University of Warsaw Faculty of Mathematics, Informatics and Mechanics

Shaohua Li

Parameterized algorithms for connectivity, separation, and modification problems in graphs

 $PhD\ dissertation$

Supervisor

dr hab. Marcin Pilipczuk Institute of Informatics University of Warsaw

April 2021

Author's declaration: I hereby declare that this dissertation is my own work.

April 7, 2021 $_{date}$

.....

 $Shaohua\ Li$

Supervisor's declaration: the dissertation is ready to be reviewed.

April 7, 2021 $_{date}$

dr hab. Marcin Pilipczuk

Abstract

This thesis contains a number of new results on a few different problems concerning connectivity and separation in graphs, including algorithmic results and hardness results.

In Chapter 3, we study the INDEPENDENT FEEDBACK VERTEX SET problem — a variant of the classic FEEDBACK VERTEX SET problem where, given a graph G and an integer k, the problem is to decide whether there exists a vertex set $S \subseteq V(G)$ such that $G \setminus S$ is a forest and S is an independent set of size at most k. We present an $\mathcal{O}^*((1 + \varphi^2)^k)$ -time FPT algorithm for this problem, where $\varphi < 1.619$ is the golden ratio. Our algorithm improves the previous fastest $\mathcal{O}^*(4.1481^k)$ -time algorithm of Agrawal et al. [3] and matches the running time of the algorithm for FVS given by Kociumaka and Pilipczuk [81].

In Chapter 4, we study multi-budgeted variants of the classic minimum cut problem (MULTI-BUDGETED CUT) and graph separation problems that turned out to be important in parameterized complexity: SKEW MULTICUT and DIRECTED FEEDBACK ARC SET. In our generalization, we assign colors $1, 2, ..., \ell$ to some edges and give separate budgets $k_1, k_2, ..., k_\ell$ for colors $1, 2, ..., \ell$. For every color $i \in \{1, ..., \ell\}$, let E_i be the set of edges of color i. The solution C for the multibudgeted variant of a graph separation problem not only needs to satisfy the usual separation requirements (i.e., be a cut, a skew multicut, or a directed feedback arc set, respectively), but also needs to satisfy that $|C \cap E_i| \leq k_i$ for every $i \in \{1, ..., \ell\}$. Contrary to the classic minimum cut problem, the multi-budgeted variant turns out to be NP-hard even for $\ell = 2$. We propose an FPT algorithm parameterized by $k = k_1 + ... + k_\ell$ and ℓ for MULTI-BUDGETED CUT, and then extend our algorithm to MULTI-BUDGETED SKEW MULTICUT and MULTI-BUDGETED DIRECTED FEEDBACK ARC SET.

In Chapter 5, we study Two DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) with transition restrictions. Given a directed graph G = (V, E), a length function $w : E \to \mathbb{R}_{\geq 0}$ and two pairs of vertices $(s_1, t_1), (s_2, t_2)$ in G, the DIRECTED TWO DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) asks to find two disjoint (vertex-disjoint or edge-disjoint) paths P_1 and P_2 in G such that P_i is a shortest path from s_i to t_i for i = 1, 2. Bérczi and Kobayashi showed that 2-DSPP is polynomial-time solvable when every directed cycle has positive length [9]. We show that 2-DSPP remains polynomial-time solvable when every directed cycle has positive length even in the presence of a *transition system*. Here, a transition system allows to use only prescribed pairs of incoming and outgoing edges as consecutive edges on the constructed paths.

In Chapter 6, we study the following variant of CLUSTER EDITING. We are given a graph G = (V, E), a packing \mathcal{H} of modification-disjoint induced P_{3} s (no pair of P_{3} s in \mathcal{H} share an edge or non-edge) and an integer ℓ . The task is to decide whether G can be transformed into a union of vertex-disjoint cliques by at most $\ell + |\mathcal{H}|$ modifications (edge deletions or insertions). We show that this problem is NP-hard even when $\ell = 0$ (in which case the problem asks to turn G into a disjoint union of cliques by performing exactly one edge deletion or insertion per element of \mathcal{H}) and when each vertex is in at most 23 P_{3} s of the packing. We also show that the two-restricted version of this problem, where every vertex belongs to at most two P_{3} s of \mathcal{H} , can be solved in $O(n^{2\ell+O(1)})$ time.

In Chapter 7, we study the problem METRIC DIMENSION. The METRIC DIMENSION problem asks for a minimum-sized *resolving set* in a given (unweighted, undirected) graph G. Here, a set $S \subseteq V(G)$ is *resolving* if no two distinct vertices of G have the same distance vector to S. The complexity of METRIC DIMENSION in graphs of bounded treewidth remained elusive in the past years. Bonnet and Purohit [IPEC 2019] showed that the problem is W[1]-hard under treewidth parameterization. In this work, we strengthen their lower bound to show that METRIC DIMENSION is NP-hard in graphs of treewidth 24.

2012 ACM Subject Classification: Graph algorithms analysis

Keywords: Feedback Vertex Set, Cluster Editing, Metric Dimension, Parameterized Algorithms, Treewidth

Abstract

Poniższa rozprawa zawiera kilka nowych wyników, zarówno algorytmicznych jak i podających dolne ograniczenia, dotyczących wybranych problemów spójności i separacji w grafach.

W Rozdziale 3 rozważamy problem INDEPENDENT FEEDBACK VERTEX SET — wariant klasycznego problemu zbioru rozcyklającego (FEEDBACK VERTEX SET) w którym, mając dany graf G i liczbę całkowitą k, pytamy, czy istnieje zbiór $S \subseteq V(G)$ taki, że $G \setminus S$ jest lasem a S jest zbiorem niezależnym wielkości co najwyżej k. Pokazujemy algorytm parametryzowany (FPT) działający w czasie $\mathcal{O}^*((1 + \varphi^2)^k)$, gdzie $\varphi < 1.619$ to proporcja złotego podziału. Nasz wynik poprawia uprzednio najszybszy algorytm Agrawal i innych [3], działający w czasie $\mathcal{O}^*(4.1481^k)$, i działa w tym samym asymptotycznym czasie co algorytm Kociumaki i Pilipczuka dla klasycznego problemu zbioru rozcyklającego [81].

W rozdziale 4 rozważamy wielobudżetowe warianty klasycznego problemu minimalnego cięcia (MULTI-BUDGETED CUT) oraz centralnych problemów złożoności parametryzowanej problemów separacji: SKEW MULTICUT oraz DIRECTED FEEDBACK ARC SET. W naszym wariancie, nadajemy kolory 1, 2, ..., ℓ niektórym krawędziom i podajemy oddzielne budżety $k_1, k_2, ..., k_\ell$ dla poszczególnych kolorów 1, 2, ..., ℓ . Dla każdego koloru $i \in \{1, ..., \ell\}$, przez E_i oznaczamy zbiór krawędzi koloru i. Rozwiązanie C wielobudżetowego wariantu rozważanego problemu nie tylko musi spełnić oryginalne warunki separacji (tj., być cięciem, wielokierunkowym cięciem, lub zbiorem rozcyklającym), ale też musi spełniać warunek $|C \cap E_i| \leq k_i$ dla każdego koloru $i \in \{1, ..., \ell\}$. W przeciwieństwie do klasycznego problemu minimalnego cięcia, wariant wielobudżetowy jest NP-trudny już od przypadku $\ell = 2$. Pokazujemy algorytmy FPT przy parametryzacji przez $k = k_1 + ... + k_\ell$ i ℓ dla problemów MULTI-BUDGETED CUT, MULTI-BUDGETED SKEW MULTICUT i MULTI-BUDGETED DIRECTED FEEDBACK ARC SET.

W rozdziałe 5 rozważamy problem Two DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) z ograniczeniami przejść. Mając dany graf skierowany G = (V, E) z długościami krawędzi $w: E \to \mathbb{R}_{\geq 0}$ oraz dwie pary wierzchołków $(s_1, t_1), (s_2, t_2)$ w G, w problemie DIRECTED TWO DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) pytamy o dwie rozłączne (wierzchołkowo lub krawędziowo) ścieżki P_1 i P_2 w G takie, że P_i jest najkrótszą ścieżką z s_i do t_i dla i = 1, 2. Bérczi i Kobayashi udowodnili, że 2-DSPP jest rozwiązywalny w czasie wielomianowym jeśli każdy cykl skierowany w G ma dodatnią długość [9]. Pokazujemy, że 2-DSPP pozostaje rozwiązywalny w czasie wielomianowym (dalej przy założeniu, że każdy cykl ma dodatnią długość) nawet, jeśli do problemu dodamy tzw. system przejść. System przejść to zestaw warunków, które pary krawędzi można użyć bezpośrednio po sobie w konstruowanych ścieżkach.

W rozdziale 6 rozważamy następujący wariant problemu CLUSTER EDITING. Dany jest graf G = (V, E), rodzina \mathcal{H} rozłącznych edycyjnie indukowanych podgrafów P_3 (tj. żadne dwa podgrafy $P_3 \le \mathcal{H}$ nie mają wspólnych dwóch lub trzech wierzchołków) oraz liczba całkowita ℓ . Zadaniem jest przekształcić G graf, którego każda spójna składowa jest kliką, używając co najwyżej $\ell + |\mathcal{H}|$ edycji (dodania lub usunięcia krawędzi). Pokazujemy, że ten p roblem jest NP-trudny nawet dla $\ell = 0$ (w którym to przypadku problem sprowadza się do przeprowadzenia jednej edycji w każdym elemencie \mathcal{H}) i gdy każdy wierzchołek zawarty jest w co najwyżej 23 podgrafów P_3 w rodzinie \mathcal{H} . Pokazujemy również, że wariant, gdzie każdy wierzchołek należy do co najwyżej dwóch podgrafów $P_3 \le \mathcal{H}$, można rozwiązać w czasie $O(n^{2\ell+O(1)})$.

W rozdziale 7 badamy problem METRIC DIMENSION. Problem ten pyta o tzw. zbiór rozwiązujący (ang. resolving set) w danym (nieskierowanym, nieważonym) grafie G. Zbiór $S \subseteq V(G)$ nazywamy rozwiązującym jeśli nie ma dwóch wierzchołków G o tym samym wektorze odległości do zbioru S. Złożoność problemu METRIC DIMENSION w grafach o ograniczonej szerokości drzewowej była otwartym problemem, kilkukrotnie wymienianym w ostatnich latach. Bonnet i Purohit [IPEC 2019] pokazali, że problem ten jest W[1]-trudny przy parametryzacji szerokością drzewową. Wzmacniamy ten wynik pokazując, że METRIC DIMENSION jest NP-trudny w grafach o szerokości drzewowej 24.

