instance.

▶ **Lemma 20.** *Let (G,k) be an instance of* CM-PM. *If G contains at least $2k+1$ bad pairs, then (G,k) is a no-instance.*

Proof. Assume that G contains at least $2k+1$ bad pairs. Let M be an optimal perfect matching of G. We shall show that $\mathtt{cr}(M) > k$. Note that corresponding to every bad pair (x_i, y_i), M contains a left- or right-leaning edge x_iy_j. Moreover, since G contains at least $2k+1$ bad pairs, at least $2k+1$ edges in M are left- or right-leaning. Then, by the pigeonhole principle, either at least $k+1$ of these edges are left-leaning or at least $k+1$ are right-leaning. Assume without loss of generality that at least $k+1$ are left-leaning, and let $M_L \subseteq M$ be the set of these left-leaning edges. Thus $|M_L| \geqslant k+1$ and note that for each $x_iy_j \in M_L$, $(j-i) \geqslant 1$. By Lemma 19, $\mathtt{cr}(M) \geqslant \sum_{x_iy_j \in M_L}(j-i) \geqslant k+1$. ◀

The above lemma leads us to the following reduction rule.

Rule 1: G contains at least $2k+1$ bad pairs. **Do:** Return that (G,k) is a no-instance.

When Rule 1 is not applicable, the number of bad pairs in G is bounded by $2k$. We now need to bound the number of good pairs. Towards that end, we introduce the following reduction rules.

Rule 2: Let (x_i, y_i) and (x_j, y_j) be two consecutive bad pairs (i.e., (x_r, y_r) is a good pair for every r, where $i < r < j$) such that $j - i > 4k+2$. **Do:** Delete vertices x_r and y_r for every $r = i+2k+2, i+2k+3, \ldots, j-2k-2$. **Parameter:** No change.

> **Rule 3:** Let (x_i, y_i) be the first bad pair (i.e., (x_r, y_r) is a good pair for every $r < i$) and (x_j, y_j) the last bad pair (i.e., (x_r, y_r) is a good pair for every $r > j$) in G.
> **Do:** If $i > 2k+1$, then delete vertices x_r and y_r for every $r < i-2k-1$. If $n-j > 2k+1$, then delete vertices x_r and y_r for every $r > j+2k+1$.
> **Parameter:** No change.

▶ **Lemma 21.** *Rules 2 and 3 are safe.*

Proof. We show safeness of Rule 2 only. The proof for Rule 3 is similar. Let (G', k') be an instance obtained from (G, k) by a single application of Rule 2 with the pair of consecutive bad pairs (x_i, y_i) and (x_j, y_j). Note that $k' = k$. We show that (G, k) is a yes-instance if and only if (G', k') is a yes-instance.

Assume that (G, k) is a yes-instance and let $M \subseteq E(G)$ be an optimal perfect matching of G. Then, $\mathtt{cr}(M) \leqslant k$. Consider the $2k+1$ edges in M that saturate (incident to) the vertices $x_{i+1}, x_{i+2}, \dots, x_{i+2k+1}$. Since $\mathtt{cr}(M) \leqslant k$, at most $2k$ of these edges can participate in a crossing. Equivalently, at least one of these edges does not participate in any crossing. But every left- or right-leaning edge in M participates in at least one crossing. Therefore, at least one of the vertices $x_{i+1}, x_{i+2}, \dots, x_{i+2k+1}$ is saturated by a vertical edge in M. Let $x_{i'}$, for some $i' \in \{i+1, i+2, \dots, i+2k+1\}$ be that vertex. That is, $x_{i'}y_{i'} \in M$ and $x_{i'}y_{i'}$ does not participate in any crossing in M.

Similarly, among the $2k+1$ edges in M that saturate the vertices $x_{j-1}, x_{j-2}, \dots x_{j-2k-1}$, at least one is a vertical edge that does not participate in any crossing. Let $x_{j'}y_{j'}$ be that edge for some $j' \in \{j-1, j-2, \dots, j-2k-1\}$. Also, note that since $j - i > 4k+2$, we have $i + 2k + 1 < j - 2k - 1$.

For r such that $i' < r < j'$, consider the edge $x_r y_s \in M$ that saturates x_r. Since the two edges $x_{i'}y_{i'}$ and $x_{j'}y_{j'}$ do not participate in any crossing, in particular, they do not cross the edge $x_r y_s$. Therefore, $i' < s < j'$. Let $\widetilde{M} \subseteq M$ be the set of edges in M that saturate the vertices $x_{i'+1}, x_{i'+2}, \dots, x_{j'-1}$. Then, $M^* = (M \setminus \widetilde{M}) \cup \{x_r y_r \mid i' < r < j'\}$ is also a perfect matching in G with $\mathtt{cr}(M^*) \leqslant \mathtt{cr}(M) \leqslant k$. Now note that the graph G' is obtained from G by deleting the vertices x_r and y_r for every $r = i+2k+2, i+2k+3, \dots, j-2k-2$. Also, note that $i' < i+2k+2$ and $j-2k-2 < j'$. Therefore, $M^* \setminus \{x_r y_r \mid i+2k+2 \leqslant r \leqslant j-2k-2\}$ is a perfect matching of G' with at most k crossings.

To see the reverse direction, assume that (G', k) is a yes-instance and let M' be an optimal matching of G'. Then, $\mathtt{cr}(M') \leqslant k$. By repeating the arguments used in the forward direction, we can show that M' contains vertical edges $x_{i'}y_{i'}$ and $x_{j'}y_{j'}$ that do not participate in any crossing, for some $i' \in \{i+1, i+2, \dots, i+2k+1\}$ and $j' \in \{j-1, j-2, \dots, j-2k-1\}$. For r such that $i' < r < j'$, consider the edge $x_r y_s \in M'$ that saturates x_r. Since the two edges $x_{i'}y_{i'}$ and $x_{j'}y_{j'}$ do not participate in any crossing, in particular, they do not cross the edge $x_r y_s$. Therefore, $i' < s < j'$. Let $\widehat{M} \subseteq M'$ be the set of edges in M' that saturate the vertices $x_{i'+1}, x_{i'+2}, \dots, x_{j'-1}$. Then, $M'' = (M' \setminus \widehat{M}) \cup \{x_r y_r \mid i' < r < j'\}$ is also a perfect matching with $\mathtt{cr}(M'') \leqslant \mathtt{cr}(M') \leqslant k$. Then, note that $M''' = M'' \cup \{x_r y_r \mid i+2k+2 \leqslant r \leqslant j-2k-2\}$ is a perfect matching of G and $\mathtt{cr}(M''') = \mathtt{cr}(M'') \leqslant k$. ◀

▶ **Lemma 22.** *Given an instance (G, k) of* CM-PM*, let (G', k) be the instance obtained from (G, k) by an exhaustive application of Rules 1 to 3. Then, $|V(G')| \leqslant (\beta(G) - 1)(4k + 2) + \beta(G) + 4k + 2$, where $\beta(G)$ is the number of bad pairs in G.*

Proof. If Rule 1 is applicable, we correctly report the answer. So we assume that Rule 1 is not applicable. After an exhaustive application of Rules 2 and 3, between two consecutive bad pairs in G', there are at most $4k+2$ good pairs, and there are most $2k+1$ good pairs between (x_1, y_1) (including (x_1, y_1)) and the first bad pair, and between the last bad pair and (x_n, y_n) (including (x_n, y_n)). Moreover, the number of bad pairs in G' is the same as the number of bad pairs in G. ◀

▶ **Theorem 23.** Crossing-Minimizing Perfect Matching, *parameterized by the number of crossings k, has a kernel of with $\mathcal{O}(k^2)$ vertices.*

Proof. Given (G, k), if G contains at least $2k+1$ bad pairs, then by Lemma 20, (G, k) is a no-insatnce. Otherwise, $\beta(G) \leqslant 2k$. When none of Rules 1 to 3 apply, we have $|V(G)| \in \mathcal{O}(k^2)$ (see Lemma 22). ◀

4 NP-hardness, FPT Algorithm and Polynomial Kernel for Crossing-Minimizing Hamiltonian Path

In this section, we show that CM-Ham Path is NPH, but can be solved in time $2^{\mathcal{O}(\sqrt{k}\log k)} n^{\mathcal{O}(1)}$ and admits a kernel with $\mathcal{O}(k^2)$ vertices. The problem Crossing-Minimizing Hamiltonian Path (CM-Ham Path) is formally defined below.

Crossing-Minimizing Hamiltonian Path (CM-Ham Path) **Parameter:** k
Input: A two-layered graph G and a non-negative integer k.
Question: Does G have a Hamiltonian path with at most k crossings?

4.1 NP-hardness of Crossing-Minimizing Hamiltonian Path

The NP-hardness of Crossing-Minimizing Hamiltonian Path follows from NP-hardness of testing if the given bipartite graph admits a Hamiltonian path. In this section, we show that even if the given instance (G, k) of CM-Ham Path, G admits a Hamiltonian path, testing if there is a Hamiltonian path in the two-layered graph G with at most k crossing is NP-hard. We call this problems Restricted CM-Ham Path, which is formally defined below.

Restricted CM-Ham Path
Input: A two-layered graph G, which admits a Hamiltonian path, vertex bipartition X, Y of $V(G)$, and an integer k.
Question: Does G have a Hamiltonian path with at most k crossings?

To establish the NP-hardness result for Restricted CM-Ham Path, we give an appropriate reduction from the Bipartite-Ham Path, which is defined below.

Bipartite-Ham Path
Input: A bipartite graph G (with maximum degree three) with vertex bipartition X, Y, and a vertex $x^* \in X$.
Question: Does G admit a Hamiltonian path with x^* as one of the end vertices?

The NP-hardness of Bipartite-Ham Path follows from the NP-hardness of Hamiltonian Path on bipartite graphs of maximum degree three [45, 46], where the goal to test if the given bipartite graph admits a Hamiltonian path.

Reduction. Let (G, X, Y, x^*) be an instance of Bipartite-Ham Path. We construct a two-layered graph H with vertex bipartition P, Q, and an integer k such that (H, P, Q, k) is a yes instance of Restricted CM-Ham Path if and only if (G, X, Y, x^*) is a yes instance of Bipartite-Ham Path. We let vertices in X to be $x_1 = x^*, x_2, \ldots, x_n$, and

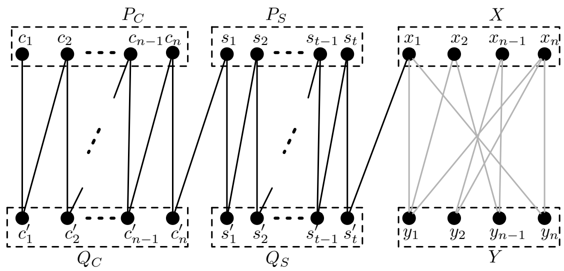


Figure 8 Partial construction of an instance of CM-Ham Path.

$\sigma_X = (x_1, x_2, \dots, x_n)$. Similarly, we let vertices in Y to be $y_1, y_2, \dots, y_n$, and $\sigma_Y = (y_1, y_2, \dots, y_n)$. Initially, $P = X$, $Q = Y$, and $E(H) = E(G)$. Next, we create sets of (new) vertices, $P_c = \{c_1, c_2, \dots, c_{n-1}, c_n\}$, $Q_c = \{c'_1, c'_2, \dots, c'_{n-1}c'_n\}$, $P_s = \{s_1, s_2, \dots, s_t\}$, and $Q_s = \{s'_1, s'_2, \dots, s'_t\}$. Here, $t = \binom{2n-1}{2} + 2$. We add all the vertices in $P_c \cup P_s$ to P, and add all the vertices in $Q_c \cup Q_s$ to Q. The vertices in $P_c \cup Q_c$ induces a path in H, namely, $P_1 = c_1c'_1c_2c'_2 \dots c_{n-1}c'_{n-1}c_nc'_n$, i.e. we add all the edges in $\{c_ic'_i \mid i \in [t]\} \cup \{c'_ic_{i+1} \mid i \in [t-1]\}$ to $E(H)$. Similarly, the vertices in $P_s \cup Q_s$ induces a path in H, namely, $P_2 = s_1s'_1s_2s'_2 \dots s_{t-1}s'_{t-1}s_ts'_t$, i.e. we add all the edges in $\{s_is'_i \mid i \in [t]\} \cup \{s'_is_{i+1} \mid i \in [t-1]\}$ to $E(H)$. We add the edge c'_ns_1 to $E(H)$, and therefore, $P_1 \bullet P_2$ induces a path in H. Next, we add all the edges in $\{y_ic_i, c_iy_{i+1} \mid i \in [n-1]\}$ to $E(H)$. The intuition behind adding these edges is to connect vertices y_i and y_{i+1} via the vertex c_i, where $i \in [n-1]$. Similarly, we add all the edges in $\{x_ic'_i, c'_ix_{i+1} \mid i \in [n-1]\}$ to $E(H)$. We add the edge $s'_tx_1 = x^*$ to $E(H)$. We also add the edge $x_nc'_n$ and y_nc_n to $E(H)$, which will be helpful in creating a Hamiltonian path in H. We let $\sigma_P = c_0c_1c_2 \dots c_{n-1} \circ s_1s_2 \dots s_t \circ \sigma_X$. Similarly, we let $\sigma_Q = c'_1c'_2 \dots c'_n \circ s'_1s'_2 \dots s'_t \circ \sigma_Y$. Next, we place vertices in P and Q in two (distinct) parallel lines L_P and L_Q, respectively. The order in which the points in P appear in L_P is given by σ_P. Similarly, the order in which the points in Q appear in L_Q is given by σ_Q. This completes the description of the two-layered graph H with vertex bipartition P and Q. Finally, we set $k = \binom{2n-1}{2}$.

In what follows, we prove some lemmata that will be helpful in establishing the equivalence of the instances (G, X, Y, x^*) of Bipartite-Ham Path and (H, P, Q, k) of Restricted CM-Ham Path.

▶ **Observation 24.** *The bipartite graph H admits a Hamiltonian path.*

Proof. Let $P_s = s_1s'_1s_2s'_2 \dots s_ts'_t$, $P_X = x_1c'_1x_2c'_2x_3 \dots x_{n-1}c'_{n-1}x_nc'_n$, and $P_Y = c_ny_n c_{n-1}y_{n-1}c_{n-2}y_{n-2} \dots y_2c_1y_1$ be paths in H. By construction the path $P_s \bullet (P_X \bullet P_Y)$ is a Hamiltonian path in H. ◀

▶ **Lemma 25.** *Let (H, P, Q, k) be a* yes *instance of* Restricted CM-Ham Path*, and S be a Hamiltonian path in H with at most k crossings. Then, $E(S) \cap (\{y_ic_j, x_ic'_j \mid i, j \in [n]\} \cap E(H)) = \emptyset$.*

Proof. Assuming a contradiction, suppose S contains an edge say, $e \in \{y_ic_j, x_ic'_j \mid i, j \in [n]\} \cap E(H)$. Note that in H, e crosses each edge in $\{s_is'_i \mid i \in [t]\} \cup \{s'_is_{i+1}) \mid i \in [t-1]\}$, where $t = \binom{2n-1}{2} + 2$. Moreover, for each $i \in [t-1]$, $N(s'_i) = \{s_i, s_{i+1}\}$. Since S is a Hamiltonian path, it follows that $|E(S) \cap \{s_is'_i, s'_is_{i+1} \mid i \in [t-1]\}| \geqslant t - 1 = \binom{2n-1}{2} + 1$. Moreover, e crosses each edge in $E(S) \cap \{s_is'_i, s'_i, s_{i+1} \mid i \in [t-1]\}$. This contradicts the fact that S is a Hamiltonian path in S with at most $k = \binom{2n-1}{2}$ crossings. ◀

▶ **Lemma 26.** *(G, X, Y, x^*) is a yes instance of* Bipartite-Ham Path *if and only if (H, P, Q, k) is a yes instance of* Restricted CM-Ham Path.

Proof. In the forward direction let S be a Hamiltonian path in G with x^* as the first vertex. Recall that $P_1 = c_1c'_1c_2c'_2 \dots c_{n-1}c'_{n-1}c_nc'_n$ and $P_2 = s_1s'_1s_2s'_2 \dots s_{t-1}s'_{t-1}s_t\ s'_t$ are paths in H, respectively. Furthermore, by construction we have that $Z = (P_1 \bullet P_2) \bullet S$ is Hamiltonian path in H. Since S is a path in H with $2n-1$ edges, it has at most $k = \binom{2n-1}{2}$ pairwise crossing edges. Moreover, no edges in $E(Z) \setminus E(S)$ crosses an edge in $E(Z)$. Therefore, Z is a Hamiltonian path in H with at most k crossings.

In the reverse direction, let Z be a Hamiltonian path in H with at most k crossings. Let $E' = \{y_ic_j, x_ic'_j \mid i, j \in [n]\} \cap E(H)$. From Lemma 25 it follows that $E(Z) \cap E' = \emptyset$. Therefore, Z is a Hamiltonian path in the graph $H' = H - E'$. Observe that for each $u \in (X \cup Y) \setminus \{x_1\}$, we have $N_{H'}(u) \subseteq X \cup Y$. Moreover, $N_{H'}(x_1) \subseteq Y \cup \{s'_t\}$. This implies that $Z[(X \cup Y) \setminus \{x_1\}]$ is an induced path. Note that in $H' - \{x_1\}$, there is not path from a vertex in $\{c_i, c'_i \mid i \in [n]\} \cup \{s_i, s'_i \mid i \in [t]\}$ to a vertex in $\{x_i, y_i \mid i \in [n]\}$. Thus $Z[X \cup Y]$ must be an induced path in H', and hence in G. This concludes the proof. ◀

Recall that in the construction of our reduction, for a graph G on n vertices with maximum degree 3 (which is an instance of Bipartite-Ham Path), we create an instance (H, k) of CM-Ham Path, such that $k \in \mathcal{O}(n^2)$ and $|V(H) + E(H)| \in \mathcal{O}(n)$. We note that Bipartite-Ham Path does not admit an algorithm running in time $2^{o(n)}n^{\mathcal{O}(1)}$ (assuming ETH). Thus, we obtain that CM-Ham Path does not admit an algorithm running in time $2^{o(\sqrt{k})}n^{\mathcal{O}(1)}$ (assuming ETH). Also, we can obtain that, unless ETH fails, CM-Ham Path does not admit an algorithm running in time $2^{o(n+m)}n^{\mathcal{O}(1)}$, where n and m are the number of vertices and edges in the input graph, respectively.

4.2 Algorithm for Crossing-Minimizing Hamiltonian Path

Let (G, k) be an instance of CM-Ham Path, with vertex bipartition X and Y. Note that if $|X| \geqslant |Y| + 2$ or $|Y| \geqslant |X| + 2$, then (G, k) is a no-instance, as it does not admit a Hamiltonian path (here we rely on the fact that G is a bipartite graph). Thus, without loss of generality, we assume that $|X| = n$, and $|Y| \in \{n-1, n\}$. We will design an FPT algorithm for CM-Ham Path running in time $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$. Our algorithm will be a dynamic programming algorithm which processes the graph from left to right. That is to say, for each $i = 1, 2, \dots, n$, at stage i, we consider the graph $G_i = G[X_i \cup Y_i]$, the graph induced by $\{x_1, \dots, x_i, y_1, \dots, y_i\}$, and solve a family of subproblems, the solution to one of which will lead to an optimal solution of the entire graph G. We will bound the number of sub-instances that we need to solve at each stage i, for $i \in [n]$, by $2^{\mathcal{O}(\sqrt{k}\log k)}$.

