Tractability of Packing Vertex-Disjoint A-Paths Under Length Constraints

Susobhan Bandopadhyay

National Institute of Science Education and Research, An OCC of Homi Bhabha National Institute, Bhubaneswar, Odisha, India

Aritra Banik

National Institute of Science Education and Research, An OCC of Homi Bhabha National Institute, Bhubaneswar, Odisha, India

Diptapriyo Majumdar

Indraprastha Institute of Information Technology Delhi, New Delhi, India

Abhishek Sahu

National Institute of Science Education and Research, An OCC of Homi Bhabha National Institute, Bhubaneswar, Odisha, India

Abstract

Given an undirected graph G and a set $A \subseteq V(G)$, an A-path is a path in G that starts and ends at two distinct vertices of A with intermediate vertices in $V(G) \setminus A$. An A-path is called an (A, ℓ)-path if the length of the path is exactly ℓ. In the (A, ℓ)-Path Packing problem (ALPP), we seek to determine whether there exist k vertex-disjoint (A, ℓ)-paths in G or not.

The problem is already known to be fixed-parmeter tractable when parameterized by $k + \ell$ via color coding while it remains Para-NP-hard when parameterized by k (Hamiltonian Path) or ℓ (P_3-Partition) alone. Therefore, a logical direction to pursue this problem is to examine it in relation to structural parameters. Belmonte et al. initiated a study along these lines and proved that ALPP parameterized by $\mathsf{pw} + |A|$ is W[1]-hard where pw is the pathwidth of G. In this paper, we strengthen their result and prove that it is unlikely that ALPP is fixed-parameter tractable even with respect to a bigger parameter ($|A| + \mathsf{dtp}$) where dtp denotes the distance between G and a path graph (distance to path). We use a randomized reduction to achieve the mentioned result. Toward this, we prove a lemma similar to the influential "isolation lemma": Given a set system $(X, \mathcal{F})$ if the elements of X are assigned a weight uniformly at random from a set of values *fairly large*, then each subset in $\mathcal{F}$ will have a unique weight with *high probability*. We believe that this result will be useful beyond the scope of this paper.

ALPP being hard even for structural parameters like distance to path+$|A|$ rules out the possibility of any FPT algorithms for many well-known other structural parameters, including FVS+$|A|$ and treewidth+$|A|$. There is a straightforward FPT algorithm for ALPP parameterized by vc, the vertex cover number of the input graph. Following this, we consider the parameters CVD (cluster vertex deletion)+$|A|$ and CVD $+|\ell|$ and show the problem to be FPT with respect to these parameters. Note that CVD is incomparable to the treewidth of a graph and has been in vogue recently.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Parameterized complexity, (A, ℓ)-Path Packing, Kernelization, Randomized-Exponential Time Hypothesis, Graph Classes

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.16

Funding *Diptapriyo Majumdar*: Supported by Science and Engineering Research Board (SERB) grant SRG/2023/001592.

Acknowledgements We express our sincere gratitude to Dr. Karthik C. S. and Professor Saket Saurabh for their invaluable suggestions and enlightening discussions.

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 16; pp. 16:1–16:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Disjoint Path problems form a fundamental class within algorithmic graph theory. These well-studied problems seek the largest collection of vertex-disjoint (or edge-disjoint) paths that satisfy specific additional constraints. Notably, in the absence of such constraints, the problem reduces to the classical maximum matching problem. One of the most well-studied variants for disjoint path problems is the Disjoint s-t Path where given a graph G and two vertices s and t, the objective is to find the maximum number of internally vertex-disjoint paths between s and t. This problem is polynomial-time solvable by reducing to max-flow problem using Menger's Theorem.

Another classical version of the disjoint path problem is Mader's $\mathcal{S}$-Path. For a graph G and a set $\mathcal{S}$ of disjoint subsets of $V(G)$, an $\mathcal{S}$-Path is a path between two vertices in different members of $\mathcal{S}$. Given G and $\mathcal{S}$, the objective of Mader's $\mathcal{S}$-Path problem is to find the maximum number of vertex-disjoint $\mathcal{S}$-paths. It is solvable in polynomial time, as demonstrated by Chudnovsky, Cunningham, and Geelen [5]. One closely related and well-known variant of Mader's $\mathcal{S}$-Path problem is the A-Path Packing problem [1, 3, 6, 14, 12]. Given a graph G and a subset of vertices A, an *A-path* is a path in G that starts and ends at two distinct vertices of A, and the internal vertices of the path are from $V(G) \setminus A$. The A-Path Packing problem aims to find the maximum number of vertex disjoint A-paths in G. A-Path Packing problem can be modelled as an $\mathcal{S}$-path problem, where for every vertex v, we create a set $\{v\}$ in $\mathcal{S}$. Consequently, the A-Path Packing problem becomes polynomial-time solvable.

Recently, Golovach and Thilikos [11], have explored an interesting variant of the classical s-t path problem known as Bounded s-t Path problem by introducing additional constraints on path lengths. In this variant, given a graph G, two distinct vertices s and t, and an integer ℓ, one seeks to find the maximum number of vertex disjoint paths between s and t of length at most ℓ. Surprisingly, the problem becomes hard with this added constraint in contrast to the classical s-t path problem. In a similar line of study, Belmonte et al. [1] considered the following variant of the A-Path Packing problem.

(A, ℓ)-Path Packing (ALPP)
Input: An undirected graph $G = (V, E)$, $A \subseteq V(G)$ and integers k and ℓ.
Question: Are there k vertex-disjoint A-paths each of length ℓ in G?

This version of A-Path Packing problem is also proved to be intractable [1]. While considering this problem in the parameterized framework, the two most natural parameters for ALPP are the solution size k and the length constraint ℓ. While parameterized by the combined parameter of $k + \ell$, the problem admits an easy FPT algorithm via *color-coding*, parameterized by the individual parameters k and ℓ, the problem becomes Para-NP-hard due to reductions from Hamiltonian Path for k and P_3-Packing for ℓ.

Although it may appear that one has exhausted the possibilities for exploration of the problem within the parameterized framework, another set of parameters, known as *structural parameters*, emerges, allowing for further investigation. Belmonte et al. [1] initiated this line of study by considering the size of set A ($|A|$) as a parameter. They proved that the problem is W[1]-hard parameterized by $\mathsf{pw}(G) + |A|$, which translates to $\mathsf{tw}(G) + |A|$ as well, where $\mathsf{pw}(G)$ and $\mathsf{tw}(G)$ denote the pathwidth and treewidth of G respectively.

The intractability result for the parameter $\mathsf{tw}(G)+|A|$ refutes the possibility of getting FPT algorithms for many well-known structural parameters. Nonetheless, one of the objectives of structural parameterization is to delimit the border of the tractability of the problem, i.e., determining the smallest parameter for which the problem becomes FPT or the largest parameters that make the problem W-hard. Therefore one natural direction is to study ALPP with respect to parameters that are either larger than $\mathsf{tw}(G) + |A|$ or incomparable with $\mathsf{tw}(G) + |A|$.

Our Contribution. As our first result, we improve upon the hardness result of Belmonte et al. [1] by showing hardness for a much larger parameter, $\mathsf{dtp}(G) + |A|$. Here $\mathsf{dtp}(G)$ denotes the distance from G to a path graph (formal definitions of all the parameters can be found in Section 2). We present a randomized reduction from a known W[1]-hard problem, which establishes the hardness of (A, ℓ)-Path Packing under the assumption of randomized Exponential Time Hypothesis (rETH). The randomized reduction technique employed in our proof is highly adaptable and can be utilized to demonstrate the hardness of various analogous problems. We use the following lemma to prove our hardness result, which can be of independent interest.

▶ **Lemma 1** (Separation Lemma). *Let $(X, \mathcal{F})$ be a set system where $\mathcal{F}$ is a family of subsets of X. For an arbitrary assignment of weights $w : X \mapsto [M]$, let $w(S) = \sum_{x \in S} w(x)$ denote the weight of the subset $S \subseteq X$. For any random assignment of weights to the elements of X independently and uniformly from $[M]$, with probability at least $1 - \frac{\binom{|\mathcal{F}|}{2}}{M}$, each set $S \in \mathcal{F}$ has a unique weight.*

In addition to refining the boundaries of hardness, we have also considered the problem with respect to the parameter of $\mathsf{cvd}(G)$ denoting the cluster vertex deletion size of G in combination with the natural parameters A and ℓ. The incomparability of $\mathsf{cvd}(G)$ with tw and pw makes it an intriguing parameter to explore. We have proved that ALPP is fixed-parameter tractable with respect to the parameters $\mathsf{cvd}(G) + |A|$ as well as $\mathsf{cvd}(G) + \ell$ where $\mathsf{cvd}(G)$ denotes the cluster vertex deletion size of G (see Section 2). Below we provide brief overviews of these two algorithms.