Contents

1	Introduction				
	1.1 Independent Feedback Vertex Set	5			
	1.2 Multi-budgeted cut	6			
	1.3 Two Disjoint Shortest Paths Problem with transition restrictions	8			
	1.4 Cluster Editing parameterized above modification-disjoint P_3 -packings	10			
	1.5 Hardness of Metric Dimension in Graphs of Constant Treewidth	12			
	1.6 Articles	12			
	1.7 Acknowledgements	13			
2	Preliminaries	15			
	2.1 Graph Notation	15			
	2.1.1 Basic Definitions	15			
	2.1.2 Neighborhoods and degrees	15			
	2.1.3 Paths and components	16			
	2.1.4 General cuts and budget-respecting cuts	16			
	2.1.5 Cluster graph and P_{3s}	17			
	2.2 Treewidth and Pathwidth	17			
	2.3 Other notations	18			
3	Independent Feedback Vertex Set	19			
	3.1 Disjoint Independent Feedback Vertex Set	19			
	3.2 Reduction Rules for DIS-IFVS	21			
	3.3 Branching for DIS-IFVS	24			
4	Multi-budgeted Directed Cuts				
	4.1 Multi-budgeted cut	28			
	4.1.1 NP-hardness of Multi-Budgeted Cut	28			
	4.1.2 FPT algorithm for Multi-Budgeted Cut	28			
	4.2 Multi-budgeted important separators with applications	30			
5	Two Disjoint Shortest Paths Problem with transition restrictions				
	5.1 Edge-disjoint case of 2DSPP	38			
	5.2 Vertex-disjoint case of 2DSPP	41			
6	Cluster Editing parameterized above modification-disjoint P_3 -packings				
	6.1 NP-hardness for tight modification-disjoint packings	47			
	6.1.1 Construction \ldots	47			
	6.1.2 Construction \ldots	48			
	6.1.3 Correctness	59			

	6.2 XP-algorithm for 2-restricted packings			63
		6.2.1	Reduction Rules for 2-restricted P_3 -packings	64
		6.2.2	Polynomial-time Reduction to 2-SAT Problem	77
7	Har	dness	of Metric Dimension in Graphs of Constant Treewidth	81
	7.1	Reduc	tion from 3-Dimensional Matching to Multicolored Resolving Set	81
7.2 Reduction from Multicolored Resolving Set to Metric Dimension		tion from Multicolored Resolving Set to Metric Dimension	83	
		7.2.1	Construction of the forced set gadgets	83
		7.2.2	Construction of the forced vertex gadgets	84
		7.2.3	Soundness of the reduction	86
		7.2.4	Treewidth bound of the graph	88
		7.2.5	Completeness of the reduction	89

Chapter 1

Introduction

Graph problems are ubiquitous in computer science. Graphs are one of the most natural models that represent the networks in real-life world and have numerous applications in different disciplines. Computer scientists are perusing faster algorithms to solve graph problems, both in practice and in theory. On the other side, there are many graph problems which are resistant to efficient algorithms. NP-completeness theory provides some clue on these problems [30, 75, 55]. If a problem is shown to be in the class of NP-complete problems, any efficient or polynomialtime algorithms for this problem imply that every NP-complete problem admits polynomial-time algorithms. In other words, there are probably no efficient algorithms for this problem. Although NP-hardness imply strong restrictions of algorithms for problems, people are still interested in how fast a problem can be solved and where the limitations of algorithms are. Exact algorithms for NP-hard problems focus mostly on reducing the exponential part of the running time as much as possible [50]. Approximation algorithms for NP-hard (optimization) problems aim to find efficient algorithms, classically polynomial-time algorithms at the cost of the optimality of the solution. Approximation algorithms try to find an *approximate solution* such that the distance between the approximate solution and the optimal solution is within a provable guarantee [121, 122].

Recently parameterized algorithms for NP-hard problems have received a lot of attention, which focus on both the input instance and the parameter. More formally, a *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed finite alphabet. An input instance of a parameterized problem is of the form $(x, k) \in \Sigma^* \times \mathbb{N}$ and k is called the *parameter*. If a parameterized problem can be solved in time bounded by $f(k)|x|^c$, where |x| is the size of the input instance, k is the parameter, $f : \mathbb{N} \to \mathbb{N}$ is a computable function, and c is a universal constant, then we say this problem is *fixed-parameter tractable* (FPT). If a parameterized problem can be solved in time bounded by $f(k)|x|^{f(k)}$, where |x| is the size of the input instance, k is the parameter and $f : \mathbb{N} \to \mathbb{N}$ is a computable function, then we say this problem can be solved in XP time. A parameterized problem admits a *kernel* of size g(k) for some computable function gif there is a polynomial-time procedure that reduces an arbitrary instance I of this problem with parameter k to an equivalent instance I' with size and parameter value bounded by g(k). We refer to the following books for a deeper introduction to parameterized algorithms [42, 49, 31, 52].

In this thesis, we study a few graph problems, mostly concerning connectivity and separation in graphs.

1.1 Independent Feedback Vertex Set

The first problem is INDEPENDENT FEEDBACK VERTEX SET, which is a variant of the classic FEEDBACK VERTEX SET problem. Given a graph G, a feedback vertex set of G is a set of vertices

 $S \subseteq V(G)$ such that $G \setminus S$ is a forest. The FEEDBACK VERTEX SET problem (FVS) asks to find a feedback vertex set of the minimum size. This problem is a classic NP-hard problem which has been studied extensively in many fields of complexity and algorithms [1]. In the context of parameterized complexity of the FEEDBACK VERTEX SET problem, there is a long line of work improving the upper bound of the FPT algorithm for the standard parameterization of the solution size [15, 19, 21, 40, 41, 62, 73, 81, 69, 92] (i.e., the input consists of a graph G and a parameter k, and the goal is to find a feedback vertex set of size at most k or show that no such set exists). At the same time, many variants of FEEDBACK VERTEX SET received significant attention, including SUBSET FVS [37, 70, 100], GROUP FVS [34, 60, 70, 88], or SIMULTANEOUS FVS [109]. In this part, we focus on the parameterized version of the INDEPENDENT FEEDBACK VERTEX SET problem (IFVS). The formal definition of this problem is as follows.

INDEPENDENT FEEDBACK VERTEX SET (IFVS)
Input: An undirected graph G and an integer k.
Question: Is there a feedback vertex set S of size at most k such that no two vertices of S are adjacent in G.

Misra et al. gave the first FPT algorithm running in time $\mathcal{O}(5^k n^{\mathcal{O}(1)})$ and an $\mathcal{O}(k^3)$ kernel for IFVS [108]. Agrawal et al. presented an improved FPT algorithm running in time $\mathcal{O}^*(4.1481^k)$ for IFVS [3]. In this part, we propose a faster FPT algorithm.

Theorem 1. The INDEPENDENT FEEDBACK VERTEX SET problem, parameterized by the solution size, can be solved in $\mathcal{O}^*((1+\varphi^2)^k) \leq \mathcal{O}^*(3.619^k)$ time, where $\varphi = \frac{1+\sqrt{5}}{2} < 1.619$ is the golden ratio.

We remark here that the exponential function of the time bound of Theorem 1 matches the one of the algorithm of Kociumaka and Pilipczuk [81] for the classic FEEDBACK VERTEX SET problem. Since FEEDBACK VERTEX SET trivially reduces to INDEPENDENT FEEDBACK VERTEX SET (subdivide each edge once), any (deterministic) improvement to the base of the exponential function of Theorem 1 would give a similar improvement for FEEDBACK VERTEX SET. Although Iwata and Kobayashi already gave a faster FPT algorithm for FEEDBACK VER-TEX SET problem [69], they use a totally different method which is involved in some sense. We believe it still makes sense if one can show an algorithm for FEEDBACK VERTEX SET which is faster than the algorithm of Kociumaka and Pilipczuk through some method different from the one of Iwata and Kobayashi.

On the technical side, we follow the standard approach of iterative compression as in [3] to reduce to a "disjoint" version of the problem. Here, our approach diverges from the one of [3]. We follow a modified measure for the subsequent branching process, somewhat inspired by the work of Kociumaka and Pilipczuk [81]. With a number of new notions (generalized W-degree, potential nice vertices and tents) and some new reduction rules, we get a clean branching algorithm for the "disjoint" version of the problem. This allows us to get an improved and also simplified algorithm for the INDEPENDENT FEEDBACK VERTEX SET problem.

The proof of Theorem 1 is covered in Chapter 3.

1.2 Multi-budgeted cut

Graph separation problems are important topics in both theoretical area and applications. Although the famous minimum cut problem is known to be polynomial-time solvable, many wellknown variants are NP-hard, which are intensively studied from the point of view of approximation [2, 20, 46, 57, 56, 74] and, what is more relevant here, parameterized complexity. The notion of important separators, introduced by Marx [103], turned out to be fundamental for a number of graph separation problems such as MULTIWAY CUT [103], DIRECTED FEEDBACK VERTEX SET [23], or ALMOST 2-CNF SAT [114]. Further work, concerning mostly undirected graphs, resulted in a wide range of involved algorithmic techniques: applications of matroid techniques [90, 89], shadow removal [28, 106], randomized contractions [25], LP-guided branching [35, 61, 71, 67], and treewidth reduction [105], among others.

From the above techniques, only the notion of important separators and the related technique of shadow removal generalizes to directed graphs, giving FPT algorithms for DIRECTED FEEDBACK ARC SET [23], DIRECTED MULTIWAY CUT [28], and DIRECTED SUBSET FEEDBACK VERTEX SET [27]. As a result, the parameterized complexity of a number of important graph separation problems in directed graphs remains open, and the quest to investigate them has been put on by Marx in a survey from 2012 [104]. Since the publication of this survey, two negative answers have been obtained. Pilipczuk and Wahlström showed that DIRECTED MULTICUT is W[1]-hard even for four terminal pairs (leaving the case of three terminal pairs open) [112], while Lokshtanov et al. [101] showed intractability of DIRECTED ODD CYCLE TRANSVERSAL.

Saurabh posed the question of parameterized complexity of a weighted variant of DIRECTED FEEDBACK ARC SET during an open problem session at Recent Advancements in Parameterized Complexity school (December 2017), where given a directed edge-weighted graph G, an integer k, and a target weight w, the goal is to find a set $X \subseteq E(G)$ such that G - X is acyclic and X is of cardinality at most k and weight at most w. Consider a similar problem WEIGHTED st-CUT: given a directed graph G with positive edge weights and two distinguished vertices $s, t \in V(G)$, an integer k and a target weight w, decide if G admits an st-cut of cardinality at most k and weight at most w. The parameterized complexity of this problem parameterized by k is open even if G is restricted to be acyclic, while with this restriction the problem can easily be reduced to DIRECTED FEEDBACK ARC SET (add an arc (t, s) of prohibitively large weight).

The WEIGHTED st-CUT problem becomes similar to another directed graph cut problem, identified in [26], namely CHAIN ℓ -SAT. While this problem is originally formulated in CSP language, the graph formulation is as follows: given a directed graph G with a partition of edge set $E(G) = P_1 \uplus P_2 \uplus \ldots \uplus P_m$ such that each P_i is an edge set of a simple path of length at most ℓ , an integer k, and two vertices $s, t \in V(G)$, find an st-cut $C \subseteq E(G)$ such that $|\{i|C \cap P_i \neq \emptyset\}| \leq k$. This problem can easily be seen to be equivalent to minimum stcut problem (and thus polynomial-time solvable) for $\ell \leq 2$, but is NP-hard for $\ell \geq 3$ and its parameterized complexity (with k as a parameter) remains an open problem.

Although the parameterized complexity of two aforementioned problems: weighted *st*-cut problem (in general digraphs, not necessary acyclic ones) and CHAIN ℓ -SAT are still open, we make some progress towards answering this question. We define a *multi-budgeted* variant of a number of cut problems (including the minimum cut problem) and show its fixed-parameter tractability. In this variant, the edges of the graph are colored with ℓ colors, and the input specifies separate budgets for each color. More formally, we primarily consider the following problem.

Multi-budgeted cut

Input: A directed graph G, two disjoint sets of vertices $X, Y \subseteq V(G)$, an integer ℓ , and for every $i \in \{1, 2, ..., \ell\}$ a set $E_i \subseteq E(G)$ and an integer k_i .

Question: Is there a set of arcs $C \subseteq \bigcup_{i=1}^{\ell} E_i$ such that there is no directed X - Y path in $G \setminus C$ and for every $i \in [\ell], |C \cap E_i| \leq k_i$.

We observe that MULTI-BUDGETED CUT for $\ell = 2$ reduces to WEIGHTED *st*-CUT as follows. Let $(G, X, Y, E_1, E_2, k_1, k_2)$ be a MULTI-BUDGETED CUT instance for $\ell = 2$. First, observe that we may assume that $E_1 \cap E_2 = \emptyset$, as we can replace every edge $e \in E_1 \cap E_2$ with two copies $e_1 \in E_1 \setminus E_2$ and $e_2 \in E_2 \setminus E_1$. Second, construct an equivalent WEIGHTED st-CUT instance (G', s, t, k, w) as follows. To construct G', first add two vertices s, t to G and edges $\{(s, x) | x \in X\}$ and $\{(y, t) | y \in Y\}$ of prohibitively large weight. Assign also prohibitively large weight to every edge $e \in E(G) \setminus (E_1 \cup E_2)$. Assign weight $(k_1 + 1)k_2 + 1$ to every edge $e \in E_1$. For every edge $e \in E_2$, add $k_1 + 1$ copies of e to G' of weight 1 each. Finally, set $k := (k_1 + 1) \cdot k_2 + k_1$ as the cardinality bound and $w := k_1((k_1+1)k_2+1) + (k_1+1)k_2$ as the target weight. The equivalence of the instances follows from the fact that the cardinality bound allows to pick in the solution at most k_2 bundles of $k_1 + 1$ copies of an edge of E_2 , while the weight bound allows to pick only k_1 edges of E_1 .