We will first explain the intuition behind our algorithm. Suppose (G, k) is a yes-instance CM-Ham Path and let H be a Hamiltonian path in G from u^* to v^* with $\mathtt{cr}(H) \leqslant k$. Note that in H, each vertex $u \in V(G) \setminus \{u^*, v^*\}$ has degree exactly 2, while u^* and v^* are vertices of degree exactly one. Fix $i \in [n]$, and consider how H saturates the "future vertices," i.e., vertices in $X_{i+1,n} \cup Y_{i+1,n}$. Consider a future vertex, say x_j for some $j > i$. Using the fact that $\mathtt{cr}(H) \leqslant k$, we will show that H cannot have a neighbor of x_j from the set Y_{i-k}. Therefore, the only vertices in $X_i \cup Y_i$ that can possibly be neighbors to vertices in the future belong to the set $X_{i-k,i} \cup Y_{i-k,i}$. Now let us further refine our observation. Let $S \subseteq X_i$ and $S' \subseteq Y_i$ be the set of vertices which have at least one neighbor in H from $X_{i+1,n}$ and $Y_{i+1,n}$, respectively. (We will argue that indeed, $S \subseteq X_{i-k,i}$ and $S' \subseteq Y_{i-k,i}$.) Consider $x_py_{p'} \in E(H)$ and $y_qx_{q'} \in E(H)$, where $p, q \leqslant i$ and $p', q' \geqslant i+1$. Note that the edges $x_py_{p'}$ and $y_qx_{q'}$ intersect each other. Thus, we can deduce that $\mathtt{cr}(H) \geqslant (|S|-1) \cdot (|S'|-1)$. From

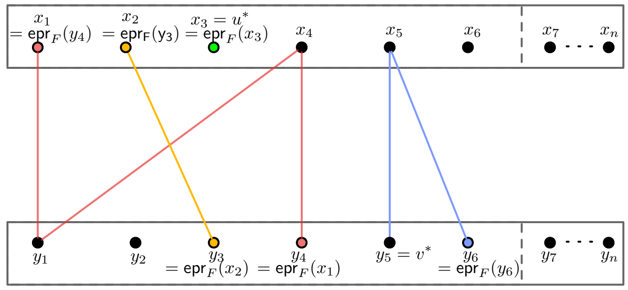


Figure 9 An illustration of a fragmented path set F in the graph G_6. The colored (other than black) vertices are the vertices from the set $\mathsf{ImpEpt}(F)$, and the vertices from $\mathsf{ImpEpt}(F)$ colored the same correspond to the "pairings" given by the function epr_F.

the above discussions we can conclude that at most one of S, S' can be of size at least $\sqrt{k}+2$ (otherwise, we will have $\texttt{cr}(H) > k$). Indeed we will argue that the sizes of both S and S', can be bounded by $\mathcal{O}(\sqrt{k})$, each. Thus, we can "guess" the sets S and S', for each $i \in [n]$, in time bounded by $2^{\mathcal{O}(\sqrt{k}\log k)}$. We note that the above step can also achieved by using the notion of "distinct-part" partitions, that we used in our FPT algorithm for CM-PM. But for the case of CM-Ham Path, this does not offer any significant improvement in the running time (the reason will be clear, when we explain from the dominant factor in the running time of our algorithm).

As was defined earlier, the subsets S and S' of X_i and Y_i, respectively, are vertices with at least one neighbor in the future. Note that some vertices from $S \cup S'$ have exactly one neighbor from the future, while others have two neighbors from the future. To define the states of our algorithm, we need to exactly know, which vertices from $S \cup S'$ have exactly one neighbor and which among them have exactly two neighbors, from the future. Thus, in our algorithm we will have pairs of subsets of X_i and Y_i, which will be determine the vertices we just described.

Note that H, when restricted to the graph G_i, is a collection of disconnected paths. In order to complete these disconnected paths to a Hamiltonian path of the whole graph, we need to remember how the endpoints of the currently "incomplete" H looks like in, the current graph, G_i. Remembering these endpoints seems to be crucial, in order to obtain a Hamiltonian path and to avoid creating cycles. As only $\mathcal{O}(k)$ vertices have neighbors from the "future", there are at most $\mathcal{O}(k)$ paths (which are sub-paths of H), whose endpoints need to be remembered, in the current graph G_i. To remember these "endpoints", we need to spend at least $2^{\mathcal{O}(\sqrt{k}\log k)}$ time. (This is the dominant factor in the running time of our algorithm.)

We start by giving some notations and preliminary results that will be helpful in designing our algorithm.

Notations and Preliminary Results

We will assume that $2 \leqslant |Y| \leqslant |X|$, as otherwise, the problem is polynomial time solvable. Also, we assume that either $|X| = |Y| = n$, or $|X| = n$ and $|Y| = n-1$. We note that if $|Y| = n-1$, then $Y_n = Y_{n-1}$. Furthermore, for $j \in [n-1]$, $Y_{j,n} = Y_{j,n-1}$. Throughout the section, we will only be seeking for a Hamiltonian path from u^* to v^* in G, with at most k crossings.

A (u^*, v^*)-*fragmented path set* (or simply, a fragmented path set) in G_i, is a subgraph of G_i, with $V(F) = V(G_i)$ and $E(F) \subseteq E(G_i)$, such that each connected component of F is a path, u^* and v^* are of degree at most one in F, and $u^*v^* \notin E(F)$. Note that in a fragmented path set, the degree of a vertex belongs to the set $\{0, 1, 2\}$. Consider a fragmented path set F in G_i, for $i \in [n]$. For $r \in \{0, 1, 2\}$, by $\mathsf{Sat}^i_r(F)$, we denote the set of vertices of degree exactly r in F, i.e., $\{v \in V(G_i) \mid d_F(v) = r\}$. For a connected component P of F, by $\mathsf{ept}_1(P)$ and $\mathsf{ept}_2(P)$, we denote the two end vertices of P (possibly $\mathsf{ept}_1(P) = \mathsf{ept}_2(P)$). We will define a set of "important" endpoints of F. In our algorithm components which are of size at least 2 and components containing u^* or v^* will be particularly important. This leads us to the following definition. Let $\mathsf{ImpEpt}(F) = \{\mathsf{ept}_1(P), \mathsf{ept}_2(P) \mid P$ is a connected component of F with at least 2 vertices or P contains u^* or $v^*\}$ (see Figure 9). We define a function which pairs up the endpoints of paths in the components of F. We let $\mathsf{epr}_F : \mathsf{ImpEpt}(F) \to \mathsf{ImpEpt}(F)$, such that $\mathsf{epr}_F(\mathsf{ept}_1(P)) = \mathsf{ept}_2(P)$ and $\mathsf{epr}_F(\mathsf{ept}_2(P)) = \mathsf{ept}_1(P)$.

Some important sets for the algorithm. Recall our assumption $|X| = n$ and $|Y| \in \{n-1, n\}$. For $i \in [n]$, we let $\widehat{X}_i = \{x_{i-k+\ell} \mid \ell \in [k]_0$ and $i - k + \ell \geqslant 1\}$ and $\widehat{Y}_i = \{y_{i-k+\ell} \mid \ell \in [k]_0$ and $i - k + \ell \geqslant 1\}$. We note that in the above definition, we have $\ell \in [k]_0$ (in contrast to $\ell \in [k]$ in a similar definition from Section 3.2). This is used to cater for the condition that sizes of X and Y need not be the same (they can differ by at most one). Roughly speaking, we will argue that in any Hamiltonian path, say H in G, with $\mathtt{cr}(H) \leqslant k$, the vertices from X_i (resp. Y_i) which are a neighbor to a vertex y_s (resp. x_s) in H, where $s \geqslant i+1$, belong to the set $\widehat{X}_i$ (resp. $\widehat{Y}_i$).

We will now associate costs to vertices (and subsets) of $\widehat{X}_i$ (resp. $\widehat{Y}_i$), which will be helpful in obtaining lower bounds on the number of crossings, the when vertices from $\widehat{X}_i$ (resp. $\widehat{Y}_i$) are adjacent to vertices y_s (resp. x_s), where $s \geqslant i+1$. To this end, consider $i \in [n]$ and a vertex $x_r \in \widehat{X}_i$. We let $\mathsf{cst}_i(x_r) = i + 1 - r$. Since $x_r \in \widehat{X}_i$, we have $r \leqslant i$, and thus, $\mathsf{cst}_i(x_r) \geqslant 1$. For a subset $Q \subseteq \widehat{X}_i$, we let $\mathsf{cst}_i(Q) = \sum_{x \in Q} \mathsf{cst}_i(x)$. Similarly, for $i \in [n]$ and a vertex $y_r \in \widehat{Y}_i$, we let $\mathsf{cst}_i(y_r) = i + 1 - r \geqslant 1$. Moreover, for a subset $Q \subseteq \widehat{Y}_i$, we let $\mathsf{cst}_i(Q) = \sum_{y \in Q} \mathsf{cst}_i(y)$. We note that, for each $i \in [n]$, we have $\mathsf{cst}_i(\emptyset) = 0$.

Now we will introduce some "special" sets of pairs of subsets of $\widehat{X}_i$ and $\widehat{Y}_i$, respectively, for each $i \in [n]$. These sets will be crucially used while creating the sub-instances in our dynamic programming based algorithm. For $i \in [n]$, let $\mathcal{S}^i_X = \{(S_1, S_2) \mid S_1 \subseteq \widehat{X}_i, S_2 \subseteq \widehat{X}_i \setminus \{u^*, v^*\}, S_1 \cap S_2 = \emptyset$, and $|S_1| + |S_2| \leqslant 2\sqrt{k}\}$. Similarly, for $i \in [n]$, let $\mathcal{S}^i_Y = \{(S'_1, S'_2) \mid S'_1 \subseteq \widehat{Y}_i, S'_2 \subseteq \widehat{Y}_i \setminus \{u^*, v^*\}, S'_1 \cap S'_2 = \emptyset$, and $|S'_1| + |S'_2| \leqslant 2\sqrt{k}\}$. In the following observation, we state a result regarding the bounds on the size and the time required for the computation of $\mathcal{S}^i_X$ and $\mathcal{S}^i_Y$, which easily follows from their definitions.

▶ **Observation 27.** *For each $i \in [n]$, the sizes and the times required for the computation of $\mathcal{S}^i_X$ and $\mathcal{S}^i_Y$ are bounded by $2^{\mathcal{O}(\sqrt{k}\log k)}$.*

Proof. The proof follows from the fact that $|\widehat{X}_i|, |\widehat{Y}_i| \leqslant k + 1$. ◀

In the following we state an observation regarding the sets $\mathcal{S}^i_X$ and $\mathcal{S}^i_Y$, which will be useful later.

▶ **Observation 28.** *Consider $i \in [n]$. Let $Q \subseteq \widehat{X}_i$, such that $\mathsf{cst}_i(Q) \leqslant 2k$. Then, for any $Q_1 \subseteq Q$, $Q_2 \subseteq Q \setminus (\{u^*, v^*\} \cup Q_1)$, we have $(Q_1, Q_2) \in \mathcal{S}^i_X$. Similarly, let $Q' \subseteq \widehat{Y}_i$, such that $\mathsf{cst}_i(Q') \leqslant k$. Then, for any $Q'_1 \subseteq Q'$, $Q'_2 \subseteq Q' \setminus (\{u^*, v^*\} \cup Q'_1)$, we have $(Q'_1, Q'_2) \in \mathcal{S}^i_Y$.*

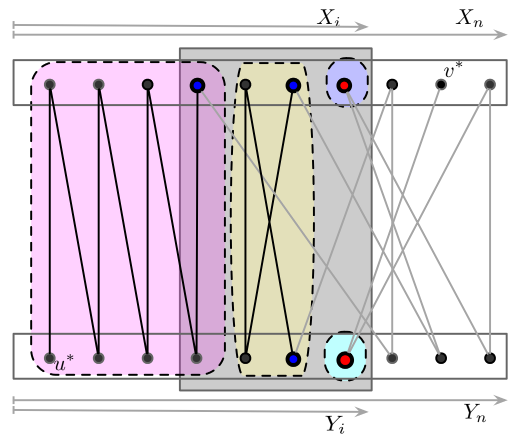


Figure 10 An illustration of the connected components of the fragmented path set F of H. The blue and red vertices are the vertices from sets $S_1 \cup S_1'$ and $S_2 \cup S_2'$, respectively.

Proof. We only prove the first statement. The proof of the second statement can be obtained by following similar arguments. Note that it is enough to show that $|Q| \leqslant 2\sqrt{k}$. If $Q = \emptyset$, then the claim trivially follows. Thus, we assume that $Q \neq \emptyset$. Let $\mathtt{P} = \{\mathsf{cst}_i(x) \mid x \in Q\}$. Note that $\mathtt{P}$ is a partition of an integer $\alpha \leqslant 2k$. Also, distinct $x_j, x_{j'} \in Q$, we have $\mathsf{cst}_i(x_j) \neq \mathsf{cst}_i(x_{j'})$. Hence, $\mathtt{P}$ is a distinct-part partition of $\alpha \leqslant 2k$. We will show that $|\mathtt{P}| \leqslant 2\sqrt{k}$, which is enough to establish the claim. Towards a contradiction, assume that $|\mathtt{P}| \geqslant 2\sqrt{k}+1$. Let $\mathtt{P} = \{\beta_1, \beta_2, \cdots, \beta_\ell\}$, where $1 \leqslant \beta_1 < \beta_2 <, \cdots, < \beta_{\ell-1} < \beta_\ell \leqslant 2k$ (recall that $\mathtt{P}$ is a distinct part partition, so no two elements in it are the same). Thus, we can obtain that $\beta_r \geqslant r$, for each $r \in [\ell]$. The above statement together with our assumption that $\ell \geqslant 2\sqrt{k}+1$, implies that $\sum_{r \in [\ell]} \beta_r \geqslant \sum_{r \in [\ell]} r \geqslant (2\sqrt{k}+1)(2\sqrt{k}+2)/2$. Hence, $\sum_{r \in [\ell]} \beta_r \geqslant (4k+6\sqrt{k}+2)/2 > 2k$. This contradicts that $\mathtt{P}$ is a distinct-part partition of α, where $\alpha \leqslant 2k$. ◀

For each $i \in [n]$, $\mathcal{S} \in \mathcal{S}_X^i$, and $\mathcal{S}' \in \mathcal{S}_X^i$, we will define a set of pairing function $\mathcal{F}_i(\mathcal{S}, \mathcal{S}')$. Roughly speaking, $\mathcal{F}_i(\mathcal{S}, \mathcal{S}')$ will give us a set of potential endpoints belonging to $\widehat{X}_i \cup \widehat{Y}_i$, of the connected components of the fragmented path set F of H, when restricted to vertices in $X_i \cup Y_i$ (see Figure 10 for an intuitive illustration of such components and their endpoints). Consider $i \in [n]$, $\mathcal{S} = (S_1, S_2) \in \mathcal{S}_X^i$ and $\mathcal{S}' = (S_1', S_2') \in \mathcal{S}_X^i$. We let $\mathcal{F}_i(\mathcal{S}, \mathcal{S}')$ be the set injective functions $\mathsf{epr} : S_1 \cup S_1' \cup (V(G_i) \cap \{u^*, v^*\}) \to S_1 \cup S_1' \cup (V(G_i) \cap \{u^*, v^*\})$, such that the following conditions are satisfied: 1) for $u, v \in S_1 \cup S_1' \cup (V(G_i) \cap \{u^*, v^*\})$, if $v = \mathsf{epr}(u)$, then $u = \mathsf{epr}(v)$, 2) for $u \in \{u^*, v^*\} \cap (S_1 \cup S_1')$, we have $\mathsf{epr}(u) = u$, and 3) $\mathsf{epr}(u^*) = v^*$, if and only if $S_1 \cup S_2 \cup S_1' \cup S_2' = \emptyset$ and $i = n$.

In the following observation, we state a result regarding a bound on the size and the time required for the computation of $\mathcal{F}_i(\mathcal{S}, \mathcal{S}')$, which easily follows from its definition.

▶ **Observation 29.** *Consider $i \in [n]$, $\mathcal{S} \in \mathcal{S}_X^i$, and $\mathcal{S}' \in \mathcal{S}_X^i$. Then, the size and the time required for the computation of $\mathcal{F}_i(\mathcal{S}, \mathcal{S}')$ is bounded by $2^{\mathcal{O}(\sqrt{k}\log k)}$.*

We will now associate a set of integers to every pair $(\mathcal{S}, \mathcal{S}') \in \mathcal{S}_X^i \times \mathcal{S}_Y^i$, for each $i \in [n]$. Intuitively speaking, these sets of integers will give the "allowed" number of crossings for the fragmented path set F_i, which is the graph H restricted to vertices of G_i. Consider $i \in [n]$, $\mathcal{S} = (S_1, S_2) \in \mathcal{S}_X^i$, and $\mathcal{S}' = (S_1', S_2') \in \mathcal{S}_Y^i$. We set $\mathsf{Alw}_i(\mathcal{S}, \mathcal{S}') = \{\ell \in [k]_0 \mid \ell \leqslant k - \max\{\mathsf{cst}_i(S_1) + 2 \cdot \mathsf{cst}_i(S_2), \mathsf{cst}_i(S_1') + 2 \cdot \mathsf{cst}_i(S_2')\}$.

Next, we will prove few observations regarding fragmented path sets in G_i. To this end, we first define the notion of a "compatible" fragmented path set.

▶ **Definition 30.** Consider $i \in [n]$, $\mathcal{S} = (S_1, S_2) \in \mathcal{S}_X^i$, $\mathcal{S}' = (S_1', S_2') \in \mathcal{S}_Y^i$, and $\mathsf{epr} \in \mathcal{F}_i(\mathcal{S}, \mathcal{S}')$. We say that a fragmented path set F is G_i is $(i, \mathcal{S}, \mathcal{S}', \mathsf{epr})$-*compatible* if the following conditions are satisfied.

1. $\mathsf{Sat}_0^i(F) = (S_2 \cup S_2') \cup (\{u^*, v^*\} \cap (S_1 \cup S_1'))$.
2. $\mathsf{Sat}_1^i(F) = ((S_1 \cup S_1') \setminus \{u^*, v^*\}) \cup ((\{u^*, v^*\} \cap (X_i \cup Y_i)) \setminus (S_1 \cup S_1'))$.
3. $\mathsf{epr}_F = \mathsf{epr}$.
4. $\mathtt{cr}(F) \in \mathsf{Alw}_i(\mathcal{S}, \mathcal{S}')$.

▶ **Observation 31.** *Consider $i \in [n]$, $\mathcal{S} = (S_1, S_2) \in \mathcal{S}_X^i$, $\mathcal{S}' = (S_1', S_2') \in \mathcal{S}_Y^i$, and $\mathsf{epr} \in \mathcal{F}_i(\mathcal{S}, \mathcal{S}')$. Let F be an $(i, \mathcal{S}, \mathcal{S}', \mathsf{epr})$-compatible fragmented path set in G_i. Then, the following holds.*

1. *If $x_i y_j \in E(F)$, where $j < i$, then $y_j \in \widehat{Y}_{i-1}$.*
2. *Similarly, if $y_i x_j \in E(F)$, where $j < i$, then $x_j \in \widehat{X}_{i-1}$.*

Proof. Let $S = S_1 \cup S_2$ and $S' = S_1' \cup S_2'$. We only prove the first statement, as the proof of the second statement is symmetric. Towards a contradiction, we assume that $y_j \notin \widehat{Y}_{i-1}$, i.e. $j \leqslant i - k - 1$ (recall that $j < i$). Now, consider the set $\widehat{Y}_{i-1}$. Note that $|\widehat{Y}_{i-1}| = k + 1$ and $S_1' \setminus \{y_i\} \subseteq \widehat{Y}_{i-1}$. Let $A = \widehat{Y}_{i-1} \setminus (S_1' \cup S_2' \cup \{u^*, v^*\})$ and $B = S' \setminus \{y_i, u^*, v^*\} \subseteq \widehat{Y}_{i-1}$.

Note that size of A is at least $k+1-(|S_1|+|S_2|+2)$ and each vertex in A has degree exactly 2 in F (see item 1 and 2 of Definition 30). Moreover, for $a \in A$, and the (distinct) edges ua and va in F intersect the edge $x_i y_j$. Similarly, the size of B is at least $|S_1|-3$ and each vertex in B has degree exactly 1 in F. Moreover, for $b \in B$, and the edge ub in F intersects the edge $x_i y_j$. Also, note that $A \cap B = \emptyset$. Thus, $\mathtt{cr}(F) \geqslant 2|A| + |B| \geqslant 2(k+1-(|S_1'|+|S_2'|+1)) + |S_1'| - 3 = 2k + 1 - (|S_1'| + 2|S_2'| + 4)$. Hence, we can obtain that $\mathtt{cr}(F) > k - (\mathsf{cst}_i(S_1') + 2 \cdot \mathsf{cst}_i(S_2'))$. Recall that $\mathsf{Alw}_i(\mathcal{S}, \mathcal{S}') = \{\ell \in [k]_0 \mid \ell \leqslant \max\{\mathsf{cst}_i(S_1) + 2 \cdot \mathsf{cst}_i(S_2), \mathsf{cst}_i(S_1') + 2 \cdot \mathsf{cst}_i(S_2')\}\}$. From the above discussions we can conclude that $\mathtt{cr}(F) \notin \mathsf{Alw}_i(\mathcal{S}, \mathcal{S}')$. This contradicts that F is $(i, \mathcal{S}, \mathcal{S}', \mathsf{epr})$-compatible (see Definition 30). ◀

For a set $\widehat{X} \subseteq X$, we let $\widehat{X}^* = \widehat{X} \setminus \{u^*, v^*\}$. Similarly, for a set $\widehat{Y} \subseteq Y$, we let $\widehat{Y}^* = \widehat{Y} \setminus \{u^*, v^*\}$.

