On the Ordered List Subgraph Embedding Problems

Olawale Hassan[1], Iyad Kanj[1], Daniel Lokshtanov[2], and Ljubomir Perković[1]

[1] School of Computing, DePaul University, Chicago, USA.
oahassan@gmail.com, ikanj@cs.depaul.edu, lperkovic@cs.depaul.edu
[2] Department of Informatics, University of Bergen, Bergen, Norway.
daniello@ii.uib.no

Abstract. In the (parameterized) Ordered List Subgraph Embedding problem (p-OLSE) we are given two graphs G and H, each with a linear order defined on its vertices, a function L that associates with every vertex in G a list of vertices in H, and a parameter k. The question is to decide if we can embed (one-to-one) a subgraph S of G of cardinality k into H such that: (1) every vertex of S is mapped to a vertex from its associated list, (2) the linear orders inherited by S and its image under the embedding are respected, and (3) if there is an edge between two vertices in S then there is an edge between their images. If we require the subgraph S to be embedded as an induced subgraph, we obtain the Ordered List Induced Subgraph Embedding problem (p-OLISE). The p-OLSE and p-OLISE problems model various problems in Bioinformatics related to structural comparison/alignment of proteins.

We investigate the complexity of p-OLSE and p-OLISE with respect to the following structural parameters: the *width* Δ_L of the function L (size of the largest list), and the maximum degree Δ_H of H and Δ_G of G. We provide tight characterizations of the classical and parameterized complexity, and approximability of the problems with respect to the structural parameters under consideration.

1 Introduction

1.1 Problem Definition and Motivation

We consider the following problem that we refer to as the parameterized Ordered List Subgraph Embedding problem, shortly (p-OLSE):

Given: Two graphs G and H with linear orders $\prec_G$ and $\prec_H$ defined on the vertices of G and H; a function $L : V(G) \longrightarrow 2^{V(H)}$; and $k \in \mathbb{N}$
Parameter: k
Question: Is there a subgraph S of G and an injective map $\varphi : V(S) \longrightarrow V(H)$ such that: (1) $\varphi(u) \in L(u)$ for every $u \in S$; (2) for every $u, u' \in S$, if $u \prec_G u'$ then $\varphi(u) \prec_H \varphi(u')$; and (3) for every $u, u' \in S$, if $uu' \in E(G)$ then $\varphi(u)\varphi(u') \in E(H)$

The parameterized Ordered List Induced Subgraph Embedding (p-OLISE) problem, in which we require the subgraph S to be embedded as an induced subgraph, is defined the same way as p-OLSE except that condition (3) is replaced with: for every $u, u' \in S$, $uu' \in E(G)$ if and only if $\varphi(u)\varphi(u') \in E(H)$. The optimization versions of p-OLSE and p-OLISE, denoted opt-OLSE and opt-OLISE, respectively, ask for a subgraph S of G with maximum number of vertices such that there exists a valid list embedding φ that embeds S into H.

The p-OLSE and p-OLISE problems have applications in the area of Bioinformatics because they provide a graph-theoretical model for numerous protein and DNA structural comparison problems (see [3, 5, 6, 16]).

In this paper we investigate the complexity of p-OLSE and p-OLISE with respect to the following structural parameters: the *width* Δ_L of the function L (*i.e.*, the size of the largest list $|L(u)|$, for $u \in G$) and the maximum degree Δ_H of H and Δ_G of G. Restrictions on the structural parameters Δ_H, Δ_G and Δ_L are very natural in Bioinformatics. The parameters Δ_H and Δ_G model the maximum number of hydrophobic bonds that an amino acid in each protein can have; on the other hand, Δ_L is usually a parameter set by the Bioinformatics practitioners when computing the top few alignments of two proteins [16].

1.2 Previous Related Results

Goldman et al. [8] studied protein comparison problems using the notion of *contact maps*, which are undirected graphs whose vertices are linearly ordered. Goldman et al. [8] formulated the protein comparison problem as a contact map overlap problem, in which we are given two contact maps and we need to identify a subset of vertices S in the first contact map, a subset of vertices S' in the second with $|S| = |S'|$, and an order-preserving bijection $f : S \longrightarrow S'$, such that the number of edges in S that correspond to edges in S' is maximized. In [8], the authors proved that the contact map overlap problem is MAXSNP-complete, even when both contact maps have maximum degree one. The main difference between the contact map overlap problem and the opt-OLSE and opt-OLISE problems under consideration is that in opt-OLSE and opt-OLISE the function is restricted to mapping a vertex to one in its list, and the goal is to maximize the size of the subset not the number of edges that can be embedded.

The p-OLISE problem generalizes the Longest Arc-Preserving Common Subsequence (LAPCS) problem, which is a well-studied problem (see [1, 5, 6, 9, 11, 13]). In LAPCS, we are given two sequences S_1 and S_2 over a fixed alphabet, where each sequence has arcs/edges between its characters, and the problem is to compute a longest common subsequence of S_1 and S_2 that respects the arcs. The p-OLISE problem generalizes LAPCS since no restriction is placed on the size of the alphabet, and a vertex can be mapped to any vertex from its list. Consequently, the “positive” results obtained in this paper about p-OLISE and opt-OLISE apply directly to their corresponding versions of LAPCS; on the other hand, we are able to borrow the $W[1]$-hardness result from [6] to conclude the W-hardness results in Proposition 3 and Proposition 4. The LAPCS problem was introduced by [5, 6] where it was shown to be $W[1]$-complete (parameterized by the length of common subsequence sought) in the case when the arcs are *crossing*. Several works studied the complexity and approximation of LAPCS with respect to various restrictions on the types of the arcs (*e.g.*, *nested*, *crossing*, etc.) [1, 5, 6, 9, 11, 13]. The work in [1, 9] considered the problem in the case of nested arcs parameterized by the total number of characters that need to be deleted from S_1 and S_2 to obtain the arc-preserving common subsequence. They showed that the problem is FPT with respect to this parameterization, and they also showed it to be FPT when parameterized by the length of the common subsequence in the case when the alphabet consists of four characters.

A slight variation of p-OLSE was considered in [3, 16], where the linear order imposed on G and H was replaced with a partial order (directed acyclic graphs); the problem was referred to as the GRAPH EMBEDDING problem in [3] and as the GENERALIZED SUBGRAPH ISOMORPHISM problem in [16]. The aforementioned problems were mainly studied in [3, 16] assuming no bound on Δ_H and Δ_G (*i.e.*, unbounded) and, not surprisingly, only hardness results were derived. In [16], a parameterized algorithm with respect to the treewidth of G and the map width Δ_L combined was given. Most of the hardness results in [3, 16] were obtained by a direct reduction from the INDEPENDENT SET or CLIQUE problems. For example, it was shown in [3] that the problem of embedding the whole graph G into H is $\mathcal{NP}$-hard, but is in $\mathcal{P}$ if $\Delta_L = 2$. It was also shown that the problem of embedding a subgraph of G of size k into H is $W[1]$-complete even when $\Delta_L = 1$, and cannot be approximated to a ratio $n^{\frac{1}{2}-\varepsilon}$ unless $\mathcal{P} = \mathcal{NP}$; we borrow these two hardness results as they also work for p-OLSE and p-OLISE.