Thus, MULTI-BUDGETED CUT for $\ell = 2$ corresponds to the case of WEIGHTED *st*-CUT where the weights are integral and both target cardinality and weight are bounded in parameter.¹ This connection was our primary motivation to study the multi-budgeted variants of the cut problems.

Contrary to the classic minimum cut problem, we note that MULTI-BUDGETED CUT becomes NP-hard for $\ell \geq 2$. We show that MULTI-BUDGETED CUT is FPT when parameterized by $k = k_1 + \ldots + k_\ell$ and ℓ . For this problem, our branching strategy is as follows. A standard application of the Ford-Fulkerson algorithm gives a minimum XY-cut C of size λ and λ edgedisjoint X - Y paths $P_1, P_2, \ldots, P_{\lambda}$. If C is a solution, then we are done. Similarly, if $\lambda > k$, then there is no solution. Otherwise, we branch which colors of the sought solution should appear on each paths P_j ; that is, for every $i \in [\ell]$ and $j \in [\lambda]$, we guess if $P_j \cap E_i$ contains an edge of the sought solution, and in each guess assign infinite capacities to the edges of wrong color. If this change increased the size of a maximum flow from X to Y, then we can charge the branching step to this increase, as the size of the flow cannot exceed k. The critical insight is that if the size of the minimum flow does not increase (i.e., P_1, \ldots, P_{λ} remains a maximum flow), then a corresponding minimum cut is necessarily a solution. As a result, we obtain the following.

Theorem 2. MULTI-BUDGETED CUT admits an FPT algorithm with running time bound $\mathcal{O}(2^{k^2\ell} \cdot k \cdot (|V(G)| + |E(G)|))$ where $k = \sum_{i=1}^{\ell} k_i$.

The charging of the branching step to a flow increase appears also in the classic argument for bound of the number of important separators [23] (see also [32, Chapter 8]). This motivates us to define multi-budgeted variants of DIRECTED FEEDBACK ARC SET and SKEW MULTICUT. We observe that our branching algorithm can be merged with this procedure, yielding a bound (as a function of k and ℓ) and enumeration procedure of naturally defined multi-budgeted important separators. This in turn allows us to generalize our FPT algorithm to MULTI-BUDGETED SKEW MULTICUT and MULTI-BUDGETED DIRECTED FEEDBACK ARC SET.

Theorem 3. MULTI-BUDGETED SKEW MULTICUT and MULTI-BUDGETED DIRECTED FEED-BACK ARC SET admit FPT algorithms with running time bound $2^{\mathcal{O}(k^3\ell \log(k\ell))}(|V(G)| + |E(G)|)$ where $k = \sum_{i=1}^{\ell} k_i$.

The proof of Theorem 2 is covered in Section 4.1 and Theorem 3 is covered in Section 4.2.

1.3 Two Disjoint Shortest Paths Problem with transition restrictions

Finding disjoint paths with specified endpoints in a given graph is a well-known problem in graph theory and combinatorial optimization. Given a graph G = (V, E) and k vertex pairs

¹For a reduction in the other direction, replace every arc e of weight $\omega(e)$ with one copy of color 1 and $\omega(e)$ copies of color 2, and set budgets $k_1 = k$ and $k_2 = w$.

 $(s_1, t_1), ..., (s_k, t_k)$, the k DISJOINT PATHS PROBLEM (k-DPP) asks whether there exist k pairwise vertex-disjoint (or edge-disjoint) paths $P_1, ..., P_k$ such that P_i starts from s_i and ends at t_i for i = 1, ..., k. If G is a digraph, k-DPP is NP-hard even when k = 2 [53]. k-DPP is NP-complete if k is part of the input, even when G is a planar undirected graph [107]. Robertson and Seymour gave an $O(n^3)$ -time algorithm for k-DPP in general undirected graphs for every constant k [115]. Later Kawarabayashi et al. gave an $O(n^2)$ -time algorithm for the same problem [77]. Chudnovsky et al. showed that there is a polynomial time algorithm for k-VERTEX-DISJOINT PATHS PROBLEM for every fixed k if G is a semicomplete digraph [29]. Here a digraph is semicomplete if for all distinct vertices u, v, at least one of uv, vu is an edge.

Researchers also studied k-DPP from the view of parameterized complexity [33, 98, 116]. Cygan et al. gave an FPT algorithm parameterized by k with running time $2^{2^{O(k^2)}} \cdot n^{O(1)}$ for k-VERTEX-DISJOINT PATHS PROBLEM when G is a directed planar graph [33]. Given a tree decomposition of width at most w for the undirected graph G, k-DPP can be solved in time $2^{O(w \log w)}$ using dynamic programming techniques on tree decompositions [116], and Lokshtanov et al. showed that there is no $2^{o(w \log w)}$ time algorithm for k-DPP assuming ETH [98].

It is natural to generalize k-DPP to k-DSPP (k-DISJOINT SHORTEST PATHS PROBLEM) with an exceptional requirement that every disjoint path is also a shortest one. More formally, given a directed graph G = (V, E), a length function $w : E \to \mathbb{R}_{\geq 0}$ and k pairs of vertices $((s_1, t_1), ..., (s_k, t_k)$ in G, the k-DISJOINT SHORTEST PATHS PROBLEM asks to find k disjoint (vertex-disjoint or edge-disjoint) paths $P_1, ..., P_k$ in G such that P_i is a shortest path from s_i to t_i for i = 1, ..., k. Eilam-Tzoreff showed that 2-DSPP in an undirected graph is polynomial-time solvable [43]. Bérczi and Kobayashi showed that 2-DSPP is NP-hard in general directed graph but polynomial-time solvable when every directed cycle has positive length [9].

In routing problems on graphs, we sometimes need to express constraints on the permitted walks that are stronger than what the standard graph model allows for. For example, in a road network, there can be a crossroad where drivers are not allowed to turn left. In this case, many walks in the underlying graph would denote routes that a driver is not allowed to use. To overcome this limitation, Kotzig introduced forbidden-transition graphs in [83]. In a directed graph G, a transition is an ordered pair of adjacent edges such that the head of the first edge is the tail of the second edge. A transition system T is a set of transitions in G. We say that a path P is T-compatible if every two consecutive edges of P form a transition of T. For notational clarity, it is sometimes useful to refer to the transitions T(v) of a specific vertex $v \in V(G)$, that is, $T(v) = \{\{e_1, e_2\} \in T \mid head(e_1) = tail(e_2) = v\}$.

In this thesis we generalize the polynomial-time algorithm of Bérczi and Kobayashi to graphs with transition restrictions. Suppose that a prescribed transition system $T = \{T(v) \mid v \in V(G)\}$ is given, we study DIRECTED TWO DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) WITH TRANSITION RESTRICTIONS. The formal definition is as follows.

DIRECTED TWO DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) WITH TRANSITION RESTRICTIONS

Input: A directed graph G = (V, E) with transition system T, a length function $w : E \to \mathbb{R}_{\geq 0}$ and two pairs of vertices $(s_1, t_1), (s_2, t_2)$ in G.

Task: Find two disjoint (vertex-disjoint or edge-disjoint) paths P_1 and P_2 in G such that for both i = 1, 2, path P_i is a shortest path (even in the graph G with no transition restrictions) from s_i to t_i and P_i is also T-compatible.

We show that finding two vertex-disjoint (edge-disjoint) *T*-compatible paths P_1 and P_2 in a digraph *G* such that P_i is a shortest path (even in the graph *G* with no transition restrictions) from s_i to t_i for i = 1, 2 can be solved in polynomial time. Roughly speaking, we show that

transition restrictions are not a barrier for using the same strategy as that in [9]. Formally, we show the following theorem.

Theorem 4. If the length of every directed cycle is positive, both edge-disjoint and vertex-disjoint variants of 2-DSPP WITH TRANSITION RESTRICTIONS can be solved in polynomial time.

Corollary 1. If the length of every edge is positive, both edge-disjoint and vertex-disjoint variant of 2-DSPP WITH TRANSITION RESTRICTIONS can be solved in polynomial time.

The proof of Theorem 4 is covered in Chapter 5.

1.4 Cluster Editing parameterized above modification-disjoint P₃packings

CORRELATION CLUSTERING is a well-known problem motivated by research in computational biology [8] and machine learning [5]. This problem aims at partitioning data points into groups or clusters according to their similarity. In this thesis, we study this problem from the view of graph theory. A graph H is called a *cluster graph* if H is a union of vertex-disjoint cliques. Given a graph G = (V, E), the CLUSTER EDITING problem asks for a *cluster editing set* S such that $G \triangle S = (V, E \triangle S)$ is a cluster graph. Here $E \triangle S$ is the symmetric difference of E and S, i.e. $E \triangle S = (E \setminus S) \cup (S \setminus E)$. The optimization version of CLUSTER EDITING asks for a cluster editing set of minimum size, which is shown to be NP-hard [117]. Given a natural number k and a graph G = (V, E), the parameterized version of CLUSTER EDITING asks if there exists a cluster editing set S such that $|S| \leq k$. A number of results were obtained for the parameterized version of CLUSTER EDITING and some of its variants [11, 13, 14, 16, 38, 47, 59, 63, 64, 82, 113, 51]. The current fastest FPT algorithm runs in time $O(1.62^k + n + m)$ [11] and it admits a kernel of 2k vertices [18, 24].

The interest in CLUSTER EDITING is not merely theoretical. Indeed, parameterized techniques are implemented in standard clustering tools [110, 123]. Although practitioners report that in particular the parameterized data-reduction techniques are effective [13, 12], the parameter k is not very small in several real-world data sets [10, 13, 120]. For instance, Böcker et al. [10, Table 2] considered 26 graphs derived from biological data with 91 to 100 vertices on which the average number of needed edits is 315, despite noting that the CLUSTER EDITING model outperformed other clustering models.

A technique to deal with such large parameters is parameterization above lower bounds. Herein, the parameter is of the form $\ell = k - h$ where h is a lower bound on the solution size, usually computable in polynomial time, and ℓ is the excess of the solution size above the lower bound. Research into parameterizations above lower bounds has been active and fruitful for several parameterized problems, not only on the theory-side (see [102, 36, 58, 99, 85], for example) but also in practice, as algorithms based on that approach yielded quite efficient implementations for VERTEX COVER [4] and among the most efficient ones for FEEDBACK VERTEX SET [68, 79]. For CLUSTER EDITING we are aware of only one research work considering parameterizations above lower bounds: Van Bevern, Froese, and Komusiewicz [120] studied edge-modification problems parameterized above the lower bound from packings of forbidden induced subgraphs and showed that CLUSTER EDITING parameterized by the excess above the size of a given packing of vertex-disjoint P_{3} s is fixed-parameter tractable. Observe that a graph is a cluster graph if and only if it does not contain any P_{3} , a path on three vertices, as an induced subgraph. Consequently, one needs to perform at least one edge deletion or insertion per element of the packing. As the P_{3s} in the above packing are vertex-disjoint, the value by which the packing can decrease the parameter is limited. In the previous example with 315 edits, subtracting the resulting lower bound would reduce the parameter by at most 33. In their conclusion, van Bevern et al. [120] asked whether CLUSTER EDITING is fixed-parameter tractable when parameterized above a stronger lower bound, the size of a modification-disjoint packing of P_{3s} . Here, a packing \mathcal{H} of induced P_{3s} in G is modification-disjoint if every two P_{3s} in \mathcal{H} do not share edges or non-edges, that is, they share at most one vertex. The formal problem definition is as follows.