$\mathbf{cvd}(\mathbf{G}) + |\mathbf{A}|$. We start by combining $|A|$ and the $\mathsf{cvd}(G)$ to form a "modulator" M. Now, each A-path comprises subpaths between certain modulator vertices. Once we guess the interaction of M with these paths, the role of the cliques in $G - M$ reduces to providing vertices for these subpaths. Next, we first employ a color-coding scheme and color the cliques. This coloring determines the role of a clique to provide required subpaths of a certain kind. And a clique within a color class is deemed *feasible* if it can provide the necessary subpaths barring the length requirements. The feasibility of a clique is identified through its color and *modulator neighborhood*. Following this, we make an important observation that the largest *feasible* cliques are the optimal choices for providing these length-constrained paths. Subsequently, we design an Integer Linear Program that checks their ability to provide subpaths with necessary length requirements.

$\mathbf{cvd}(\mathbf{G}) + \ell$. In the context of a modulator M of size m, each clique in $G - M$ is termed "local". A path is labeled "local" if it contains no vertices from M and lies entirely within a local component (clique). It is worth noting that there are at most m non-local paths in any optimal solution. From each clique, we designate a few vertices with a *marking scheme* that are utilized in providing these non-local paths. The remaining vertices from cliques are utilized in providing local paths and possess very specific characteristics. Exploiting this property, we can extract such local paths from unmarked vertices of cliques, effectively bounding the size of each clique. Subsequently, we identify *equivalent*/indistinguishable cliques based on their *modulator neighborhood* and only need to retain a few of the equivalent cliques (as almost all of them are used in providing local paths, and we can extract these local paths). This bounds the size of the instance as the clique size, the number of equivalent classes, and the number of equivalent cliques inside a class are bounded.

We design a cubic kernel for a larger parameter (than $\mathsf{tw}(G)$) vertex cover $\mathsf{vc}(G)$ in a manner very similar to the algorithm designed for $\mathsf{cvd}(G) + \ell$.

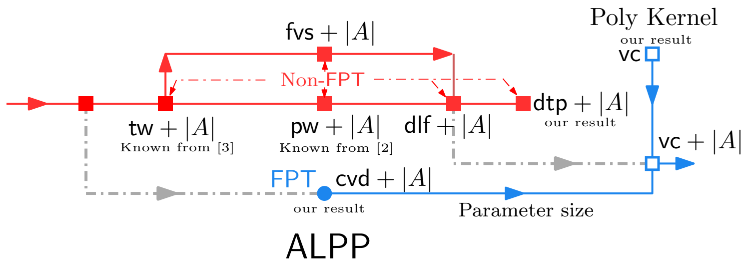

Figure 1 Structural Parameterizations of ALPP. The arrow represents the hierarchy of different structural parameters, while the dashed line represents the parameters that have yet to be explored in the context of our problems.

2 Preliminaries

Sets, Numbers and Graph Theory. We use $\mathbb{N}$ to denote the set of all natural numbers and $[r]$ to denote the set $\{1, \ldots, r\}$ for every $r \in \mathbb{N}$. Given a finite set S and $r \in \mathbb{N}$, we use $\binom{S}{r}$ and $\binom{S}{\leq r}$ to denote the collection of subsets of S with exactly r elements and at most r elements respectively. We use standard graph theoretic notations from the book by Diestel [9]. For any two vertices x, y and a path P, we denote $V[x, y]$ as the number of vertices in the subpath between x and y in P. And for a path P we denote the set of vertices in P by $V(P)$. Further, for a collection $\mathcal{P}$ of paths, $V(\mathcal{P}) = \{\bigcup_{P_i \in \mathcal{P}} V(P_i)\}$. In a graph G, let $P_i(v_1, v_2, \cdots, v_j)$ be a path and $X \subseteq V$ be a set. An *ordered intersection* of P_i with X, denoted as $X_i = (v_a, v_b, \cdots, v_p)$, is defined as $V(P_i) \cap X = X_i$, where the ordering of the vertices in X_i is the same as that in P_i. Additionally, we define $X_1, X_2, \cdots, X_x$ as an *ordered partition* of X if each X_i is an ordered set and $\bigcup_{i=1}^{x} X_i = X$. Given a path $P_i(v_1, v_2, \cdots, v_j)$, we denote $P_i(v_1, v_2, \cdots, v_{j-1})$ as $P_i \setminus (v_{j-1,v_j})$.

Structural Parameters. Given a graph class $\mathcal{H}$ and a graph G, we define the *distance* of G to $\mathcal{H}$ as the minimum number of vertices that need to be deleted to obtain a graph in class $\mathcal{H}$, denoted by $d^{\mathcal{H}}(G)$. For instance, the vertex cover size $\mathsf{vc}(G)$ represents the distance to the class of edgeless graphs or independent sets, the feedback vertex set size $\mathsf{fvs}(G)$ is the distance to the class of forests and cluster vertex deletion set size $\mathsf{cvd}(G)$ denotes the distance to the class of cluster graphs (collection of disjoint cliques). Furthermore, a graph is called a *path graph* if it has only one connected component and that connected component is an induced path. The class of all path graphs and all linear forests are denoted by Γ and $\mathcal{F}$, respectively. We denote $d^{\Gamma}(G)$ and $d^{\mathcal{F}}(G)$ by $\mathsf{dtp}(G)$ and $\mathsf{dlf}(G)$, respectively. Formal definitions of all these parameters can be found in [7].

3 (A, ℓ)-Path Packing Parameterized by $\mathsf{dtp}(G) + |A|$

This section establishes that ALPP is unlikely to be fixed-parameter tractable with respect to the combined parameter $(\mathsf{dtp}(G) + |A|)$ under standard complexity theoretic assumptions. We begin by presenting some key ideas that will be instrumental in our subsequent hardness reduction.

ALPP **Parameter:** $a + m$
Input: A graph G, two subsets $A, M \subseteq V$ of cardinality a and m respectively, and integers k and ℓ such that $G - M \in \Gamma$, where Γ denotes the family of paths.
Question: Are there k vertex-disjoint A-paths each of length exactly ℓ in G?

3.1 Essential Results

We show the hardness for ALPP under the assumption that the randomized Exponential Time Hypothesis (rETH) holds. The concept of rETH was introduced by Dell et al. [8], which states the following. There exists a constant $c > 0$, such that no randomized algorithm can solve 3-SAT in time $\mathcal{O}^*(2^{cn})$ with a (two-sided) error probability of at most $\frac{1}{3}$, where n represents the number of variables in the 3-SAT instance. The $\mathcal{O}^*$ notation hides polynomial factors in the input size.

In the realm of parameterized complexity, rETH has been widely employed to prove hardness for many well-known parameterized problems. The following theorem can be derived from Theorem 12 in [4] when $\epsilon = 1/m$ in Conjecture 5.

▶ **Theorem 2.** *Unless rETH fails, there is no randomized algorithm that decides k-Clique in time $f(k) \cdot n^{o(k)}$ correctly with probability at least $2/3$.*

We establish the intractability result for ALPP parameterized by $(|A| + \mathsf{dtp}(G))$ through a "parameter preserving reduction" from the k-Independent Set problem on a 2-interval graph, which in turn was shown to be W[1]-complete following a reduction from the k-Clique problem by Fellows et al. [10]. A 2-*interval* $\mathcal{I}_i$ is a disjoint pair of intervals $\{I_i^a, I_i^b\}$ on a real line. We say that a pair of 2-intervals, $\mathcal{I}_i$ and $\mathcal{I}_j$ *intersect* if they have at least one point in common, that is $\{I_i^a \cup I_i^b\} \cap \{I_j^a \cup I_j^b\} \neq \emptyset$. Conversely, if two 2-intervals do not intersect, they are called *disjoint*.

A 2-*interval representation* of a graph G is a set of two intervals $\mathcal{J}$ such that there is a one to one correspondence between $\mathcal{J}$ and $V(G)$ such that there exists an edge between u and v if and only if the 2-intervals corresponding to u and v intersect. A graph is a 2-*interval graph* if there is a 2-*interval representation* for G. For a graph G, a set of vertices $W \subseteq V(G)$ is said to be independent if for any pair of vertices u and v in W, $(u, v) \notin E(G)$. Given a graph G, the k-Independent Set problem asks whether there exists a k size independent set in G. Observe that given a 2-interval graph G and a 2-interval representation $\mathcal{J}$ of G a k-independent set W for G corresponds to a set of pairwise disjoint 2-intervals in $\mathcal{J}$.

Fellows et al. [10] presented a parameterized reduction from an arbitrary instance (G, k) of the k-Clique problem to an instance $(\mathcal{J}, k')$ of the k'-Independent Set problem on a 2-interval graph such that there exists a k size clique in G if and only if there exists a $k' = k + 3\binom{k}{2}$ sized independent set in $\mathcal{J}$. Thus, we have the following theorem.

▶ **Theorem 3.** *Unless rETH fails, there is no randomized algorithm that decides k-Independent Set on a 2-interval graph in $f(k) \cdot n^{o(k)}$-time correctly with probability at least $2/3$.*

Next, we present a lemma, which we will use later on. We believe that this can be of independent interest and applicable to various other problem domains. This lemma is similar to the well-known isolation lemma [13]. Recall that $[r]$ denotes the set $\{1, \dots, r\}$ where $r \in \mathbb{N}^+$.