Finally, one can draw some similarities between p-OLISE and the celebrated SUBGRAPH ISOMORPHISM and GRAPH EMBEDDING problems. The main differences between p-OLISE and the aforementioned problems are: (1) in p-OLSE we have linear orders on G and H that need to be respected by the map sought, (2) we ask for an embedding of a subgraph of G rather than the whole graph G, and (3) each vertex must be mapped to a vertex from its list. In particular, requirement (1) above precludes the application of well-known (logic) meta-theorems (see [7]) to the restrictions of p-OLISE that are under consideration in this paper.

1.3 Our Results and Techniques

We draw a complete complexity landscape of p-OLSE and p-OLISE with respect to the computational frameworks of classical complexity, parameterized complexity, and approximation, in terms of the structural parameters Δ_H, Δ_G and Δ_L. Table 1 outlines the obtained results about p-OLSE and p-OLISE and their optimization versions. Note that even though our hardness results are for specific values of the parameters Δ_H, Δ_G, and Δ_L, these results certainly hold true for restrictions of the problems to instances in which the corresponding parameters are upper bounded by (or equal to — by adding dummy vertices) any constants larger than these specific values. Observe also that the results we obtain *completely and tightly* characterize the complexity (with respect to all frameworks under consideration) of the problems with respect to Δ_H, Δ_G and Δ_L (unbounded vs. bounded, and when applicable, for different specific values).

Section 2 presents various complexity and approximation results. The NP-hardness results are obtained by a reduction from the k-Multi-Colored Independent Set problem, and the $W[1]$-hardness results are obtained by tweaking the $W[1]$-hardness results given in the literature [3, 6], or by simple known reductions from the Independent Set problem. Section 3 presents FPT algorithms for various restrictions of p-OLSE and p-OLISE. The FPT results in Theorem 2 for p-OLSE, when $\Delta_L = O(1)$, $\Delta_G = O(1)$, and $\Delta_H = \infty$, are derived using the random separation method. This method is applied after transforming the problem — via reduction operations — to the Independent Set problem on a graph composed of (1) a permutation graph and (2) a set of additional edges between the permutation graph vertices such that the number of additional edges incident to any vertex is at most a constant; Lemma 4 then shows that the Independent Set problem on such graphs is FPT. On the other hand, the FPT results in Proposition 6, when $\Delta_H = 0$, $\Delta_G = O(1)$ (resp. $\Delta_G = 0$ and $\Delta_H = O(1)$ for p-OLISE by symmetry) and $\Delta_L = \infty$, are also derived using the random separation method, but the argument is simpler.

To cope with the W-hardness of p-OLSE in certain cases, we consider a different parameterization of the problem, namely the parameterization by the vertex cover number, and denote the associated problem by p-VC-OLSE. This parameterization is not interesting for p-OLISE since we proved that p-OLISE is NP-complete in the case when $\Delta_G = 0$, $\Delta_H = 1$ and $\Delta_L = 1$, and hence the problem is *para-NP-hard* with respect to this parameterization. Proposition 7 shows that p-VC-OLSE is $W[1]$-complete in the case when $\Delta_H = 1$, $\Delta_G = 1$ and Δ_L is unbounded (note that if either $\Delta_H = 0$ or $\Delta_G = 0$ then p-OLSE is FPT when $\Delta_L = \infty$). So we restrict our attention to the case when $\Delta_L = O(1)$, and show in this case that the problem is FPT even when both Δ_H and Δ_G are unbounded; the method relies on a bounded search tree approach, combined with the dynamic programming algorithm described in Proposition 1.

1.4 Background and Terminologies

We provide the necessary background on approximation in the appendix.

Graphs. For a graph H we denote by $V(H)$ and $E(H)$ the set of vertices and edges of H, respectively; we write $|H|$ for $|V(H)|$. For a set of vertices $S \subseteq V(H)$, we denote by $H[S]$ the subgraph of H induced by the vertices in S. For a subset of edges $E' \subseteq E(H)$, we denote by $H - E'$ the graph $(V(H), E(H) \setminus E')$. For a vertex $v \in H$, $N(v)$ denotes the set of neighbors of v in H. The *degree* of a vertex v in H, denoted $deg_H(v)$, is $|N(v)|$. A vertex v is *isolated* in H if $deg_H(v) = 0$. The *degree* of H, denoted $\Delta(H)$, is $\Delta(H) = \max\{deg_H(v) : v \in H\}$. An *Independent Set* of a graph H is a set of vertices I such that no two vertices in I are adjacent. A *vertex cover* of H is a set of vertices such that each edge in H is incident to at least one vertex in this set; we denote by $\tau(H)$ the cardinality of a minimum vertex cover of H. Let L and L' be two parallel lines in the plane. A *permutation graph* P is the intersection graph of a set of line segments D such that one endpoint of each of those segments lies on L and the other endpoint lies on L'.

p-OLSE	Δ_H	Δ_G	Δ_L	**Complexity**
Classical	∞	0	∞	$\mathcal{P}$
	0	1	1	$\mathcal{NP}$-complete
Approximation	∞	$O(1)$	∞	$\mathcal{APX}$-complete
	0	∞	1	not approximable to $n^{\frac{1}{2}-\varepsilon}$
Parameterized	∞	$O(1)$	$O(1)$	FPT
	1	1	∞	$W[1]$-complete
	0	∞	1	$W[1]$-complete
	0	$O(1)$	∞	FPT
	∞	0	∞	FPT (even in $\mathcal{P}$)

p-OLISE	Δ_H	Δ_G	Δ_L	**Complexity**
Classical	0	0	∞	$\mathcal{P}$
	0	1	1	$\mathcal{NP}$-complete
	1	0	1	$\mathcal{NP}$-complete
Approximation	$O(1)$	$O(1)$	∞	$\mathcal{APX}$-complete
	0	∞	1	not approximable to $n^{\frac{1}{2}-\varepsilon}$
	∞	0	1	not approximable to $n^{\frac{1}{2}-\varepsilon}$
Parameterized	∞	0	1	$W[1]$-complete
	0	∞	1	$W[1]$-complete
	0	$O(1)$	∞	FPT
	$O(1)$	0	∞	FPT
	1	1	1	$W[1]$-complete

Table 1. Classical, approximation, and parameterized complexity maps of p-OLSE and p-OLISE with respect to Δ_H, Δ_G and Δ_L. The inapproximability results are under the assumption that $\mathcal{P} \neq \mathcal{NP}$. The symbol ∞ stands for unbounded degree, and the results with the $O(1)$ upper bound on the degree hold true for *any* fixed degree.

Parameterized Complexity. A *parameterized problem* is a set of instances of the form (x, k), where $x \in \Sigma^*$ for a finite alphabet set Σ, and k is a non-negative integer called the *parameter*. A parameterized problem Q is *fixed parameter tractable* (FPT), if there exists an algorithm that on input (x, k) decides if (x, k) is a yes-instance of Q in time $f(k)n^{O(1)}$, where f is a computable function independent of $n = |x|$; we will denote by *fpt-time* a running time of the form $f(k)n^{O(1)}$. A hierarchy of fixed-parameter intractability, *the W-hierarchy* $\bigcup_{t\geq 0} W[t]$, was introduced based on the notion of *fpt-reduction*, in which the 0-th level $W[0]$ is the class FPT. It is commonly believed that $W[1] \neq$ FPT. The asymptotic notation $O^*()$ suppresses a polynomial factor in the input length.