Cluster Editing above modification-disjoint P_3 packing (CEAMP)

Input: A graph G = (V, E), a packing \mathcal{H} of modification-disjoint induced P_3 s of G, and a non-negative integer ℓ .

Question: Is there a cluster editing set, i.e. a set of vertex pairs $S \subseteq {\binom{V}{2}}$ so that $G \triangle S$ is a union of disjoint cliques, with $|S| - |\mathcal{H}| \leq \ell$?

We also say that a set S as above is a *solution*.

At Shonan Meeting no. 144 [72] Christian Komusiewicz re-iterated the question of van Bevern et al. [120] and it was also asked in Vincent Froese's dissertation [54]. In this thesis, we answer this question negatively by showing the following.

Theorem 5. CLUSTER EDITING ABOVE MODIFICATION-DISJOINT P_3 PACKING is NP-hard even for $\ell = 0$ and when each vertex in the input graph is incident with at most 23 P_3s of \mathcal{H} .

In other words, given a graph G and a packing \mathcal{H} of modification-disjoint P_3 s in G, it is NP-hard to decide if one can delete or insert exactly one edge per element of \mathcal{H} to obtain a cluster graph.

Our NP-hardness result implies that CEAMP is probably not FPT or even in XP unless P = NP. This motivates us to study a more restrictive variant of CEAMP in which every vertex is incident with **at most 2** packed P_{3s} . Call a modification-disjoint P_{3} packing *two-restricted* if each vertex is in at most two packed P_{3s} . The problem CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_{3} PACKING (CEATMP) is defined in the same way as CEAMP except that the input packing \mathcal{H} is two-restricted.

Cluster Editing above two-restricted modification-disjoint P_3 packing (CEATMP)

Input: A graph G = (V, E), a packing \mathcal{H} of modification-disjoint induced P_3 s of G such that every vertex $v \in V(G)$ is incident with **at most 2** P_3 s of \mathcal{H} , and a nonnegative integer ℓ .

Question: Is there a cluster editing set, i.e. a set of vertex pairs $S \subseteq \binom{V}{2}$ so that $G \triangle S$ is a union of disjoint cliques, with $|S| - |\mathcal{H}| \leq \ell$?

It turns out that the complexity of the problem indeed drops when making the packing two-restricted.

Theorem 6. CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_3 PACK-ING can be solved in $O(n^{2\ell+O(1)})$ time.

The main ingredient for the XP algorithm is the following theorem.

Theorem 7. CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_3 PACK-ING can be solved in polynomial time when $\ell = 0$.

The proof of Theorem 5 is covered in Section 6.1 and the proofs of Theorem 6 and 7 are covered in Section 6.2.

1.5 Hardness of Metric Dimension in Graphs of Constant Treewidth

Let G be an unweighted and undirected graph and let $S \subseteq V(G)$. For a vertex $v \in V(G)$, the distance vector from v to S is the assignment $S \ni w \mapsto \text{dist}_G(v, w)$, where dist_G denotes the distance in the graph G. The set S is resolving if a distance vector to S uniquely determines the source vertex; that is, no two vertices of G have the same distance vector to S. The METRIC DIMENSION problem asks for a resolving set of minimum possible size; such a set is sometimes called the *metric basis* of G. The formal definition of the decision version of METRIC DIMENSION is as follows.

METRIC DIMENSION

Input: An undirected graph G and an integer k. **Question:** Is there a resolving set $S \subseteq V(G)$ such that $|S| \leq k$?

METRIC DIMENSION has already been introduced in 1970s [65, 118]. Determining its computational complexity turned out to be quite challenging. It is polynomial-time solvable on trees [65, 118, 78], outerplanar graphs [39], and chain graphs [48], but NP-hard for example on planar graphs [39] or split graphs [45]. From the parameterized complexity point of view, the FPT status of the METRIC DIMENSION parameterized by the solution size has been open for a while and finally resolved in negative by Hartung and Nichterlein [66]. In the search of a tractable structural parameterization, FPT algorithms has been shown for parameters: treelength plus maximum degree [7], vertex cover number [66], max leaf number [44], and modular-width [7].

The above list misses probably the most important graph width measure, namely treewidth. Determining the complexity of METRIC DIMENSION, parameterized by treewidth, remained elusive in the last years, and has been asked a few times [7, 39, 44]. Bonnet and Purohit in a paper presented at IPEC 2019 [17] showed that the problem is W[1]-hard, even with pathwidth parameterization. In this work we strengthed their result by proving para-NP-hardness of this parameterization.

Theorem 8. METRIC DIMENSION, restricted to graphs of treewidth at most 24, is NP-hard.

Theorem 8 brings us much closer to closing (unfortunately mostly in negative) the question of the complexity of METRIC DIMENSION in graphs of bounded treewidth. The remaining gap is to determine the exact treewidth value where the problem becomes hard: note that it is open if METRIC DIMENSION is polynomial-time solvable on graphs of treewidth 2.

The proof of Theorem 8 is covered in Chapter 7.

1.6 Articles

This thesis is based on the following articles and preprints:

- Chapter 3 is based on An improved FPT algorithm for Independent Feedback Vertex Set, which is a joint work with Marcin Pilipczuk, published at Theory Comput. Syst. 2020 [95]. The extended abstract of the publication was published in the 44th International Workshop on Graph-Theoretic Concepts in Computer Science, WG, 2018 [94].
- Chapter 4 is based on *Multi-budgeted Directed Cuts*, which is a joint work with Stefan Kratsch, Dániel Marx, Marcin Pilipczuk and Magnus Wahlström, published at Algorithmica, 2020 [87]. The extended abstract of the publication was published in 13th International Symposium on Parameterized and Exact Computation, IPEC, 2018 [86].
- Chapter 5 is based on Section 4 of *The Complexity of Connectivity Problems in Forbidden-Transition Graphs And Edge-Colored Graphs*, which is a joint work with Thomas Bellitto,

Karolina Okrasa, Marcin Pilipczuk and Manuel Sorge, published at 31st International Symposium on Algorithms and Computation, ISAAC, 2020 [6].

- Chapter 6 is based on *Cluster Editing Parameterized Above Modification-Disjoint P*₃-*Packings*, which is a joint work with Marcin Pilipczuk and Manuel Sorge, published at 38th International Symposium on Theoretical Aspects of Computer Science, STACS, 2021 [97].
- Chapter 7 is based on *Hardness of Metric Dimension in Graphs of Constant Treewidth*, which is a joint work with Marcin Pilipczuk, CoRR, 2021 [96].

The following are the papers I coauthored during PhD study but they are not included in this thesis.

- Many Visits TSP Revisited. This is a joint work with Lukasz Kowalik, Wojciech Nadara, Marcin Smulewicz and Magnus Wahlström, published at 28th Annual European Symposium on Algorithms, ESA, 2020 [84].
- An Improved FPT Algorithm for the Flip Distance Problem. This is a joint work with Qilong Feng, Xiangzhong Meng and Jianxin Wang [93], published at 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS, 2017. Its journal version is currently accepted to Information and Computation.

1.7 Acknowledgements

First of all, I would like to thank my supervisor, Marcin Pilipczuk, for all his support and encouragement in solving problems, in the whole PhD study and in my everyday life as a foreign student in Poland. I learnt a lot during the four years and I believe it will be a treasure in my future career and life.

I would like to thank all my coauthors Marcin Pilipczuk, Stefan Kratsch, Dániel Marx, Magnus Wahlström, Thomas Bellitto, Karolina Okrasa, Manuel Sorge, Lukasz Kowalik, Wojciech Nadara and Marcin Smulewicz for the fruitful discussion and inspiring ideas.

I would also like to thank Adam Karczmarz, Michal Pilipczuk, Krzysztof Fleszar, Karol Wegrzycki, Arkadiusz Socala, Pawel Komosa, Anna Zych-Pawlewicz, Jiehua Chen, Tomasz Kociumaka, Jana Novotná, Tomás Masarík, Lukasz Bozyk, Rahul C.S., Irene Muzi, Sebastian Siebertz, Maciej Matraszek and Michal Kutwin, who worked in the same department with me and gave me a lot of help or suggestions for my work and life in Warsaw.

Chapter 2

Preliminaries

2.1 Graph Notation

2.1.1 Basic Definitions

A simple undirected graph is a pair G = (V, E), where V is the set of vertices and $E \subseteq {V \choose 2}$ is the set of edges. ${V \choose 2} \setminus E$ is the set of non-edges. An undirected edge between two vertices u and v will be denoted by uv where we put uv = vu. An undirected non-edge between two vertices x and y will be denoted by xy, where we put xy = yx, and we will explicitly mention that xy is a non-edge in case of confusion with the notation of an edge. If uv is an edge in the graph, we say u and v are adjacent. We also use V(G) and E(G) to denote the vertex set and the edge set of G respectively. G is a simple graph if there are no loops or multiple edges (parallel edges). If we allow loops or multiple edges in G, then we call G a multigraph. We also use V(G) and E(G)to denote the vertex set and the edge multiset of G respectively.

We denote a bipartite graph by B = (U, W, E), where U, W are the two parts of the vertex set of B and E is the set of edges of B. We say that a bipartite graph is *complete* if for every pair of vertices $u \in U$ and $w \in W$, $uw \in E$.

A simple directed graph is a pair G = (V, E), where V is the set of vertices and E is the set of directed edges or arcs. If we allow loops or multiple edges (parallel edges) in G, then we call G a directed multigraph. We also use V(G) and E(G) to denote the vertex set and the directed edge multiset of G respectively. A directed edge e from u to v is denoted by uv, where u = tail(e) is called the *tail* of e and v = head(e) is called the *head* of e.

For a non-empty subset of vertices $X \subseteq V(G)$, we denote the *induced subgraph* of X by G[X]. For simplicity, we use $G \setminus X$ to denote $G[V(G) \setminus X]$. A *clique* Q in a graph G is a subgraph of G in which any two distinct vertices are adjacent.

2.1.2 Neighborhoods and degrees

For a vertex $v \in V(G)$, we use $N(v) = \{u \in V(G) : uv \in E(G)\}$ to denote the *neighborhood* of v. We define the *closed neighborhood* of v as $N[v] = N(v) \cup \{v\}$. In an undirected simple graph, the degree of a vertex v is $\deg(v) = |N(v)|$. For a vertex set $X \subseteq V(G)$, the neighborhood of X is $N(X) = \bigcup_{v \in X} N(v) \setminus X$. For a vertex set $X \subseteq V(G)$ and $v \in V(G)$, we define X-degree of v as the number of edges with one endpoint being v and the other lying in X, and we denote it by $\deg_X(v)$. Note that the X-degree counts edges with multiplicities if the corresponding graph is a multigraph. We say that two distinct vertices u, u' are *false twins* if N[u] = N[u'].

2.1.3 Paths and components

A path in G is a sequence $(v_1, e_1, v_2, e_2, \ldots, e_\ell, v_{\ell+1})$, where v_i s are distinct vertices of G, e_i s are edges of G, and for every $1 \leq i \leq \ell$, the vertices v_i and v_{i+1} are the two endpoints of the edge e_i . A cycle is a closed path in which no vertex occurs twice except the first and last vertex. For a path $P = (v_1, e_1, v_2, e_2, \ldots, e_\ell, v_{\ell+1})$, sometimes we denote it as $P = e_1, e_2, \ldots, e_\ell$. For two directed paths P and Q with consecutive edges $e_1^p, e_2^p, \ldots, e_{|P|}^p$ and, respectively, $e_1^q, e_2^q, \ldots, e_{|Q|}^q$ such that $w = \text{head}(e_{|P|}^p) = \text{tail}(e_1^q)$ is the only vertex that P and Q have in common, by $P \cdot Q$ we denote the concatenation of paths P and Q, i.e., $P \cdot Q = e_1^p, e_2^p, \ldots, e_{|P|}^p, e_1^q, e_2^q, \ldots, e_{|Q|}^q$. Here |P| and |Q| denote the number of edges of paths P and Q respectively. Similarly for two undirected paths P and Q which have only one vertex in common, we denote the concatenation of paths P and P_1, P_2, P_3 such that $P_1 \cdot P_2 \cdot P_3 = P$, then we say that P_1, P_2, P_3 are subpaths of P. Note that we allow P_1, P_2 or P_3 to be paths with one vertex.