▶ **Lemma 1** (Separation Lemma). *Let $(X, \mathcal{F})$ be a set system where $\mathcal{F}$ is a family of subsets of X. For an arbitrary assignment of weights $w : X \mapsto [M]$, let $w(S) = \sum_{x \in S} w(x)$ denote the weight of the subset $S \subseteq X$. For any random assignment of weights to the elements of X independently and uniformly from $[M]$, with probability at least $1 - \frac{\binom{|\mathcal{F}|}{2}}{M}$, each set $S \in \mathcal{F}$ has a unique weight.*

Proof. Let w be a random assignment of weights to the elements of X independently and uniformly from $[M]$ and S_1 and S_2 be two arbitrary sets in $\mathcal{F}$. Our objective is to find the probability of the event "$w(S_1) = w(S_2)$". Observe that if $S_1 \setminus S_2 = \emptyset$ or $S_2 \setminus S_1 = \emptyset$, then $\mathbb{P}\big(w(S_1) = w(S_2)\big) = 0$. Let $S_1 \setminus S_2 = \{x_1, x_2, \cdots, x_a\}$ and $S_2 \setminus S_1 = \{y_1, y_2, \cdots y_b\}$. We define a random variable W_{12} as follows.

$$W_{12} = \{w(x_1) + \cdots + w(x_a)\} - \{w(y_2) + \cdots + w(y_b)\} = w(S_1) - w(S_2) + w(y_1)$$

From the law of total probability, we have the following.

$$\begin{aligned}
&\mathbb{P}\big(w(S_1) = w(S_2)\big) = \sum \mathbb{P}\big(w(S_1) = w(S_2) | W_{12} = z\big) \cdot \mathbb{P}(W_{12} = z) \\
&= \sum \mathbb{P}\big(w(y_1) = z\big) \cdot \mathbb{P}(W_{12} = z) \\
&= \sum_{z \in [M]} \mathbb{P}\big(w(y_1) = z\big) \cdot \mathbb{P}(W_{12} = z) \qquad \text{if } z \notin [M] \text{ then } \mathbb{P}\big(w(y_1) = z\big) = 0 \\
&= \sum_{z \in [M]} \frac{1}{M} \mathbb{P}(W_{12} = z) = \frac{1}{M} \sum_{z \in [M]} \mathbb{P}(W_{12} = z) \leq \frac{1}{M}
\end{aligned}$$

Using Boole's inequality, we can prove that none of the two sets in $\mathcal{F}$ are of equal weight with probability at least $1 - \frac{\binom{|\mathcal{F}|}{2}}{M}$. Thus, the claim holds. ◀

3.2 Hardness Proof

We are ready to present a randomized reduction from the k-Independent Set problem in 2-interval graphs to the ALPP problem. Throughout this section, we denote the family of path graphs by Γ. Let $(G_{\mathcal{J}}, k)$ be an instance of k-Independent Set problem in 2-interval graph where the set of 2-intervals representing $G_{\mathcal{J}}$ be $\mathcal{J}$.

Now, we present a randomized construction of an ALPP problem instance (G, A, M, ℓ, k) from $(G_{\mathcal{J}}, k)$ where $H = G \setminus M$ is in Γ and $|M| = 4k$. We assume that we are given $\mathcal{J}$. The construction of G is done in two phases. In the first phase, we generate a set of points P on the real line $\mathbb{R}$. In the second phase, we construct the graph G. Observe that the points in P naturally induce a path graph H which is defined as follows. Corresponding to each point in P, we define a vertex in $V(H)$, and there is an edge between two vertices if the points corresponding to them are adjacent in $\mathbb{R}$. We additionally add $4k$ vertices. Before detailing our construction, let us establish a few notations and assumptions that can be accommodated without changing the combinatorial structure of the problem. Let $\mathcal{J} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_n\}$ where each 2-interval $\mathcal{I}_j$ is a collection of two intervals I_j^a and I_j^b. We presume that all the intervals in $\mathcal{J}$ are inside the interval $[0, 1]$ in the real line. Furthermore, we assume that all the endpoints of the intervals are distinct, and the distance between any two consecutive endpoints is at least 2ϵ, where ϵ is an arbitrarily small constant. We use $L(I)$ and $R(I)$ to denote the left and right endpoints of an interval I, respectively.

Construction Phase 1. We place a set of points P on $\mathbb{R}$ as follows. For each interval I_j^c, where $c \in \{a, b\}$ and $j \in [n]$, we generate two random numbers $n_r(L(I_j^c))$ and $n_r(R(I_j^c))$ between 1 and N. We will decide on the value of N at a later stage.

Let $P(L(I_j^c))$ be the set of $n_r(L(I_j^c))$ equally spaced points in the interval $[L(I_j^c), L(I_j^c)+\epsilon]$. The first point of $P(L(I_j^c))$ coincides with $L(I_j^c)$, and the last point coincides with $L(I_j^c) + \epsilon$. Let $P_L = \bigcup_{c\in\{a,b\},j\in[n]} P(L(I_j^c))$. Similarly, let $P(R(I_j^c))$ be the set of $n_r(R(I_j^c))$ equally spaced points in the interval $[R(I_j^c) - \epsilon, R(I_j^c)]$. The first point of $P(R(I_j^c))$ coincides with $R(I_j^c) - \epsilon$ and the last point coincides with $R(I_j^c)$. Let $P_R = \bigcup_{c\in\{a,b\},j\in[n]} P(R(I_j^c))$. An illustration of this process can be seen in Figure 2. Observe that the cardinality of $P_L \cup P_R$ is at most $4nN$.

Consider $\mathcal{I}_j = (I_j^a, I_j^b)$ be a 2-interval, and n_j be the total number of points from $P_L \cup P_R$ that are inside $\mathcal{I}_j$. Let $\overline{n_j} = 8nN - n_j$ for all $j \in [n]$. Define C_j as the collection of $\overline{n_j}$ points evenly distributed within the interval $[2j, 2j + 1]$ and let $P_C = \cup_{j\in[n]} C_j$. To ease the notations, we denote $L_j = 2j$, $R_j = 2j + 1$ and I_j as the interval (L_j, R_j). Observe that the total number of points inside I_j^a, I_j^b and I_j is $8nN$.

We add a large number of points (exactly $8nN + 4$ many) between R_j and L_{j+1} for $j \in [n]$ and denote all these points by P_X. Let $P = P_L \cup P_R \cup P_C \cup P_X$.

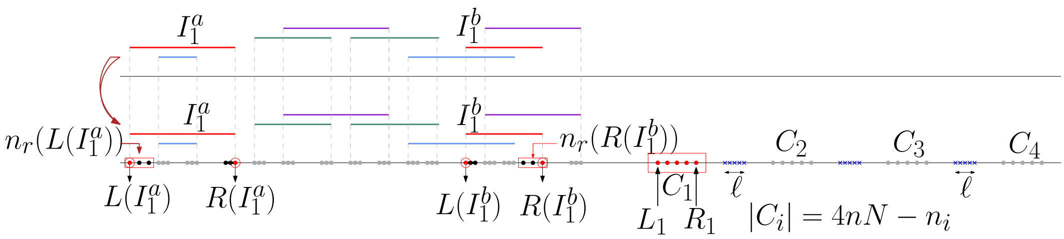

Figure 2 Illustration of Construction Phase 1. Note that $|C_i| = \overline{n_i} = 8nN - n_i$.

Construction Phase 2. Consider the set of points P in $\mathbb{R}$. Observe that there is a natural ordering among the points in P. We define adjacency based on this ordering. Consider the path graph G_P induced by P. Specifically, we introduce a vertex for each point in P and add an edge between two vertices if and only if the points corresponding to them are adjacent. With slight abuse of notation, we denote the vertex corresponding to a point p by p. For any interval I, let $V(I)$ denote the number of points/vertices within the interval I. For two points a and b in P, $\lambda[a, b]$ denotes the path from a to b in G_P.

We construct G, by setting $V(G) = V(G_P) \cup V_M$ where $V_M = \{a_i, b_i, c_i, d_i | \ i \in [k]\}$. And, $E(G) = E(G_P) \cup E_a \cup E_b \cup E_c \cup E_d$, where $E_a = \{(a_i, L(I_j^a)) \mid j \in [n] \text{ and } i \in [k]\}$, $E_b = \{(b_i, R(I_j^a)), (b_i, L_j) \mid j \in [n] \text{ and } i \in [k]\}$, $E_c = \{(c_i, L(I_j^b)), (c_i, R_j) \mid j \in [n] \text{ and } i \in [k]\}$, $E_d = \{(d_i, R(I_j^b)) \mid j \in [n] \text{ and } i \in [k]\}$.
Informally, the edges are defined as follows. Each $a_i \in V_M$ is adjacent to $L(I_j^a)$ for all $j \in [n]$ (see Figure 3). Similarly each $b_i \in V_M$ is adjacent to $R(I_j^a)$ for all $j \in [n]$. Additionally, each b_i is adjacent to every other L_j for all $j \in [n]$. Each c_i is adjacent to $L(I_j^b)$ and R_j, and each d_i is adjacent to $R(I_j^b)$.
We denote $A = \{a_i, d_i | \ i \in [k]\}$ and set $\ell = 8nN + 4$.