We will denote an instance of p-OLSE or p-OLISE by the tuple $(G, H, \prec_G, \prec_H, L, k)$. We shall call an injective map φ satisfying conditions (1)-(3) in the definition of p-OLSE and p-OLISE (given in Section 1) for some subgraph S of G, a *valid list embedding*, or simply a *valid embedding*. Constraint (3) will be referred to as the *embedding constraint* (note that constraint (3) is different in the two problems). We define the *width* of L, denoted Δ_L, as $\max\{|L(v)| \mid v \in G\}$. It is often more convenient to view/represent the map L as a set of edges joining every vertex $u \in G$ to the vertices of H that are in $L(u)$.

2 Complexity Results

Consider the restrictions of the opt-OLSE and opt-OLISE problems to instances in which $\Delta_G = \Delta_H = 0$ (for p-OLSE we can even assume that $\Delta_H = \infty$ as the edges in H do not play any role when $\Delta_G = 0$). This version of the problem can be easily shown to be solvable in polynomial time by dynamic programming:

Proposition 1. *(Proposition 8, Appendix) The opt-OLSE and opt-OLISE problems (and hence p-OLSE and p-OLISE) restricted to instances in which $\Delta_G = \Delta_H = 0$ are solvable in $O(|V(G)| \cdot |V(H)|)$ time (and hence are in $\mathcal{P}$).*

Proposition 1 will be useful for Proposition 2 and Theorem 3.

If $\Delta_G > 0$, the p-OLSE and p-OLISE problems become $\mathcal{NP}$-complete, even in the simplest case when $\Delta_G = 1$, $\Delta_H = 0$ and $\Delta_L = 1$. For p-OLISE, the same proof by symmetry shows the NP-completeness of the problem when $\Delta_H = 1$, $\Delta_G = 0$ and $\Delta_L = 1$ (this version of p-OLSE is in $\mathcal{P}$).

Theorem 1. *(Theorem 4, Appendix) The p-OLSE and p-OLISE problems restricted to instances in which $\Delta_G = 1$, $\Delta_H = 0$ and $\Delta_L = 1$ are $\mathcal{NP}$-complete.*

Proposition 2. *The opt-OLSE problem restricted to instances in which $\Delta_G = O(1)$ has an approximation algorithm of ratio $(\Delta_G + 1)$, and the opt-OLISE problem restricted to instances in which $\Delta_G = O(1)$ and $\Delta_H = O(1)$ has an approximation algorithm of ratio $(\Delta_H + 1) \cdot (\Delta_G + 1)$.*

Proof. Let $(G, H, \prec_G, \prec_H, L)$ be an instance of opt-OLSE, and consider the following algorithm. Apply the dynamic programming algorithm in Proposition 1 to $(G, H, \prec_G, \prec_H, L)$ after removing the edges of G and the edges of H, and let S and φ be the subgraph and map obtained, respectively. Apply the following trivial approximation algorithm to compute an independent set I of S: pick a vertex v in S, include v in I, remove v and $N(v)$ from S, and repeat until S is empty. Return the subgraph $G[I] = I$, and the restriction of φ to I, φ_I. Clearly, we have $|I| \geq |S|/(\Delta_G + 1)$.

Since φ is a valid list embedding of S with respect to G and H with their edges removed, and since I is an independent set of $G[S]$, it is clear that φ_I is a valid list embedding of $G[I]$ into H. Therefore, the algorithm is an approximation algorithm. Now let S_{opt} be an optimal solution of the instance. Clearly, we have $|S_{opt}| \leq |S|$. Therefore, $|S_{opt}|/|I| \leq |S|/|I| \leq (\Delta_G + 1)$.

For opt-OLISE, we follow the same steps to obtain I, and then we apply a similar approximation algorithm to H, to retain in I a subset whose image under φ_I is an independent set of H. At least $|I|/(\Delta_H + 1)$ vertices are retained. □

The inapproximability results outlined in Table 1 follow by simple reductions from the Maximum Independent Set problem. The APX-hardness for the restrictions of opt-OLSE and opt-OLISE considered in Proposition 2 follow by a reduction from Maximum Independent Set on bounded-degree graphs. On the other hand, the inapproximability results for opt-OLSE and opt-OLISE when $\Delta_G = \infty$, $\Delta_L = 1$ and $\Delta_H = 0$ (and the symmetric case for opt-OLISE when $\Delta_G = 0$, $\Delta_L = 1$ and $\Delta_H = \infty$) follow by a reduction from the (general) Maximum Independent Set that was given in [3] for a variation of p-OLSE.

Evans [6] proved that the Longest Arc-Preserving Common Subsequence (LAPCS) is $W[1]$-complete. LAPCS is a special case of p-OLISE, and it turns out that the reduction in [6] results in an instance that can be modeled by an instance of either p-OLSE or p-OLISE in which $\Delta_H = 1$, and $\Delta_G = 1$:

Proposition 3. *The p-OLSE and p-OLISE problems restricted to instances in which $\Delta_H = 1$ and $\Delta_G = 1$ are $W[1]$-complete.*

The reduction in [6] can also be tweaked to show that p-OLISE, restricted to instances in which $\Delta_H = O(1)$, $\Delta_G = O(1)$, and $\Delta_L = O(1)$ is $W[1]$-complete. To see this, observe first that the result would follow if we proved the $W[1]$-hardness of p-OLISE restricted to instances in which $\Delta_H = O(1)$, $\Delta_G = O(1)$, and the number of vertices in G that have the same vertex $v \in H$ in their list is also $O(1)$, since that would correspond to $\Delta_L = O(1)$ if we switched G and H (by symmetry). Using color-coding, the reduction in [6] can be tweaked to a Turing fpt-reduction in which $\Delta_G = 1$, $\Delta_H = 1$, and every vertex $v \in H$ appears in the list of exactly one vertex in G:

Proposition 4. *The p-OLISE problem restricted to instances in which $\Delta_H = 1$, $\Delta_G = 1$, and $\Delta_L = 1$ is $W[1]$-complete.*

The result below follows from a reduction given in [3] for a variant of p-OLSE:

Proposition 5. *([3], Theorem 0.3) The p-OLSE problem restricted to instances in which $\Delta_G = \infty$, $\Delta_H = 0$ and $\Delta_L = 1$ is $W[1]$-complete, and the p-OLISE problem restricted to instances in which $\Delta_G = \infty$ (resp. $\Delta_H = \infty$ by symmetry), $\Delta_H = 0$ (resp. $\Delta_G = 0$) and $\Delta_L = 1$ is $W[1]$-complete.*

3 FPT results

We start by discussing the results for p-OLSE when both Δ_G and Δ_L are $O(1)$ (Δ_H may be unbounded). Let $(G, H, \prec_G, \prec_H, L, k)$ be an instance of p-OLSE in which both Δ_G and Δ_L are upper bounded by a fixed constant. Consider the graph $\mathcal{G}$ whose vertex-set is $V(G) \cup V(H)$ and whose edge-set is $E(G) \cup E(H) \cup E_L$, where $E_L = \{uv \mid u \in G, v \in H, v \in L(u)\}$; that is, $\mathcal{G}$ is the union of G and H plus the edges that represent the mapping L. We perform the following *splitting* operation on the vertices of $\mathcal{G}$ (see Figure 1 for illustration):