▶ **Observation 32.** *Consider $i \in [n]$, $\mathcal{S} = (S_1, S_2) \in \mathcal{S}_X^i$, $\mathcal{S}' = (S_1', S_2') \in \mathcal{S}_Y^i$, and $\mathsf{epr} \in \mathcal{F}_i(\mathcal{S}, \mathcal{S}')$. Let F be an $(i, \mathcal{S}, \mathcal{S}', \mathsf{epr})$-compatible fragmented path set in G_i. If $x_j y_i \in E(F)$, where $j \leqslant i$, then $x_j y_i$ crosses exactly $2|X_{j+1,i}^* \setminus (S_1 \cup S_2)| + |X_{j+1,i}^* \cap S_1| + |(X_{j+1,i} \cap \{u^*, v^*\}) \setminus S_1|$ many edges of F. Similarly, if $x_i y_j \in E(F)$, where $j \leqslant i$, then $x_i y_{j'}$ crosses exactly $2|Y_{j+1,i}^* \setminus (S_1' \cup S_2')| + |Y_{j+1,i}^* \cap S_1'| + |(Y_{j+1,i} \cap \{u^*, v^*\}) \setminus S_1'|$ many edges of F.*

Dynamic Programming Algorithm for CM-Ham Path

We are now ready to define the states of our dynamic programming table. For each $i \in [n]$, $\mathcal{S} = (S_1, S_2) \in \mathcal{S}_X^i$, $\mathcal{S}' = (S_1', S_2') \in \mathcal{S}_Y^i$, $\mathsf{epr} \in \mathcal{F}[i, S, S']$, and $\ell \in \mathsf{Alw}_i(\mathcal{S}, \mathcal{S}')$, we define $T[i, \mathcal{S}, \mathcal{S}', \mathsf{epr}, \ell]$

$$T[i, \mathcal{S}, \mathcal{S}', \mathsf{epr}, \ell] = \begin{cases} 1, & \text{if there is a fragmented path set } F \text{ in } G_i \text{, such that} \\ & F \text{ is } (i, \mathcal{S}, \mathcal{S}', \mathsf{epr})\text{-compatible and } \mathtt{cr}(F) = \ell, \\ 0, & \text{otherwise.} \end{cases}$$

A fragmented path set F in G_i is said to *realizes* $T[i, \mathcal{S}, \mathcal{S}', \mathsf{epr}, \ell]$, if $\mathtt{cr}(F) = \ell$ and F is $(i, \mathcal{S}, \mathcal{S}', \mathsf{epr})$-compatible.

In the following observation we show how we can use the table entries to resolve the instance (G, k) of CM-Ham Path.

▶ **Observation 33.** *G admits a Hamiltonian path from u^* to v^* with $\mathtt{cr}(H) \leqslant k$ if and only if there is $\widehat{\ell} \in [k]_0$, such that $T[n, (\emptyset, \emptyset), (\emptyset, \emptyset), \mathsf{epr}^*, \widehat{\ell}] = 1$, where $\mathsf{epr}^* : \{u^*, v^*\} \to \{u^*, v^*\}$, such that $\mathsf{epr}(u^*) = v^*$ (and $\mathsf{epr}(v^*) = u^*$).*

Proof. Consider $\widehat{\ell} \in [k]_0$, such that $T[n, (\emptyset, \emptyset), (\emptyset, \emptyset), \mathsf{epr}^*, \widehat{\ell}] = 1$. Furthermore, let F be a fragmented path set that realizes $T[n, (\emptyset, \emptyset), (\emptyset, \emptyset), \mathsf{epr}^*, \widehat{\ell}]$. As $S = S' = \emptyset$, by item 1 and 3 of Definition 30, it follows that each vertex in $V(G) \setminus \{u^*, v^*\}$ has degree exactly two in F. Moreover, from item 2 of the definition, it follows that the degrees of u^* and v^* are exactly one in F. From item 5 of the definition, it follows that $\mathtt{cr}(F) = \widehat{\ell} \leqslant k$. Note that as F is a fragmented path set, no component of it contains a cycle. From the above discussions we can conclude that F is a Hamiltonian path in G from u^* to v^* with at most k crossings.

For the other direction, let H be a Hamiltonian path from u^* to v^* in G with $\mathtt{cr}(H) = \widehat{\ell} \leqslant k$. Observe that $\mathsf{Alw}_n((\emptyset, \emptyset), (\emptyset, \emptyset)) = [k]_0$, and hence $\widehat{\ell} \in \mathsf{Alw}_n((\emptyset, \emptyset), (\emptyset, \emptyset))$. Also, $(\emptyset, \emptyset) \in \mathcal{S}_X^n \cap \mathcal{S}_Y^n$, and the function epr^* belongs to $\mathcal{F}[i, (\emptyset, \emptyset), (\emptyset, \emptyset)]$. It is easy to see that H realizes $T[n, (\emptyset, \emptyset), (\emptyset, \emptyset), \mathsf{epr}_\emptyset : \emptyset \to \emptyset, \widehat{\ell}]$, and thus $T[n, (\emptyset, \emptyset), (\emptyset, \emptyset), \mathsf{epr}^*, \widehat{\ell}] = 1$. ◀

We compute the entries of our dynamic programming table, recursively. The base case occurs when $i = 1$, in which case we can fill each of the entries in polynomial time. Then, we fill all the other entries of our table by using recursive formulae. This can be achieved by an exhaustive case analysis by considering how the vertices x_i and y_i "look like" in the graph G_i, for $i \in [n]$ and $i > 1$.

From Observation 27 and 29 it follows that the number of entries in our table is bounded by $2^{\mathcal{O}(\sqrt{k} \log k)}$. Moreover, an entry of the table can be computed in time bounded by $2^{\mathcal{O}(\sqrt{k} \log k)} n^{\mathcal{O}(1)}$. Thus, the running time of our algorithm is bounded by $2^{\mathcal{O}(\sqrt{k} \log k)} n^{\mathcal{O}(1)}$.

4.3 Kernel for Crossing-Minimizing Hamiltonian Path

We now move on to designing a kernel with $\mathcal{O}(k^2)$ vertices, for CM-Ham Path. Let (G, k) be an instance of CM-Ham Path. Our strategy is to first identify a set of "bad structures" in the graph G. We shall see that the number of bad structures must be $\mathcal{O}(k)$, for otherwise (G, k) would be a no-instance. We then apply a set of reduction rules to bound the number of vertices between two bad structures by $\mathcal{O}(k)$.

We start with the following definitions. For $i \in [n]$, the set $\{x_i, y_i\}$ is called a *duo at index* i; and $\{x_i, y_i\}$ is said to be a *good duo* if $x_i y_i \in E(G)$, and a *bad duo* otherwise. For $i \in [n-1]$, the set $\{x_i, y_i, x_{i+1}, y_{i+1}\}$ is called a *quartet at index i* if both $\{x_i, y_i\}$ and $\{x_{i+1}, y_{i+1}\}$ are good duos, and i is called an index ; and it is said to be a *good quartet* if either $x_i y_{i+1} \in E(G)$ or $x_{i+1} y_i \in E(G)$, and a *bad quartet* otherwise. In the above definitions, the index i is referred to as the index corresponding to the duo or quartet, as the case may be.

For i, j, where $1 \leqslant i < j \leqslant n$ and $|j - i| \geqslant 3$, the set $X_{i,j} \cup Y_{i,j}$ is said to be an *ensemble at* (i, j) if exactly one of the following holds: (i) $x_r y_r \in E(G)$ for every $i \leqslant r \leqslant j$, $x_r y_{r+1}, x_r y_{r-1} \in E(G)$ for every $i + 1 \leqslant r \leqslant j - 1$, but $x_i y_{i+1}, x_j y_{j-1} \notin E(G)$, or (ii) $x_r y_r \in E(G)$ for every $i \leqslant r \leqslant j$, $x_{r-1} y_r, y_r x_{r+1} \in E(G)$ for every $i + 1 \leqslant r \leqslant j - 1$, but $x_{i+1} y_i, x_{j-1} y_j \notin E(G)$.

▶ **Observation 34.** *In polynomial time, we can determine whether G contains a bad duo, a bad quartet, or an ensemble. The cases of duo and quartet must be straightforward as each*

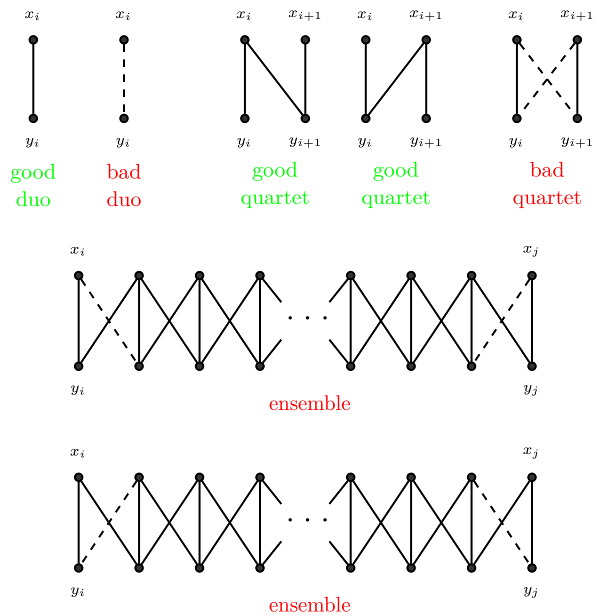


Figure 11 Duos, quartets and ensembles. A dashed line segment shows a non-edge.

such structure has a constant size. As for testing whether G contains an ensemble, we can go over all pairs of indices (i, j) and check whether $X_{i,j} \cup Y_{i,j}$ is an ensemble in polynomial time. ◀

We shall show that the number of ensembles, bad duos and bad quartets cannot exceed $\mathcal{O}(k)$. We need the following two lemmas for that.

▶ **Lemma 35.** *Let (G, k) be an instance of* CM-Ham Path *and let P be a Hamiltonian path in G. If $x_iy_j \in E(P)$, then $\texttt{cr}(P) \geqslant 2|j - i| - 3$. In particular, if $j \geqslant i + 2$, then edge x_iy_j intersects at least $2(j - i) - 3$ edges $x_ry_s \in E(P)$, where $r > i$ and $s < j$; and if $i \geqslant j + 2$, then the edge x_iy_j crosses at least $2(i - j) - 3$ edges $x_ry_s \in E(P)$, where $r < i$ and $s > j$.*

Proof. Assume that $x_iy_j \in E(P)$. If $|j - i| \leqslant 1$, then there is nothing to prove. So, assume that $|j - i| \geqslant 2$, where $j \geqslant i + 2$. (The case where $i \geqslant j + 2$ is symmetric.) Consider the sets Y_{j-1} and $X \setminus X_i$. We claim that $E(P)$ contains at least $2(j - i) - 3$ edges between $X \setminus X_i$ and Y_{j-1}, i.e., edges x_ry_s with $r > i$ and $s < j$. Before moving on to the proof of the above statement, we explain how to use it to obtain the desired result. Note that each edge x_ry_s, with $r > i$ and $s < j$ crosses the edge x_iy_j. Thus, using our claim, we can obtain that the edge x_iy_j crosses at least $2(j - i) - 3$ edges $x_ry_s \in E(P)$, where $r > i$ and $s < j$.

We now prove our claim. Note that each vertex in Y_{j-1}, except possibly two of them (the terminal vertices of P), has degree 2 in P. Therefore, $\sum_{y \in Y_{j-1}} d_P(y) \geqslant 2(j - 1) - 2$. Each vertex in X_i, except x_i, can have at most two neighbors in Y_{j-1}; x_i can have at most one neighbor in Y_{j-1} (as $x_iy_j \in E(P)$). That is, for each $x \in X_{i-1}$, $|N_P(x) \cap Y_{j-1}| \leqslant 2$, and $|N_P(x_i) \cap Y_{j-1}| \leqslant 1$. Therefore, $\sum_{x \in X_i} |N_P(x) \cap Y_{j-1}| \leqslant 2(i - 1) + 1$. In other words, $E(P)$ contains at most $2(i - 1) + 1$ edges between X_i and Y_{j-1}. The remaining $\sum_{y \in Y_{j-1}} d_P(y) - \sum_{x \in X_i} |N_P(x) \cap Y_{j-1}| \geqslant 2(j - 1) - 2 - 2(i - 1) + 1 = 2(j - i) - 3$ edges incident on Y_{j-1} are between Y_{j-1} and $X \setminus X_i$. ◀

▶ **Lemma 36.** *Let (G, k) be an instance of* CM-Ham Path *and let P be a Hamiltonian path in G. If $x_i y_i \notin E(P)$ for some $i \in [n]$, then there is an edge in P incident to exactly one of x_i or y_i that participates in a crossing.*

Proof. Let $x_i y_j \in E(P)$. Consider the case when $j \neq i-1, i+1$, i.e., $|j-i| \geqslant 2$. Now from Lemma 35 the edge $x_i y_j$ crosses at least $2|j-i|-3 \geqslant 1$ edges in P. So now we assume that $N_P(x_i) \subseteq \{y_{i-1}, y_{i+1}\}$. If x_i is not a terminal vertex of P, then it is adjacent to both y_{i-1} and y_{i+1} in P, and then one of the edges incident on y_i intersects either $x_i y_{i-1}$ or $x_i y_{i+1}$. So assume that x_i is a terminal vertex of P. Assume without loss of generality that y_{i-1} is the unique neighbor of x_i in P.

By symmetric arguments, either y_i participates in at least one crossing, or y_i is a terminal vertex of P with either x_{i-1} or x_{i+1} as its unique neighbor. So, assume that y_i is a terminal vertex of P. If x_{i-1} is the unique neighbor of y_i, then the edges $x_i y_{i-1}$ and $x_{i-1} y_i$ intersect each other. So, assume that x_{i+1} is the unique neighbour of y_i.

Consider the $y_{i-1} - x_{i+1}$ subpath of P. Let $x_r y_s$ be the first edge on this subpath with either $r \geqslant i+1$ and $s \leqslant i-1$ or $r \leqslant i-1$ and $s \geqslant i+1$. Such an edge exists as P is connected. So assume that $r \geqslant i+1$ and $s \leqslant i-1$. (The other case is symmetric.) First, note that it cannot be the case that $(r, s) = (i+1, i-1)$, for this would imply that $x_{i+1} y_{i-1} \in E(P)$, and hence $P = x_i y_{i-1} x_{i+1} y_i$, which is not a Hamiltonian path. Therefore, either $r > i+1$, in which case $x_r y_s$ intersects $x_{i+1} y_i$; or $s < i-1$, in which case $x_r y_s$ intersects $x_i y_{i-1}$. ◀

We are now ready to bound the number of vertex disjoint ensembles, bad duos and bad quartets.

▶ **Lemma 37.** *Let (G, k) be an instance of* CM-Ham Path*. If G contains at least $4k+1$ bad duos, then (G, k) is a no-instance.*

Proof. Let P be an optimal Hamiltonian path in G. We will show that if G contains $4k+1$ bad duos, then $\mathtt{cr}(P) \geqslant k+1$. Assume that G contains at least $4k+1$ bad duos. Let $i_1, i_2, \ldots, i_d \in [n]$ be the indices corresponding to bad duos, where $d \geqslant 4k+1$. Then, by Lemma 36, the bad duo $\{x_{i_j}, y_{i_j}\}$ participates in at least one crossing for every $j \in [d]$. Also, note that every crossing in P can involve at most four bad duos. (If $\{e, e'\}$ is a crossing, where $e, e' \in E(P)$, then the four endpoints of e and e' can belong to four different bad duos.) Therefore, the number of distinct crossings involving the d bad duos is at least $\lceil d/4 \rceil \geqslant \lceil (4k+1)/4 \rceil \geqslant k+1$. ◀

▶ **Lemma 38.** *Let (G, k) be an instance of* CM-Ham Path*. If G contains at least $4k+3$ vertex disjoint bad quartets, then (G, k) is a no-instance.*

Proof. Let P be an optimal Hamiltonian path in G. We will show that if G contains $4k+3$ vertex disjoint bad quartets, then $\mathtt{cr}(P) \geqslant k+1$. Assume that G contains at least $4k+3$ vertex disjoint bad quartets. Let $i_1, i_2, \ldots, i_q \in [n]$ be the indices corresponding to them, where $q \geqslant 4k+3$. Consider the set of vertices $\{x_{i_j}, y_{i_j}, x_{i_j+1}, y_{i_j+1} \mid j \in [q]\}$. Each vertex, except two, are of degree 2 in P. Without loss of generality, assume that $d_P(x_{i_j}) = d_P(y_{i_j}) = 2$ and $d_P(x_{i_j+1}) = d_P(y_{i_j+1}) = 2$ for every $j = 3, 4, \ldots, q$.

Consider the edge $x_{i_3} y_{i_3}$. We claim that (at least) one of x_{i_3} or x_{i_3+1} participates in at least one crossing. If $x_{i_3} y_s \in E(P)$ for some $s \neq i_3, i_3-1$, then $|i_3 - s| \geqslant 1$ and hence $2|i_3-1|-3 \geqslant 1$. As shown in the proof of Lemma 35, the edge $x_{i_3} y_s$ participates in at least one crossing. So, assume that y_{i_3} and y_{i_3-1} are the two neighbors of x_{i_3} in P. Similarly, either x_{i_3+1} participates in at lest one crossing, or y_{i_3+1} and y_{i_3+2} are the two neighbors of x_{i_3+1} in P. Now let x_r be a neighbor of y_{i_3}, where $r \neq i_3$. If $r < i_3$, then $x_r y_{i_3}$ and

$x_{i_3}y_{i_3-1}$intersect each other, in which case x_{i_3} participates in a crossing. Note that $r \neq i_3+1$, because $\{x_{i_3}, y_{i_3}, x_{i_3+1}, y_{i_3+1}\}$ is a bad quartet. If $r > i_3+1$, then $x_r y_{i_3}$ intersect both $x_{i_3+1}y_{i_3+1}$ and $x_{i_3+1}y_{i_3+2}$, in which case x_{i_3+1} participates in two crossings. This proves the claim.

The same argument applies to x_{i_j} and x_{i_j+1} as well, for every $3 \leqslant j \leqslant q$. Thus, every bad quartet contains either a terminal vertex of P, or it participates in at least one crossing. Therefore, if there are at least $4k+3$ vertex disjoint bad quartets, then at most two of them contain terminal vertices of P, and hence least $4k+1$ of them participate in at least one crossing. Any crossing in P can involve at most four distinct quartets. Hence, $\texttt{cr}(P) \geqslant k+1$. ◀

▶ **Lemma 39.** *Let (G,k) be an instance of* CM-Ham Path. *If G contains at least $4k+3$ vertex disjoint ensembles, then (G,k) is a no-instance.*

Proof. Let P be an optimal Hamiltonian path in G. We will show that if G contains $4k+3$ vertex disjoint ensembles, then $\texttt{cr}(P) \geqslant k+1$. Assume that G contains at least $4k+3$ vertex disjoint bad ensembles.

Let $X_{i,j} \cup Y_{i,j}$ be an ensemble such that $x_i y_{i+1}, x_j y_{j-1} \notin E(G)$. Also assume that this ensemble does not contain any terminal vertex of P, so that $d_P(v) = 2$ for every $v \in X_{i,j} \cup Y_{i,j}$. We shall show that $X_{i,j} \cup Y_{i,j}$ participates in at least one crossing. Assume that $x_r y_r \in E(P)$ for every r, where $i \leqslant r \leqslant j$, for otherwise, by Lemma 36, $X_{i,j} \cup Y_{i,j}$ would participate in a crossing. Consider the vertex y_{i+1}. Since $x_i y_{i+1} \notin E(G)$, we can assume that $x_{i+2}y_{i+1} \in E(P)$, for otherwise, y_{i+1} would have to be adjacent (in P) to x_r for some r with $|r-(i+1)| \geqslant 2$. Then, by Lemma 35, y_{i+1} would participate in a crossing. We thus have that $x_{i+1}y_{i+1}x_{i+2}y_{i+2}$ is a subpath of P. Observe then that either $x_{i+3}y_{i+2} \in E(P)$, or by Lemma 35, y_{i+2} participates in a crossing. Proceeding this way, we can show that for every $s \geqslant i+1$, either $x_{s+1}y_s \in E(P)$ or y_s participates in a crossing. Then, note that y_{j-1} (for $s = j-1$) must participate in a crossing, since $x_j y_{j-1} \notin E(G)$. ◀

We now proceed as follows. We have already identified certain "bad" sets of vertices, sets of vertices that participate in at least one crossing. As Lemmas 37 - 39 show, there are only $\mathcal{O}(k)$ many of such sets. We mark them. We then show that we can bound the number of vertices in between two consecutive marked sets. Specifically, we do the following. First, mark all the bad duos. Then, we mark bad quartets and ensembles, in that order. (A set of vertices, say $S \subset V(G)$ is marked only if none of its elements is already marked.) For $i \in [n]$, i is said to be the index of a marked set if the bad duo/quartet is marked. Moreover, $i, j \in [n]$ are said to be indices of a marked set, if $X_{i,j} \cup Y_{i,j}$ is a marked ensemble. We now introduce reduction rules that are to applied exhaustively. After an exhaustive application of these rules, the number of unmarked vertices shall be bounded by $\mathcal{O}(k^2)$. The number of vertices in every ensemble shall be bounded by $\mathcal{O}(k)$. In particular, Rule 1, among other things, bounds the number of vertices in ensembles. (See Lemma 41 for precise bounds.)