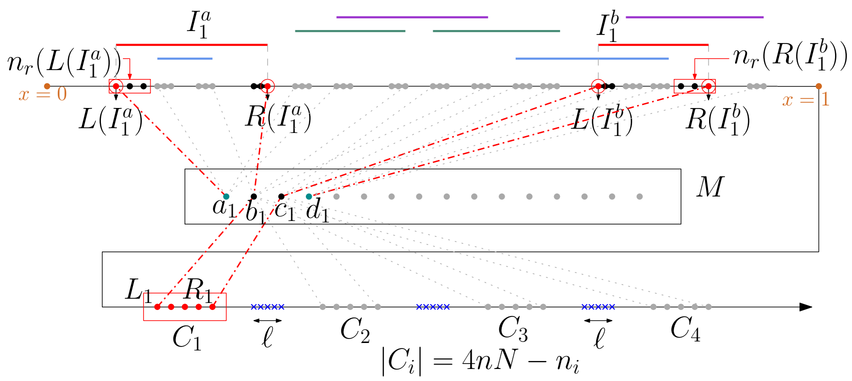

Figure 3 Illustration of construction of (A, ℓ)-PATH PACKING. Note that $|C_i| = \overline{n_i} = 8nN - n_i$.

Next, we demonstrate that if there exists a k-size independent set in $G_{\mathcal{J}}$, then with probability 1 there exist k many vertex-disjoint (A, ℓ)-paths in G where $\ell = 8nN + 4$ (Lemma 4). Following this, we show that if there exist k many vertex-disjoint (A, ℓ)-paths in G where $\ell = 8nN + 4$, then with *high* probability there is a k-size independent set in $G_{\mathcal{J}}$.

▶ **Lemma 4.** *If there are k disjoint 2-intervals in $\mathcal{J}$, then there are k vertex-disjoint (A, ℓ)-paths in G.*

Proof. Without loss of generality, let us assume that the k disjoint 2-intervals are $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$. Recall that for two vertices a and b in the path graph G_P, $\lambda[a, b]$ denotes the path from a to b in G_P. Consider the following set of paths, defined for $1 \le i \le k$,

$$\lambda_i = a_i \cdot \lambda[L(I_i^a), R(I_i^a)] \cdot b_i \cdot \lambda[L_i, R_i] \cdot c_i \cdot \lambda[L(I_i^b), R(I_i^b)] \cdot d_i$$

Observe that the paths in $\{\lambda_i | 1 \le j \le k\}$ are pairwise vertex-disjoint, each having ℓ vertices, with two endpoints at two different vertices in A (See Figure 4 for an illustration). By construction, they contain $\ell = 8nN + 4$ many vertices. Thus, the claim holds. ◀

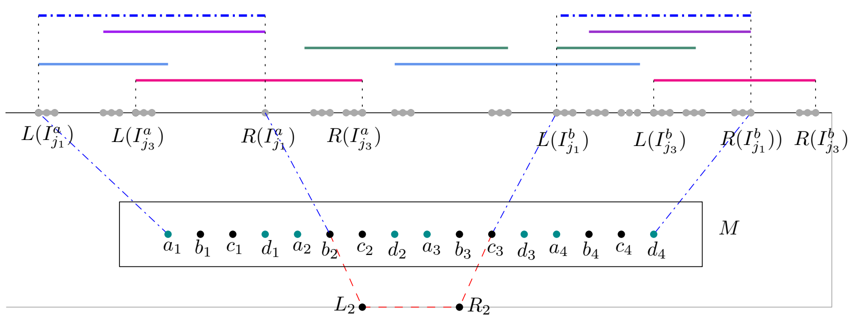

Figure 4 Construction of the paths λ_i in the proof of Lemma 4.

Next, we show that if there exist k many vertex-disjoint (A, ℓ)-paths in G where $\ell = 8nN + 4$, then with *high* probability there is a k size independent set in $G_{\mathcal{J}}$.

Recall that the set of intervals $\{I_j | j \in [n]\}$ where $I_j = [L_j, R_j]$. Let $V_j = V(I_j)$. And, the path corresponding to I_j is $\lambda[L_j, R_j]$. Let us define $\mathcal{C} = \{C_i | 1 \le i \le n\}$. Next, we have the following observation.

▶ **Observation 5.** *For any A-path L of length exactly* $\ell = 8nN + 4$ *in G, there exists an integer* $y \in [n]$ *such that* $V_y \subset V(L)$ *and for any* $i \neq y$, $V_i \cap V(L) = \emptyset$. *Further L contains the subpath* $b_x\cdot \lambda[L_y, R_y]\cdot c_z$ *for some* $x, z \in [k]$.

Proof. As the total number of points/vertices in $[0, 1]$ is at most $4nN < \ell$, note that every (A, ℓ)-path should contain all points/vertices from at least one set of points from $\mathcal{C}$. Observe that any path with a length of exactly ℓ contains exactly one set of points $C_i \in \mathcal{C}$ as between two consecutive set of points in $\mathcal{C}$ there are more than ℓ points. As only neighbors of C_i which is not in H is b_x and c_z, where $x, z \in [k]$, therefore, the path must include a subpath of the form $b_x \cdot \lambda[L_y, R_y] \cdot c_z$ for some $y \in [n]$ and $x, z \in [k]$. ◀

Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ be a set of k vertex-disjoint A-paths of length exactly ℓ in G. We show that with *high* probability, there is a k size independent set in $G_{\mathcal{J}}$.

▶ **Observation 6.** *Each path* $P_i \in \mathcal{P}$ *is of the form* $p\cdot \lambda[r, s]\cdot b_x\cdot \lambda[L_y, R_y]\cdot c_z\cdot \lambda[t, u] \cdot q$ *where* $p, q \in A$, $r, s, t, u \in M$, $y \in [n]$ *and* $x, z \in [k]$.

Proof. Since the total number of points in A is $2k$, each path must contain exactly two points from A. It follows from Observation 5 that any such path contains a subpath of the form $b_x\cdot \lambda[L_y, R_y]\cdot c_z$. By construction, for any path P, that connects p with b_x, $P \setminus \{p, b_x\}$ induces a continuous set of points on the path graph with endpoints in M (see Figure 4). A similar argument can be made for c_z and q as well. Thus, the claim holds. ◀

Observation 6 indicates that from the disjoint paths in $\mathcal{P}$, it is possible to construct a disjoint set of 3-intervals $\mathcal{J} = \{((x_i, y_i), (z_i, w_i), I_j)|1 \leq i \leq k\}$ where $x_i, y_i, z_i, w_i \in M$ and $I_j = (L_j, R_j)$. Next we prove that with *high* probability the interval $(x_i, y_i) = I^a_{y_i}$ and the interval $(z_i, w_i) = I^b_{y_i}$, which will give us the desired set of k disjoint 2-intervals.

Let a, b, c, d be any four points in M. Without loss of generality, assume that $a < b < c < d$. Consider the 2-interval $((a, b), (c, d))$ defined by a, b, c, d. Let $\mathcal{J}_{\mathcal{F}}$ be the set of all such intervals, formally defined as $\mathcal{J}_{\mathcal{F}} = \{((a, b), (c, d))|a, b, c, d \in$ and $a < b < c < d\}$. Note that $|\mathcal{J}_{\mathcal{F}}| < n^4$. From Lemma 1, we know that each 2-interval in $\mathcal{J}_{\mathcal{F}}$ contains a unique number of points with probability at least $1 - \frac{\binom{n^4}{2}}{N}$. If we set $N = 3\binom{n^4}{2}$, with probability at least $\frac{2}{3}$ every 2-interval in $\mathcal{J}_{\mathcal{F}}$ contains a unique number of points. Thus with probability at least $\frac{1}{3}$, for every $1 \leq j \leq n$, no other 2-interval except (I^a_j, I^b_j) contains $\ell - V(I_j) - 4$ many points where $V(I)$ denotes the number of points in the interval I. Therefore with probability at least $\frac{2}{3}$, for every 3-interval $((x_i, y_i), (z_i, w_i), I_{y_i})$ in $\mathcal{J}$, $(x_i, y_i) = I^a_{y_i}$ and $(z_i, w_i) = I^b_{y_i}$. Hence we have the following lemma.

▶ **Lemma 7.** *If there are k vertex-disjoint* (A, ℓ)*-paths in G, then there are k disjoint 2-intervals in* $\mathcal{J}$.

The following conclusive theorem arises from the combination of Theorem 3, Lemma 4, and Lemma 7.

▶ **Theorem 8.** *Unless rETH fails, there is no randomized algorithm for* (A, ℓ)-Path Packing *which runs in* $f(\mathsf{dtp}(G) + |A|) \cdot n^{o(\mathsf{dtp}(G)+|A|)}$*-time correctly with probability at least* $2/3$.