Definition 1. *Let u be a vertex in $\mathcal{G}$ and assume that $u \in G$ (the operation is similar when $u \in H$). Suppose that the vertices of G are ordered as $\langle u_1, \ldots, u_n \rangle$ with respect to $\prec_G$, and suppose that $u = u_i$, for some $i \in \{1, \ldots, n\}$. Let $e_1 = uv_1, \ldots, e_r = uv_r$ be the edges incident to u in E_L, and assume that $v_1 \prec_H v_2 \prec_H \ldots \prec_H v_r$. By* splitting *vertex u we mean: (1) replacing u in $\mathcal{G}$ with vertices $u_i^1, \ldots, u_i^r$ such that the resulting ordering of the vertices in G with respect to $\prec_G$ is $\langle u_1, \ldots, u_{i-1}, u_i^1, \ldots, u_i^r, u_{i+1}, \ldots, u_n \rangle$; (2) removing all the edges $e_1, \ldots, e_r$ from $\mathcal{G}$ and replacing them with the edges $u_i^1 v_r, u_i^2 v_{r-1}, \ldots, u_i^r v_1$; and (3) replacing every edge uu_j in G with the edges $u_i^s u_j$, for $s = 1, \ldots, r$.*

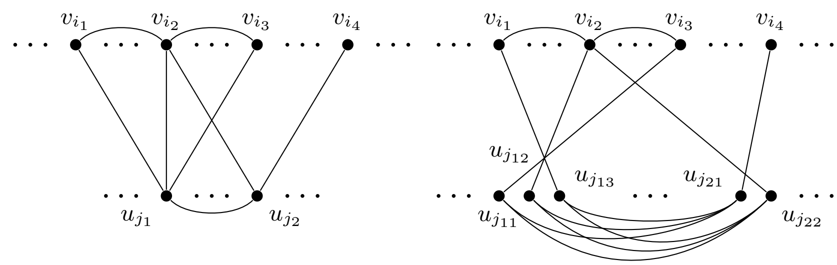

Fig. 1. Illustration of the splitting operation when applied to vertices u_{j_1} and u_{j_2}.

Let $\mathcal{G}_{split}$ be the graph resulting from $\mathcal{G}$ by splitting every vertex in G and every vertex in H (in an arbitrary order), where G_{split} is the graph resulting from splitting the vertices in G and H_{split} that resulting from splitting the vertices of H. Let E_{split} be the set of edges having one endpoint in G_{split} and the other in H_{split}, $L_{split} : G_{split} \longrightarrow 2^{V(H_{split})}$ defined by $L_{split}(u) = \{v \mid uv \in E_{split}\}$ for $u \in V(G_{split})$, and let $\prec_{G_{split}}$ and $\prec_{H_{split}}$ be the orders on G_{split} and H_{split}, respectively, resulting from $\prec_G, \prec_H$ after the splitting operation. The following lemma can be easily verified:

Lemma 1. *The graph $\mathcal{G}_{split}$ satisfies the properties: (i) for every $u \in V(G_{split})$ we have $deg_{G_{split}}(u) \leq \Delta_L \cdot \Delta_G$;*[3] *(ii) in the graph $(V(G_{split}) \cup V(H_{split}), E_{split})$ every vertex has degree exactly 1 (in particular $|L_{split}(u)| = 1$ for every $u \in V(G_{split})$), and (iii) the instance $(G, H, \prec_G, \prec_H, L, k)$ is a yes-instance of p-OLSE if and only if $(G_{split}, H_{split}, \prec_{G_{split}}, \prec_{H_{split}}, L_{split}, k)$ is.*

Next, we perform the following operation, denoted **Simplify**, to $\mathcal{G}_{split}$. Since every vertex in $(V(G_{split}) \cup V(H_{split}), E_{split})$ has degree 1 by part (ii) of Lemma 1, if two vertices $u, u' \in G_{split}$ are such that either (1) $uu' \notin E(G_{split})$ but $vv' \in E(H_{split})$ or (2) both $uu' \in E(G_{split})$ and $vv' \in E(H_{split})$, where $\{v\} = L_{split}(u)$ and $\{v'\} = L_{split}(u')$, then we can remove edge vv' from $E(H_{split})$ in case (1) and we can remove both edges uu' and vv' in case (2) without affecting any embedding constraint. Without loss of generality, we will still denote by $(G_{split}, H_{split}, \prec_{G_{split}}, \prec_{H_{split}}, L_{split}, k)$ the resulting instances after the removal of the edges satisfying cases (1) and (2) above. Note that $E(H_{split}) = \emptyset$ at this point, and hence if $uu' \in E(G_{split})$ then no valid list embedding can be defined on a subset that includes both u and u'. Note also that $\mathcal{G}_{split} - E(G_{split})$ is a realization of a permutation graph P in which the vertices of G_{split} can be arranged on one line according to the order induced by $\prec_{G_{split}}$, and the vertices of H_{split} can be arranged on a parallel line according to the order induced by $\prec_{H_{split}}$. The vertex-set of P corresponds to the edges in E_{split}, and two vertices in P are adjacent if and only if their two corresponding edges cross. Note that two vertices in P correspond to two edges of the form $e = uv$ and $e' = u'v'$, where $u, u' \in G_{split}$ and $v, v' \in H_{split}$. Let $\mathcal{I}$ be the graph whose vertex-set is $V(P)$ and whose edge set is $E(P) \cup E_c$, where $E_c = \{ee' \mid e, e' \in V(P), e = uv, e' = u'v', uu' \in E(G_{split})\}$

[3] Note that the degree of a vertex in H_{split} may be unbounded.

is the set of *conflict edges*; that is, $\mathcal{I}$ consists of the permutation graph P plus the set of conflict edges E_c, where each edge in E_c joins two vertices in P whose corresponding endpoints in G_{split} cannot both be part of a valid solution.

Lemma 2. *(Lemma 5, Appendix) For every vertex $e \in \mathcal{I}$, the number of conflict edges incident to e in $\mathcal{I}$, denoted $deg_c(e)$, is at most $\Delta_L \cdot \Delta_G$.*

Lemma 3. *The instance $(G_{split}, H_{split}, \prec_{G_{split}}, \prec_{H_{split}}, L_{split}, k)$, and hence $(G, H, \prec_G, \prec_H, L, k)$, before* **Simplify** *is applied is a yes-instance of p-OLSE if and only if $\mathcal{I}$ has an independent set of size k.*

Proof. A size-k independent set I in $\mathcal{I}$ corresponds to a set of k edges $u_{i_1}v_{j_1}, \ldots, u_{i_k}v_{j_k}$ in $\mathcal{G}_{split}$ such that $u_{i_1} \prec_{G_{split}} \ldots \prec_{G_{split}} u_{i_k}$, $v_{j_1} \prec_{H_{split}} \ldots \prec_{H_{split}} v_{j_k}$, and $S = G_{split}[\{u_{i_1}, \ldots, u_{i_k}\}]$ is a subgraph in G_{split} whose vertices form an independent set. Clearly, the embedding $\varphi(u_{i_s}) = \{v_{j_s}\}$, $s = 1, \ldots, k$, is a valid embedding that embeds S into H_{split} because it respects both $\prec_{G_{split}}, \prec_{H_{split}}$, and because it respects the embedding constraints. To see why the latter statement is true, note that, for any two vertices u_{i_s} and u_{i_r} $(r \neq s)$ in S, either there was no edge between u_{i_s} and u_{i_r} before the application of the operation **Simplify**, or there was an edge and got removed by **Simplify**, and in this case there must be also an edge between v_{j_s} and v_{j_r} in H_{split}; in either case, φ respects the embedding constraints.