> **Rule 1:** Let i and j be the indices of two consecutive marked sets such that $j-i > 8k+3$ (or $i=1$ and j be the index corresponding to the first marked index, or $j=n$ and i be the index of the last marked set, or i and j be such that $X_{i,j} \cup Y_{i,j}$ is a marked ensemble.) And, $x_r y_{r+1}, x_{r+1} y_r \in E(G)$ for every r, where $i+4k+1 \leqslant r \leqslant j-4k-1$.
> **Do:** Delete vertices x_r and y_r for every $i+4k+2 \leqslant r \leqslant j-4k-2$, and add edges $x_{i+k+1}y_{j-k-1}$ and $x_{j-k-1}y_{i+k+1}$.
> **Parameter:** No change.

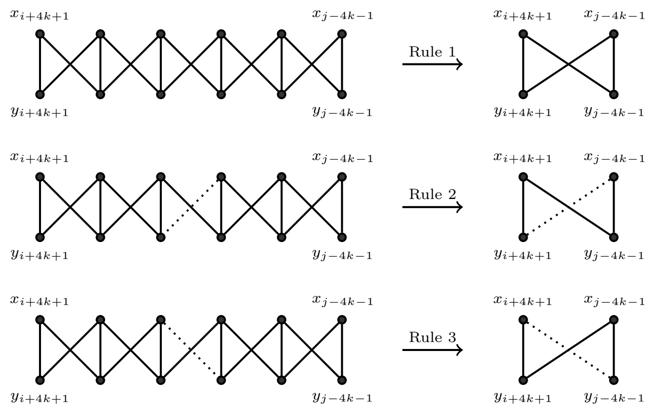


Figure 12 Reduction Rules 1,2 and 3. A dotted segment shows a non-edge.

Rule 2: Let i and j be two consecutive marked indices with $j-i > 8k+3$ (or $i = 1$ and j be the first marked index, or $j = n$ and i be the last marked index.) And, $x_{r+1}y_r \notin E(G)$ for some r, where $i+k+1 \leqslant r \leqslant j-k-1$
Do: Delete vertices x_r and y_r for every $i+k+2 \leqslant r \leqslant j-k-2$, and add edge $x_{i+k+1}y_{j-k-1}$.
Parameter: No change.

Rule 3: Let i and j be two consecutive marked indices with $j-i > 8k+3$ (or $i = 1$ and j be the first marked index, or $j = n$ and i be the last marked index.) And, $x_r y_{r+1} \notin E(G)$ for some r, where $i+k+1 \leqslant r \leqslant j-k-1$
Do: Delete vertices x_r and y_r for every $i+k+2 \leqslant r \leqslant j-k-2$, and add edge $x_{j-k-1}y_{i+k+1}$.
Parameter: No change.

▶ **Lemma 40.** *Rules 1,2 and 3 are safe.*

Proof. We show safeness of Rule 1 only. The proofs for Rules 2 and 3 are analogous. Let (G', k') be the instance obtained from (G, k) by a single application of Rule 1. We shall show that (G, k) is a yes-instance if and only if (G', k') is a yes-instance. Let P be an optimal Hamiltonian path in G with terminal vertices u and v such that $\texttt{index}(u) \leqslant \texttt{index}(v)$.

Note first that $k' = k$. Assume that (G, k) is a yes-instance. Then, $\texttt{cr}(P) \leqslant k$. Consider the $4k+1$ duos $\{x_{i+1}, y_{i+1}\}, \{x_{i+2}, y_{i+2}\} \dots, \{x_{i+4k+1}, y_{i+4k+1}\}$. Since $\texttt{cr}(P) \leqslant k$, and since any crossing can involve at most four of these duos, at least one of these $4k+1$ duos does not participate in any crossing. Let i' be the index corresponding to that duo. Then, by Lemma 36, $x_{i'}y_{i'} \in E(P)$. Similarly, there exists $j' \in \{j-1, j-2, \dots, j-4k-1\}$ such that the duo $\{x_{j'}y_{j'}\}$ does not participate in any crossing and hence $x_{j'}y_{j'} \in E(P)$. Note that in P, no vertex in $X_{i',j'} \cup Y_{i',j'}$ is adjacent to any vertex in $V(G) \setminus (X_{i',j'} \cup Y_{i',j'})$, for otherwise, $\{x_{i'}, y_{i'}\}$ or $\{x_{j'}, y_{j'}\}$ would participate in a crossing. Let $P_{i'j'}$ be the path $x_{i'}y_{i'}x_{i'+1}y_{i'+1}, \dots, x_{j'-1}y_{j'-1}x_{j'}y_{j'}$.

Traverse along P from u to v. Note that on this traversal, among the two edges $x_{i'}y_{i'}$ and $x_{j'}y_{j'}$, the edge $x_{i'}y_{i'}$ appears first. For otherwise, at least one of the duos $\{x_{i'}, y_{i'}\}$ or $\{x_{j'}, y_{j'}\}$ would participate in a crossing. Assume without loss of generality that $x_{i'}$ appears first, followed by $y_{i'}$ in P. Then, P must be as follows: starts from u, passes through all vertices x_r, y_r with $r < i$, enters $x_{i'}$, goes to $y_{i'}$ along the edge $x_{i'}y_{i'}$, passes through all vertices of $(X_{i',j'} \cup Y_{i',j'})$, reaches $y_{j'}$, then passes through all vertices x_r, y_r with $r > j'$, and finally ends at v. Otherwise, at least one of the duos $\{x_{i'}, y_{i'}\}$ or $\{x_{j'}, y_{j'}\}$ would participate in a crossing. Since P passes through all vertices of $(X_{i',j'} \cup Y_{i',j'})$, and since P contains no edge with one endpoint in $(X_{i',j'} \cup Y_{i',j'})$ and the other in $V(G) \setminus (X_{i',j'} \cup Y_{i',j'})$, and since the path $P_{i'j'}$ has no crossings, we can assume that $P_{ij'}$ is a subpath of P. Let P' be the path obtained from P by replacing the x_{i+4k+2}-y_{j-4k-2} subpath with the edge $x_{i+4k+1}y_{j-4k-1}$. Then P' is a Hamiltonian path in G' and $\mathtt{cr}(P) = \mathtt{cr}(P')$.

Conversely, assume that (G', k') is a yes-instance and let P'' be an optimal Hamiltonian path in G'. By repeating the arguments used above, we can show the following: (i) there exist $i' \in \{i+1, i+2, \ldots, i+4k+1\}$ and $j' \in \{j-1, j-2, \ldots, j-4k-1\}$ such that the duos $\{x_{i'}y_{i'}\}$ and $\{x_{j'}, y_{j'}\}$ do not participate in any crossing, (ii) $x_r y_r \in E(P'')$ for every $i' \leqslant r \leqslant j'$, and (iii) either the edge $e = x_{i+4k+1}y_{j-4k-1} \in E(P''')$ or the edge $e' = x_{j-4k-1}y_{i+4k+1} \in E(P'')$. Construct a Hamiltonian path P''' of G by replacing the edge e or e', (whichever is present in P'') with $P_{i',j'}$. Note that the path $P_{i',j'}$ contains no crossings, and therefore, $\mathtt{cr}(P''') = \mathtt{cr}(P'') \leqslant k' = k$. ◀

Rules 1-3 show that we can safely remove vertices between two consecutive marked sets as well as between the boundaries of a marked ensemble, if their number exceeds $\mathcal{O}(k)$. This leads us to the following result.

▶ **Lemma 41.** *Given an instance (G, k) of* CM-Ham Path*, let (G', k') be the instance obtained from (G, k) by an exhaustive application of Rules 1-3. We have the following.*

1. *The number of marked sets in G is at most $3(4k+3)$.*
2. *The number of marked vertices is at most $64k^2 + 104k + 38$.*
3. *The number of vertices between two marked sets (and between (x_1, y_1) and the first marked set, and between the last marked set and (x_n, y_n)) is at most $2(8k+2)$.*
4. *Total number of unmarked vertices in G' is at most $(4(4k+3)+1)(2(8k+2)) + 1 = 256k^2 + 272k + 53$.*
5. *$|V(G')| \leqslant 320k^2 + 376k + 91$.*

Proof. **1.** There are three types of marked sets - bad duos, bad quartets and ensembles. By Lemmas 37 - 39, there can be at most $4k+3$ of each of them. Therefore, there can be at most $3(4k+3)$ marked sets.

2.
- The number of marked duos is at most $4k+1$, and there are 2 vertices in each duo. So, the number of vertices in marked duos is at most $2(4k+1)$.
- The number of marked quartets is at most $4k+3$, and there are 4 vertices in each quartet. So, the number of vertices in marked quartets is at most $4(4k+3)$.
- The number of marked ensembles is at most $4k+3$. Since Rule 1 has been applied exhaustively, there are $2(8k+4)$ vertices in each marked ensemble. If i and j are the indices of a marked ensemble, then there are $8k+4 = (8k+2)$ vertices between x_i and x_j. These vertices, along wih the two vertices x_i and x_j contribute $8k+4$ to the sum. Similarly, y_i and y_j, and the $8k+2$ vertices in between them contribute $8k+4$. Thus, the number of vertices in each marked ensemble is at most $2(8k+4)$. Hence, the number of vertices in marked ensembles is at most $2(4k+3)(8k+4)$.

Adding these bounds, we get that the number of marked vertices is at most $2(4k+1) + 4(4k+3) + 2(4k+3)(8k+4) = 64k^2 + 104k + 38$.

3. Since Rules 1-3 have been applied exhaustively, we must have that the number of vertices between two marked sets (and between (x_1, y_1) and the first marked set, and between the last marked set and (x_n, y_n)) is at most $2(8k+2)$.

4. There are at most $(4k+3)+1$ unmarked “regions” - $(4k+3)-1$ regions between the marked sets, and two additional regions - the one that precedes the first marked set and the one that follows the last marked set. Each such region contains $2(8k+2)$ vertices, (except possibly the last region, which may contain $2(8k+2)+1$ vertices, the “plus 1” owing to the fact that $|X|$ could be $1+|Y|$. Summing up, we get that the number of unmarked vertices is at most $(4(4k+3)+1)(2(8k+2))+1 = 256k^2+272k+53$.
5. Summing up the number of marked and unmarked vertices, we have $|V(G')| \leqslant 320k^2 + 376k+91$.

◀

We have thus proved the following result.

▶ **Theorem 42.** Crossing-Minimizing Hamiltonian Path*, parameterized by the number of crossings k, has a kernel with $\mathcal{O}(k^2)$ vertices.*

5 XP algorithm and W[1]-hardness for Crossing-Minimizing Path

In this section, we show that CM-Path is W[1]-hard, but can be solved in time $n^{\mathcal{O}(k)}$. The problem is formally stated below.

> Crossing-Minimizing Path (CM-Path) **Parameter:** k
> **Input:** A two-layered graph G, vertices $s,t \in V(G)$ and a non-negative integer k.
> **Question:** Does G contain a path from s to t with at most k crossings?

5.1 XP Algorithm for CM-Path

We first consider the case when $k=0$. We show that if $k=0$, then CM-Path can be solved in polynomial time. (We will use this fact while designing the XP algorithm for the general case.) Specifically, we consider the following problem.

> Zero-Crossing Path
> **Input:** A two-layered graph G, vertices $s,t \in V(G)$.
> **Question:** Does G contain a path from s to t with no crossings?

5.1.1 Algorithm for Zero-Crossing Path

Consider an instance (G,s,t) of Zero-Crossing Path. An s-t path in G with no crossings is called a feasible path. We now state and justify some assumptions that we make regarding the vertices s and t.

1. We assume that $s \in X$ and $t \in Y$. If this were not true, then we can arrive at an instance where our assumption is satisfied as follows. If $s \in Y$, then by exchanging the roles of X and Y, we can satisfy our assumption that $s \in X$. Now consider the case when $t \in X$. Note that in the above case, it is enough to find a path between s and a vertex $v \in N(t)$, whose edges do not cross the edge (v,t). If we have an algorithm $\mathcal{A}$ that finds a path with no crossings when our assumption is satisfied, then we can use $\mathcal{A}$ to find a path with no crossings in the case when $s \in X$ and $t \in X$ as follows. For each $(v,t) \in E(G)$, delete all the edges in G that cross the edge (v,t) in G and then delete the edge (v,t). In the resulting graph (with the same two-layer drawing as G) use $\mathcal{A}$ to find a path P_v (if it exists) with zero crossing from s to v. Now add the edge (v,t) to P_v and return it. If for

each $v \in N(t)$, the path P_v does not exist, then report that such a path does not exist. The correctness of the above procedure is clear from its description.

2. $\texttt{index}(s) \leqslant \texttt{index}(t)$. Otherwise, we can reverse the ordering of vertices in X and Y in the two-layer drawing of G, and arrive at our assumption.
3. $s = x_1$ and $t = y_{n_Y}$. To see this, consider a feasible path P. We first claim that P cannot contain two distinct vertices x_i and x_j with $i < \texttt{index}(s)$ and $j > \texttt{index}(s)$. That is, either $\texttt{index}(v) \leqslant \texttt{index}(s)$ for every vertex $v \in V(P) \cap X$, or $\texttt{index}(v) \geqslant \texttt{index}(s)$ for every vertex $v \in V(P) \cap X$. Suppose not. Let $x_i, x_j \in V(P)$ with $i < \texttt{index}(s)$ and $j > \texttt{index}(s)$. Traverse from s to t along P. Assume that in this traversal, x_i appears first and then x_j. (The other case is symmetric.) Let y_j be the unique neighbor of s in P. Then, the x_i-x_j subpath of P must cross the edge sy_s. But this is not possible as P is a feasible path. In light of this claim, we can reduce the given instance of our problem to two instances of the same problem such that the given instance is a yes-instance if and only if at least one of the reduced instances is a yes-instances. To create the first instance, we delete from X all vertices x_i with $i < \texttt{index}(s)$. To create the second instance, we delete from X all vertices x_i with $i > \texttt{index}(s)$ and reverse the orderings of vertices in X and Y. In both the reduced instances, we have $\texttt{index}(s) = 1$. Using symmetric arguments, we can show that $\texttt{index}(t) = n_Y$.

We use a simple dynamic programming algorithm to solve Zero-Crossing Path. For every $i = 1, 2, \ldots, n_X$, $j = 1, 2, \ldots, n_Y$ and $\ell =, 1, 2, \ldots, n-1$, we define $A[i, j, \ell]$ and $B[i, j, \ell]$ as follows.

$$A[i,j,\ell] = \begin{cases} 1, & \text{if } G \text{ contains a feasible path } P \text{ of length } \ell \text{ from } s(= x_1) \text{ to } y_j \\ & \quad \text{such that } (x_i, y_j) \text{ is the last edge of } P. \\ 0, & \text{otherwise.} \end{cases}$$

$$B[i,j,\ell] = \begin{cases} 1, & \text{if } G \text{ contains a feasible path } P \text{ of length } \ell \text{ from } s(= x_1) \text{ to } x_i \\ & \quad \text{such that } (y_j, x_i) \text{ is the last edge of } P. \\ 0, & \text{otherwise.} \end{cases}$$

In the above, the length of a path is the number of edges in it. We start by stating our base cases for the computation of $A[., ., .]$ and $B[., ., .]$. Note that $A[i, j, \ell] = 0$ if ℓ is even, and $B[i, j, \ell] = 0$ if ℓ is odd. Also, $A[i, j, 1] = 1$ if $i = 1$ and $x_1 y_j \in E(G)$, and $A[i, j, 1] = 0$ otherwise. In what follows, consider an odd $1 < \ell \leqslant n - 1$ and an even $1 \leqslant \ell' \leqslant n - 1$. We recursively compute (in order of increasing ℓ and ℓ') $A[., ., .]$ and $B[., ., .]$ as follows.

$$B[i,j,\ell'] = \bigvee_{\substack{i' < i \\ x_{i'} \in N(y_j)}} A[i', j, \ell' - 1] \tag{1}$$

$$A[i,j,\ell] = \bigvee_{\substack{j' < j \\ y_{j'} \in N(x_i)}} B[i, j', \ell - 1] \tag{2}$$

Correctness of the recursive formulae. We show the correctness of Equation 1 (using similar arguments we can establish the correctness of Equation 2). For the forward direction, suppose there is a feasible path P of length ℓ' from x_1 to x_i, where (y_j, x_i) is the last edge.

Consider the neighbor $x_{i^*} \in \{x_1, x_2, \cdots, x_{i-1}\}$ of y_j other than x_i in P. Note that x_{i^*} exists as P is a feasible path from x_1 to x_i and G is a bipartite graph. But then, we have $A[i^*, j, \ell'-1] = 1$, as $P - \{x_i\}$ is a desired type of feasible path. Now for the reverse direction, consider an integer $1 \leqslant i^* < i$, such that $x_{i^*} \in N(y_j)$ and $A[i^*, j, \ell' - 1] = 1$. Let P be a feasible path from x_1 to y_j, where (x_i^*, y_j) is the last edge of P. Furthermore, let P' be the path obtained from P by adding the edge (y_j, x_i). Note that no edge in P crosses the edge (y_j, x_i) as P is a feasible path with (x_i^*, y_j) as the last edge. This implies that P' is a feasible path from x_1 to x_i of length ℓ' with (y_j, x_i) as the last edge. Thus, $B[i, j, \ell'] = 1$.

Note that each $A[i, j, \ell]$ and $B[i, j, \ell]$ can be computed in $\mathcal{O}(n)$ time, and $(G, s = x_1, t = y_{n_Y})$ of Zero-Crossing Path is a yes-instance if and only if $\vee_{i,\ell} A[i, n_Y, \ell] = 1$. Since the number of choices for (i, j, ℓ) is bounded by n^3, we can solve Zero-Crossing Path in $\mathcal{O}(n^4)$ time. For future reference, we state this result as follows.

▶ **Lemma 43.** Zero-Crossing Path*, on an instance (G, s, t) can be solved in time $\mathcal{O}(n^4)$, where $n = |V(G)|$.*

We note that although we gave an algorithm for the decision version of Zero-Crossing Path, we can use memoization to find an $s - t$ path with no crossings, if it exists. This leads us to the following result.

▶ **Lemma 44.** Zero-Crossing Path*, on an instance (G, s, t) can be solved in time $\mathcal{O}(n^4)$, where $n = |V(G)|$. Furthermore, for a yes-instance we can compute an $s - t$ path with no crossings in time $\mathcal{O}(n^4)$.*

5.1.2 Algorithm for Crossing-Minimizing Path

Let (G, s, t, k) be an instance of CM-Path. We first describe the intuition behind the algorithm. Since the desired running time of the algorithm is $n^{\mathcal{O}(k)}$, we have a lot of leeway in "guessing" how a prospective solution looks like. Assume that (G, s, t, k) is a yes-instance of CM-Path, and let P be an $s - t$ path with $\texttt{cr}(P) \leqslant k$. We start by analysing how the path P looks like, in the graph G. Some edges of P are involved in crossings, and some are not. Let E_{crs} be the set of edges in P that participate in at least one crossing. Note that $|E_{\mathsf{crs}}| \leqslant 2k$. Consider the graph $H = P - E_{\mathsf{crs}}$ (where we delete only edges, and not the vertices). Each connected component of H is a path (or an isolated vertex). As $|E_{\mathsf{crs}}| \leqslant 2k$, the number of connected components in H is bounded by $2k + 1$. Consider a connected component $\widehat{P}$ (which is a path) of H which has at least 2 vertices. Let x_a and x_b be the vertices with smallest and largest index in $V(\widehat{P}) \cap X$, respectively. Similarly, let y_c and y_d be the vertices with smallest and largest index in $V(\widehat{P}) \cap Y$, respectively. Note that $\widehat{P}$ is a path in the graph $G[X_{a,b} \cup Y_{c,d}]$, where $x_a, x_b, y_c, y_d \in V(\widehat{P})$ (these vertices need not be all distinct). We will show that no edge in $E(P) \setminus E(\widehat{P})$ has an endpoint from $X_{a+1,b-1} \cup Y_{c+1,d-1}$ and at most two vertices from $\{x_a, x_b, y_c, y_d\}$ can be an endpoint of an edge from $E(P) \setminus E(\widehat{P})$. Recall that edges from $\widehat{P}$ do not participate in any crossings. The above mentioned properties help us to argue that any path $\widetilde{P}$ (with same endpoint as $\widehat{P}$ and no crossings) from the graph $G[X_{a,b} \cup Y_{c,d}]$ can be used instead of $\widehat{P}$, and such a path can easily be computed using the algorithm for Zero-Crossing Path from Section 5.1.1. Roughly speaking, the above arguments allow us to "guess" the endpoints (which are at most $4k + 2$) and (four indices of) the regions in which the paths in $P - E_{\mathsf{crs}}$ are contained. As the size of E_{crs} is bounded by $2k$, we can afford to "guess" the set E_{crs}. Finally, we argue that the computed paths (for some guess of regions and endpoints) and (for some guess of) the edges participating in crossings can be "sewn" together to give an $s - t$ path with at most k crossings (in the case of a yes-instance).