4 (A, ℓ)-Path Packing Parameterized by $\mathsf{cvd}(G)+|A|$

In this section, we design an FPT algorithm for the (A, ℓ)-Path Packing problem parameterized by combining the two following parameters: the size of a cluster vertex deletion set and $|A|$. Our algorithm operates under the assumption that we are provided with a minimum size set M as input, satisfying the conditions $A \subseteq M$ and $G - M$ forms a cluster graph. This assumption is justified by the fact that one can efficiently find the smallest size cluster vertex deletion set of $G - A$ in time $1.9102^k \cdot n^{\mathcal{O}(1)}$ [2]. We restate the problem definition and illustrate a brief sketch of our algorithm.

ALPP **Parameter:** $m = |M|$
Input: A graph G, two subsets $A, M \subseteq V$ of cardinality a and m respectively, and integers k and ℓ such that $A \subseteq M$ and $G - M$ is a cluster graph.
Question: Does there exist k vertex-disjoint paths of length exactly ℓ that have endpoints in A?

Overview of the Algorithm. Our algorithm starts by making an educated guess regarding the precisely ordered intersection of each path within an optimal solution ($\mathcal{P}$) with the modulator set M. This involves exploring a limited number of possibilities, specifically on the order of $f(|M|) \cdot n^{\mathcal{O}(1)}$ choices. Once we fix a choice, the problem reduces to finding subpaths (of any given path in $\mathcal{P}$) between modulator vertices satisfying certain length constraints. To provide a formal description, let $P \in \mathcal{P}$ be a path of the form $m_1 P_{1,2} m_2 m_3 m_4 P_{4,5} m_5$ (our guess for $P \in \mathcal{P}$) where each $m_i \in M$ and each $P_{i,i+1}$ is contained in $G - M$. Subsequently, our algorithm proceeds to search for the subpaths $P_{1,2}$ and $P_{4,5}$ each contained in cliques of $G - M$ with endpoints adjacent to vertices m_1, m_2 and m_4, m_5 respectively. A collection of constraints within our final *integer linear program (ILP)* guarantees that the combined length of the paths $P_{1,2}$ and $P_{4,5}$ precisely matches $\ell - 5$, satisfying the prescribed length requirement. Before presenting the ILP formulation, we make informed decisions about the cliques that are well-suited and most appropriate for providing these subpaths. Towards that, we partition the at most $|M|$ many subpaths (originating from all the paths in $\mathcal{P}$), with each partition containing subpaths exclusively from a single clique of $G - M$. Moreover, no two subpaths in separate partitions come from the same clique. Once such a choice is fixed, we apply a color coding scheme on the cliques in $G - M$ where we color the cliques with the same number of colors as the number of sets in the mentioned partition of subpaths into sets. With a *high* probability, each clique involved in the formation of $\mathcal{P}$ is assigned a distinct color. These assigned colors play a crucial role in determining the roles of the cliques in providing subpaths and, consequently, in constructing the final solution $\mathcal{P}$. We show that among all the cliques colored with a single color, a largest size *feasible* clique is an optimal choice for providing the necessary subpaths of $\mathcal{P}$. A *feasible* clique is a clique that is able to provide the necessary subpaths determined by its assigned color, barring the length requirements, and, is identified by its adjacency relation with M. Therefore, we keep precisely one feasible clique of the maximum size for each color and eliminate the others. Thus the number of cliques in the reduced instance is bounded by a function $f(m)$. Following these steps, our problem reduces to finding required subpaths (with length constraints) for which we design a set of ILP equations where the number of variables is a function of m. Below, we give a detailed description of our algorithm.

Algorithm.

Phase 1: The Guessing Phase

1. Find a cluster vertex deletion set S of the minimum size in $G - A$. Then, set $M = A \cup S$.
2. Generate all $M' \subseteq M$. For a fixed M', generate all its *ordered partitions* such that only the first and last vertices of every set of the partitions are from A. Let $\mathcal{M} = \{M'_1, \ldots, M'_{|\mathcal{M}|}\}$ be a fixed such partition of M'.

3. Without loss of generality, let $M'_i = (m_{g(i)}, m_{g(i)+1}, \ldots, m_{l(i)})$. For any two consecutive vertices m_j and m_{j+1} where $j \in [g(i), l(i)-1]$, we introduce a variable $P_{j,j+1}$. These variables serve as placeholders representing subpaths within the optimal solution that we are aiming to find. We use $\mathbb{P}$ to denote the collection of $P_{j,j+1}$.
4. We enumerate all partitions of $\mathbb{P}$. Let $\mathcal{X}_\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_x\}$ be a partition of $\mathbb{P}$ with $x = |\mathcal{X}_\mathbb{P}|$.
5. Additionally, we generate all *valid* functions $h : \mathbb{P} \to \{0, 1, > 1\}$, i.e., we guess whether the length of each subpath is exactly 0, 1, or more than 1. The function h is *valid* if $h(P_{j,j+1}) = 0$, implies m_j and m_{j+1} are adjacent. The validity of h concerning the other two values (1 and > 1) is inherently assured by the presence of a feasible clique.
6. We create a function $T : \mathcal{X}_\mathbb{P} \to 2^{2^M}$ as follows. For each part $\mathcal{P}_i \in \mathcal{X}_\mathbb{P}$, create the set $T(\mathcal{P}_i) = \{\{m_j, m_{j+1}\} : P_{j,j+1} \in \mathcal{P}_i \text{ and } h(P_{j,j+1}) = 1\} \cup \{\{m_j\} \cup \{m_{j+1}\} : P_{j,j+1} \in \mathcal{P}_i \text{ and } h(P_{j,j+1}) > 1\}$.

At the conclusion of Step 6, we have generated all the tuples denoted as $\tau = (M', \mathcal{M}, \mathbb{P}, \mathcal{X}_\mathbb{P}, h, T)$. For each specific τ, we proceed to Phase 2 in order to bound the number of cliques and subsequently generate a set of ILPs.

Phase 2: Bounding the number of Cliques Phase

1. We color all the cliques in $G - M$ with x many colors uniformly at random. From a set of cliques colored with color i, we choose a largest *feasible* clique Q_i. A clique with color i is *feasible* if and only if it has $|T(\mathcal{P}_i)|$ distinct vertices, each being a neighbor to a different set in $T(\mathcal{P}_i)$. Also, we denote the above coloring function by $\mathcal{C}_\tau$.
2. Following the Algorithm, we construct the following set of ILPs.

$$\begin{aligned} ILP(\tau, \mathcal{C}_\tau) : \quad & \sum_{j=g(i)}^{l(i)-1} x_{j,j+1} = \ell - |M'_i|, && \forall M'_i \in \mathcal{M} \\ & \sum_{P_{j,j+1} \in \mathcal{P}_i} x_{j,j+1} \leq |Q_i|, && \forall \mathcal{P}_i \in \mathcal{X}_\mathbb{P} \\ & x_{j,j+1} = 1, && \text{iff } \; h(P_{j,j+1}) = 1 \end{aligned}$$

Correctness of The Guessing Phase (Steps 1 to 6). Let $\mathcal{P} = \{P_1, \ldots, P_p\}$ be an optimal solution of size p where any ℓ-path in $\mathcal{P}$ by definition has both its endpoints in A. In the above ILP, note that the variable $x_{j,j+1}$ represents the path $P_{j,j+1}$. The M' generated in the Step 2 is $V(\mathcal{P}) \cap M$. Each M'_i is the *ordered intersection* of a path P_i with M' , i.e., the sequence of vertices of $V(P_i) \cap M$ appearing in the path is given by M'_i (Step 2). In Step 3, we create the variables (corresponding to the subpaths of $\mathcal{P}$) for each pair of consecutive vertices from M'_i for every $M'_i \in \mathcal{M}$ (Step 3). Any such subpath with a non-zero length is contained in exactly one of the cliques in $G - M$. The subpaths of $\mathcal{P}$ that come from single cliques together are denoted by the partition $\mathcal{X}_\mathbb{P}$, i.e., the subpaths (in $\mathcal{P}$) in a part of the partition are exactly the subpaths that are contained in a single clique of $G - M$ (Step 4). We further divide these subpaths into three groups ($h^{-1}(0)$, $h^{-1}(1)$, $h^{-1}(> 1)$) based on whether their lengths are exactly 0, 1, or more than 1 (Step 5). If $h(P_{j,j+1}) = 0$, then the corresponding subpath has length *zero* implying m_j and m_{j+1} are adjacent in $\mathcal{P}$. If $h(P_{j,j+1}) = 1$, then the corresponding subpath has length exactly one and the lone vertex in the subpath is adjacent to both m_j and m_{j+1} in $\mathcal{P}$. When $h(P_{j,j+1}) > 1$, the correcponding subpath has length *more than one* and has two vertices, one is adjacent to m_j while another is adjacent to m_{j+1}. The set $T(\mathcal{P}_i)$ basically stores the adjacency relations (required) between the endpoints of non-zero length subpaths of $\mathcal{P}$ and M. The correctness of the first 6 steps follows directly because of the fact that we exhaust all possible choices at each step.