Conversely, let φ be a valid embedding that embeds a subgraph S of size k where $V(S) = \{u_{i_1}, \ldots, u_{i_k}\}$, and $\varphi(u_{i_s}) = v_{j_s}$, for $s = 1, \ldots, k$. We claim that the set of vertices $I = \{e_1 = u_{i_1}v_{j_1}, \ldots, e_k = u_{i_k}v_{j_k}\}$ is an independent set in $\mathcal{I}$. Since φ is a valid list embedding, no edge in P exists between any two vertices in I. Let $e_r = u_{i_r}v_{j_r}$ and $e_s = u_{i_s}v_{j_s}$ be two vertices in I, where $r \neq s$. If there is no edge between u_{i_r} and u_{i_s} in $E(G_{split})$, then no edge exists between e_s and e_r in $\mathcal{I}$. On the other hand, if there is an edge between u_{i_r} and u_{i_s} in $E(G_{split})$, then because φ is a valid embedding, there must be an edge as well between v_{i_r} and v_{i_s}. After applying **Simplify**, the edge between u_{i_r} and u_{i_s} will be removed, and hence no edge exists between e_s and e_r in $\mathcal{I}$. It follows that I is a size-k independent set in $\mathcal{I}$. □

Lemma 4. *Let $\mathcal{C}$ be a hereditary class of graphs on which the* Independent Set *problem is solvable in polynomial time, and let $\Delta \geq 0$ be a fixed integer constant. Let $\mathcal{C}' = \{\mathcal{I} = (V(P), E(P) \cup E_c) \mid P \in \mathcal{C}, E_c \subseteq V(P) \times V(P)\}$, where at most Δ edges in E_c are incident to any vertex in $\mathcal{I}$. Assuming that a graph in $\mathcal{C}'$ is given as $(V(P), E(P) \cup E_c)$ (*i.e., *E_c is given), the* Independent Set *problem can be solved in fpt-time on graphs in the class $\mathcal{C}'$.*

Proof. Let $(\mathcal{I} = (V(P), E(P) \cup E_c), k)$ be an instance of Independent Set, where $\mathcal{I} \in \mathcal{C}'$. We use the random separation method introduced by Cai et al. [4]; this method can be de-randomized in fpt-time using the notion of universal sets and perfect hash functions [2, 14, 15]. Therefore, we only present the randomized algorithm here, and the deterministic fpt-time algorithm follows from de-randomizing the algorithm we present.

We apply the random separation method to the subgraph $(V(P), E_C)$, and color the vertices in $V(P)$ with two colors, "green" and "red", randomly and independently. If I is an independent set of size k in $\mathcal{I}$, since there are at most Δ edges of E_c that are incident to any vertex in $\mathcal{I}$, the probability that all vertices in I are colored green and all their neighbors along the edges in E_c are colored red, is at least $2^{-k+\Delta k} = 2^{-(\Delta+1)k}$. Using universal sets and perfect hash functions, by trying FPT-many 2-colorings, if a size-k independent set exists, then there is a 2-coloring among the ones we try that will result in the independent set vertices being colored green, and all their neighbors along edges in E_c being colored red. Therefore, it suffices to determine, given a 2-colored graph $\mathcal{I}$, whether there is an independent set of size k consisting of green vertices whose neighbors along the edges in E_c are red vertices. We explain how to do so next.

Suppose that the vertices in $\mathcal{I}$ are colored green or red, and $\mathcal{I}_g$ be the subgraph of $\mathcal{I}$ induced by the green vertices, and $\mathcal{I}_r$ that induced by the red vertices. Notice that if there is an independent set I consisting of k green vertices whose neighbors along the edges in E_c are red, then for each vertex u in I, u is an isolated vertex in the graph $(V(\mathcal{I}_g), E_c)$. Moreover, since I is an independent set, then no edge in $E(P)$ exists between any two vertices in I. Therefore, if we form the subgraph $G_0 = (V_0, E_0)$, where V_0 is the set of vertices in $\mathcal{I}_g$ that are isolated with respect to the set of edges E_c, and E_0 is the set of edges in $E(P)$ whose both endpoints are in V_0, then I is an independent set in G_0. On the other hand, any independent set of G_0 is also an independent set of $\mathcal{I}$. Since G_0 is a subgraph of $P \in \mathcal{C}$ and $\mathcal{C}$ is hereditary, it follows that $G_0 \in \mathcal{C}$ and we can compute a maximum independent set I_{max} in G_0 in polynomial time. If $|I_{max}| \geq k$, then we accept the instance; otherwise, we try the next 2-coloring. If no 2-coloring results in an independent set of size at least k, we reject. □

Theorem 2. *The p-OLSE problem restricted to instances in which $\Delta_G = O(1)$ and $\Delta_L = O(1)$ is FPT.*

Proof. Let $(G, H, \prec_G, \prec_H, L, k)$ be an instance of p-OLSE in which both Δ_G and Δ_L are upper bounded by a fixed constant. We form the graph $\mathcal{G}$ and perform the splitting operation described in Definition 1 to obtain the instance $(G_{split}, H_{split}, \prec_{G_{split}}, \prec_{H_{split}}, L_{split}, k)$. By Lemma 1, $(G, H, \prec_G, \prec_H, L, k)$ is a yes-instance of p-OLSE if and only if $(G_{split}, H_{split}, \prec_{G_{split}}, \prec_{H_{split}}, L_{split}, k)$ is. We now apply the operation **Simplify** to the instance and construct the graph $\mathcal{I} = (V(P), E(P) \cup E_c)$ as described above, where P is a permutation graph. Note that the set of edges E_c is known to us. By Lemma 3, $(G, H, \prec_G, \prec_H, L, k)$ is a yes-instance of p-OLSE if and only if $\mathcal{I}$ has an independent set of size k. Since the Independent Set problem is solvable in polynomial time on the class of permutation graph (*e.g.*, see [12]), the class of permutation graph is hereditary, and every vertex is in $\mathcal{I}$ has at most $\Delta_L \cdot \Delta_G$ edges in E_c incident to it by Lemma 2, it follows from Lemma 4 that we can decide if $\mathcal{I}$ has an independent set of size k in fpt-time. □

Unfortunately, the above result does not hold true for p-OLISE, even when $\Delta_G = O(1)$, $\Delta_H = O(1)$, and $\Delta_L = O(1)$, because the number of vertices in G that have the same vertex $v \in H$ in their list may be unbounded.

From Proposition 3, we know that in the case when Δ_L is unbounded, and both $\Delta_G = 1$ and $\Delta_H = 1$, p-OLSE is W[1]-hard. The following proposition says that the condition $\Delta_H = 1$ is essential for this W-hardness result:

Proposition 6. *The p-OLSE and p-OLISE problems restricted to instances in which $\Delta_H = 0$, $\Delta_G = O(1)$ (resp. $\Delta_G = 0$ and $\Delta_H = O(1)$ for p-OLISE by symmetry) and $\Delta_L = \infty$ are FPT.*

Proof. We prove the result for p-OLSE. The proof is exactly the same for p-OLISE. The proof uses the random separation method, but is simpler than the proof of Lemma 4. Let $(G, H, \prec_G, \prec_H, L, k)$ be an instance of p-OLSE. Observe that if S is the solution that we are looking for then $G[S]$ must be an independent set since $\Delta_H = 0$. Use the random separation method to color G with green or red. Since $\Delta_G = O(1)$, if a solution S exists, then in fpt-time (deterministic) we can find a 2-coloring in which all vertices in S are green and their neighbors in G are red. So we can work under this assumption. Let G_g be the subgraph of G induced by the green vertices, and G_r that induced by the red vertices. Observe that any green vertex in G_g that is not isolated in G_g can be discarded by our assumption (since all neighbors of a vertex in S must be in G_r). Therefore, we can assume that G_g is an independent set. We can now compute a maximum cardinality subgraph of G_g that can be (validly) embedded into H using the dynamic programming algorithm in Proposition 1; if the subgraph has size at least k we accept; otherwise, we try another 2-coloring of G. If no 2-coloring of G results in a solution of size at least k, we reject. □