In an undirected graph G, let G[X] be an induced subgraph of G. If for any two vertices $u, v \in X$, there is a path in G connecting u and v, then we say that G[X] is connected. We say that G[X] is a connected component of G if X is inclusion-wise maximal and G[X] is connected.

2.1.4 General cuts and budget-respecting cuts

Let X and Y be two disjoint vertex sets in a directed graph G, an XY-cut of G is a set of edges C such that every directed path from a vertex in X to a vertex in Y contains an edge of C. A cut C is minimal if no proper subset of C is an XY-cut, and minimum if C is of minimum possible cardinality. Let C be an XY-cut and let R be the set of vertices reachable from X in $G \setminus C$. We define $\delta^+(R) = \{(u, v) \in E(G) | u \in R \text{ and } v \notin R\}$ and note that if C is minimal, then $\delta^+(R) = C$. For two distinct (inclusion-wise) minimal XY-cuts C_1 , C_2 we say that C_1 is closer to Y than C_2 if every vertex reachable from X in $G - C_2$ is also reachable from X in $G - C_1$. A classic submodularity argument implies that there is exactly one minimal and any inclusion-wise minimal XY-cut C' which is closer to Y than C has strictly larger size than C or such a cut C' does not exist.

In our study for multi-budgeted problems, the directed graph G comes with sets $E_i \subseteq E(G)$ for $i \in [\ell]$ which we refer as *colors*. That is, an edge e is *of color* i if $e \in E_i$, and of *no color* if $e \in E(G) \setminus \bigcup_{i=1}^{\ell} E_i$. Note that an edge may have many colors, as we do not insist on the sets E_i being pairwise disjoint.

Let $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ be a MULTI-BUDGETED CUT instance and let C be an XY-cut. We say that C is budget-respecting if $C \subseteq \bigcup_{i=1}^{\ell} E_i$ and $|C \cap E_i| \leq k_i$ for every $i \in [\ell]$. For a set $Z \subseteq E(G)$ we say that C is Z-respecting if $C \subseteq Z$. In such contexts, we often call Z the set of deletable edges. An XY-cut C is a minimum Z-respecting cut if it is a Z-respecting XY-cut of minimum possible cardinality among all Z-respecting XY-cuts.

We will invoke the classic Ford-Fulkerson algorithm in our FPT algorithms for multi-budgeted problems. We encapsulate our use of the classic Ford-Fulkerson algorithm in the following statement.

Theorem 9. Given a directed graph G, two disjoint sets $X, Y \subseteq V(G)$, a set $Z \subseteq E(G)$, and an integer k, one can in $\mathcal{O}(k(|V(G)| + |E(G)|))$ time either find the following objects:

- λ paths $P_1, P_2, \ldots, P_{\lambda}$ such that every P_i starts in X and ends in Y, and every edge $e \in Z$ appears on at most one path P_i ;
- a set B ⊆ Z consisting of all edges of G that participate in some minimum Z-respecting XY-cut;

• a minimum Z-respecting XY-cut C of size λ that is closest to Y among all minimum Z-respecting XY-cuts;

or correctly conclude that there is no Z-respecting XY-cut of cardinality at most k.

Proof. Assing capacity 1 to every edge of Z and capacity $+\infty$ to every edge not in Z. Run k + 1 rounds of the Ford-Fulkerson algorithm. If the final flow exceeded k, return that there is no Z-respecting XY-cut of cardinality at most k. Otherwise, decompose the final flow into unit flow paths P_1, \ldots, P_{λ} in a standard manner. For the set B, observe that B consists of exactly those edges that are fully saturated in the flow network, and their reverse counterparts in the residual network are not contained in a single strongly connected component of the residual network (and thus can be discovered in linear time). Finally, observe that the sought cut C consists of the last edge of B on every path P_i .

2.1.5 Cluster graph and P_3 s

A clique Q in an undirected graph G is a subgraph of G in which any two distinct vertices are adjacent. A cluster graph is an undirected graph in which every connected component is a clique. A connected component in a cluster graph is called a *cluster*.

Let G' be a cluster graph and let S be a cluster editing set S, that is, a set $S \subseteq \binom{V}{2}$ of vertex pairs such that $G \triangle S = G'$. We say that two cliques Q_1 and Q_2 of G are *merged* (in G') if they belong to the same cluster in G'. We say that Q_1 and Q_2 are *separated* (in G') if they belong to two different clusters in G'. When mentioning the edges or non-edges between the vertices of the clique Q_1 and the vertices of the clique Q_2 , we refer to the edges or non-edges between the clique Q_1 and the clique Q_2 for short. Let $\ell, r \in \mathbb{N}$. We denote a path with ℓ vertices by P_{ℓ} and a cycle with r vertices by C_r .

Let x, y, z be vertices in a graph G. We say that xyz is an induced P_3 of G if $xy, yz \in E(G)$ and $xz \notin E(G)$. Vertex y is called the *center* of xyz. We say that vertices x, y, z belong to xyz or x, y, z are *incident with* xyz. We also say that xyz is *incident with* the vertices x, y and z. Here all P_{3s} we mention are induced P_{3s} ; we sometimes skip the qualifier "induced" for convenience.

Given an instance (G, \mathcal{H}, ℓ) of CEAMP, if xyz is a P_3 in G and $xyz \in \mathcal{H}$, we say that xyz is packed, and we say that the edges xy, yz are covered by xyz and the non-edge xz is covered by xyz. If an edge xy is covered by some P_3 of \mathcal{H} , we say that xy is a packed edge. Otherwise we say that xy is a non-packed edge. If a non-edge uv is covered by some P_3 of \mathcal{H} , we say that uv is a packed non-edge. Otherwise we say that uv is a non-packed non-edge. Otherwise we say that uv is a non-packed non-edge. If none of the edges of a path P is packed, we say that the path P is non-packed. If xyz is a P_3 in G and Q_1, Q_2 , and Q_3 are pair-wise non-intersecting vertex sets of G, we say that xyz connects Q_1 and Q_3 via Q_2 if the center y of xyz belongs to Q_2 and x, z belong to Q_1 and Q_3 , respectively.

We will use finite fields of prime order in the NP-hardness proof of CLUSTER EDITING ABOVE MODIFICATION-DISJOINT P_3 PACKING. Let p be some prime. By \mathbb{F}_p we denote the finite field with the p elements $0, \ldots, p-1$ with addition and multiplication modulo p. Let $x \in \mathbb{F}_p$. Where it is not ambiguous, -x and x^{-1} will denote the additive and multiplicative inverse, respectively, of x in \mathbb{F}_p .

2.2 Treewidth and Pathwidth

For an undirected graph G, a path decomposition of G is a sequence $(X_1, ..., X_t)$ of subsets of V(G), i.e. $X_i \subset V(G)$ for i = 1, ..., t, such that the following holds:

- (i) for every edge $uv \in E(G)$, there exists an integer $i \in [t]$ such that $u, v \in X_i$.
- (ii) for every $v \in V(G)$, if $v \in X_i$ and $v \in X_k$ $(i \le k)$, then $v \in X_j$ for every $i \le j \le k$.

The width of a path decomposition $(X_1, ..., X_t)$ is defined as $\max_{1 \le i \le t} |X_i| - 1$.

For an undirected graph G, a *tree decomposition* of G is a pair (\mathcal{T}, β) where \mathcal{T} is a tree and $\beta : V(\mathcal{T}) \to 2^{V(G)}$ such that the following holds:

- (i) for every $v \in V(G)$, the set $\{t \in V(\mathcal{T}) \mid v \in \beta(t)\}$ induces a nonempty connected subtree of \mathcal{T} , and
- (ii) for every $uv \in E(G)$, there exists $t \in V(\mathcal{T})$ with $u, v \in \beta(t)$. That is, the function β assigns to every node $t \in V(\mathcal{T})$ a subset $\beta(t) \subseteq V(G)$, often called a *bag*.

It is often convenient to root \mathcal{T} at an arbitrary vertex. The width of a tree decomposition (\mathcal{T}, β) equals $\max_{t \in V(\mathcal{T})} |\beta(t)| - 1$, and the treewidth of a graph is the minimum possible width of its tree decomposition.

In this thesis, we will use the alternative characterization of pathwidth, i.e. the pathwidth of a graph G equals the node search number of G minus one [80]. The node search number is defined as follows. We can imagine that the edges of a graph G are tunnels which are contaminated by a poisonous gas and we need to use some searchers to clean the gas. An edge is cleaned when its two endpoints are occupied by searchers at the same time. However, a cleaned edge will be recontaminated by the gas immediately if we remove a searcher guarding at one of its endpoints and there is a contaminated path which allows the gas to come into this edge through the nonguarded endpoint. We can put a searcher on any non-guarded vertex of the graph or remove a searcher from a vertex of the graph (the removed searchers can be put on other non-guarded vertices of the graph). The node search number is the minimum number of searchers required to clean the whole graph G.

2.3 Other notations

We use \mathbb{N} to denote the set of nonnegative integers. For an integer n, we denote $[n] = \{1, 2, \ldots, n\}$. The \mathcal{O}^* -notation suppresses factors that are polynomial in the input size.

Chapter 3

Independent Feedback Vertex Set

In this chapter, we present an FPT algorithm running in time $\mathcal{O}^*(3.619^k)$ for INDEPENDENT FEEDBACK VERTEX SET problem. Recall that INDEPENDENT FEEDBACK VERTEX SET problem is a variant of the classic FEEDBACK VERTEX SET problem where, given a graph G and an integer k, the task is to decide whether there exists a vertex set $S \subseteq V(G)$ such that $G \setminus S$ is a forest and S is an independent set of size at most k.

Organization of this chapter. In Section 3.1, we present the top-level procedure of the algorithm and introduce the problem DISJOINT INDEPENDENT FEEDBACK VERTEX SET, which is the main subproblem to be solved by the algorithm. In Section 3.2, we give the reduction rules for DISJOINT INDEPENDENT FEEDBACK VERTEX SET. In Section 3.3, we give the branching rules for DISJOINT INDEPENDENT FEEDBACK VERTEX SET, thus completing the algorithm for DISJOINT INDEPENDENT FEEDBACK VERTEX SET.

3.1 Disjoint Independent Feedback Vertex Set

Given an instance (G, k), we first invoke the $\mathcal{O}^*((1 + \varphi^2)^k)$ -time FPT algorithm for the classic FEEDBACK VERTEX SET problem [81]. If the algorithm returns NO, we conclude that there is no independent feedback vertex set of size at most k since an independent feedback vertex set is also a feedback vertex set. Otherwise, the algorithm returns a feedback vertex set Z such that $|Z| \leq k$. Obviously, $F = G \setminus Z$ is a forest.

Suppose there is a solution S for the input instance (G, k). The algorithm branches into $2^{|Z|}$ directions, guessing a subset Z' of Z such that $S \cap Z = Z'$. Let $W = Z \setminus Z'$. If G[Z'] is not an independent set or G[W] is not a forest, the algorithm rejects this guess. Hence, we can assume that G[Z'] is an independent set and $G[Z \setminus Z']$ is a forest. Let $R = N(Z') \cap F$. Since the solution S is an independent set and $Z' \subseteq S$, we have $R \cap S = \emptyset$. Then the algorithm tries to find an independent feedback vertex set $S' \subseteq F$ for $G \setminus Z'$ such that $S' \cap R = \emptyset$ and $|S'| \leq k - |Z'|$. Following [3], we call this subproblem DISJOINT INDEPENDENT FEEDBACK VERTEX SET (DIS-IFVS for short). We give a faster FPT algorithm for DIS-IFVS in the next section. The algorithm finds a YES instance of DIS-IFVS, then it returns YES for the instance (G, k) of IFVS. Otherwise, if the algorithm tries every possible $Z' \subseteq Z$ and obtains a NO answer for every corresponding instance of DIS-IFVS, it reports that (G, k) is a NO instance.