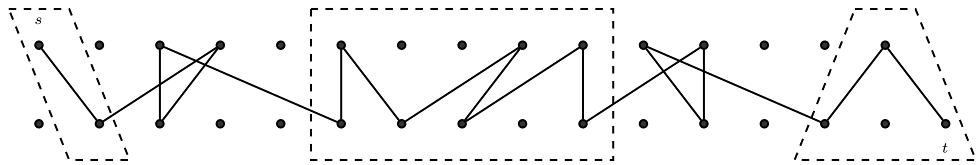


Figure 13 An s-t path. Sets of vertices encased by dashed rectangles form the ℓ-regioning.

Before moving to the formal description of the algorithm, we introduce some notations that we will follow in the remainder of the section.

Notations. For a subset $E' \subseteq E(G)$, $G[\![E']\!]$ denotes the graph with vertex set $V(E')$ and edge set E'. Suppose that P is an $s-t$ path in G that we are seeking for, with $E_{\mathsf{crs}} \subseteq E(P)$ as the set of edges participating in some crossing (in P). Consider an integer $1 \leqslant \ell \leqslant 2k+1$. (Roughly, ℓ is the number of connected components with at least one edge in the graph $P - E_{\mathsf{crs}}$.) Let $A = \{(a,b,c,d,u,v) \mid a,b \in [n_X], c,d \in [n_Y], a \leqslant b, c \leqslant d, u,v \in \{x_a, x_b, y_c, y_d\}, \text{and } u \neq v\}$. (Tuples from A will be used to obtain "regions" and endpoints for paths with at least two vertices in $P - E_{\mathsf{crs}}$.)

For each $i \in [\ell]$, consider some $\mathsf{r}^i = (a^i, b^i, c^i, d^i, u^i, v^i) \in A$. We say that the collection $\{\mathsf{r}^i \mid i \in [\ell]\}$ is an *ℓ-region* (or simply region, when the context is clear) if for every distinct $i,j \in [\ell]$, (exactly) one of the following holds: 1) $a^i \leqslant b^i < a^j \leqslant b^j$ and $c^i \leqslant d^i < c^j \leqslant d^j$, or 2) $a^j \leqslant b^j < a^i \leqslant b^i$ and $c^j \leqslant d^j < c^i \leqslant d^i$. Let $\mathcal{R}_\ell$ be the set of all ℓ-regions. Note that $|\,\mathcal{R}_\ell\,|$ and the time required to compute $\mathcal{R}_\ell$, are both bounded by $n^{\mathcal{O}(k)}$ (as $\ell \leqslant 2k+1$).

Consider an ℓ-region $R = \{\mathsf{r}_i \mid i \in [\ell]\} \in \mathcal{R}_\ell$, where for $i \in [\ell]$, we have $\mathsf{r}_i = (a^i, b^i, c^i, d^i, u^i, v^i)$. R is an *ℓ-important region* (or simply, important region, when the context is clear) if for every $i \in [\ell]$, $(G[X_{a^i,b^i}, Y_{c^i,d^i}], u^i, v^i)$ is a yes-instance of Zero-Crossing Path. In the above, for $i \in [\ell]$, the graph $G[X_{a^i,b^i}, Y_{c^i,d^i}]$ is the two-layered graph with vertex bipartition X_{a^i,b^i} and Y_{c^i,d^i}, where the two-layer drawing is obtained by restricting the two-layer drawing of G to vertices in $X_{a^i,b^i} \cup Y_{c^i,d^i}$. Let $\mathcal{I}_\ell \subseteq \mathcal{R}_\ell$ be the set of all ℓ-important regions. Note that $|\mathcal{I}_\ell|$ is bounded by $n^{\mathcal{O}(k)}$. Moreover, as Zero-Crossing Path admits a polynomial time algorithm (see Section 5.1.1, Lemma 43), we can compute $\mathcal{I}_\ell$ in time bounded by $n^{\mathcal{O}(k)}$.

Algorithm. We are now ready to describe our algorithm. If there is a subset $E' \subseteq E(G)$, such that $G[\![E']\!]$ is an $s-t$ path with $|E'| \leqslant 2k$ and $\texttt{cr}(G[\![E']\!]) \leqslant k$, then return Yes. Hereafter, we assume that such a set E' does not exist. Thus, for any $s-t$ path $\widehat{P}$, such that $\texttt{cr}(\widehat{P}) \leqslant k$ (if it exists), we have $E(\widehat{P}) > 2k+1$, and there is at least one edge in $E(\widehat{P})$ which does not participate in any crossing in $\widehat{P}$.

Consider an integer $1 \leqslant \ell \leqslant 2k+1$, and $R = \{\mathsf{r}^i \mid i \in [\ell]\} \in \mathcal{I}_\ell$, where for $i \in [\ell]$, $\mathsf{r}^i = (a^i, b^i, c^i, d^i, u^i, v^i)$. Using Lemma 44, for each $i \in [\ell]$, we compute a path P^i with endpoints u^i and v^i with zero crossings in the two-layered graph $G[X_{a^i,b^i}, Y_{c^i,d^i}]$. ($P^i$s exist by the definition of important regions.) Let $\widetilde{E} = \cup_{i \in [\ell]} E(P^i)$. Let $\widehat{E}$ be the set of all edges which are in $\widetilde{E}$ or intersect an edge in $\widetilde{E}$. If there is a subset $E' \subseteq E(G) \setminus \widehat{E}$ of size at most $2k$, such that $G[\![\widetilde{E} \cup E']\!]$ is an $s-t$ path with $\texttt{cr}(G[\![\widetilde{E} \cup E']\!]) \leqslant k$, then return Yes.

Otherwise, for no integer $1 \leqslant \ell \leqslant 2k$ and $R \in \mathcal{I}_\ell$, there is $E' \subseteq E(G) \setminus \widehat{E}$ of size at most $2k$, such that $G[\![\widetilde{E} \cup E']\!]$ is an $s-t$ path with $\texttt{cr}(G[\![\widetilde{E} \cup E']\!]) \leqslant k$. In this case, the algorithm return No.

In the following lemma, we show that the algorithm is correct.

▶ **Lemma 45.** *The algorithm presented for* CM-Path *is correct.*

Proof. Notice that if the algorithm returns Yes, then indeed there is an $s-t$ path with at most k crossings. We will now show that if (G, s, t, k) is a yes-instance of CM-Path, then the algorithm returns Yes. If there is an $s-t$ path $\widehat{P}$ with at most $2k$ edges, such that $\mathtt{cr}(\widehat{P}) \leqslant k$, then the algorithm always reports Yes. Otherwise, every $s-t$ path in G with at most k crossings has at least $2k+1$ edges. Let P be an $s-t$ path in G, such that $\mathtt{cr}(P) \leqslant k$ and $E(P) \geqslant 2k+1$. Let $E^* \subseteq E(P)$ be the set of edges which participate in some crossing in P. Note that at most $2k$ edges of P can participate in a crossing, and thus $|E^*| \leqslant 2k$. Let $E_1 = E(P) \setminus E^*$. Let $\mathcal{C}$ be the set of connected components in $G[\![E_1]\!]$, and $\ell^* = |\mathcal{C}|$. As $|E(P)| \geqslant 2k+1$ and $|E^*| \leqslant 2k$, we have $E_1 \neq \emptyset$. Thus, $1 \leqslant \ell^* \leqslant 2k+1$. Each $C \in \mathcal{C}$ is a path on at least 2 vertices, as it is a subgraph of P and contains at least one edge. Let $\mathcal{C} = \{P_1, P_2, \cdots, P_{\ell^*}\}$. Consider $i \in [\ell^*]$. Let u^i and v^i be the end vertices of P^i, where u^i comes before v^i in the path P. Furthermore, let a^i and b^i be the lowest and highest indices of vertices in $V(P^i) \cap X$, respectively (possibly $a^i = b^i$). We note that a^i and b^i exist as G is a bipartite graph and P^i is a path with at least one edge. Similarly, we let c^i and d^i be the lowest and highest indices of vertices in $V(P^i) \cap Y$, respectively. For $i \in [\ell^*]$, we let $\mathsf{r}^i = (a^i, b^i, c^i, d^i, u^i, v^i)$, and $R = \{\mathsf{r}^i \mid i \in [\ell^*]\}$.

We will argue that $R \subseteq A$. (Recall that $A = \{(a, b, c, d, u, v) \mid a, b \in [n_X], c, d \in [n_Y], a \leqslant b, c \leqslant d, u, v \in \{x_a, x_b, y_c, y_d\}, \text{and } u \neq v\}$.) To this end, consider $i \in [\ell^*]$. By construction, we have $a^i \leqslant b^i$, $c^i \leqslant d^i$, $a^i, b^i \in [n_X]$ and $c^i, d^i \in [n_Y]$. As P^i has at least one edge, we have $u^i \neq v^i$. Let $Z^i = \{x_{a^i}, x_{b^i}, y_{c^i}, y_{d^i}\}$. We will now argue that $u^i, v^i \in Z^i$. Towards a contradiction, assume that $u^i \notin Z^i$. (Similar arguments can be given for the case when $v^i \notin Z^i$.) Note that $u^i \in X_{a^i+1, b^i-1} \cup Y_{c^i+1, d^i-1}$. Suppose that $u^i \in X_{a^i+1, b^i-1}$ (the other case is symmetric). Let y_j be the neighbor of u^i in P^i. Note that $y_j \in Y_{c^i, d^i}$. As $u^i \in X_{a^i+1, b^i-1}$, we have $a^i < b^i$. Assume that x_{a^i} is the first vertex in the subpath of P^i from u^i to v^i (the other case is symmetric). Let P' be the subpath of P^i from x_{a^i} to x_{b^i}. As $a^i < b^j$, there is an edge in P' which intersects the edge $u^i y_j$. This contradicts the fact that $\mathtt{cr}(P^i) = 0$. From the above discussions we can conclude that for each $i \in [\ell^*]$, we have $\mathsf{r}^i \in A$.

We now argue that R is an ℓ^*-region. To this end, consider distinct $i, j \in [\ell^*]$. Without loss of generality, we assume that $a^i \leqslant a^j$. By construction, we have $x_{a^i}, x_{b^i}, y_{c^i}, y_{d^i} \in V(P^i)$ and $x_{a^j}, x_{b^j}, y_{c^j}, y_{d^j} \in V(P^j)$. Also, P^i and P^j are distinct connected components in $\mathcal{C}$ with at least one edge each. We recall that edges in P^i and P^j do not participate in any crossings (in P). From the above discussion, we can conclude that $a^i \leqslant b^i < a^j \leqslant b^j$. Now we will argue that $c^i \leqslant d^i < c^j \leqslant c^j$. If $c^j < c^i$, then there will be an edge in P^i and an edge in P^j which will intersect. Note that $c^i \neq c^j$ as $y_{c^i} \in V(P^i)$ and $y_{c^j} \in V(P^j)$, and P^i and P^j are connected components in $\mathcal{C}$. Similarly, we have that $d^i \neq c^j$. If $c^i < c^j < d^i$, then we can obtain a pair of edges in $E(P^i) \cap E(P^j)$ which intersect each other. Thus, we conclude that $c^i \leqslant d^i < c^j \leqslant d^j$. From the above discussions we can conclude that R is an ℓ^*-region.

As R is an ℓ^*-region, and for each $i \in [\ell^*]$, the path P^i is a path from u^i to v^i in the graph $G[X_{a^i, b^i} \cup Y_{c^i, d^i}]$ with $\mathtt{cr}(P^i) = 0$, we can conclude that $R \in \mathcal{I}_{\ell^*}$. In what follows, we observe some properties of edges in E^* (the set of edges participating in a crossing in P) which will be useful later. Consider $xy \in E^*$ and $i \in [\ell^*]$. Observe that $x \notin X_{a^i, b^i} \setminus \{u^i, v^j\}$ and $y \notin Y_{c^i, d^i} \setminus \{u^i, v^j\}$. Furthermore, if $u^i v^i \in E(G)$, we have $xy \neq u^i v^j$. From the above discussions, we can conclude that the edge xy does not belong to the graph $G[X_{a^i, b^i} \cup Y_{c^i, d^i}]$. We also note that the edge xy does not cross any edge in the graph $G[X_{a^i, b^i} \cup Y_{c^i, d^i}]$.

For $i \in [\ell^*]$, let $\widehat{P}^i$ be the path from u^i and v^i in the graph $G[X_{a^i, b^i} \cup Y_{c^i, d^i}]$ with $\mathtt{cr}(\widehat{P}^i) = 0$, computed by the algorithm ($\widehat{P}^i$ exists as $R \in \mathcal{I}_{\ell^*}$). Let $\widetilde{E} = \cup_{i \in [\ell^*]} E(\widehat{P}^i)$. Note that by the properties of E^* discussed earlier, we have $\widetilde{E} \cap E^* = \emptyset$ and no edge of E^* crosses an edge in $\widetilde{E}$. But then, $G[\![\widetilde{E} \cup E^*]\!]$ is an $s-t$ path with $\mathtt{cr}(G[\![\widetilde{E} \cup E^*]\!]) \leqslant k$. Thus, the algorithm will return Yes. This concludes the proof. ◀

▶ **Lemma 46.** *The algorithm presented for* CM-Path *is correct, and runs in time $n^{\mathcal{O}(k)}$, where n is the number of vertices in the input graph.*

Proof. The claimed running time of the algorithm follows from the following facts. The number of important regions, $|\mathcal{I}_\ell|$, and the time required to compute $\mathcal{I}_\ell$ are both bounded by $n^{\mathcal{O}(k)}$. The size of the subsets $E' \subseteq E(G)$ considered by the algorithm is bounded by $2k$. Moreover, Zero-Crossing Path admits an algorithm running in polynomial time (Section 5.1.1, Lemma 44). ◀

Lemma 45 and 46 immediately lead us to the following result.

▶ **Theorem 47.** CM-Path *admits an algorithm running in time $n^{\mathcal{O}(k)}$, where n is the number of vertices in the input graph.*

5.2 W[1]-hardness of Crossing-Minimizing Path

In this section, we show that Crossing-Minimizing Path, when parameterized by the number of crossings is W[1]-hard.

We prove the W[1]-hardness of Crossing-Minimizing Path by giving an appropriate reduction from the problem Multi-Colored Clique, which is known to be W[1]-hard [22]. The Multi-Colored Clique problem is formally defined below.

Multi-Colored Clique	**Parameter:** k
Input: A k-partite graph G with a partition $V_1, V_2, \ldots, V_k$ of $V(G)$ such that for all $i, j \in [k]$, $|V_i| = |V_j|$.	
Question: Is there $X \subseteq V(G)$ such that, for all $i \in [k]$, $|X \cap V_i| = 1$ and $G[X]$ is a clique?	

Let $(G, V_1, V_2, \ldots, V_k)$ be an instance of Multi-Colored Clique. We create an instance (G', X, Y, s, t, k') of Crossing-Minimizing Path such that $(G,\ V_1,\ V_2,\ \ldots,\ V_k)$ is a yes-instance of Multi-Colored Clique if and only if (G', X, Y, s, t, k') is a yes-instance of Crossing-Minimizing Path. Here, G' is a two-layered graph.

The intuitive description of the reduction is as follows (see Figure 14). Let $\varphi : \{(i,j) \mid i,j \in [k], i < j\} \to [\binom{k}{2}]$ be the lexicographic ordering of elements in $\{(i,j) \mid i,j \in [k], i < j\}$. Also, for $r \in [\binom{k}{2}]$, we let $\varphi(r(1), r(2)) = r$. We note that the only use of φ is to order the elements of $\{(i,j) \mid i,j \in [k], i < j\}$, which will be helpful in describing the construction. The main idea behind the construction is to create two special vertices s and t, and create an $s-t$ path in G', which selects a vertex from each V_i, for $i \in [k]$ and an edge between each pair of color classes. Moreover, the number of crossings in such a path will ensure that the selected set of vertices form a clique in G. Towards this, for each V_i, where $i \in [k]$, we have an axis-parallel box $\mathcal{V}_i$, containing an edge (a vertical line) corresponding to each vertex in V_i. Similarly, for each V_i, V_j, where $i, j \in [k]$ and $i < j$, we have an axis-parallel box $\mathcal{E}_{ij}$, containing a pair of non-adjacent vertices corresponding to each edge between V_i and V_j. The boxes $\mathcal{V}_i$, where $i \in [k]$ and $\mathcal{E}_{ij}$, where $i < j$ and $i, j \in [k]$ are arranged in a linear fashion to create an $s-t$ path in G' (see Figure 14). We note that the ordering among boxes $\mathcal{E}_{ij}$s is obtained by using the function φ. In the construction, we added an edge in $\mathcal{V}_i$ corresponding to each vertex in V_i, while we added a pair of (non-adjacent) vertices in $\mathcal{E}_{ij}$ for an edge between V_i and V_j. The motivation behind this is to add a path between the pair of

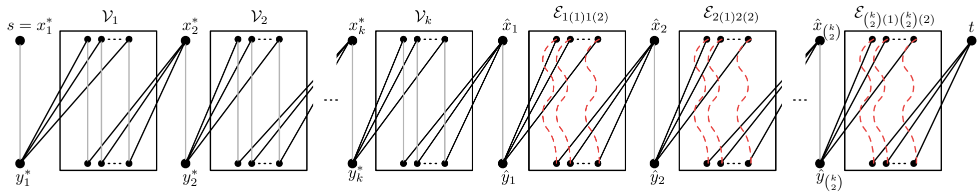


Figure 14 A schema of the reduction. Here, red dotted paths are pairwise vertex disjoint and have vertices outside the box they are drawn in.

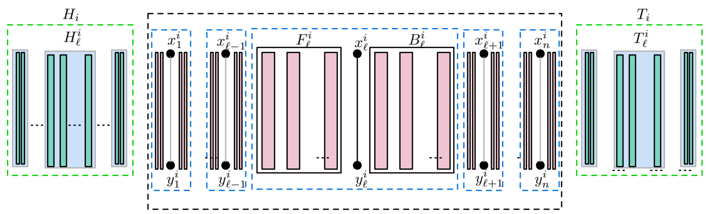


Figure 15 An illustration of various slots and boxes in a vertex selection gadget.

vertices (corresponding to an edge between V_i and V_i), where the edges of this path crosses the boxes $\mathcal{V}_i$ and $\mathcal{V}_j$ so as to ensure that the vertices and edges selected are compatible. Now we move to the formal description of the reduction.

For $i \in [k]$, we let the vertices in V_i to be $\{v_1^i, v_2^i, \ldots, v_n^i\}$. Consider $i, j \in [k]$, where $i \neq j$. We let $E_{ij} = \{e_1^{ij}, e_2^{ij}, \ldots, e_{m_{ij}}^{ij}\}$ be the edges between V_i and V_j, where m_{ij} is the number of edges between V_i and V_j. Note that E_{ij} and E_{ji} are the same sets. Whenever we are considering the vertex set V_i and the edge set E_{ij} $(= E_{ji})$, we will use the lexicographic ordering of edges in E_{ij} whose first coordinate is given by the index of vertex in V_i and the second coordinate is given by the index of vertex in V_j. We will denote such a lexicographic ordering by lex_i^{ij} $(= \mathsf{lex}_i^{ji})$. For $\ell \in [n]$, all the edges in $E(G) \cap \{v_\ell^i v_p^j \mid p \in [n]\}$ appear consecutively in the ordering lex_i^{ij}. Therefore, by $\mathsf{lex}_i^{ij}[\ell]$, we denote the sub-ordering obtained from lex_i^{ij} of edges in $E(G) \cap \{v_\ell^i v_p^j \mid p \in [n]\}$. Also, by $m_{ij}[\ell]$ we denote $|E(G) \cap \{v_\ell^i v_p^j \mid p \in [n]\}|$

Vertex selection gadget. Consider $i \in [k]$. We construct the vertex selection gadget $\mathcal{V}_i$ that will be responsible for selecting a vertex from the color class V_i. The gadget $\mathcal{V}_i$ will be placed in an axis-parallel rectangle (for ease of description). Consider $\ell \in [n]$. Corresponding to the vertex v_ℓ^i, we add an edge $x_\ell^i y_\ell^i$ to $E(H)$ (and to $\mathcal{V}_i$), and add vertices x_ℓ^i and y_ℓ^i to X and Y, respectively. There are two axis-parallel rectangles (often referred to as boxes) F_ℓ^i and B_ℓ^i in the front and the back of the edge $x_\ell^i y_\ell^i$, respectively (see Figure 15). Boxes F_ℓ^i and B_ℓ^i contains $m_{ij}[\ell]$ many slots (small rectangular axis-parallel boxes), where some portion of the Vertex-Edge compatibility gadgets will be placed. We let $\sigma^{\mathcal{V}}(i, X) = (x_1^i, x_2^i, \ldots, x_n^i)$ and $\sigma^{\mathcal{V}}(i, Y) = (y_1^i, y_2^i, \ldots, y_n^i)$. In the rectangle $\mathcal{V}_i$, vertices in $\{x_\ell^i \mid \ell \in [n]\}$ and $\{y_\ell^i \mid \ell \in [n]\}$ are placed in the order given by $\sigma^{\mathcal{V}}(i, X)$ and $\sigma^{\mathcal{V}}(i, Y)$, respectively. The gadget $\mathcal{V}_i$ comprises of two additional rectangular boxes, namely H_i and T_i each containing m_i slots, where $m_i = |\{(v, u) \mid v \in V_i, u \in V(G) \setminus V_i\}|$. These m_i slots are classified into $(k-1)$ groups corresponding to each E_{ij}, where $j \in [k] \setminus \{i\}$. A group of slots allocated for $j \in [k] \setminus \{i\}$ in H_i and T_i will be denoted by H_j^i and T_j^i, respectively. Moreover, H_j^i (and T_j^i) contains m_{ij} consecutive slots, the first group (starting from left) being assigned to the smallest element

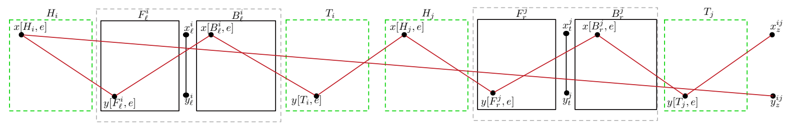


Figure 16 Vertex-Edge compatibility.

in $[k] \setminus \{i\}$. In the slots of H_i and T_i, we place some portion of the Vertex-Edge compatibility gadget for V_i and E_{ij} in the order given by lex_i^{ij}.