Correctness of Phase 2. We apply the *color-coding* scheme in the first step of second phase of our algorithm. Each clique $Q_i \in \{Q_1, \ldots, Q_x\}$ that contains vertices from $\mathcal{P}$ gets a different color with *high* probability. Moreover, each Q_i colored with color i exactly contains the subpaths denoted by $\mathcal{P}_i$. We compute this exact probability later in the runtime analysis of our algorithm. Notice the role of the cliques in $G - M$ is to provide subpaths of certain lengths between the vertices from M. And, a *feasible* clique of color i is able to provide all the subpaths in $\mathcal{P}_i$ between the vertices of M, barring the length requirements. Thus, given a feasible clique Q_i' of maximum size, we can reconstruct an equivalent optimal solution $\mathcal{P}'$ in which all its paths within $\mathcal{P}_i$ are entirely contained within Q_i', all the while sticking to the specified length requirements. This reconstruction can be systematically applied to guarantee the existence of an optimal solution where all its subpaths are derived from a collection of feasible cliques with the largest size available from each color class. Let τ be a correctly guessed tuple, $\mathcal{C}_\tau$ be a correct coloring scheme (coloring each of the x cliques involved in the solution distinctly), and Q_i' be a feasible clique of the largest possible size colored with color i respecting the guessed tuple for each color i. Then, there exists an optimal solution that is entirely contained in the subgraph $G[\bigcup_{i=1}^{x} Q_i' \cup M]$. To obtain the desired solution for a YES instance, we narrow our attention to the subgraph and formulate the specified ILP denoted as ILP$(\tau, \mathcal{C}_\tau)$. The primary objective of the ILP equations is to guarantee that every path we are seeking has an exact length of ℓ. The first set of constraints enforces the specified length requirements for the subpaths, ensuring that each subpath adheres to its designated length. And the second set of constraints ensures that the combined total of all vertices to be utilized from a clique (across all subpaths) in a solution to the ILP does not surpass the total number of vertices within the largest feasible clique.

Runtime Analysis. The total number of ordered partitions generated in Step 2 is $\mathcal{O}(2^m \cdot m^m)$. In Step 3, $|M_i'|$ can be of $\mathcal{O}(m)$. Hence for a fixed $\mathcal{M}$, the number of permutations enumerated is of $\mathcal{O}(m!) \cdot m$. Notice the number of $P_{j,j+1}$ ($|\mathbb{P}|$) is bounded by m. Therefore in the next step, $\mathcal{X}_\mathbb{P}$ can be partitioned in m^m ways. In Step 5, each $P_{j,j+1} \in \mathbb{P}$ takes one of the three values. Thus there can be at most 3^m assignments for a fixed $\mathbb{P}$. Hence total number of tuples generated at the end of Step 6 is bounded by $\mathcal{O}(2^m \cdot m^m \cdot m! \cdot m \cdot m^m \cdot 3^m) \equiv 2^{\mathcal{O}(m \log m)}$. Since $x \leq m$, the probability that we get a coloring that colors all the x cliques properly and distinctly is at least $\frac{1}{m!}$. Once we have a good coloring instance, we formulate the ILP$(\tau, \mathcal{C}_\tau)$ to solve the problem. Since both $|\mathcal{M}|$ and $|\mathcal{X}_\mathbb{P}|$ are bounded by $\mathcal{O}(m)$, the ILP can be solved in time $m^{\mathcal{O}(m)}$. This immediately implies a randomized FPT algorithm running in time $2^{\mathcal{O}(m \log m)}$. Notice the randomization step (Phase 2) can be derandomized using (m, x)-universal family [7]. And we have the following theorem.

▶ **Theorem 9.** (A, ℓ)-Path Packing *is FPT parameterized by* $\mathsf{cvd}(G) + |A|$.

5 (A, ℓ)-Path Packing Parameterized by $\mathsf{cvd}(G) + \ell$

In this section, we design an FPT algorithm for the instance (G, S, A, k, ℓ) of ALPP parameterized by the combined parameter $\mathsf{cvd}(G) + \ell$.

(A, ℓ)-Path Packing Problem **Parameter:** $|M| (= m) + \ell$
Input: A graph G, two subsets $A, M \subseteq V$ of cardinality a and m respectively, and integers k and ℓ such that $G - M$ is a cluster graph.
Question: Does there exist k vertex-disjoint paths of length exactly ℓ that have endpoints in A?

We denote the set of cliques in $G - M$ by $\mathcal{Q}$ and the vertices in the cliques by $V_{\mathcal{Q}} = \cup_{Q \in \mathcal{Q}} V(Q)$. Let $\mathcal{I} = (G, M, A, k, \ell)$ be a YES instance of ALPP and let $\mathcal{P}$ be any arbitrary solution for $\mathcal{I}$. We denote the set paths in $\mathcal{P}$ that contain at least one vertex from M by $\mathcal{P}^{\mathrm{M}}$ and the set of paths in $\mathcal{P}$ that are completely inside a clique by $\mathcal{P}^{\mathrm{Q}}$. Note that $\mathcal{P}^{\mathrm{M}} \cap \mathcal{P}^{\mathrm{Q}} = \emptyset$ and $\mathcal{P} = \mathcal{P}^{\mathrm{M}} \cup \mathcal{P}^{\mathrm{Q}}$.

▶ **Observation 10.** *The total number of vertices present in the paths $\mathcal{P}^{M}$ is at most $\ell \cdot m$, i.e. $|\cup_{P \in \mathcal{P}^M} V(P)| \leq \ell \cdot m$.*

Next, we present a marking procedure followed by a few reduction rules to bound the size of each clique.

Marking Procedure.

1. For each vertex $u \in M$, mark $\ell m + 1$ many of its neighbors from both $A \cap V(Q)$ and $V(Q) \setminus A$ for each clique $Q \in \mathcal{Q}$. If any clique does not contain that many neighbors of u, we mark all the neighbors of u in that clique.
2. For each pair of vertices u, v in M, mark $\ell m + 1$ many common neighbors of u and v outside A, in every clique of $\mathcal{Q}$.
3. Additionally, mark $\ell m + 1$ many vertices from both $A \cap V(Q)$ and $V(Q) \setminus A$ for each clique $Q \in \mathcal{Q}$.

In the marking procedure the upper bound on the number of marked vertices for each clique Q from A (in $A \cap V(Q)$) is $f_1(\ell, m) = (m + 1)(\ell m + 1)$ and the number of marked vertices outside A (in $V(Q) \setminus A$) is $f_2(\ell, m) = (m^2 + m + 1)(\ell m + 1)$.

Exchange Operation. Consider any two arbitrary paths $P_1, P_2 \in \mathcal{P}$ and $< a_1, a_2, a_3 >$ and $< b_1, b_2, b_3 >$ be any two subsequences of vertices in P_1 and P_2 respectively. Let a_2 be a neighbour of b_1 and b_3 and b_2 be a neighbour of a_1 and a_3. We define the operation *exchange* with respect to P_1, P_2, a_2, and b_2 as follows. We create the path P_1' by replacing the vertex a_2 with b_2, and we create the path P_2' by replacing the vertex b_2 with a_2. Observe that $\mathcal{P} \setminus \{P_1, P_2\} \cup \{P_1', P_2'\}$ also forms a solution.

▶ **Lemma 11.** *There exists a solution $\mathcal{P}$ for (G, M, A, k, ℓ) such that all the vertices in $\mathcal{P}^M$ are either from M or are marked.*

Proof. Suppose there is a path P in $\mathcal{P}^{\mathrm{M}}$ that contains an unmarked vertex w. There are at most two neighbors of w in P. We assume here that there are exactly two neighbors of w. The case when w has only one neighbor in P can be argued similarly. Let the neighbors of w in P be w_1 and w_2. We have the following three exhaustive cases.

$w_1, w_2 \in V(Q)$: Recall that we have marked an additional $\ell m + 1$ many vertices from outside A in each clique (vertices that w may be replaced with) and from Observation 10, we know at most $m\ell$ many of them are contained in $\mathcal{P}^{\mathrm{M}}$. Thus, there is at least one marked vertex, say, w' in $V(Q)$, that is not contained in any path of $\mathcal{P}^{\mathrm{M}}$. If w' is also not contained in any path of $\mathcal{P}^{\mathrm{Q}}$, we simply replace w by w' in P. If it is in a path $P' \in \mathcal{P}^{\mathrm{Q}}$, we do an *exchange* operation with respect to P, P', w and w' and reconstruct a new solution.

$w_1 \in V(Q)$ **and** $w_2 \in M$: Recall that we have marked $\ell m + 1$ many vertices from $N(w_2) \cap V(Q) \setminus A$. From Observation 10, at most $m\ell$ many of them are contained in $\mathcal{P}^{\mathrm{M}}$. Hence, there is at least one marked vertex in $V(Q) \setminus A$ that is not contained in any path from $\mathcal{P}^{\mathrm{M}}$. Similar to the arguments outlined in the previous case, we replace the vertex w by

w' when w' is not contained in any path of $\mathcal{P}^{\text{Q}}$, or perform an *exchange* operation with respect to P, P', w and w' and reconstruct a new solution when w' is contained in some $P' \in \mathcal{P}^{\text{Q}}$.

$\boldsymbol{w_1, w_2 \in M}$**:** Using arguments similar to the previous case, we can again replace an unmarked vertex in $\mathcal{P}^{\text{M}}$ with a marked vertex and reconstruct a new solution for this case as well.