We give the formal definition of DISJOINT INDEPENDENT FEEDBACK VERTEX SET as follows.

DISJOINT INDEPENDENT FEEDBACK VERTEX SET

Input: An undirected (multi)graph G, a feedback vertex set W of G, $R \subseteq V(G) \setminus W$, and an integer k.

Question: Is there an independent feedback vertex set $X \subseteq V(G) \setminus (W \cup R)$ for G such that $|X| \leq k$?

Let $F = V(G \setminus W)$. Obviously, G[F] is a forest since W is a feedback vertex set of G. A vertex $v \in F \setminus R$ is a *nice vertex* if $\deg_W(v) = 2$ and v has no neighbors in F. A vertex $v \in F \setminus R$ is a *tent* if $\deg_W(v) = 3$ and v has no neighbors in F.

As mentioned earlier, we rely on a measure different from the one in [3]. The measure μ of an instance (G, W, R, k) is defined as

$$\mu = k + \rho - (\eta + \tau).$$

Here, ρ represents the number of connected components of G[W], η is the number of nice vertices in $F \setminus R$ and τ is the number of tents in $F \setminus R$.

We remark that the distinction between sets W and R is purely for the sake of complexity of the algorithm. The set of feasible solutions to a DISJOINT INDEPENDENT FEEDBACK VERTEX SET instance (G, W, R, k) would be the same if we move vertices from R to W. However, the notions of tents, nice vertices, and the measure μ strongly depends on the distinction between the sets W and R. The algorithm maintains this distinction to ensure the promised running time bound.

Our main technical result is the following.

Lemma 1. A DISJOINT INDEPENDENT FEEDBACK VERTEX SET instance I with measure μ can be solved in time $\mathcal{O}^*(\varphi^{\mu})$, where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

Theorem 1 follows by standard analysis as in [3]:

Proof of Theorem 1. The algorithm for FVS of [81] runs in time $\mathcal{O}^*((1 + \varphi^2)^k)$. In a branch with a set $Z' \subseteq Z$ the routine for DIS-IFVS is passed an instance with both $W = Z \setminus Z'$ and the parameter bounded by k - |Z'|, and hence with measure bounded by 2(k - |Z'|). Since the algorithm for DIS-IFVS runs in time $O^*(\varphi^{\mu})$, the total running time of its applications is bounded by

$$\sum_{i=0}^{k} \binom{k}{i} \mathcal{O}^{*}(\varphi^{2(k-i)}) = \mathcal{O}^{*}((1+\varphi^{2})^{k}) \le \mathcal{O}^{*}(3.619^{k}).$$

This completes the proof.

The remainder of this section is devoted to the proof of Lemma 1. We start with showing that μ is nonnegative on YES instances.

Lemma 2. Let I = (G, W, R, k) be a YES instance of DISJOINT FEEDBACK VERTEX SET. Then $\mu \ge 0$.

Proof. Let X be a solution to the instance I. Thus $G' = G \setminus X$ is a forest. Let $N \subseteq V(G) \setminus (W \cup R)$ be the set of nice vertices and $T \subseteq V(G) \setminus (W \cup R)$ be the set of tents. Since $X \cap W = \emptyset$, we have that $H := G[W \cup (N \setminus X) \cup (T \setminus X)]$ is a forest. Now we contract each component in H[W]into a single vertex and get a forest \tilde{H} . Since there are at most $\rho + |N \setminus X| + |T \setminus X|$ vertices in \tilde{H} , there are at most $\rho + |N \setminus X| + |T \setminus X| - 1$ edges in \tilde{H} . According to the definition of tents

and nice vertices, $(N \cup T) \setminus X$ is an independent set. Moreover, since the degree of any vertex in $N \setminus X$ and $T \setminus X$ is 2 and 3, respectively, we get the following inequality:

 $2|N\setminus X|+3|T\setminus X|\leq |E(\tilde{H})|\leq \rho+|N\setminus X|+|T\setminus X|-1.$

It follows that:

$$|N \setminus X| + |T \setminus X| \le |N \setminus X| + 2|T \setminus X| \le \rho.$$

Hence, as $|X| \leq k$,

$$|N| + |T| \le \rho + k.$$

As a result, $\mu = \rho + k - (\eta + \tau) \ge 0$.

A small comment is in place. Our measure μ is different from the one of [3]: $\mu' = 2k + \rho - (\eta + 2\tau)$. The change in the measure is one of the critical insights in this result: while it sometimes leads to weaker branching vectors as compared to [3], the "starting value" in an application in the above proof of Theorem 1 is 2(k - |Z'|), not 3(k - |Z'|) as in [3]. Thus, to obtain the promised running time bound, we are fine with branching vectors of the form (1, 2); that is, we are fine with branching steps in two directions, where in one direction the measure drops by at least one, and in the other direction by at least two. The change in the measure is similar to the one that happened in the work of Kociumaka and Pilipczuk for FEEDBACK VERTEX SET [81], as compared to a previous champion of Cao, Chen, and Liu [19].

We introduce now some definitions that will help us streamline later arguments. Let (G, W, R, k) be an instance of DIS-IFVS and let $F = V(G) \setminus W$. We say that $u \in F \setminus R$ is a *potential nice* vertex or *P*-nice if u is of degree 2 and exactly one of its incident edges has a second endpoint in W. For a vertex v in G[F], we define the nice degree of v, denoted by Ndeg(v), as the number of P-nice neighbors of v. A generalized degree of v is $Gdeg_W(v) = Ndeg(v) + deg_W(v)$. We say that $u \in F \setminus R$ is a potential tent or *P*-tent if $Gdeg_W(u) = 2$ and deg(u) = 3. For a vertex v in F, we define the tent degree of v, denoted by Tdeg(v), as the number of neighbors of v that are P-tents.

3.2 Reduction Rules for DIS-IFVS

Now we introduce some reduction rules for DIS-IFVS. We always apply the applicable reduction rule of the lowest number. First, let us introduce five reduction rules from [3].

Reduction Rule 1: Delete any vertex of degree at most one.

Reduction Rule 2: Let u, v be two adjacent vertices of degree two in $G \setminus W$ which are not nice vertices in F. Besides, u is adjacent to x while v is adjacent to y (x and y could be the same vertex). If neither u nor v is in R or both are in R, then delete one vertex in $\{u, v\}$ arbitrarily and connect the neighbors of the deleted vertex with a new edge. If exactly one of u and v is in R, say $v \in R$, then delete v and add an edge between its neighbors (i.e., an edge uy).

Reduction Rule 3: If k < 0 or $\mu < 0$, return that the input instance is a NO instance.

Reduction Rule 4: If there is a vertex $v \in R$ such that v has two incident edges with the second endpoints in the same component of W, then return that the input instance is a NO instance.

Reduction Rule 5: If there is a vertex $v \in F \setminus R$ such that v has at least two incident edges with the second endpoints in the same component of W, then remove v from G and add all

vertices in $F \cap N(v)$ to R. In this case, k decreases by one.

It is not difficult to verify the safeness of Reduction Rules 1-5 as shown in [3]. But when analyzing Reduction Rules 1 and 5, we need to be careful since we use a different measure $\mu = k + \rho - (\eta + \tau)$. In Reduction Rule 1, if one deletes a neighbor w of a tent or a nice vertex v, then v stops being a tent or a nice vertex $(\eta + \tau \text{ could decrease by one})$, but also $\{w\}$ stops being a connected component of G[W] (decreasing ρ by one). For Reduction Rule 5, it may happen that v is a tent or a nice vertex, and its deletion decreases $\eta + \tau$ by one. However, the removal of v also decreases k by one. Thus μ does not increase.

Now we introduce two new reduction rules.

Reduction Rule 6: If there is a vertex $v \in R$ such that $Gdeg_W(v) \ge 1$ or $Tdeg(v) \ge 1$, then remove v from R and add v to W.



Figure 3.1: Reduction Rule 6

Reduction Rule 7: If there is a vertex $v \in F \setminus R$ such that every neighbor $w \in N(v) \setminus (W \cup R)$ is of degree 2, and at least one such neighbor exists, then put $N(v) \setminus (W \cup R)$ into R.



Figure 3.2: Reduction Rule 7

We first show their safeness.

Claim 1. Reduction Rules 6 and 7 are safe.

Proof. The safeness of Reduction Rule 6 is straightforward. For the safeness of Reduction Rule 7, suppose that (G, W, R, k) is an input instance. Let v be the vertex satisfying the condition of Reduction Rule 7 and $(G, W, R \cup (N(v) \cap F), k)$ be the instance obtained after applying Reduction Rule 7. We claim that (G, W, R, k) is a YES instance if and only if $(G, W, R \cup (N(v) \cap F), k)$ is a YES instance. The "if" direction is straightforward, since we only increased the set R.

For the "only if" direction, let X be a solution of size at most k to the instance (G, W, R, k). If $X \cap N(v) = \emptyset$, then X is also a solution to $(G, W, R \cup (N(v) \cap F), k)$. Otherwise, we construct a vertex set $X' = (X \cup \{v\}) \setminus (N(v) \cap F)$. Obviously $|X'| \leq k$. We will show that X' is a solution to $(G, W, R \cup (N(v) \cap F), k)$. Clearly, it is disjoint with $W \cup R \cup N((v) \cap F)$ and independent, as it is disjoint with N(v). To show that X' is a feedback vertex set in G, observe that since every vertex $w \in N(v) \setminus (W \cup R)$ is of degree 2, every cycle passing through w in G passes also through v.

Since Reduction Rule 7 only moves vertices to R, its application does not change the measure; note that the neighbors of a vertex affected by Reduction Rule 7 can be neither a nice vertex nor a tent. However, the situation is not that easy for Reduction Rule 6, and we need to show that its application does not increase μ . To this end, we show a number of generic observations on how the measure μ changes if we modify a neighbor of a P-nice vertex or a P-tent.

Observation 1. Let $v \in F$ be a vertex with a P-nice neighbor w. Consider the operation of moving v to W. Then, the vertex w becomes nice and η goes up at least by one.

Observation 2. Let $v \in F$ be a vertex with a P-tent neighbor w such that v is not P-nice. Consider the operation of putting v in a solution: deleting it from G and putting $N(v) \cap F$ into R. Then the application of reduction rules on w and its (possible) other neighbors in F decreases μ by at least one.

Proof. The operation moves w to R and decreases its degree to 2. Since w is a P-tent and v is not a P-nice vertex, every neighbor $u \in (N(w) \cap F) \setminus \{v\}$ is a P-nice vertex. Consequently, Reduction Rule 2 reduces $(N[w] \cap F) \setminus \{v\}$ to a single vertex w', which is in R if $(N(w) \cap F) \setminus \{v\} \subseteq$ R. Furthermore, $\deg(w') = \deg_W(w') = 2$. If w' has both neighbors in the same connected component of G[W], then either Reduction Rule 4 rejects the instance or Reduction Rule 5 decreases k by one. Otherwise, if $w' \in R$, Reduction Rule 6 moves w' to W, decreasing ρ by one. If $w' \notin R$, then w' becomes a nice vertex, increasing η by one. Thus, in all cases, μ decreases by at least one.

Observation 3. Let $v \in F$ be a vertex with a P-tent neighbor w such that v is not P-nice. Consider the operation of moving v into W. Then the application of reduction rules on w and its (possible) other neighbors in F decrease μ by at least one.