Edge selection gadget. Consider $i, j \in [k]$, where $i < j$. We construct the edge selection gadget $\mathcal{E}_{ij}$ that will be responsible for selecting an edge from E_{ij}. The gadget $\mathcal{E}_{ij}$ will be contained in an axis-parallel rectangle, and we would refer to this rectangle by $\mathcal{E}_{ij}$ as well. For each $\ell \in [m_{ij}]$, corresponding to the edge e_ℓ^{ij}, we add two non-adjacent vertices x_ℓ^{ij}, y_ℓ^{ij} to $V(H)$ (and $\mathcal{E}_{ij}$) and add x_ℓ^{ij} and y_ℓ^{ij} to X and Y, respectively. The pairs of vertices x_ℓ^{ij} and y_ℓ^{ij} are placed according to the ordering lex_i^{ij} of edges.

Vertex-Edge compatibility gadgets. As described above, the edge selection gadget consists of a pair of non-adjacent vertices for every edge of G. In order to ensure compatibility between the vertex selection and the edge selection gadgets, we add a path between the two vertices of the edge selection gadget. Consider $i, j \in [k]$, where $i < j$, and an edge $e = e_z^{ij} = v_\ell^i v_r^j \in E_{ij}$. We add a path $P(e)$ between x_z^{ij} and y_z^{ij} with 8 internal vertices as follows (see Figure 16). Towards this we add 8 new vertices as follows. For each $Z \in \{H_i, H_j, B_\ell^i, B_r^j\}$, we add a vertex $x[Z, e]$ (and add it to X). Similarly, for each $Z \in \{T_i, T_j, F_\ell^i, F_r^j\}$, we add a vertex $y[Z, e]$ (and add it to Y). Next, the path $P(e)$ is set to be $y_z^{ij}, x[H_i, e], y[F_\ell^i, e], x[B_\ell^i, e], y[T_i, e], x[H_j, e], y[F_r^j, e], x[B_r^j, e], y[T_j, e], x_z^{ij}$ (see Figure 16).

Overall connections. For each $i \in [k]$, we add an edge $x_i^* y_i^*$, and add the vertex x_i^* to X and y_i^* to Y. The edge $x_i^* y_i^*$ is placed right before the rectangle $\mathcal{V}_i$. Next, we describe the connection between various vertex selection gadgets. For $i \in [k]$, we add all the edges in $\{y_i^* x_j^i \mid j \in [n]\}$ to $E(H)$. For each $i \in [k] \setminus \{1\}$, we add all the edges in $\{y_j^{i-1} x_i^* \mid j \in [n]\}$ to $E(H)$.

Recall that φ is the lexicographic ordering of elements in $\{(i,j) \mid i, j \in [k], i < j\}$. Consider $r \in [\binom{k}{2}]$, and let $(i,j) = \varphi(r)$. Note that $i, j \in [k]$ and $i < j$. We add an edge $\hat{x}_r \hat{y}_r$ (placed before the rectangle $\mathcal{E}_{ij}$), and add the vertex $\hat{x}_r$ to X and $\hat{y}_r$ to Y. Next, we describe the connection between various edge selection gadgets. For $r \in [\binom{k}{2}]$, we add all the edges in $\{\hat{y}_r x_j^{r(1)r(2)} \mid j \in [m_{r(1)r(2)}]\}$ to $E(H)$. For each $r \in [\binom{k}{2}] \setminus \{1\}$, we add all the edges in $\{y_j^{r-1(1)r-1(2)} \hat{x}_r \mid j \in [m_{r(1)r(2)}]\}$ to $E(H)$.

We add a new vertex t to $V(H)$, and make it adjacent to every vertex in $\{y_\ell^{\binom{k}{2}(1)\binom{k}{2}(2)} \mid \ell \in [m_{\binom{k}{2}(1)\binom{k}{2}(2)}]\}$ in H. Also, we set $s = x_1^*$. This completes the description of G', X, Y, s, t. We postpone the description of k', and proceed to prove some structural lemmata which will be useful in determining the appropriate value of k', as well as establishing the equivalence of the instances $(G, V_1, V_2, \ldots, V_k)$ of Multi-Colored Clique and $(G', X, Y, x^\star, y^\star, k')$ of Crossing-Minimizing Path.

▶ **Observation 48.** *For any $s - t$ (simple) path $P^\star$ in G', the following properties hold.*

1. $\{x_i^* y_i^* \mid i \in [k]\} \cup \{\hat{x}_i \hat{y}_i \mid i \in [\binom{k}{2}]\} \subseteq E(P^\star)$.
2. *For each $i \in [k]$, there is a unique $i^* \in [n]$ such that $y_i^* x_{i^*}^i, x_{i^*}^i y_{i^*}^i, y_{i^*}^i x_{i+1}^* \in E(P^\star)$. Here, $x_{i+1}^* = \hat{x}_1$, when $i = k$.*

3. *Consider $r \in [\binom{k}{2}]$, and let $\varphi(i,j) = r$. There is a unique $\ell^*_{ij} \in [m_{ij}]$ such that $\hat{y}_r x^{ij}_{\ell^*_{ij}} \in E(P^\star)$, $P(e^{ij}_{\ell^*_{ij}}) \subseteq P^\star$, and $y^{ij}_{\ell^*_{ij}} \hat{x}_{r+1} \in E(P^\star)$. Here, $\hat{x}_{r+1} = t$, when $r = \binom{k}{2}$.*

Moreover, each edge in $E(P^\star)$ is present in one of the above items.

In the following, let $P^\star$ be an $s - t$ (simple) path. We define the following integers that satisfy the conditions of Observation 48. For $i \in [k]$, we let $i^* \in [n]$ be the integer given by item 2 of Observation 48. Similarly, for each $i, j \in [k]$, where $i < j$, we let $\ell^*_{ij} \in [m_{ij}]$ to be the integer given by item 3 of Observation 48.

In the following, we prove some properties of the path $P^\star$.

▶ **Lemma 49.** *For $\tilde{i} \in [k]$, the number of edges in $P^\star$ that cross the edge $x^*_{\tilde{i}} y^*_{\tilde{i}}$ is exactly $2(\tilde{i} - 1)(k - \tilde{i} + 1) + 2\binom{\tilde{i}-1}{2}$.*

Proof. Consider $\tilde{i} \in [k]$. By construction, the only edges that can potentially cross $x^*_{\tilde{i}} y^*_{\tilde{i}}$, are edges in paths $P(e^{ij}_\ell)$, where $i, j \in [k]$, $i < j$ and $\ell \in [m_{ij}]$. Consider $i, j \in [k]$, where $i < j$, and let $r = \varphi(i,j)$. From Observation 48 (item 3), we know that $\ell^*_{ij} \in [m_{ij}]$ is the unique integer such that $\hat{y}_t x^{ij}_{\ell^*_{ij}} \in E(P^\star)$, $P(e^{ij}_{\ell^*_{ij}}) \subseteq P^\star$, and $y^{ij}_{\ell^*_{ij}} \hat{x}_{t+1} \in E(P^\star)$. Here, $\hat{x}_{t+1} = y^\star$, if $t = \binom{k}{2}$. From the above discussion, there can be no $\ell \neq \ell^*_{ij}$ such that an edge in the path $P(e^{ij}_\ell)$ crosses the edge $x^*_{\tilde{i}} y^*_{\tilde{i}}$. Moreover, some edges in $P(e^{ij}_{\ell^*_{ij}})$ can potentially cross the edge $x^*_{\tilde{i}} y^*_{\tilde{i}}$. In the following, we consider cases based on where $\tilde{i}$ lies in the linear ordering to count the number of edges in $P(e^{ij}_{\ell^*_{ij}})$ that cross the edge $x^*_{\tilde{i}} y^*_{\tilde{i}}$.

- $i < \tilde{i} \leqslant j$. By construction the only edges in $P(e^{ij}_{\ell^*_{ij}})$ that cross $x^*_{\tilde{i}} y^*_{\tilde{i}}$, are $y[T_i, e^{ij}_{\ell^*_{ij}}]$ $x[H_j, e^{ij}_{\ell^*_{ij}}]$ and $x[H_i, e^{ij}_{\ell^*_{ij}}] y^{ij}_{\ell^*_{ij}}$ (see Figure 16 for reference). Therefore, in this case there are two edges in $P(e^{ij}_{\ell^*_{ij}})$ that cross $x^*_{\tilde{i}} y^*_{\tilde{i}}$.
- $\tilde{i} \leqslant i < j$. In this case, by our construction, no edge in $P(e^{ij}_{\ell^*_{ij}})$ crosses $x^*_{\tilde{i}} y^*_{\tilde{i}}$.
- $i < j < \tilde{i}$. By construction the only edges in $P(e^{ij}_{\ell^*_{ij}})$ that cross $x^*_{\tilde{i}} y^*_{\tilde{i}}$, are $x[H_i, e^{ij}_{\ell^*_{ij}}] y^{ij}_{\ell^*_{ij}}$ and $y[T_j, e^{ij}_{\ell^*_{ij}}] x^{ij}_{\ell^*_{ij}}$ (see Figure 16 for reference). Therefore, in this case there are two edges in $P(e^{ij}_{\ell^*_{ij}})$ that cross $x^*_{\tilde{i}} y^*_{\tilde{i}}$.

Hence, the number of edges in $P^\star$ that cross the edge $x^*_{\tilde{i}} y^*_{\tilde{i}}$ is $2(\tilde{i}-1)(k-\tilde{i}+1)+2\binom{\tilde{i}-1}{2}$. ◀

▶ **Lemma 50.** *For $\tilde{r} \in [\binom{k}{2}]$, the number of edges in $P^\star$ that cross the edge $\hat{x}_{\tilde{r}} \hat{y}_{\tilde{r}}$ is exactly $2(\binom{k}{2} - \tilde{r} + 1)$.*

Proof. Consider $\tilde{r} \in [\binom{k}{2}]$. By construction, the only edges that can potentially cross $\hat{x}_{\tilde{r}} \hat{y}_{\tilde{r}}$, are edges in paths $P(e^{ij}_\ell)$, where $i, j \in [k]$, $i < j$ and $\ell \in [m_{ij}]$. Consider $i, j \in [k]$, where $i < j$, and let $r = \varphi(i,j)$.

From Observation 48 (item 3), we know that $\ell^*_{ij} \in [m_{ij}]$ is the unique integer such that $\hat{y}_r x^{ij}_{\ell^*_{ij}} \in E(P^\star)$, $P(e^{ij}_{\ell^*_{ij}}) \subseteq P^\star$, and $y^{ij}_{\ell^*_{ij}} \hat{x}_{r+1} \in E(P^\star)$. Here, $\hat{x}_{r+1} = t$, if $r = \binom{k}{2}$. From the above discussion, there can be no $\ell \neq \ell^*_{ij}$ such that an edge in the path $P(e^{ij}_\ell)$ crosses the edge $\hat{x}_{\tilde{r}} \hat{y}_{\tilde{r}}$ in $P^\star$. Moreover, some edges in $P(e^{ij}_{\ell^*_{ij}})$ can potentially cross the edge $\hat{x}_{\tilde{r}} \hat{y}_{\tilde{r}}$ in $P^\star$. In the following, we consider cases based on where $\tilde{r}$ lies in the linear ordering to count the number of edges in $P(e^{ij}_{\ell^*_{ij}})$ that cross the edge $x^*_{\tilde{r}} y^*_{\tilde{r}}$.

- $r < \tilde{r}$. By construction, there is no edge in $P(e^{ij}_{\ell^*_{ij}})$ that crosses the edge $\hat{x}_{\tilde{r}} \hat{y}_{\tilde{r}}$.

- $r \geqslant \tilde{r}$. In this case, by construction there are two edges namely, $x[H_i, e^{ij}_{\ell^*_{ij}}]y^{ij}_{\ell^*_{ij}}$ and $y[T_j, e^{ij}_{\ell^*_{ij}}]x^{ij}_{\ell^*_{ij}}$ (see Figure 16 for reference) that cross the edge $\hat{x}_{\tilde{r}}\hat{y}_{\tilde{r}}$.

Hence, the number of edges in $P^\star$ that cross the edge $\hat{x}_{\tilde{r}}\hat{y}_{\tilde{r}}$ is $2(\binom{k}{2} - \tilde{r} + 1)$. ◀

▶ **Lemma 51.** *Consider $r \in [\binom{k}{2}]$, and let $\varphi(i,j) = r$. Then the number of edges in $P^\star$ that cross the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$ is exactly $2(\binom{k}{2} - r) + 1$.*

Proof. Consider $r \in [\binom{k}{2}]$, and let $\varphi(i,j) = r$. By construction, the only edges that can potentially cross $\hat{y}_r, x^{ij}_{\ell^*_{ij}}$, are edges in paths $P(e^{i'j'}_{\ell})$, where $i', j' \in [k]$, $i' < j'$ and $\ell \in [m_{i'j'}]$. Consider $i', j' \in [k]$, where $i' < j'$, and let $r' = \varphi(i', j')$. From Observation 48 (item 3), we know that $\ell^*_{i'j'} \in [m_{i'j'}]$ is the unique integer such that $\hat{y}_{r'} x^{i'j'}_{\ell^*_{i'j'}} \in E(P^\star)$, $P(e^{i'j'}_{\ell^*_{i'j'}}) \subseteq P^\star$, and $y^{i'j'}_{\ell^*_{i'j'}} \hat{x}_{r'+1} \in E(P^\star)$. Here, $\hat{x}_{r'+1} = t$, if $r' = \binom{k}{2}$. Thus there is no $\ell \neq \ell^*_{i'j'}$ such that an edge in the path $P(e^{i'j'}_{\ell})$ crosses the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$ in $P^\star$. Moreover, some edges in $P(e^{i'j'}_{\ell^*_{i'j'}})$ can potentially cross the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$ in $P^\star$. In the following, we consider cases based on where r lies in the linear ordering to count the number of edges in $P(e^{i'j'}_{\ell^*_{i'j'}})$ that cross the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$.

- $r' < r$. By construction, there is no edge in $P(e^{i'j'}_{\ell^*_{i'j'}})$ that crosses the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$.
- $r' > r$. In this case, by construction there are two edges namely, $x[H_{i'}, e^{i'j'}_{\ell^*_{i'j'}}]y^{i'j'}_{\ell^*_{i'j'}}$ and $y[T_{j'}, e^{i'j'}_{\ell^*_{i'j'}}]x^{i'j'}_{\ell^*_{i'j'}}$ (see Figure 16 for reference) that cross the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$.
- $r' = r$. The only edge that crosses the $\hat{y}_r x^{ij}_{\ell^*_{ij}}$ is $x[H_i, e^{ij}_{\ell^*_{ij}}]y^{ij}_{\ell^*_{ij}}$.

Hence, the number of edges in $P^\star$ that cross the edge $\hat{y}_r x^{ij}_{\ell^*_{ij}}$ is $2(\binom{k}{2} - r + 1) - 1 = 2(\binom{k}{2} - r) + 1$. ◀

▶ **Lemma 52.** *Consider $r \in [\binom{k}{2}]$, and let $\varphi(i,j) = r$. Then the number of edges in $P^\star$ that cross the edge $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$ is exactly $2(\binom{k}{2} - r)$. Here, $\hat{x}_{r+1} = y^*$, when $r = \binom{k}{2}$.*

Proof. Consider $r \in [\binom{k}{2}]$, and let $\varphi(i,j) = r$. By construction, the only edges that can potentially cross $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$, are edges in paths $P(x^{i'j'}_{\ell})$, where $i', j' \in [k]$, $i' < j'$ and $\ell \in [m_{i'j'}]$. Consider $i', j' \in [k]$, where $i' < j'$, and let $r' = \varphi(i', j')$. From Observation 48 there is no $\ell \neq \ell^*_{i'j'}$ such that an edge in the path $P(e^{i'j'}_{\ell})$ crosses the edge $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$ in $P^\star$. Moreover, some edges in $P(e^{i'j'}_{\ell^*_{i'j'}})$ can potentially cross $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$ in $P^\star$. In the following, we consider cases based on where r lies in the linear ordering to count the number of edges in $P(e^{i'j'}_{\ell^*_{i'j'}})$ that cross the edge $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$.

- $r' \leqslant r$. By construction, there is no edge in $P(e^{i'j'}_{\ell^*_{i'j'}})$ that crosses the edge $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$.
- $r' > r$. In this case, by construction there are two edges namely, $x[H_{i'}, e^{i'j'}_{\ell^*_{i'j'}}]y^{i'j'}_{\ell^*_{i'j'}}$ and $y[T_{j'}, e^{i'j'}_{\ell^*_{i'j'}}]x^{i'j'}_{\ell^*_{i'j'}}$ (see Figure 16 for reference) that cross the edge $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$.

Hence, the number of edges in $P^\star$ that cross the edge $\hat{x}_{r+1} y^{ij}_{\ell^*_{ij}}$ is $2(\binom{k}{2} - r)$. ◀

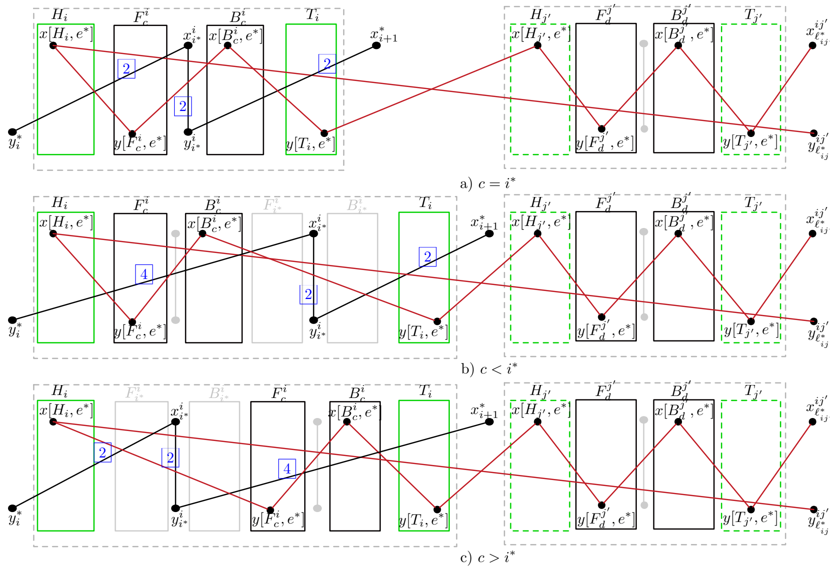


Figure 17 Counting number of edges crossing.