After exhaustively replacing unmarked vertices of $\mathcal{P}^{\text{M}}$ (that are not in M), we derive a solution $\mathcal{P}$ in which paths from $\mathcal{P}^{\text{M}}$ do not include unmarked vertices from the cliques. ◀

Henceforth, we seek for a solution $\mathcal{P}$ for (G, M, A, k, ℓ) such that all the vertices in any path of $\mathcal{P}^{\text{M}}$ are either from M or marked. Next, we have the following reduction rule.

▶ **Reduction Rule 1.** *If there exists a clique Q containing a pair of unmarked vertices $u, v \in A$ and a set X of $(\ell - 2)$ unmarked vertices outside A, then delete u, v along with X and return the reduced instance $(G - \{X \cup \{u, v\}\}, M, A \setminus \{u, v\}, k - 1, \ell)$.*

We prove the safeness of the reduction rule below.

▶ **Lemma 12.** *(G, M, A, k, ℓ) is a* YES *instance if and only if $(G - \{X \cup \{u, v\}\}, M, A \setminus \{u, v\}, k - 1, \ell)$ is a* YES *instance.*

Proof. If $(G - \{X \cup \{u, v\}\}, M, A \setminus \{u, v\}, k - 1, \ell)$ is a YES instance with a solution $\mathcal{P}^R$, then (G, M, A, k, ℓ) is also a YES instance as $\mathcal{P}^R$ along with the path formed by $X \cup \{u, v\}$ forms a solution to the instance.

Conversely, let (G, M, A, k, ℓ) be a YES instance with a solution $\mathcal{P}$. Now, we will obtain a solution $\mathcal{P}'$ for $G - \{X \cup \{u, v\}\}$ of size at least $k - 1$. We denote the paths in $\mathcal{P}$ that intersect with $X \cup \{u, v\}$ by $\mathcal{P}_X$ (with slight abuse of notation). If $|\mathcal{P}_X| \leq 1$ then $\mathcal{P}' = \mathcal{P} \setminus \mathcal{P}_X$ is the desired solution. From now on, we assume that $|\mathcal{P}_X| > 1$. Observe that as the vertices in $X \cup \{u, v\}$ are unmarked, $\mathcal{P}_X \cap \mathcal{P}^{\text{M}} = \emptyset$ (from Lemma 11). We can reconstruct a new solution $\mathcal{P}_1$ from $\mathcal{P}$ where exactly one path P in $\mathcal{P}_1$ intersects $X \cup \{u, v\}$, and the rest of the paths in $\mathcal{P}_1$ do not intersect with $X \cup \{u, v\}$, by repeatedly utilizing the *exchange* operation among the paths in $\mathcal{P}_X$. Observe that $\mathcal{P}' = \mathcal{P}_1 \setminus \{P\}$ is the desired solution. Thus, the claim holds. ◀

Note that the upperbound on the number of marked vertices from $A \cap V(Q)$ is $f_1(\ell, m) = (m+1)(\ell m+1)$ and the number of marked vertices $V(Q) \setminus A$ is $f_2(\ell, m) = (m^2+m+1)(\ell m+1)$. And, after exhaustive application of Reduction Rule 1, in any clique Q, either there are at most $\ell - 3$ unmarked vertices in $V(Q) \setminus A$ or at most one unmarked vertex in $A \cap V(Q)$.

Case (i): There is at most one unmarked vertex in $A \cap V(Q)$.
Case (ii): There are at most $\ell - 3$ unmarked vertices in $V(Q) \setminus A$.

Based on the aforementioned cases, we introduce two reduction rules – one for each case – that help us limit the overall number of unmarked vertices in Q, thereby bounding the size of each clique in $G - M$. First we consider the Case (i) when the number of unmarked vertices form $A \cap V(Q)$ is bounded by one and bound the number of the unmarked vertices in $V(Q) \setminus A$ with the following reduction rule.

▶ **Reduction Rule 2.** *If there exists a clique Q containing at most one unmarked vertex from A and at least $(f_1(\ell, m) + 1) \cdot \frac{\ell}{2} + 1$ unmarked vertices outside A, then delete one unmarked vertex $u \in V(Q) \setminus A$ and return the reduced instance $(G - \{u\}, M, A, k, \ell)$.*

Let $G' = G - \{u\}$ be the new graph following an application of Reduction Rule 2. We prove the safeness of the reduction rule in the following lemma.

▶ **Lemma 13.** *(G, M, A, k, ℓ) is a* YES *instance if and only if (G', M, A, k, ℓ) is a* YES *instance.*

Proof. If (G', M, A, k, ℓ) is a YES instance, then (G, M, A, k, ℓ) is a YES instance since G' is a subgraph of G. Conversely, suppose (G, M, A, k, ℓ) is a YES instance, and $\mathcal{P}$ is a solution. If u does not belong to any path in $\mathcal{P}$, then $\mathcal{P}$ is a solution to (G', M, A, k, ℓ) as well. Otherwise, let $P \in \mathcal{P}^{\mathrm{Q}}$ contain u. This is true since any unmarked vertex can only be used in a path in $\mathcal{P}^{\mathrm{Q}}$. But any such path uses exactly 2 vertices from $V(Q) \cap A$. Hence we can upper bound the number of unmarked vertices outside A that are contained in $\mathcal{P}^{\mathrm{Q}}$ and hence $\mathcal{P}$ by $(f_1(\ell, m) + 1) \cdot \frac{\ell}{2}$. Hence, there is at least one unmarked vertex $u' \neq u$ in $V(Q) \setminus A$ which is not used by any path in $\mathcal{P}$. We replace u with u' in P to get a desired solution to (G', M, A, k, ℓ). ◀

For the Case (ii) when the number of unmarked vertices form $V(Q) \setminus A$ is bounded by $l - 3$ and we bound the number of the unmarked vertices in $V(Q) \cap A$ with the following reduction rule.

▶ **Reduction Rule 3.** *If there exists a clique Q containing at most $\ell - 3$ unmarked vertices from $V(Q) \setminus A$ and at least $(f_2(\ell, m) + (\ell - 3)) \cdot \frac{1}{\ell-2} + 1$ many unmarked vertices in A, then delete an unmarked vertex $u \in A \cap Q$ and return the reduced instance $(G - \{u\}, M, A \setminus \{u\}, k, \ell)$.*

Proof. If $(G - \{u\}, M, A \setminus \{u\}, k, \ell)$ is a YES instance, then (G, M, A, k, ℓ) is trivially a YES instance since G' is a subgraph of G. Conversely, suppose (G, M, A, k, ℓ) is a YES instance, and $\mathcal{P}$ is a solution. If u does not belong to any path in $\mathcal{P}$, then $\mathcal{P}$ is a solution to (G', M, A, k, ℓ) as well. Otherwise, let $P \in \mathcal{P}^{\mathrm{Q}}$ contain u. This is true since any unmarked vertex can only be used in a path in $\mathcal{P}^{\mathrm{Q}}$. But any such path uses exactly $\ell - 2$ vertices from $V(Q) \cap A$. Hence we can upper bound the number of unmarked vertices from A that are contained in $\mathcal{P}^{\mathrm{Q}}$ and hence $\mathcal{P}$ by $(f_2(\ell, m) + (\ell - 3)) \cdot \frac{1}{\ell-2}$. Hence, there is at least one unmarked vertex $u' \neq u$ in $V(Q) \cap A$ which is not used by any path in $\mathcal{P}$. We replace u with u' in P to get a desired solution to (G', M, A, k, ℓ). ◀

After exhaustive application of Reduction Rules 2 and 3, the upper bound on the number of vertices of different types in each clique is as follows:

- Marked vertices in A: $f_1(\ell, m) = (m + 1)(\ell m + 1)$
- Marked vertices in $V(Q) \setminus A$: $f_2(\ell, m) = (m^2 + m + 1)(\ell m + 1)$
- Unmarked vertices in A: $(f_2(\ell, m) + (\ell - 3)) \cdot \frac{1}{\ell-2} + 2$
- Unmarked vertices in $V(Q) \setminus A$: $(f_1(\ell, m) \cdot \ell + 1$

Hence the total number of vertices in each clique is bounded by $\mathcal{O}(\ell^2 m^2 + \ell m^3)$.

Equivalent cliques. Now we aim to bound the number of cliques by introducing the concept of *equivalent* cliques. Two cliques Q_i and Q_j, are equivalent (belong to the same *equivalent class*) if and only if the number of vertices from the cliques that are in A, and that are outside A with an exact neighborhood of $M' \subseteq M$ is same for each $M' \in 2^M$. Two cliques Q_i, Q_j in an equivalence class are essentially *indistinguishable* from each other, i.e., there is a bijective mapping $g_{ij} : V(Q_i) \mapsto V(Q_j)$, so that $N(u) \cap M = N(g(u)) \cap M$, for all $u \in V(Q_i)$. This fact is crucial in the construction of our next reduction rule. Observe that the number of equivalence classes is at most $\mathcal{O}(\ell^2 m^2 + \ell m^3)^{2^m} = f(\ell, m)$. The following reduction rule bounds the number of cliques in each equivalent class.