Proof. Since w is a P-tent and v is not P-nice, every neighbor $u \in (N(w) \cap F) \setminus \{v\}$ is P-nice. Consider such a vertex u; note that $u \in F \setminus R$ by the definition of P-nice. Reduction Rule 7 is applicable to w; this rule would move u to R and then Reduction Rule 6 would move u to W. Along this process, Reduction Rule 4 or 5 can be triggered on w, either rejecting the instance or decreasing k by one. Otherwise, if $w \in R$, Reduction Rule 6 moves w to W, decreasing ρ by two. Finally, in the last case we are left with $w \in F \setminus R$ with $\deg_W(w) = \deg(w) = 3$, that is, w becomes a tent and increases τ by one. Thus, in all cases, μ decreases by at least one.

Armed with the above observations (see Fig. 3.3), we can now show that Reduction Rule 6 on its own does not increase the measure.

Claim 2. An application of Reduction Rule 6 does not increase the measure.



Figure 3.3: Observation 1-3

Proof. If v is a tent or a nice vertex, then η or τ decreases by one but ρ decreases by at least one because Reduction Rule 4 or 5 is not applicable. In this case, μ does not increase. If v is neither a tent nor a nice vertex and $\deg_W(v) \ge 1$, ρ does not increase, and η and τ do not decrease. In this case, μ does not increase.

We are left with the case $\deg_W(v) = 0$, and then ρ increases by one. If $\operatorname{Gdeg}_W(v) \ge 1$ but $\deg_W(v) = 0$, we have a P-nice neighbor w of v. Then, after v is moved to W, Observation 1 asserts that future application of reduction rules on w cause a measure decrease of at least one, offsetting the increase of ρ . Otherwise, $\operatorname{Tdeg}(v) \ge 1$, and we have a neighbor w of v that is a P-tent. Then, after v is moved to W, Observation 3 asserts that future application of reduction rules on w and its possible neighbors in F cause measure decrease of at least one. This finishes the proof.

3.3 Branching for DIS-IFVS

Now we are ready to introduce the branching algorithm. We assume that all reduction rules have been applied exhaustively. As a branching pivot, we pick a vertex $v \in F$ that is neither a nice vertex, nor a tent, nor a *P*-nice vertex, and satisfies one of the following three cases:

Case A: $\operatorname{Gdeg}_W(v) \ge 3$. Case B: $\operatorname{Gdeg}_W(v) \ge 1$ and $\operatorname{Tdeg}(v) \ge 1$. Case C: $\operatorname{Tdeg}(v) \ge 2$.

In case of more than one vertices of F satisfying one of the above cases, we prefer to pick a vertex v that satisfies an earlier case.

First, note that the non-applicability of Reduction Rule 6 implies that the chosen branching pivot v does not lie in R.

No matter which case the chosen branching pivot v satisfies, we branch into two cases. In one case we include v into the solution: we delete v from the graph, include $N(v) \cap F$ into R, and decrease k by one. In the other case, we move v to W.

We now show that in each of the cases, the branching gives a branching vector (1, 2) or better with respect to the measure μ . That is, in one of the branches the measure drops by at least one, and in the other by at least two.

Case A: $Gdeg_W(v) \ge 3$.

- (i) Branch where v is deleted and all vertices in N(v) ∩ F are added to R. k decreases by 1, *ρ* stays the same, and η and ρ does not decrease as v is neither a nice vertex nor a tent. Thus, μ decreases by at least one.
- (ii) Branch where v is moved from F to W. ρ decreases by $\deg_W(v) 1$ (which may be -1 if $\deg_W(v) = 0$) and η increases by $\operatorname{Ndeg}(v)$. Since $\operatorname{Gdeg}_W(v) = \deg_W(v) + \operatorname{Ndeg}(v) \geq 3$ and τ does not decrease, μ decreases by at least two.

Case B: $Gdeg_W(v) \ge 1$ and $Tdeg(v) \ge 1$.

- (i) Branch where v is deleted and all vertices in $N(v) \cap F$ are added to R. First, k decreases by one. Furthermore, v has a P-tent neighbor w and Observation 2 asserts that future applications of reduction rules on w and its remaining neighbors in F decrease the measure by at least one. Thus, in total μ decreases by at least two.
- (ii) Branch where v is moved from F to W. For every P-tent neighbor w of v, Observation 3 asserts that the application of reduction rules to w and its remaining neighbors in F cause a measure decrease of at least 1. If $\deg_W(v) \ge 1$, then moving v to W does not increase ρ , and we are done. Otherwise, if $\deg_W(v) = 0$, moving v to W increases ρ by 1 but the assumption $\operatorname{Gdeg}_W(v) \ge 1$ implies that there also exists a P-nice neighbor w of v. For every such P-nice neighbor w of v, Observation 1 asserts that the future application of reduction rules on w and its remaining neighbors in F cause measure drop by at least 1. Consequently, in this case we also have a measure drop of at least 1.

Case C: $Tdeg(v) \ge 2$.

- (i) Branch where v is deleted and all vertices in $N(v) \cap F$ are added to R. First, k decreases by one. Furthermore, for every P-tent neighbor w of v, Observation 2 asserts that the application of reduction rules on w and its remaining neighbors in F cause measure drop by at least one. Since $\text{Tdeg}(v) \geq 2$, together with the decrease of k we have a total measure decrease of at least 3.
- (ii) Branch where v is moved from F to W. The move itself may increase ρ by one. For every P-tent neighbor w of v, Observation 3 asserts that the future application of reduction rules on w and its remaining neighbors in F cause measure drop by at least 1. Since $T \deg(v) \geq 2$, in total we have a measure decrease by at least 1.

We are left with analyzing what happens if no vertex of F satisfies any of the three cases for the choice of the branching pivot. As in [3], we rely on the following base case.

Lemma 3 ([3]). Let (G, W, R, k) be an instance of DIS-IFVS where every vertex in $V(G) \setminus W$ is either a nice vertex or a tent. Then we can find an independent feedback vertex set $X \subseteq V(G) \setminus (W \cup R)$ for G of the minimum size in polynomial time.

Lemma 3 follows from the observation by Cao et al. [19] and the fact that all nice vertices and tents form an independent set.

We show the following.

Lemma 4. If no reduction rule can be applied and every vertex of F does not satisfy any of the cases for the choice of the branching pivot, then the remaining instance of DIS-IFVS can be solved in polynomial time.

Proof. We claim that every vertex in F of the remaining graph G is either a tent or a nice vertex; the claim then follows by Lemma 3.

For contradiction, suppose that there is a connected component D of G[F] that is not a singleton with a tent or a nice vertex. Since no vertex of D falls into Case A, $\operatorname{Gdeg}_W(v) \leq 2$ for every $v \in D$; in particular, every leaf (a vertex in F that has only exactly one neighbor in F) $v \in D$ satisfies $\operatorname{deg}_W(v) \in \{1, 2\}$. Root the tree G[D] at an arbitrary vertex, and consider a leaf $v \in D$ that is furthest from the root in G[D] and, among such leaves, choose one maximizing $\operatorname{deg}_W(v)$. Note that $v \notin R$ as otherwise Reduction Rule 6 would move v to W.

First, assume $\deg_W(v) = 2$. Since v is a leaf of D and is not nice, v has exactly one neighbor $u \in D$, and v is a P-tent. Hence, $\operatorname{Tdeg}(u) \ge 1$. If $\deg(u) = 2$, then Reduction Rule 7 applies to v if $u \notin R$ and once u is in R, then Reduction Rule 6 applies to u, making v a tent. Consequently, $\deg(u) \ge 3$. However, by the choice of v, $\deg_W(u) \ge 1$ or u is adjacent to another leaf v' of D. However, this implies that $\operatorname{Gdeg}_W(u) \ge 1$ (if $\deg_W(u) \ge 1$ or v' exists and $\deg_W(v') = 1$) or $\operatorname{Tdeg}(u) \ge 2$ (if v' exists and $\deg_W(v') = 2$), and Case B or C applies to u.

Second, assume $\deg_W(v) = 1$, and again let u be the unique neighbor of v in G[D]. If $\deg(u) = 2$, then Reduction Rule 2 is applicable. By the choice of v, every other leaf v' adjacent to u also satisfies $\deg_W(v') = 1$; that is, every child of u is P-nice as $u \notin R$. If $\operatorname{Gdeg}_W(u) \ge 3$, then Case A applies to u. Hence, $\deg(u) = 3$ and $\operatorname{Gdeg}_W(u) = 2$: u has a parent x in G[D] and either one more child v' that is P-nice or a neighbor in W. In particular, u is a P-tent, and $\operatorname{Tdeg}(x) \ge 1$.

If $\deg(x) = 2$, then Reduction Rule 7 would apply to u and move v to R, and consequently Reduction Rule 6 would move v to W. If $\operatorname{Gdeg}_W(x) \ge 1$, then Case B applies to x. Hence, x has another child u' that is not P-nice. By the choice of v, the connected component of $G[D] \setminus \{x\}$ containing u' is a star with u' as a center. Furthermore, every child w of u' is P-nice (i.e., $\deg_W(w) = 1$). Since Case A is not applicable to u', we have $\operatorname{Gdeg}_W(u') \le 2$. If $\deg(u') = 2$, then either u' is P-nice (if $\deg_W(u') = 1$) or Reduction Rule 2 is applicable to u' and its child (if $\deg_W(u') = 0$). We infer that $\deg(u') = 3$ and $\operatorname{Gdeg}_W(u') = 2$; in particular, u' is a P-tent. Hence, $\operatorname{Tdeg}(x) \ge 2$ and case C applies to x. This completes the proof of the lemma.

Every step of the reduction rules and branching can be executed in polynomial time. In every case of branching, the branching vector is (1,2). Thus we get the following recurrence: $T(\mu) = T(\mu - 1) + T(\mu - 2)$. As a result, the running time of the algorithm for DIS-IFVS is $O^*(\varphi^{2k})$. This concludes the proof of Lemma 1 and thus of the whole Theorem 1.

Chapter 4

Multi-budgeted Directed Cuts

In this chapter, we give an FPT algorithm parameterized by $k = \sum_{i=1}^{\ell} k_i$ and ℓ for the MULTI-BUDGETED CUT problem and show some of its applications. Recall that the MULTI-BUDGETED CUT problem is a generalization of the classic MINIMUM CUT problem. In MULTI-BUDGETED CUT problem, we assign colors $1, 2, ..., \ell$ to some edges and give separate budgets $k_1, k_2, ..., k_{\ell}$ for colors $1, 2, ..., \ell$. For every color $i \in \{1, ..., \ell\}$, let E_i be the set of edges of color i. The solution C for MULTI-BUDGETED CUT not only needs to be a cut satisfying the usual separation requirements, but also needs to satisfy that $|C \cap E_i| \leq k_i$ for every $i \in \{1, ..., \ell\}$. Contrary to the classic minimum cut problem, the MULTI-BUDGETED CUT problem is NP-hard even for $\ell = 2$. The DIRECTED FEEDBACK ARC SET problem is a classic problem that played major role in the development of parameterized complexity. In this problem, given a directed graph G and an integer k, the problem is to decide if there exists an arc set S of size at most k such that G - S has no directed cycles. In a similar way we define the problem MULTI-BUDGETED DIRECTED FEEDBACK ARC SET as follows.

MULTI-BUDGETED DIRECTED FEEDBACK ARC SET **Input:** A directed graph G, an integer ℓ , and for every $i \in \{1, 2, ..., \ell\}$ a set $E_i \subseteq E(G)$ and an integer k_i .

Question: Is there a set of arcs $S \subseteq \bigcup_{i=1}^{\ell} E_i$ such that there is no directed cycle in G - S and for every $i \in [\ell], |S \cap E_i| \leq k_i$.

The first FPT algorithm for the DIRECTED FEEDBACK ARC SET problem is given by Chen et al. [23]. In their algorithm, they use iterative compression and reduce the DIRECTED FEEDBACK ARC SET compression problem to the SKEW EDGE MULTICUT problem. They propose a pushing lemma for SKEW EDGE MULTICUT and solve SKEW EDGE MULTICUT through enumerating important cuts. We show that for the multi-budgeted variant, a similar strategy enumerating multi-budgeted important cuts works. Formally, the MULTI-BUDGETED SKEW EDGE MULTICUT problem is defined as follows.