▶ **Lemma 53.** *Consider $i, i', j' \in [k]$ such that $i' < j'$. Let $v_c^{i'} v_d^{j'}) = e^{i'j'}_{\ell^*_{i'j'}} = e^*$. Also, let $\theta = |\{e \mid e \in E(P(e^*))$ and e crosses $y_i^* x^i_{i^*}\}| + |\{e \mid e \in E(P(e^*))$ and e crosses $x^i_{i^*} y^i_{i^*}\}| + |\{e \mid e \in E(P(e^*))$ and e crosses $y^i_{i^*} x^*_{i+1}\}|$. Here, $x^*_{i+1} = \hat{x}_1$ if $i = k$. Then the following conditions hold.*

1. *Consider the case when $i \notin \{i', j'\}$. If $i < i'$ then $\theta = 0$, and otherwise $\theta = 2$.*
2. *Consider the case when $i = i'$. If $c = i^*$ then $\theta = 6$, otherwise, $\theta = 8$.*
3. *Consider the case when $i = j'$. If $d = i^*$ then $\theta = 6$, otherwise, $\theta = 8$.*

Proof. Item 1 follows from the construction. We only consider the case when $i = i'$. The case when $i = j'$ follows from a similar a argument. Next, we consider the following cases based relation between c and i^* (see Figure 17).

- $c = i^*$. In this case, by the construction, edges in $P(e^*)$ which cross:
  - $y_i^* x^i_{i^*}$ are $x[H_i, e^*]y[F^i_c, e^*]$ and $x[H_i, e^*]y^{ij'}_{\ell^*_{ij'}}$;
  - $x^i_{i^*} y^i_{i^*}$ are $y[F^i_c, e^*]x[B^i_c, e^*]$ and $x[H_i, e^*]y^{ij'}_{\ell^*_{ij'}}$;
  - $y^i_{i^*} x^*_{i+1}$ are $x[B^i_c, e^*]y[T_i, e^*]$ and $x[H_i, e^*]y^{ij'}_{\ell^*_{ij'}}$.

  Hence, $\theta = 6$.
- $c < i^*$. In this case, by the construction, edges in $P(e^*)$ which cross:
  - $y_i^* x^i_{i^*}$ are $x[H_i, e^*]y[F^i_c, e^*]$, $x[H_i, e^*]y^{ij'}_{\ell^*_{ij'}}$, $y[F^i_c, e^*]x[B^i_c, e^*]$, and $x[B^i_c, e^*]$ $y[T_i, e^*]$;
  - $x^i_{i^*} y^i_{i^*}$ are $x[B^i_c, e^*]y[T_i, e^*]$ and $x[H_i, e^*]y^{ij'}_{\ell^*_{ij'}}$;
  - $y^i_{i^*} x^*_{i+1}$ are $x[B^i_c, e^*]y[T_i, e^*]$ and $x[H_i, e^*]y^{ij'}_{\ell^*_{ij'}}$.

  Hence, $\theta = 8$.
- $c > i^*$. In this case, by the construction, edges in $P(e^*)$ which cross:

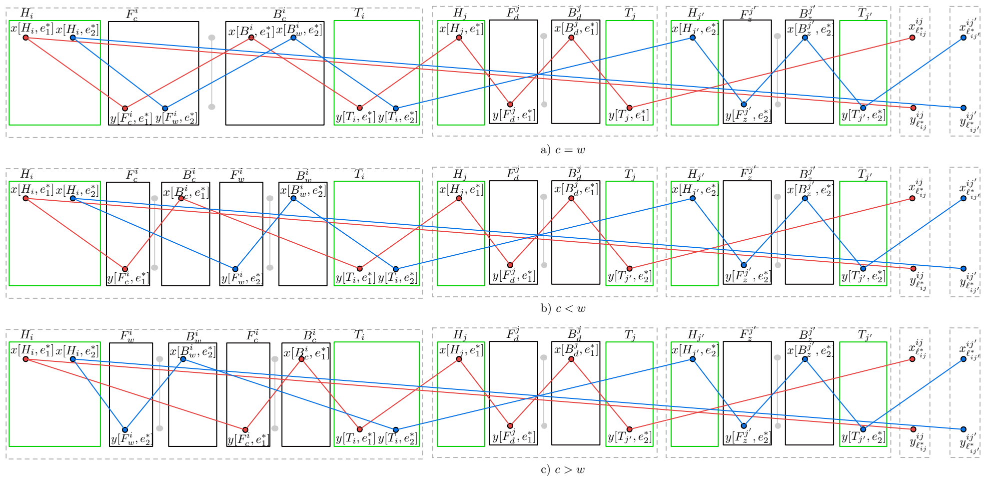


Figure 18 Counting number of edges crossing.

- $y_i^* x_{i^*}^i$ are $x[H_i, e^*]y[F_c^i, e^*]$ and $x[H_i, e^*]y_{\ell^*_{ij'}}^{ij'}$;
- $x_{i^*}^i y_{i^*}^i$ are $x[H_i, e^*]y[F_c^i, e^*]$ and $x[H_i, e^*]y_{\ell^*_{ij'}}^{ij'}$;
- $y_{i^*}^i x_{i+1}^*$ are $x[H_i, e^*]y[F_c^i, e^*]$, $x[H_i, e^*]y_{\ell^*_{ij'}}^{ij'}$, $y[F_c^i, e^*]x[B_v^i, e^*]$ and $x[B_v^i, e^*]$ $y[T_i, e^*]$.

Hence, $\theta = 8$.

◀

In the lemmata that we proved till now, the only pair of edges whose crossing have not been considered belong to paths $P(e_{\ell^*_{ij}}^{ij}) \subseteq P^\star$, $P(e_{\ell^*_{i'j'}}^{i'j'}) \subseteq P^\star$, where $i, i', j, j' \in [k]$, $i < j$ and $i' < j'$. In the following proposition and lemmata, we count such pairs of crossing edges.

▶ **Proposition 54.** *Consider $i, j \in [k]$ with $i < j$. Then the number of pairwise edge crossings in $P(e_{\ell^*_{ij}}^{ij})$ is 7.*

▶ **Lemma 55.** *Consider $i, j, j' \in [k]$, such that $i < j < j'$. Let $e_{\ell^*_{ij}}^{ij} = v_c^i v_d^j$ and $e_{\ell^*_{ij'}}^{ij'} = v_w^i v_z^{j'}$, and $\omega = |\{(e, e') \mid e \in P(e_{\ell^*_{ij}}^{ij}), e' \in P(e_{\ell^*_{ij'}}^{ij'})$ and e crosses $e'\}|$. Exactly one of the following conditions hold.*

1. *If $c \leqslant w$ then $\omega = 24$;*
2. *otherwise, $\omega = 26$.*

Proof. Let $e_1^* = e_{\ell^*_{ij}}^{ij}$ and $e_2^* = e_{\ell^*_{ij'}}^{ij'}$. Since $j < j'$, therefore, all slots in T_j^i lie strictly to the left of slots in $T_{j'}^i$. Therefore, the vertex $x[H_i, e_1^*]$ lies strictly to the left of $x[H_i, e_2^*]$. Similarly, $y[T_i, e_1^*]$ lies strictly to the left of $y[T_i, e_2^*]$. This implies that $x[H_i, e_1^*]y_{\ell^*_{ij}}^{ij}$ crosses every edge in $E(P(e_2^*)) \setminus \{x[H_i, e_2^*]y_{\ell^*_{ij'}}^{ij'}\}$ and does not cross the edge $x[H_i, e_2^*]y_{\ell^*_{ij'}}^{ij'}$. Therefore, $x[H_i, e_1^*]y_{\ell^*_{ij}}^{ij}$ crosses 8 edges in $E(P(e_2^*))$ (see Figure 18). Similarly, the edge $x[H_i, e_2^*]y_{\ell^*_{ij'}}^{ij'}$ crosses every edge in $E(P(e_1^*)) \setminus \{x[H_i, e_1^*]y[F_c^i, e_1^*], x[H_i, e_1^*]y_{\ell^*_{ij}}^{ij}\}$, does not cross the edges $x[H_i, e_1^*]y[F_c^i, e_1^*]$ and $x[H_i, e_1^*]y_{\ell^*_{ij}}^{ij}$, and therefore, it crosses 7 edges in $E(P(e_1^*))$.

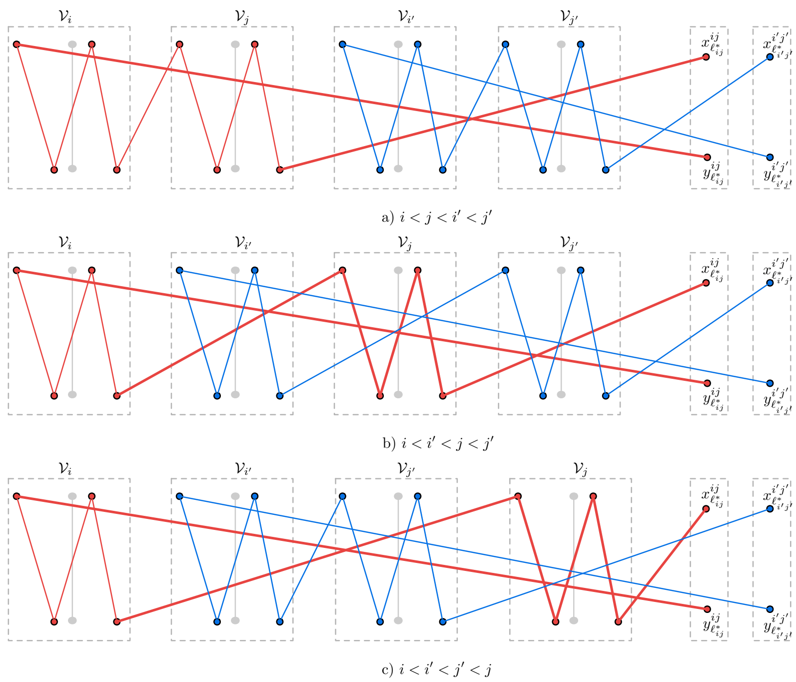


Figure 19 Counting number of edges crossing.

Next, consider the subpaths P_1^* of $P(e_1^*)$ between $x^{ij}_{\ell^*_{ij}}$ and $y[T_i, e_1^*]$ and P_2^* of $P(e_2^*)$ between $x^{ij'}_{\ell^*_{ij'}}$ and $y[T_i, e_2^*]$. By the construction of G' and our assumption that $j < j'$, $|\{(e, e') \mid e \in P_1^*, e' \in P_2^* \text{ and } e \text{ crosses } e'\}|$ is 6. Also, no edge in P_2^* crosses an edge in $E(P(e_1^*)) \setminus (E(P_1^*) \cup \{x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}\})$, and no edge in P_1^* crosses an edge in $E(P(e_2^*)) \setminus (E(P_2^*) \cup \{x[H_i, e_2^*]y^{ij'}_{\ell^*_{ij'}}, x[B^i_w, e_2^*]y[T_i, e_2^*]\})$. By the ordering of vertices T_i, we have that $y[T_i, e_1^*]x[H_j, e_1^*]$ crosses the edge $x[B^i_w, e_2^*]y[T_i, e_2^*]$. Moreover, no edge in $E(P_1^*)\setminus\{y[T_i, e_1^*]x[H_j, e_1^*]\}$ crosses an edge in $E(P(e_2^*)) \setminus (E(P_2^*) \cup \{x[H_i, e_2^*]y^{ij'}_{\ell^*_{ij'}}\})$. Let $\hat{P}_1 = E(P_1^*) \cup \{x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}\}$ and $\hat{P}_2 = E(P_2^*) \cup \{x[H_i, e_2^*]y^{ij'}_{\ell^*_{ij'}}\}$. From the above we have that, $|\{(e, e') \mid e, e' \in \hat{P}_1 \cup \hat{P}_2 \text{ and } e \text{ crosses } e'\}| + |\{(e, e') \mid e \in \hat{P}_1 \cup \hat{P}_2, e' \in (E(P(e_1^*)) \setminus \hat{P}_1) \cup (E(P(e_2^*)) \setminus \hat{P}_2) \text{ and } e \text{ crosses } e'\}| = 22$. In the following we only need to count those crossing edge pairs e, e' such that $e \in E(P(e_1^*)) \setminus \hat{P}_1$ and $e' \in E(P(e_2^*)) \setminus \hat{P}_2$. We consider the following cases based on whether or not $c \leqslant w$.

- $c \leqslant w$. In this case, $y[F^i_c, e_1^*]$ is to the left of $y[F^i_w, e_2^*]$, and the number of desired type of crossing edge pairs is 2.
- $c > w$. In this case, $y[F^i_c, e_1^*]$ is to the right of $y[F^i_w, e_2^*]$, and the number of desired type of crossing edge pairs is 4.

This concludes the proof. ◀

▶ **Lemma 56.** *Consider $i, i', j, j' \in [k]$, where $i < j$, $i' < j'$, and $i < i'$. Let $e_1^* = e^{ij}_{\ell^*_{ij}} = v^i_c v^j_d$, $e_2^* = e^{i'j'}_{\ell^*_{i'j'}} = v^{i'}_w v^{j'}_z$, and $\delta = |\{(e, e') \mid e \in P(e_1^*), e' \in P(e_2^*) \text{ and } e \text{ crosses } e'\}|$. Then the* *following holds.*

1. *If $i < j < i' < j'$, then $\delta = 16$.*
2. *If $i < i' < j < j'$, then $\delta = 21$.*
3. *If $i < i' < j' < j$, then $\delta = 24$.*

Proof. Observe first that the paths $P(e_1^*)$ and $P(e_2^*)$ have nine edges each. (See Figure 19. The red path is $P(e_1^*)$ and the blue path $P(e_2^*)$.)

1. Suppose $i < j < i' < j'$. Then, only two edges of $P(e_1^*)$ – $x^{ij}_{\ell^*_{ij}} y[T_j, e_1^*]$ and $y^{ij}_{\ell^*_{ij}} x[H_i, e_1^*]$ – cross $P(e_2^*)$. Edge $x^{ij}_{\ell^*_{ij}} y[T_j, e_1^*]$ crosses eight of the nine edges of $P(e_2^*)$ – all edges except $x^{i'j'}_{\ell^*_{i'j'}} y[T_{j'}, e_2^*]$. Similarly, edge $y^{ij}_{\ell^*_{ij}} x[H_i, e_1^*]$ crosses eight of the nine edges of $P(e_2^*)$ – all edges except $y^{i'j'}_{\ell^*_{i'j'}} x[H_{i'}, e_2^*]$. Thus, $\delta = 8 + 8 = 16$.
2. Suppose $i < i' < j < j'$. Then six edges of $P(e_1^*)$ cross edges of $P(e_2^*)$.
   - Edge $x^{ij}_{\ell^*_{ij}} y[T_j, e_1^*]$ crosses 3 edges of $P(e_2^*)$. Those three edges are $y[T_{j'}, e_2^*]x[B^{j'}_z, e_2^*]$, $x[B^{j'}_z, e_2^*]y[F^{j'}_z, e_2^*]$ and $y[F^{j'}_z, e_2^*], x[H_{j'}, e_2^*]$.
   - Each of the three edges $y[T_j, e_1^*]x[B^j_d, e_1^*]$, $x[B^j_d, e_1^*]y[F^j_d, e_1^*]$ and $y[F^j_d, e_1^*]x[H_j, e_1^*]$ of $P(e_1^*)$ crosses both the edges $x[H_{j'}, e_2^*], y[T_{i'}, e_2^*]$ and $x[H_{I'}, e_2^*]y^{i'j'}_{\ell^*_{i'j'}}$ of $P(e_2^*)$, thus resulting in $3 \times 2 = 6$ crossings.
   - Edge $x[H_j, e_1^*]y[T_i, e_1^*]$ crosses 4 edges of $P(e_2^*)$. Those four edges are $y[T_{i'}, e_2^*]x[B^{i'}_w, e_2^*]$, $x[B^{i'}_w, e_2^*]y[F^{i'}_w, e_2^*]$, $y[F^{i'}_w, e_2^*]x[H_{i'}, e_2^*]$ and $x[H_{i'}, e_2^*]y^{i'j'}_{\ell^*_{i'j'}}$.
   - Edge $x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}$ of $P(e_1^*)$ crosses eight of the nine edges – all except $x[H_{i'}, e_2^*]y^{i'j'}_{\ell^*_{i'j'}}$ – of $P(e_2^*)$.

   Thus, $\delta = 3 + 6 + 4 + 8 = 21$.
3. Suppose $i < i' < j' < j$. Six edges of $P(e_1^*)$ cross edges of $P(e_2^*)$.
   - Each of the four edges $x^{ij}_{\ell^*_{ij}} y[T_j, e_1^*]$, $y[T_j, e_1^*]x[B^j_d, e_1^*]$, $x[B^j_d, e_1^*]y[F^j_d, e_1^*]$ and $y[F^j_d, e_1^*]x[H_j, e_1^*]$ of $P(e_1^*)$ crosses the two edges $x^{i'j'}_{\ell^*_{i'j'}} y[T_{j'}, e_2^*]$ and $x[H_{I'}, y^{i'j'}_{\ell^*_{i'j'}}]$ of $P(e_2^*)$, thus resulting in $4 \times 2 = 8$ crossings.
   - Edge $x[H_j, e_1^*]y[T_i, e_1^*]$ of $P(e_1^*)$ crosses eight of the nine edges of $P(e_2^*)$ – all edges except $x^{i'j'}_{\ell^*_{i'j'}} y[T_{j'}, e_2^*]$.
   - $x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}$ of $P(e_1^*)$ crosses eight of the nine edges – all except $x[H_{i'}, e_2^*]y^{i'j'}_{\ell^*_{i'j'}}$ – of $P(e_2^*)$.

   Thus, $\delta = 8 + 8 + 8 = 24$.

◀

▶ **Lemma 57.** *Consider $i, j, j' \in [k]$, where $i < j < j'$. Let $e_1^* = e^{ij}_{\ell^*_{ij}} = v^i_c v^j_d$, $e_2^* = e^{jj'}_{\ell^*_{jj'}} = v^j_w v^{j'}_z$, and $\beta = |\{(e, e') \mid e \in P(e_1^*), e' \in P(e_2^*)$ and e crosses $e'\}|$. Exactly one of the following holds.*

1. *If $d = w$, then $\beta = 18$.*
2. *If $d < w$, then $\beta = 18$.*
3. *If $d > w$, then $\beta = 20$.*

Proof. The paths $P(e_1^*)$ and $P(e_2^*)$ have nine edges each. (See Figure 20. Red path is $P(e_1^*)$ and Blue path $P(e_2^*)$.)

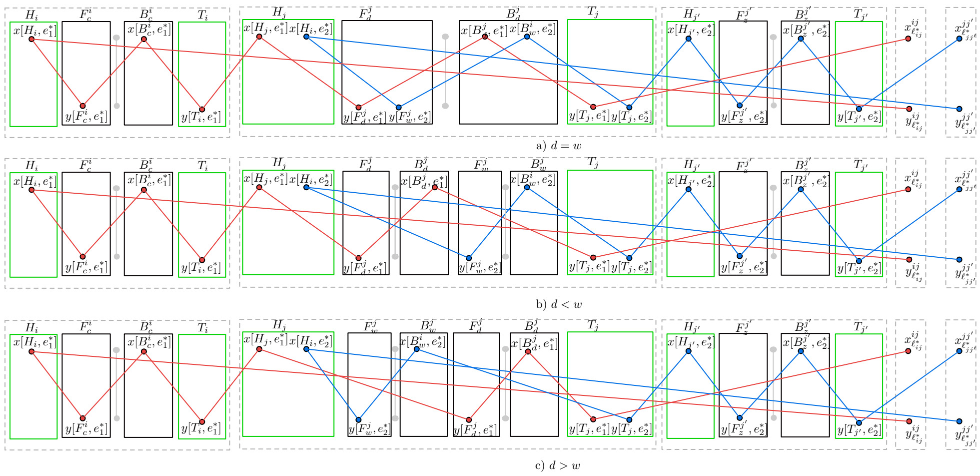


Figure 20 Counting number of edges crossing.

1. Suppose $d = w$. Then four edges of $P(e_1^*)$ cross edges of $P(e_2^*)$.
   - Edge $x^{ij}_{\ell^*_{ij}} y[T_j, e_1^*]$ crosses six of the nine edges of $P(e_2^*)$ – all edges except $x[B^i_w, e_2^*]y[F^j_w, e_2^*]$, $y[F^j_w, e_2^*], x[H_i, e_2^*]$ and $x^{jj'}_{\ell^*_{jj'}} y[T_{j'}, e_2^*]$.
   - Edge $y[T_j, e_1^*]x[B^j_d, e_1^*]$ crosses two edges of $P(e_2^*)$ – $x[B^i_w, e_2^*]y[F^j_w, e_2^*]$ and $x[H_i, e_2^*]y^{jj'}_{\ell^*_{jj'}}$.
   - $x[B^j_d, e_1^*]y[F^j_d, e_1^*]$ crosses two edges of $P(e_2^*)$ – $y[F^j_w, e_2^*], x[H_i, e_2^*]$ and $x[H_i, e_2^*]y^{jj'}_{\ell^*_{jj'}}$.
   - Edge $x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}$ crosses eight of the nine edges of $P(e_2^*)$ – all edges except $x[H_i, e_2^*]y^{jj'}_{\ell^*_{jj'}}$.