▶ **Reduction Rule 4.** *If there exists an equivalent class $\mathcal{C}$ with at least $\ell m + 1$ cliques, then delete one of the cliques $Q_i \in \mathcal{C}$ and return the reduced instance $(G - Q_i, M, A \setminus (A \cap V(Q_i)), k - x_i, \ell)$ where, $x_i = \min\left\{\frac{|A \cap V(Q_i)|}{2}, \frac{|V(Q_i) \setminus A|}{\ell - 2}\right\}$.*

▶ **Lemma 14.** *(G, M, A, k, ℓ) is a* YES *instance if and only if $(G - Q_i, M, A \setminus (A \cap V(Q_i)), k - x_i, \ell)$ is a* YES *instance.*

Proof. In the forward direction, let (G, M, A, k, ℓ) be a YES instance. Recall that the number of the vertices contained in paths of $\mathcal{P}_M$ for any optimal solution $\mathcal{P}$ is bounded by ℓs. Thus, there are at most ℓm many cliques in total and also from any equivalence class that has vertices in paths from $\mathcal{P}_M$. Let Q_j be one such clique in the equivalence class $\mathcal{C}$ that does not contain any vertex in the paths from $\mathcal{P}_M$. From the definition of an equivalence class, it is evident that the two cliques Q_i, Q_j in the equivalence class $\mathcal{C}$ are *indistinguishable* from each other, i.e., there is a bijective mapping $g_{ij} : V(Q_i) \mapsto V(Q_j)$, so that $N(u) \cap M = N(g(u)) \cap M$, for all $u \in V(Q_i)$. Let $X_i = V(\mathcal{P}) \cap V(Q_i)$ and $X_j = V(\mathcal{P}) \cap V(Q_j)$, i.e, the set of vertices from Q_i and Q_j that are used in paths from $\mathcal{P}$, respectively. We construct an alternate solution, $\mathcal{P}'$, where we replace X_i with $g_{ij}(X_i)$ and X_j with $g_{ij}^{-1}(X_j)$ in $\mathcal{P}$. Since $X_j \cap M = \emptyset$, we have $g_{ij}^{-1}(X_j) \cap M = \emptyset$. Therefore in $\mathcal{P}'$, there is no path that contain vertices from both M and $V(Q_i)$. In other words, vertices in Q_i can only be contained in paths from $\mathcal{P}' \setminus \mathcal{P}'_M$ (paths that are completely contained inside the clique). And, the number of such paths is bounded by $x_i = \min\left\{\frac{|A \cap V(Q_i)|}{2}, \frac{|V(Q_i) \setminus A|}{\ell - 2}\right\}$. Hence $(G - Q_i, M, A \setminus (A \cap V(Q_i)), k - x_i, \ell)$ is a YES instance.

In the reverse direction, let $(G - Q_i, M, A \setminus (A \cap V(Q_i)), k - x_i, \ell)$ be a YES instance with a solution $\mathcal{P}$. But there are x_i many paths (say $\mathcal{P}_i$) that are completely contained in Q_i.Hence, $\mathcal{P} \cup \mathcal{P}_i$ is a set of k vertex-disjoint (A, ℓ)-paths contained in G, making (G, M, A, k, ℓ) a YES instance. ◀

After exhaustively applying all the aforementioned reduction rules, the following bounds hold.

- The number of vertices in each clique is bounded by $\mathcal{O}(\ell^2 m^2 + \ell m^3)$.
- The number of equivalence classes is at most $\mathcal{O}(\ell^2 m^2 + \ell m^3)^{2^m}$.
- The number of cliques in each equivalence class is at most $\ell m + 1$.

Consequently, the size of the reduced instance is upper-bounded by a computable function of ℓ and m, thus directly implying the following theorem.

▶ **Theorem 15.** ALPP *parameterized by* $\mathsf{cvd}(G) + \ell$ *admits an algorithm running in FPT time.*

6 (A, ℓ)-Path Packing parameterized by $\mathsf{vc}(G)$

In this section, we design a polynomial kernel for (A, ℓ)-Path Packing parameterized by the size of a vertex cover of the graph.

> ALPP **Parameter:** $m = |M|$
> **Input:** An undirected graph $G = (V, E)$, $A, M \subseteq V(G)$ such that M is a vertex cover of G and integers k and ℓ.
> **Question:** Are there k vertex-disjoint A-paths each of length ℓ in G?

For a YES instance (G, M, A, k), a solution $\mathcal{P}$ contains at most $3m$ vertices. This limitation arises because there are no consecutive vertices from $I = V(G) - M$ in any path within $\mathcal{P}$.

▶ **Observation 16.** *Any solution* $\mathcal{P}$ *to a* YES *instance of* (G, M, A, k, ℓ)*, has at most* $3m$ *vertices.*

Our kernelization approach comprises the following marking process followed by a reduction rule that bounds the instance size by a polynomial function of m.

Marking Procedure.
1. For each vertex $u \in M$, mark $3m + 1$ many of its neighbors in $I \cap A$. If any vertex $u \in M$ has less than $3m + 1$ neighbors, we mark all of them.
2. For each pair of vertices $u, v \in M$, mark $3m + 1$ many common neighbors in $I \setminus A$ for each clique. If any pair $u, v \in M$ has less than $3m + 1$ common neighbors, we mark all of them.

Now we apply the following reduction rule to eliminate unmarked vertices in I.

▶ **Reduction Rule 5.** *We delete any unmarked vertex* $u \in I$ *from* G*, and return the reduced instance* $(G - \{u\}, M, A, k, \ell)$*.*

Let $G' = G - \{u\}$ be the new graph obtained after an application of Reduction Rule 5. The safeness of the reduction rule is not very difficult to see and will be provided in the full version.

Following the exhaustive application of the Reduction Rule 5, there are $3m + 1$ vertices marked in I for each pair of vertices as well as each individual vertex in M. Consequently, in the reduced instance, $|I|$ is bounded by $\mathcal{O}(m^3)$. As a result, we have the following theorem.

▶ **Theorem 17.** ALPP *parameterized by* $\mathsf{vc}(G)$ *admits a kernel with* $\mathcal{O}(m^3)$ *vertices.*

7 Conclusion

Our results have extended the works of Belmonte et al. [1] by addressing the parameterized complexity status of (A, ℓ)-PATH PACKING (ALPP) across numerous structural parameters. It was known from Belmonte et al. [1] that ALPP is W[1]-complete when parameterized by $\mathsf{pw} + |A|$. We prove an intractability result for a much larger parameter of $\mathsf{dtp}(G) + |A|$. Also, the parameterized complexity of ALPP when parameterized by the combined parameter of cliquewidth and ℓ was an open question [1]. While that problem still remains open, we have been successful in making slight progress by obtaining an FPT algorithm for the problem when parameterized by the combined parameter of $\mathsf{cvd}(G)$ and ℓ. Another direction to explore would be to determine the fixed-parameter tractability status of the problem when parameterized by $\mathsf{cvd}(G)$ only. It would be interesting to explore if this FPT result can be generalized to the combined parameter of cograph vertex deletion set size and ℓ since cographs are graphs of cliquewidth at most two. We believe that the positive results presented in this paper are not optimal and some of those results can be improved with more involved structural analysis. Therefore, improving the efficiency of our positive results are exciting research direction for future works.

References

1 Rémy Belmonte, Tesshu Hanaka, Masaaki Kanzaki, Masashi Kiyomi, Yasuaki Kobayashi, Yusuke Kobayashi, Michael Lampis, Hirotaka Ono, and Yota Otachi. Parameterized Complexity of (A, l)-Path Packing. *Algorithmica*, 84(4):871–895, 2022.

2 A. Boral, M. Cygan, T. Kociumaka, and M. Pilipczuk. A fast branching algorithm for cluster vertex deletion. *Theory of Computing Systems*, 58(2):357–376, 2016.

3 Henning Bruhn and Arthur Ulmer. Packing a-paths of length zero modulo four. *Eur. J. Comb.*, 99:103422, 2022.

4 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-eth to fpt-inapproximability: Clique, dominating set, and more. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 743–754, 2017.

5 M. Chudnovsky, William H. Cunningham, and James F. Geelen. An algorithm for packing non-zero a-paths in group-labelled graphs. *Combinatorica*, 28:145–161, 2008.

6 Maria Chudnovsky, Jim Geelen, Bert Gerards, Luis A. Goddyn, Michael Lohman, and Paul D. Seymour. Packing non-zero a-paths in group-labelled graphs. *Comb.*, 26(5):521–532, 2006.

7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

8 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the tutte polynomial. *ACM Trans. Algorithms*, 10(4):21:1–21:32, 2014.

9 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

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

11 Petr A. Golovach and Dimitrios M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. *Discret. Optim.*, 8:72–86, 2011.

12 Hiroshi Hirai and Gyula Pap. Tree metrics and edge-disjoint s-paths. *Math. Program.*, 147(1-2):81–123, 2014.

13 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 345–354. Association for Computing Machinery, 1987.

14 Gyula Pap. Packing non-returning a-paths algorithmically. *Discret. Math.*, 308(8):1472–1488, 2008.