4 Parameterization by the Vertex Cover Number

We study the parameterized complexity of p-OLSE parameterized by the size of a vertex cover ν in the graph G; we denote the corresponding problem with p-VC-OLSE. The reduction in [6], which can be used to prove the $W[1]$-hardness of p-OLSE when restricted to instances in which $\Delta_H \leq 1$, $\Delta_G \leq 1$ and Δ_L is unbounded, results in an instance in which the number of vertices in G, and hence $\tau(G)$, is upper bounded by a function of the parameter. Therefore:

Proposition 7. *p-VC-OLSE restricted to instances in which $\Delta_H = 1$ and $\Delta_G = 1$ is $W[1]$-complete.*

Therefore, we can focus our attention on studying the complexity of p-VC-OLSE restricted to instances in which $\Delta_L \in O(1)$.

Theorem 3. *p-VC-OLSE restricted to instances in which $\Delta_L = O(1)$ is FPT.*

Proof. Let $\Delta \geq 0$ be any fixed integer, and suppose that $\Delta_L \leq \Delta$. Let $(G, H, \prec_G, \prec_H, L, k, \nu)$ be an instance of p-VC-OLSE, where k is the desired solution size and ν is the size of a vertex cover in G. In fpt-time (in ν) we can compute a vertex cover C of G of size ν (if no such vertex cover exists we reject the instance). Let $I = V(G) \setminus C$, and note that I is an independent set of G. Suppose that the solution we are seeking (if it exists) is S, and the valid mapping of S is φ. Let $S_C = S \cap C$, where S_C is possibly empty, and let φ_C be the restriction of φ to S_C. We enumerate each subset of C as S_C, enumerate each possible mapping from S_C to $L(S_C)$ as φ_C, and check the validity of φ_C (if no valid φ_C exists, we reject the enumeration); since $\Delta_L \leq \Delta$, the enumeration can be carried out in fpt-time in ν, and more specifically, in time $O^*((2\Delta)^\nu)$. Therefore, we will work under the assumption that the desired solution intersects C at a known subset S_C, and that the restriction of φ to S_C is a known map φ_C, and reject the instance if this assumption is proved to be wrong.

We now remove all vertices in $C \setminus S_C$ from G (together with their incident edges) and update L accordingly; without loss of generality, we will still use G to refer to the resulting graph whose vertex-set at this point is $I \cup S_C$. Let $u_{i_1}, \ldots, u_{i_r}$ be the vertices in S_C, where $u_{i_1} \prec_G \ldots \prec_G u_{i_r}$. Since φ_C is valid, we have $\varphi(u_{i_1}) \prec_H \ldots \prec_H \varphi(u_{i_r})$. We now perform the following operation. For each vertex u in I and each vertex $v \in L(u)$, if setting $\varphi(u) = v$ violates the embedding constraint in the sense that either (1) there is a vertex $u_{i_j} \in S_C$ such that $uu_{i_j} \in E(G)$ but $v\varphi_C(u_{i_j}) \notin E(H)$ or (2) there is a vertex $u_{i_j} \in S_C$ such that $u \prec u_{i_j}$ (resp. $u_{i_j} \prec u$) but $\varphi_C(u_{i_j}) \prec v$ (resp. $v \prec \varphi_C(u_{i_j})$), then remove v from $L(u)$. Afterwards, partition the vertices in G into at most $r+1$ intervals, $I_0, \ldots, I_r$, where I_0 consists of the vertices preceding u_{i_1} (with respect to $\preceq_G$), I_r consists of those vertices following u_{i_r}, and I_j consists of those vertices that fall strictly between vertices $u_{i_{j-1}}$ and u_{i_j}, for $j = 1, \ldots, r$. Similarly, partition H into $r+1$ intervals, $I'_0, \ldots, I'_r$, where I'_0 consists of the vertices preceding $\varphi_C(u_{i_1})$ (with respect to $\prec_H$), I'_r those vertices following $\varphi_C(u_{i_r})$, and I'_j those vertices that fall strictly between vertices $\varphi_C(u_{i_{j-1}})$ and $\varphi_C(u_{i_j})$, for $j = 1, \ldots, r$. Clearly, any valid mapping φ that respects $\prec_G$ and $\prec_H$ must map vertices in the solution that belong to I_j to vertices in I'_j, for $j = 0, \ldots, r$, in a way that respects the restrictions of $\prec_G$ and $\prec_H$ on I_j and I'_j, respectively. On the other hand, since after the above operation every vertex u in I can be validly mapped to any vertex $v \in L(u)$, any injective mapping φ_j that maps a subset of vertices in I_j to a subset in I'_j in a way that respects the restrictions of $\prec_G$ and $\prec_H$ to I_j and I'_j, respectively, can be extended to a valid embedding whose restriction to S_C is φ_C. Therefore, our problem reduces to determining whether there exist injective maps φ_j, $j = 0, \ldots, r$, mapping vertices in I_j to vertices in I'_j, such that the total number of vertices mapped in the I_j's is $k - r$. Consider the subgraphs $G_j = G[I_j]$ and $H_j = H[I'_j]$, for $j = 0, \ldots, r$. Since G_j is an independent set, the presence of edges in H_j does not affect the existence of a valid list mapping from vertices in G_j to H_j, and hence those edges can be removed. Therefore, we can solve the opt-OLSE problem using Proposition 1 on the two graphs G_j and H_j to compute a maximum cardinality subset of vertices S_j in G_j that can be validly embedded into H_j via an embedding φ_j. If the union of the S_j's with S_C has cardinality at least k, we accept. If after all enumerations of S_C and φ_C we do not accept, we reject the instance. This algorithm runs in time $O^*((2\Delta)^\nu)$. □

References

1. J. Alber, J. Gramm, J. Guo, and R. Niedermeier. Computing the similarity of two sequences with nested arc annotations. *Theoretical Computer Science*, 312(2-3):337–358, 2004.
2. N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of the ACM*, 42:844–856, 1995.
3. C. Ashby, X. Huang, D. Johnson, I. Kanj, K. Walker, and G. Xia. New algorithm for protein structure comparison and classification. *BMC Genomics*, 14(S2:S1), 2012.
4. L. Cai, S.-M. Chan, and S.-O. Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *IWPEC*, pages 239–250, 2006.
5. P. Evans. Algorithms and complexity for annotated sequence analysis. Technical report, Ph.D. thesis, University of Victoria, 1999.
6. P. Evans. Finding common subsequences with arcs and pseudoknots. In *CPM*, pages 270–280, 1999.
7. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-verlag, Berlin, Germany, 2010.
8. D. Goldman, S. Istrail, and C.H. Papadimitriou. Algorithmic aspects of protein structure similarity. In *FOCS*, pages 512–522, 1999.
9. J. Gramm, J. Guo, and R. Niedermeier. On exact and approximation algorithms for distinguishing substring selection. In *FCT*, pages 195–209, 2003.
10. Sepp Hartung and Rolf Niedermeier. Incremental list coloring of graphs, parameterized by conservation. 6108:258–270, 2010.
11. T. Jiang, G. Lin, B. Ma, and K. Zhang. The longest common subsequence problem for arc-annotated sequences. *J. Discrete Algorithms*, 2(2):257–270, 2004.
12. H. Kim. Finding a maximum independent set in a permutation graph. *Information Processing Letters*, 36(1):19–23, 1990.
13. G.-H. Lin, Z.-Z. Chen, T. Jiang, and J. Wen. The longest common subsequence problem for sequences with nested arc annotations. *Journal of Computer and System Sciences*, 65(3):465–480, 2002.
14. M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *FOCS*, pages 182–191, 1995.
15. J. Schmidt and Alan Siegel. The spatial complexity of oblivious k-probe hash functions. *SIAM Journal on Computing*, 19(5):775–786, 1990.
16. Y. Song, C. Liu, X. Huang, R. Malmberg, Y. Xu, and L. Cai. Efficient parameterized algorithms for biopolymer structuresequence alignment. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 3(4):423–432, 2006.