   Thus, $\beta = 6 + 2 + 2 + 8 = 18$.
2. Suppose $d < w$. This case is identical to the previous one and we have $\beta = 18$.
3. Suppose $d > w$. In this case, five edges of $P(e_1^*)$ cross edges of $P(e_2^*)$. Four of them are exactly as in the case when $d = w$, thus resulting in 18 crossings. In addition, the edge $y[F^j_d, e_1^*]x[H_i, e_1^*]$ crosses two edges of $P(e_2^*)$ – $x[B^j_w, e_2^*]y[F^j_w, e_2^*]$ and $y[F^j_w, e_2^*]x[H_i, e_2^*]$. Thus, $\beta = 18 + 2 = 20$.

◀

▶ **Lemma 58.** *Consider $i, i', j \in [k]$, where $i < i' < j$. Let $e_1^* = e^{ij}_{\ell^*_{ij}} = v^i_c v^j_d$, $e_2^* = e^{i'j}_{\ell^*_{i'j}} = v^{i'}_w v^j_z$, and $\alpha = |\{(e, e') \mid e \in P(e_1^*), e' \in P(e_2^*)$ and e crosses $e'\}|$. Exactly one of the following holds.*

1. *If $d \leqslant z$, then $\alpha = 20$;*
2. *otherwise $d > z$ $\alpha = 22$.*

Proof. By the construction and the assumption that $i < i' < j$, the edge $x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}$ crosses every edge in $E(P(e_2^*)) \setminus \{x[H_i, e_2^*]y^{i'j}_{\ell^*_{i'j}}\}$ and does not cross $\{x[H_i, e_2^*]y^{i'j}_{\ell^*_{i'j}}\}$. Therefore, $x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}$ crosses 8 edges in P_2^* (see Figure 21). Let $E_1^* = \{x[H_i, e_1^*]y[F^i_c, e_1^*]$, $y[F^i_c, e_1^*]x[B^i_c, e_1^*], x[B^i_c, e_1^*]y[T_i, e_1^*]\}$. None of the edges in E_1^* crosses an edge in P_2^*. The

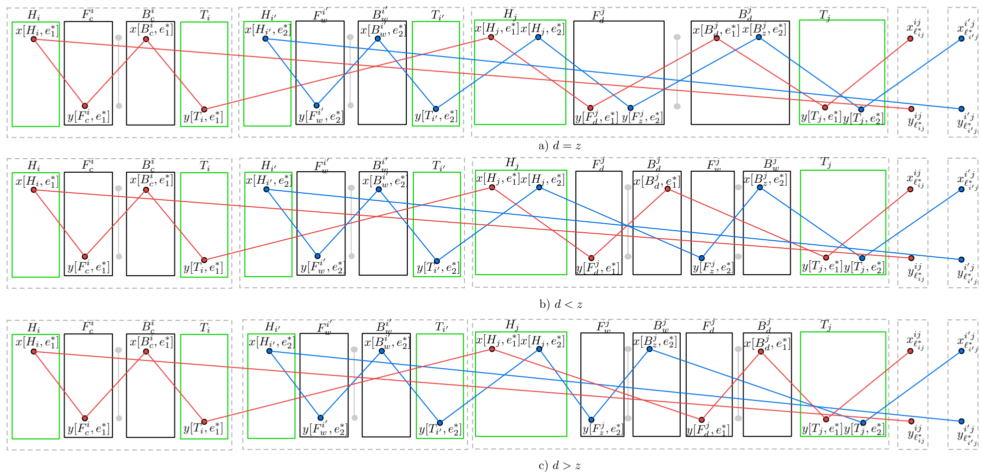


Figure 21 Counting number of edges crossing.

edge $x[H_{i'}, e_2^*], y^{i'j}_{\ell^*_{i'j}}$ crosses each edge in $E(P_1^*) \setminus (E_1^* \cup \{x[H_i, e_1^*]y^{ij}_{\ell^*_{ij}}\})$, and gives is 5 additional pairwise crossings. The edge $y[T_i, e_1^*, x[H_j, e_1^*]] \in E(P_1^*)$ crosses each edge in $\{x[H_{i'}, e_2^*]y[F^{i'}_w, e_2^*], y[F^{i'}_w, e_2^*]x[B^{i'}_w, e_2^*], x[B^{i'}_w, e_2^*]y[T_{i'}, e_2^*]\}$, giving 3 more crossing edge pairs. By ordering of vertices in H_j, we have that the edge $y[T_{i'}, e_2^*]x[H_j, e_2^*]$ crosses the edge $x[H_j, e_1^*]y[F^j_d, e_1^*]$, giving one additional crossing edge pair. Next, we consider cases based on whether or not $d \leqslant z$.

- $d \leqslant z$. By ordering of vertices in H_j, we have 3 additional crossing edges, namely, $\{x[H_j, e_2^*]y[F^j_z, e_2^*], y[F^j_d, e_1^*]x[B^j_d, e_1^*]\}$, $\{x[B^j_d, e_1^*]y[T_j, e_1^*], y[F^j_z, e_2^*]x[B^j_z, e_2^*]\}$, and $\{y[T_j, e_1^*]x^{ij}_{\ell^*_{ij}}, x[B^j_z, e_2^*]y[T_j, e_2^*]\}$. Hence, the total number of crossing edge pairs is 20.
- $d > z$. This together with the ordering of vertices in H_j gives 5 additional crossing edge pairs as follows. The edge $x[H_j, e_1^*]y[F^j_d, e_1^*]$ crosses each of the edges $x[H_j, e_2^*]y[F^j_z, e_2^*]$, $y[F^j_z, e_2^*]x[B^j_z, e_2^*]$, and the edge $x[B^j_z, e_2^*]y[T_j, e_2^*]$ crosses each of the edges $y[F^j_d, e_1^*]$ $x[B^j_d, e_1^*], x[B^j_d, e_1^*]y[T_j, e_1^*], y[T_j, e_1^*]x^{ij}_{\ell^*_{ij}}$. Hence, the total number of crossing edge pairs is 22.

◀

In the following table (Figure 22), we set the value of k' using Lemma 49 to 58. Note that $k' = \mathcal{O}(k^4)$.

▶ **Lemma 59.** *$(G, V_1, V_2, \ldots, V_k)$ is a yes-instance of* Multi-Colored Clique *if and only if (G', X, Y, s, t, k') is a yes-instance of* Crossing-Minimizing Path.

Proof. Suppose that $(G, V_1, V_2, \ldots, V_k)$ is a yes-instance of Multi-Colored Clique, and let H be a clique in G that contains exactly one vertex from each V_i. Then, for each $i \in [k]$, H contains a unique vertex $v^i_{i^*} \in V_i$ ("the selected vertex"), and for every $i, j \in [k], i < j$, H contains the edge $v^i_{i^*}v^j_{j^*}$ ("the selected edge"). The required an (s,t)-path in G' starts at s and traverses along the gadgets corresponding to each of the selected vertices and edges, and finally ends at t.

Crossing with edge(s)	Contribution to the sum ($k' = \sum \cdot$)	Lemma
(x_i^*, y_i^*)	$\sum_{i\in[k]} 2(i-1)(k-i+1) + 2\binom{i-1}{2}$	Lemma 49
$(\hat{x}_i, \hat{y}_i)$	$\sum_{i\in[\binom{k}{2}]} 2(\binom{k}{2} - i + 1)$	Lemma 50
$(\hat{y}_r, x^{ij}_{\ell^*_{ij}})$ Here, $\varphi(i,j) = r$	$\sum_{r\in[\binom{k}{2}]} 2(\binom{k}{2} - r) + 1$	Lemma 51
$(y^{ij}_{\ell^*_{ij}}, \hat{x}_{r+1})$ Here, $\varphi(i,j) = r$	$\sum_{r\in[\binom{k}{2}]} 2(\binom{k}{2} - r)$	Lemma 52
Path $(y_i^*, x^*_{i^*}, y^*_{i^*}, x^*_{i+1})$	$\sum_{i\in[k]} 2\binom{k-i}{2} + 6(k-1)$	Lemma 53
$\mathcal{E}_{ij}$ with $\mathcal{E}_{ij}$ $i < j$	$7\binom{k}{2}$	Proposition 54
$\mathcal{E}_{ij}$ with $\mathcal{E}_{ij'}$s $i < j < j'$	$\sum_{i,j\in[k],i<j} 24(k-j)$	Lemma 55
$\mathcal{E}_{ij}$ with $\mathcal{E}_{i'j'}$s $i < j < i' < j'$	$\sum_{i,j\in[k],i<j} 16\binom{k-j}{2}$	Lemma 56 (item 1)
$\mathcal{E}_{ij}$ with $\mathcal{E}_{i'j'}$s $i < i' < j < j'$	$\sum_{i,j\in[k],i<j} 21(j-i-1)(k-j)$	Lemma 56 (item 2)
$\mathcal{E}_{ij}$ with $\mathcal{E}_{i'j'}$s $i < i' < j' < j$	$\sum_{i,j\in[k],i<j} 24\binom{j-i-1}{2}$	Lemma 56 (item 3)
$\mathcal{E}_{ij}$ with $\mathcal{E}_{jj'}$s $i < j < j'$	$\sum_{i,j\in[k],i<j} 18(k-j)$	Lemma 57
$\mathcal{E}_{ij}$ with $\mathcal{E}_{i'j}$s $i < i' < j$	$\sum_{i,j\in[k],i<j} 20(j-i-1)$	Lemma 58

Figure 22 Setting value of k'.

To see the reverse direction, suppose that (G', X, Y, s, t, k') is a yes-instance of Crossing-Minimizing Path, and let $P^\star$ be an $(s,t) - path$ in G' with at most k' crossings. Then, by Observation 48, $P^\star$ contains the following.

1. $\{x_i^* y_i^* \mid i \in [k]\} \cup \{\hat{x}_i \hat{y}_i \mid i \in [\binom{k}{2}]\} \subseteq E(P^\star)$.
2. For each $i \in [k]$, there is a unique $i^* \in [n]$ such that $y_i^* x^i_{i^*}, x^i_{i^*} y^i_{i^*}, y^i_{i^*} x^*_{i+1} \in E(P^\star)$. Here, $x^*_{i+1} = \hat{x}_1$, when $i = k$.
3. Consider $r \in [\binom{k}{2}]$, and let $\varphi(i,j) = r$. There is a unique $\ell^*_{ij} \in [m_{ij}]$ such that $\hat{y}_r x^{ij}_{\ell^*_{ij}} \in E(P^\star)$, $P(e^{ij}_{\ell^*_{ij}}) \subseteq P^\star$, and $y^{ij}_{\ell^*_{ij}}, \hat{x}_{r+1} \in E(P^\star)$. Here, $\hat{x}_{r+1} = t$, when $r = \binom{k}{2}$.

That is, $P^\star$ can be thought of as selecting one vertex from each V_i and one edge between every pair V_i and V_j, where $i < j$. We claim that the required clique in G is the subgraph of G induced on $\{v^i_{i^*} \mid i \in [k]\}$. In order to see that this graph is indeed a clique, consider $i, j \in [k]$, where $i < j$. We shall show that $v^i_{i^*}$ and $v^j_{j^*}$ are adjacent in G. We have $P(e^{ij}_{\ell^*_{ij}}) \subseteq P^\star$. Suppose $e^{ij}_{\ell^*_{ij}} = (v^i_c, v^j_d)$. Then, because of our choice of k' and parts 2 and 3 of Lemma 53, it must be the case that $c = i^*$ and $d = j^*$. That is, $e^{ij}_{\ell^*_{ij}}$ is the edge between $v^i_{i^*}$ and $v^j_{j^*}$. This completes the proof. ◀

▶ **Theorem 60.** Crossing-Minimizing Path *is both* NP*-hard and* W[1]-hard *when parameterized by the number of crossings.*

References

1. Swapping labeled tokens on graphs. *Theoretical Computer Science*, 586:81 – 94, 2015. Fun with Algorithms.
2. A. Karim Abu-Affash, Ahmad Biniaz, Paz Carmi, Anil Maheshwari, and Michiel H. M. Smid. Approximating the bottleneck plane perfect matching of a point set. *Comput. Geom.*, 48(9):718–731, 2015.
3. A. Karim Abu-Affash, Paz Carmi, Matthew J. Katz, and Yohai Trabelsi. Bottleneck non-crossing matching in the plane. *Comput. Geom.*, 47(3):447–457, 2014.
4. Jihad Al-Oudatallah, Fariz Abboud, Mazen Khoury, and Hassan Ibrahim. Overlapping signal separation method using superresolution technique based on experimental echo shape. *Advances in Acoustics and Vibration*, pages 1–9, 2017.
5. Victor Alvarez, Karl Bringmann, Radu Curticapean, and Saurabh Ray. Counting crossing-free structures. In *Symposuim on Computational Geometry 2012, SoCG '12, Chapel Hill, NC, USA, June 17-20, 2012*, pages 61–68, 2012.
6. Marc Benkert, Herman J. Haverkort, Moritz Kroll, and Martin Nöllenburg. Algorithms for multi-criteria one-sided boundary labeling. In *Graph Drawing, 15th International Symposium, GD 2007, Sydney, Australia, September 24-26, 2007. Revised Papers*, pages 243–254, 2007.
7. Therese C. Biedl, Franz-Josef Brandenburg, and Xiaotie Deng. Crossings and permutations. In *Proceeding of the 13th International Symposium on Graph Drawing, GD*, volume 3843 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.
8. Édouard Bonnet, Tillmann Miltzow, and Paweł Rzążewski. Complexity of token swapping and its variants. *Algorithmica*, Oct 2017.
9. Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.*, 42(5):1803–1829, 2013.
10. Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Proceedings of the 4th International Workshop on Parameterized and Exact Computation, IWPEC*, volume 5917 of *Lecture Notes in Computer Science*, pages 75–85. Springer, 2009.
11. John Gunnar Carlsson, Benjamin Armbruster, Saladi Rahul, and Haritha Bellam. A bottleneck matching problem with edge-crossing constraints. *Int. J. Comput. Geometry Appl.*, 25(4):245–262, 2015.
12. Xuanwu Chen and Ming S. Lee. A case study on multi-lane roundabouts under congestion: Comparing software capacity and delay estimates with field data. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(2):154–165, 2016.
13. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition.* MIT Press, 2009.
14. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015.
15. Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry*, 4(5):235–282, 1994.
16. Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics.* Springer, 2012.
17. Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.
18. Vida Dujmovic, Michael R. Fellows, Michael T. Hallett, Matthew Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond,

Matthew Suderman, Sue Whitesides, and David R. Wood. On the parameterized complexity of layered graph drawing. In *Algorithms - ESA 2001, 9th Annual European Symposium, Aarhus, Denmark, August 28-31, 2001, Proceedings*, pages 488–499, 2001.

19 Peter Eades and Sue Whitesides. Drawing graphs in two layers. *Theoretical Computer Science*, 131(2):361 – 374, 1994.

20 Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994.

21 Jack Edmonds. Paths, trees and flowers. *Canadian Journal OF Mathematics*, pages 449–467, 1965.

22 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical computer science*, 410(1):53–61, 2009.

23 Per Garder. Pedestrian safety at traffic signals: A study carried out with the help of a traffic conflicts technique. *Accident Analysis & Prevention*, 21(5):435–444, 1989.

24 M R Garey and D S Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, New York, 1979.

25 Michael R Garey and David S Johnson. Crossing number is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.

26 Martin Grohe. Computing crossing numbers in quadratic time. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 231–236, 2001.

27 Magnús M. Halldórsson, Christian Knauer, Andreas Spillner, and Takeshi Tokuyama. Fixed-parameter tractability for non-crossing spanning trees. In *Algorithms and Data Structures, 10th International Workshop, WADS 2007, Halifax, Canada, August 15-17, 2007, Proceedings*, pages 410–421, 2007.

28 Godfrey H Hardy and Srinivasa Ramanujan. Asymptotic formulaæ in combinatory analysis. *Proceedings of the London Mathematical Society*, 2(1):75–115, 1918.

29 Petr Hlinený. Crossing number is hard for cubic graphs. *J. Comb. Theory, Ser. B*, 96(4):455–471, 2006.

30 Petr Hlinený and Marek Dernár. Crossing number is hard for kernelization. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 42:1–42:10, 2016.

31 John E. Hopcroft and Robert Endre Tarjan. Efficient algorithms for graph manipulation [H] (algorithm 447). *Commun. ACM*, 16(6):372–378, 1973.

32 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. URL: `http://dx.doi.org/10.1006/jcss.2000.1727`, `doi:10.1006/jcss.2000.1727`.

33 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. URL: `http://dx.doi.org/10.1006/jcss.2001.1774`, `doi:10.1006/jcss.2001.1774`.

34 Klaus Jansen and Gerhard J. Woeginger. The complexity of detecting crossingfree configurations in the plane. *BIT*, 33(4):580–595, 1993.

35 Michael Junger and Petra Mutzel. *Graph Drawing Software*. Springer-Verlag, Berlin, Heidelberg, 2003.

36 Ken-ichi Kawarabayashi and Bruce A. Reed. Computing crossing number in linear time. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 382–390, 2007.

37 Fabian Klute and Martin Nöllenburg. Minimizing crossings in constrained two-sided circular graph layouts. In *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, pages 53:1–53:14, 2018.

38 Jan Kratochvíl, Anna Lubiw, and Jaroslav Nešetřil. Noncrossing subgraphs in topological layouts. *SIAM J. Discret. Math.*, 4(2):223–244, March 1991.

39 Mukkai S Krishnamoorthy and Narsingh Deo. Node-deletion np-complete problems. *SIAM Journal on Computing*, 8(4):619–625, 1979.

40 Martin I Krzywinski, Jacqueline E Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, 2009.

41 J. Malik, J. Weber, Q. T. Luong, and D. Roller. Smart cars and smart roads. In *Proceedings 6th. British Machine Vision Conference*, pages 367–381, 1995.

42 Dániel Marx and Tillmann Miltzow. Peeling and nibbling the cactus: Subexponential-time algorithms for counting triangulations and related problems. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 52:1–52:16, 2016.

43 Tillmann Miltzow. Subset token swapping on a path and bipartite minimum crossing matchings. In *Order and Geometry Workshop, Problem booklet. http://orderandgeometry2016.tcs.uj.edu.pl/docs/OG2016-ProblemBooklet.pdf.*, pages 5–6, 2016.

44 Tillmann Miltzow, Lothar Narins, Yoshio Okamoto, Günter Rote, Antonis Thomas, and Takeaki Uno. Approximation and hardness of token swapping. In *24th Annual European Symposium on Algorithms, ESA 2016*, pages 66:1–66:15, 2016.

45 Haiko Müller. Hamiltonian circuits in chordal bipartite graphs. *Discrete Mathematics*, 156(1-3):291–298, 1996.

46 C St JA Nash-Williams. Hamiltonian circuits in graphs and digraphs. In *The many facets of graph theory*, pages 237–243, 1969.

47 Monroe M. Newborn and William O. J. Moser. Optimal crossing-free hamiltonian circuit drawings of $\mathrm{k_n}$. *J. Comb. Theory, Ser. B*, 29(1):13–26, 1980.

48 Michael Osigbemeh, Michael Onuu, and Olumuyiwa Asaolu. Design and development of an improved traffic light control system using hybrid lighting system. *Journal of Traffic and Transportation Engineering (English Edition)*, 4(1):88–95, 2017. Special Issue: Driver Behavior, Highway Capacity and Transportation Resilience.

49 Marcus Schaefer. The graph crossing number and its variants: A survey. *The Electronic Journal of Combinatorics*, 20, 04 2013.

50 Carl Sechen. *VLSI placement and global routing using simulated annealing*, volume 54. Springer Science & Business Media, 2012.

51 Micha Sharir and Emo Welzl. On the number of crossing-free matchings, cycles, and partitions. *SIAM J. Comput.*, 36(3):695–720, 2006.

52 Paul Turán. A note of welcome. *Journal of Graph Theory*, 1(1):7–9, 1997.

53 Manuel Wettstein. Counting and enumerating crossing-free geometric graphs. *JoCG*, 8(1):47–77, 2017.

54 Katsuhisa Yamanaka, Erik D. Demaine, Takehiro Ito, Jun Kawahara, Masashi Kiyomi, Yoshio Okamoto, Toshiki Saitoh, Akira Suzuki, Kei Uchizawa, and Takeaki Uno. Swapping labeled tokens on graphs. In Alfredo Ferro, Fabrizio Luccio, and Peter Widmayer, editors, *Fun with Algorithms*, pages 364–375, 2014.

55 Lanbo Zheng and Christoph Buchheim. A new exact algorithm for the two-sided crossing minimization problem. In *Proceedings of the First International Conference on Combinatorial Optimization and Applications, COCOA*, volume 4616 of *Lecture Notes in Computer Science*, pages 301–310. Springer, 2007.