5 Appendix

5.1 Approximation Algorithms Terminologies

An *NP optimization problem* Q is a 4-tuple (I_Q, S_Q, f_Q, g_Q). I_Q is the set of input instances, which is recognizable in polynomial time. For each instance $x \in I_Q$, $S_Q(x)$ is the set of feasible solutions for x, which is defined by a polynomial p and a polynomial-time computable predicate π (p and π depend only on Q) as $S_Q(x) = \{y : |y| \leq p(|x|) \ \wedge \ \pi(x,y)\}$. The function $f_Q(x,y)$ is the objective function mapping a pair $x \in I_Q$ and $y \in S_Q(x)$ to a non-negative integer. The function f_Q is computable in polynomial time. The function g_Q is the *goal* function, which is one of the two functions $\{\max, \min\}$, and Q is called a *maximization problem* if $g_Q = \max$, or a *minimization problem* if $g_Q = \min$. We will denote by $opt_Q(x)$ the value $g_Q\{f_Q(x,z) \mid z \in S_Q(x)\}$, and if there is no confusion about the underlying problem Q, we will write $opt(x)$ to denote $opt_Q(x)$.

An algorithm A is an *approximation algorithm* for a (maximization) problem Q if for each input instance $x \in I_Q$ the algorithm A returns a feasible solution $y_A(x) \in S_Q(x)$. The solution $y_A(x)$ has an *approximation ratio* $r(|x|)$ if it satisfies the following condition:

$$opt_Q(x)/f_Q(x, y_A(x)) \leq r(|x|).$$

The approximation algorithm A has an *approximation ratio* $r(|x|)$ if for every instance x in I_Q the solution $y_A(x)$ constructed by the algorithm A has an approximation ratio bounded by $r(|x|)$.

An optimization problem Q has a *constant-ratio approximation algorithm* if it has an approximation algorithm whose ratio is a constant (i.e., independent from the input size).

5.2 Omitted Proofs

Proposition 8. *The opt-OLSE and opt-OLISE problems (and hence p-OLSE and p-OLISE) restricted to instances in which $\Delta_G = \Delta_H = 0$ are solvable in $O(|V(G)| \cdot |V(H)|)$ time (and hence are in $\mathcal{P}$).*

Proof. Assume that the vertices of G are ordered as $u_1, \ldots, u_n$ with respect to $\prec_G$, and that the vertices of H are ordered as $v_1, \ldots v_N$ with respect to $\prec_H$. We maintain a two-dimensional table T, in which $T[i,j]$ is the maximum cardinality of a subset of $\{u_1, \ldots, u_i\}$ from which there exists a valid list embedding into $\{v_1, \ldots, v_j\}$. The entry $T[i,j]$ can be computed recursively as follows: $T[i,j] = 1 + T[i-1, j-1]$ if $v_j \in L(u_i)$, and $T[i,j] = \max\{$T[i, j-1],T[i-1, j]$\}$ otherwise. We can now use a standard dynamic programming approach to compute T and to construct the subgraph S and the embedding φ. The running time of the algorithm is $O(|V(G)| \cdot |V(H)|)$. □

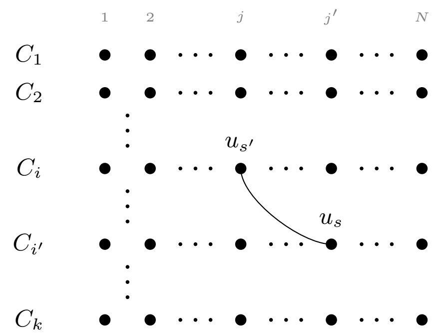

Fig. 2. An instance of k-MCIS consists of kN vertices partitioned into k color classes $C_1, \ldots, C_k$ each of size N. All edges are between vertices in different color classes. The edge $(u_{s'}, u_s)$, for example, connects vertices $u_{s'}$ and u_s where $s' = (i-1)N + j$ and $s = (i'-1)N + j'$.

Theorem 4. *The p-OLSE and p-OLISE problems restricted to instances in which $\Delta_H = 0$, $\Delta_G = 1$, and $\Delta_L = 1$ are $\mathcal{NP}$-complete.*

Proof. We present the proof only for p-OLSE because the same proof works for p-OLISE as well.

It is easy to see that p-OLSE $\in \mathcal{NP}$, so it suffices to show that p-OLSE is $\mathcal{NP}$-hard. We do so by providing a polynomial time reduction from the k-Multi-Colored Independent Set (k-MCIS) problem: decide whether for a given graph $M = (V(M), E(M))$ and k-coloring of the vertices $f : V(M) \longrightarrow C$ (where $C = \{1, 2, ..., k\}$) there exists an independent set $I \subseteq V(M)$ of size k such that $\forall u, v \in I : f(u) \neq f(v)$ [10].

Let (M, f) be an instance of k-MCIS. Without loss of generality, we assume that the set C_i of vertices that are mapped to color $i \in C$ is of size N, and we label the vertices of M $u_1, \ldots, u_{kN}$ such that $C_i = \{u_{(i-1)N+1}, \ldots, u_{iN}\}$ (see Figure 2). We now describe how we construct the corresponding instance $(G, H, \prec_G, \prec_H, L, k)$ of p-OLSE.

To construct $G = (V(G), E(G))$, we associate with every color class C_i a block B_i of vertex sequences $b_i^1, \ldots, b_i^N$ such that each b_i^j is a sequence of kN vertices $u_{i,1}^j, \ldots, u_{i,kN}^j$. $V(G)$ is then the set of all vertices $u_{i,s}^j$ for $i = 1, \ldots, k$, $j = 1, \ldots, N$, and $s = 1, \ldots, kN$. The ordering $\prec_G$ of vertices in G is defined as follows: for all $u_{i,s}^j, u_{i',s'}^{j'} \in G$

$$u_{i,s}^j \prec_G u_{i',s'}^{j'} \iff i < i' \text{ or } (i = i' \text{ and } j < j'), \text{ or } (i = i' \text{ , } j = j' \text{ and } s < s').$$

(See Figure 3-(a).) Lastly we define the set $E(G)$ as follows: $(u_{i,s}^j, u_{i',s'}^{j'})$ is an edge in $E(G)$ if and only if $i < i'$, $s = (i'-1)N + j'$, $s' = (i-1)N + j$, and $(u_s, u_{s'}) \in E(M)$. (See Figure 3-(b).)

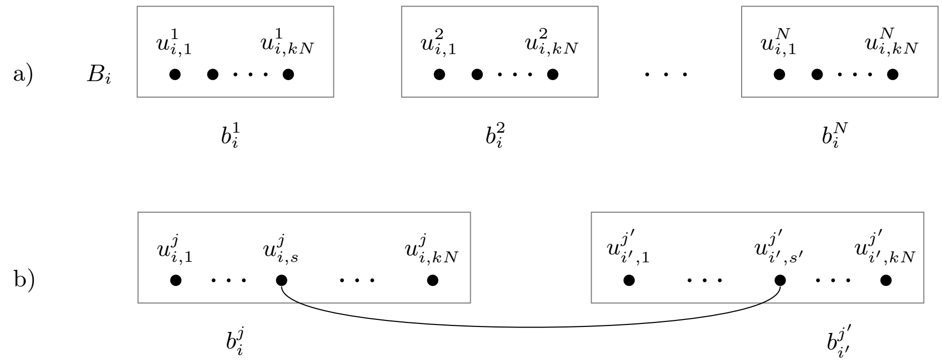

Fig. 3. a) Associated with every color class C_i is a block B_i of vertex sequences $b_i^1, \ldots, b_i^N$ such that each b_i^j is a sequence of kN vertices $u_{i,1}^j, \ldots, u_{i,kN}^j$. The vertex sequences $b_i^1, \ldots, b_i^N$ correspond to the N vertices of C_i. b) $(u_{i,s}^j, u_{i',s'}^{j'})$ is an edge in $E(G)$ if and only if $s = (i'-1)N + j'$, $s' = (i-1)N + j$, and $(u_s, u_{s'}) \in E(M)$.

To construct $H = (V(H), E(H))$, we again associate with every color class C_i a block B_i' of vertex sequences $b_i^{'1}, \ldots, b_i^{'N}$ such that each $b_i^{'j}$ is a sequence of kN vertices $v_{i,1}^j, \ldots, v_{i,kN}^j$. $V(H)$ is the set of all vertices $v_{i,r}^j$ for $i = 1, \ldots, k$, $j = 1, \ldots, N$, and $r = 1, \ldots, kN$. The ordering $\prec_H$ of vertices in H is defined just as it was in G. The edge set $E(H)$ is simply $\emptyset$: in this way, for any valid list embedding $\varphi : S \rightarrow V(H)$, where S is a subgraph of G, S must be an independent set of G.

We complete the construction of the instance of p-OLSE by defining the mapping $L : G \rightarrow 2^{V(H)}$. L is a bijection that maps the vertices in each block $B_i \in G$ to the block $B_i' \in H$ as follows:

$$\forall u_{i,s}^j \in b_i^j,\ L(u_{i,s}^j) = \{v_{i,s}^{(N-j+1)}\}.$$

This mapping is illustrated in Figure 4.

This completes the construction of the instance of p-OLSE. Observe that in the constructed instance we have $\Delta_H = 1$, $\Delta_G = 1$, and $\Delta_L = 1$. We show next that M has a k-MCIS of size k if an only if the constructed instance of p-OLSE has a subgraph S of G of size k^2N, and a valid embedding $\varphi : S \rightarrow V(H)$.

Let I be a k-MCIS in M. Each vertex $u_s \in I$ corresponds to a sequence $b_i^j \in B_i$ where $s = (i-1)N + j$. For any two sequences $b_i^j \in B_i, b_{i'}^{j'} \in B_{i'}$, there is an edge between a vertex of b_i^j and a vertex of $b_{i'}^{j'}$ if and only if $(u_s, u_{s'}) \in E(M)$ for $s' = (i-1)N + j$ and $s = (i'-1)N + j'$ $(i < i')$. Since for every pair $u_s, u_{s'} \in I, (u_s, u_{s'}) \notin E(M)$ and since $f(u_s) \neq f(u_{s'})$, there are k sequences b_i^j with the property that for any two vertices $u_{i,s}^j \in b_i^j$ and $u_{i',s'}^{j'} \in b_{i'}^{j'}$, $(u_{i,s}^j, u_{i',s'}^{j'}) \notin E(G)$. By construction of G, the vertices of any two sequences $b_i^j, b_{i'}^{j'}$ $(i \neq i')$ that do not have edges between them are simultaneously embeddable in H. Also by construction of G, the vertices of each sequence b_i^j form an independent set, and

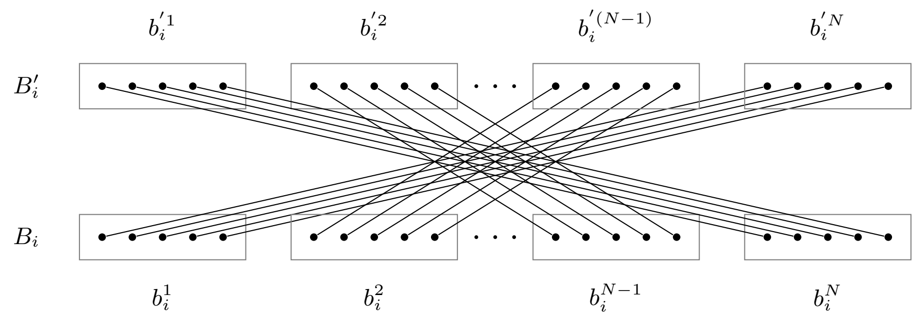

Fig. 4. The mapping L maps vertices of B_i to vertices of B_i'. More precisely, it maps vertices of b_i^j to vertices of $b_i^{\prime N-j+1}$ in a way that forbids an embedding that would map a vertex in b_i^j and a vertex in $b_i^{j'}$ simultaneously, for any $i = 1, \ldots, k$, $j, j' = 1, \ldots, N$, $j \neq j'$).

by construction of L, all vertices in the same sequence b_i^j can be simultaneously embeddable. Therefore, we can define a valid embedding $\varphi : S \to V(H)$, where $|S| = k \cdot (kN) = k^2N$ consists of all the vertices in each b_i^j that corresponds to a vertex in I.

Conversely, let $\varphi : S \to V(H)$, $|S| = k^2N$, be a valid embedding from a subgraph S of G into H. By construction of L, φ can embed at most kN vertices from any block B_i into H and, furthermore, all vertices in B_i embedded by φ must belong to the same sequence b_i^j of B_i. Therefore, exactly k sequences, one from each of the k blocks, are fully embedded by φ. Moreover, because $\Delta_H = 0$, φ embeds all kN vertices of the k sequences b_i^j in S if and only if for any two vertices $u_{i,s}^j, u_{i',s'}^{j'} \in S, (u_{i,s}^j, u_{i',s'}^{j'}) \notin E(G)$. By construction of G, each sequence b_i^j corresponds to the j^{th} vertex of the color class C_i. Therefore, we can find a subset $I \subseteq M, |I| = k$ such that for any two vertices $u_s, u_{s'} \in I, f(u_s) \neq f(u_{s'})$ and $(u_s, u_{s'}) \notin E(M)$; that is, the set I forms a k-MCIS in M. □

Lemma 5. *For every vertex $e \in \mathcal{I}$, the number of conflict edges incident to e in $\mathcal{I}$, denoted $deg_c(e)$, is at most $\Delta_L \cdot \Delta_G$.*

Proof. Let $e = uv \in \mathcal{I}$. By definition, there is an edge in E_c between e and a vertex $e' = u'v' \in \mathcal{I}$ if $uu' \in E(G_{split})$. By part (i) of Lemma 1, the degree of a vertex $u \in G_{split}$ is at most $\Delta_L \cdot \Delta_G$, and by part (ii) of Lemma 1 every vertex has degree exactly 1 in $(V(G_{split}) \cup V(H_{split}), E_{split})$. □