Multi-budgeted Skew Edge Multicut

Input: A directed graph G, an integer ℓ , for every $i \in \{1, 2, ..., \ell\}$ a set $E_i \subseteq E(G)$ and an integer k_i , and a sequence $(s_i, t_i)_{i=1}^q$ of terminal pairs.

Question: Is there a set of arcs $C \subseteq \bigcup_{i=1}^{\ell} E_i$ such that there is no directed path from s_i to t_j for any $i \ge j$ in G - C and for every $i \in [\ell], |C \cap E(i)| \le k_i$?

Organization of this chapter. In Section 4.1, we show the NP-hardness of MULTI-BUDGETED CUT problem when $\ell \geq 2$ and give an FPT algorithm for the MULTI-BUDGETED CUT problem parameterized by $k = \sum_{i=1}^{\ell} k_i$ and ℓ . In Section 4.2, we give FPT algorithms for the multi-budgeted versions of SKEW EDGE MULTICUT and FEEDBACK ARC SET parameterized by $k = \sum_{i=1}^{\ell} k_i$ and ℓ .

4.1 Multi-budgeted cut

4.1.1 NP-hardness of MULTI-BUDGETED CUT

Although it is well-known that the minimum cut problem is polynomial-time solvable, we prove that the MULTI-BUDGETED CUT problem is NP-hard for $\ell \geq 2$.

Lemma 5. MULTI-BUDGETED CUT problem is NP-hard for every $\ell \geq 2$.

Proof. We prove this lemma by making a reduction from constrained minimum vertex cover problem on bipartite graphs (MIN-CBVC), which is proved to be NP-hard by Chen and Kanj [22]. In the MIN-CBVC problem the input consists of a bipartite graph $G = (U \uplus L, E)$ and integers k_U, k_L ; the goal is to find a vertex cover $X \subseteq U \cup L$ such that $|X \cap U| \leq k_U$ and $|X \cap L| \leq k_L$.

Given an instance (G, k_U, k_L) of MIN-CBVC, where $G = (U \cup L, E)$ is a bipartite graph, we construct an instance $(G', X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ of MULTI-BUDGETED CUT as follows. We take $V(G') = V(G) \cup \{s, t\}$, where s and t are two new vertices, and set $X = \{s\}$ and $Y = \{t\}$. Then for each vertex $u \in U$, we add an arc (s, u) with color 1 to G' and for each vertex $v \in L$, we add an arc (v, t) with color 2 to G'. For each undirected edge $(u, v) \in E(G)$ such that $u \in U$ and $v \in L$, we add an arc (u, v) with no color. Let E_1 be the set of arcs of color 1 in G' and E_2 be the set of arcs of color 2 in G'. Let $E_i = \emptyset$ for $i = 3, ..., \ell$. Let $Z = E_1 \cup E_2$ be the deletable arcs. Let the budgets of the MULTI-BUDGETED CUT instance be $k_1 = k_U, k_2 = k_L, k_3 = 0, ..., k_\ell = 0$. This completes the construction.

Now we show that (G, k_U, k_L) is a yes instance if and only if $(G', X, Y, k_1, k_2, ..., k_\ell)$ is a yes instance. Suppose (G, k_U, k_L) is a yes instance. Then there exists a vertex cover $U' \cup L'$ of G such that $U' \subseteq U$, $L' \subseteq L$, $|U'| \leq k_U$ and $|L'| \leq k_L$. Let $C_1 = \{(s, u) | u \in U'\}$ and $C_2 = \{(v, t) | v \in L'\}$. We claim that $C_1 \cup C_2$ is a solution for $(G', X, Y, k_1, k_2, ..., k_\ell)$. Obviously $|C_1| \leq k_1, |C_2| \leq k_2$ and C is Z-respecting. For contradiction, suppose that there is a directed path su'v't in $G' \setminus (C_1 \cup C_2)$. It follows that $u' \notin U'$ and $v' \notin L'$. Thus there is an edge u'v' which is not covered by $U' \cup L'$ in G, contradicting that $U' \cup L'$ is a vertex cover of G.

Suppose that $(G', X, Y, k_1, k_2, ..., k_\ell)$ is a yes instance. Then there is a Z-respecting budgetrespecting st-cut $C = C_1 \cup C_2$ such that C_1 is a set of arcs of color 1 of size at most k_1 and C_2 is a set of arcs of color 2 of size at most k_2 . Obviously any arc between U and V in G' is not in the solution because they are not deletable. Let $U' = \{u|(s, u) \in C_1\}$ and $L' = \{v|(v, t) \in C_2\}$. We get that $U' \subseteq U$, $L' \subseteq L$, $|U'| \leq k_1 = k_U$ and $|L'| \leq k_2 = k_L$. We claim that $U' \cup L'$ is a solution for (G, k_U, k_L) . For contradiction, suppose that there is an edge u'v' not covered by $U' \cup L'$. It follows that su'v't is a directed path in $G' \setminus C$, contradicting that $C = C_1 \cup C_2$ is a solution for $(G', X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$. This completes the proof.

4.1.2 FPT algorithm for MULTI-BUDGETED CUT

We now give an FPT algorithm parameterized by $k = \sum_{i=1}^{\ell} k_i$ and ℓ for the MULTI-BUDGETED CUT problem. We follow a branching strategy that recursively reduces a set Z of deletable edges. That is, we start with $Z = \bigcup_{i=1}^{\ell} E_i$ (so that every solution is initially Z-respecting) and in each recursive step, we look for a Z-respecting solution and reduce the set Z in a branching step. Consider a recursive call where we look for a Z-respecting solution to the input MULTI-BUDGETED CUT instance $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$. That is, we look for a Z-respecting budgetrespecting cut. We apply Theorem 9 to it. If it returns that there is no Z-respecting XY-cut, we terminate the current branch, as there is no solution. Otherwise, we obtain the paths $P_1, P_2, \ldots, P_{\lambda}$, the set B (which we will not use in this section), and the cut C.

If C is budget-respecting, then it is a solution and we can return it. Otherwise, we perform the following branching step. We iterate over all tuples $(A_1, ..., A_\ell)$ such that for every $i \in [\ell]$, $A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$. A_i represents the subset of paths $P_1, ..., P_\lambda$ on which at least one edge of color i is in the solution for each $i \in [\ell]$. For those edges of color i which are on the paths not indicated by A_i , they are not in the solution. Thus we can safely delete them from Z. More formally, for every $i \in [\ell]$ and $j \in [\lambda] \setminus A_i$, we remove from Z all edges of $E(P_j) \cap E_i$. We recurse on the reduced set Z. A pseudocode is available in Figure 6.2.



Figure 4.1: FPT algorithm for MULTI-BUDGETED CUT

Theorem 10. The algorithm in Figure 6.2 for MULTI-BUDGETED CUT is correct and runs in time $O(2^{\ell k^2} \cdot k \cdot (|V(G)| + |E(G)|))$ where $k = \sum_{i=1}^{\ell} k_i$.

Proof. We prove the correctness of the algorithm by showing that it returns a solution if and only if the input instance is a yes-instance. The "only if" direction is obvious, as the algorithm returns only Z-respecting budget-respecting XY-cuts and $Z \subseteq \bigcup_{i=1}^{\ell} E_i$ in each recursive call.

We prove the correctness for the "if" direction. Let C_0 be a solution, that is, a budgetrespecting XY-cut. In the initial call to **Solve**, C_0 is Z-respecting. It suffices to inductively show that in each call to **Solve** such that C_0 is Z-respecting, either the call returns a solution, or C_0 is \widehat{Z} -respecting for at least one of the subcalls. Since C_0 is Z-respecting, the application of Theorem 9 returns objects $(P_i)_{i=1}^{\lambda}$, B, and C. If C is budget-respecting, then the algorithm returns it and we are done. Otherwise, consider the branch $(A_1, A_2, \ldots, A_\ell)$ where $A_i = \{j | E(P_j) \cap C_0 \neq \emptyset\}$. Since C_0 is budget-respecting, $C_0 \subseteq Z$, and no edge of Z appears on more than one path P_j , we have $|A_i| \leq k_i$ for every $i \in [\ell]$. Thus, $(A_1, A_2, \ldots, A_\ell)$ is a branch considered by the algorithm. In this branch, the algorithm refines the set Z to \widehat{Z} . By the definition of A_i , for every $i \in [\ell]$ and $j \in [\lambda] \setminus A_i$, we have $C_0 \cap E_i \cap E(P_j) = \emptyset$. Consequently, C_0 is \widehat{Z} -respecting and we are done.

For the time bound, the following observation is crucial.

Claim 3. Consider one recursive call Solve(Z) where the application of Theorem 9 in line a returned objects $(P_i)_{i=1}^{\lambda}$, B, and C. Assume that in some recursive subcall $Solve(\widehat{Z})$ invoked in line d.3 (Figure 6.2), the subsequent application of Theorem 9 in line a of the subcall returned a cut of the same size, that is, the algorithm of Theorem 9 returned a cut \widehat{C} of size $\widehat{\lambda} = \lambda$. Then the cut \widehat{C} is budget-respecting and, consequently, is returned in line c of the subcall.

Proof. Since $|\hat{C}| = \lambda$ is a \hat{Z} -respecting XY-cut, $\hat{Z} \subseteq Z$, and every edge $e \in Z$ appears on at most one path P_i , we have that \hat{C} consists of exactly one edge of \hat{Z} on every path P_i , that is, $\hat{C} = \{e_1, e_2, \ldots, e_{\lambda}\}$ and $e_j \in E(P_j) \cap \hat{Z}$ for every $j \in [\lambda]$. (I.e., the paths $(P_j)_{j=1}^{\lambda}$ still correspond to a maximum flow from X to Y with edges of \hat{Z} being of unit capacity and edges outside \hat{Z} of infinite capacity.) If $e_j \in E_i$ for some $j \in [\lambda]$ and $i \in [\ell]$, then by the construction of set \hat{Z} , we have $j \in A_i$. Consequently, $|\{j|e_j \in E_i\}| \leq |A_i| \leq k_i$ for every $i \in [\ell]$, and thus \hat{C} is budget-respecting.

Claim 3 implies that the depth of the search tree is bounded by k, as the algorithm terminates when λ exceeds k. At every step, there are at most $(2^{\lambda})^{\ell} \leq (2^{k})^{\ell}$ different tuples $(A_{1}, ..., A_{\ell})$ to consider. Consequently, there are $\mathcal{O}(2^{(k-1)k\ell})$ nodes of the search tree that enter the loop in line d and $\mathcal{O}(2^{k^{2}\ell})$ nodes that invoke the algorithm of Theorem 9. As a result, the running time of the algorithm is $\mathcal{O}(2^{\ell k^{2}} \cdot k \cdot (|V(G)| + |E(G)|))$.

4.2 Multi-budgeted important separators with applications

Similar to the concept of important separators proposed by Marx [103] (see also [32, Chapter 8]), we define *multi-budgeted important separators* as follows.

Definition 1. Let $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ be a MULTI-BUDGETED CUT instance and let $Z \subseteq \bigcup_{i=1}^{\ell} E_i$ be a set of deletable edges. Let C_1, C_2 be two minimal Z-respecting budget-respecting XY-cuts. We say that C_1 dominates C_2 if

- 1. every vertex reachable from X in $G C_2$ is also reachable from X in $G C_1$;
- 2. for every $i \in [\ell], |C_1 \cap E_i| \le |C_2 \cap E_i|$.

We say that \widehat{C} is an important Z-respecting budget-respecting XY-cut if \widehat{C} is a minimal Zrespecting budget-respecting XY-cut and no other minimal Z-respecting budget-respecting XYcut dominates \widehat{C} . \widehat{C} is an important budget-respecting XY-cut if it is an important Zrespecting budget-respecting XY-cut for $Z = \bigcup_{i=1}^{\ell} E_i$.

Chen et al. [23] showed an enumeration procedure for (classic) important separators using similar charging scheme as the one of the previous section. Our main result in this section is a merge of the arguments from the previous section with the arguments of Chen et al., yielding the following theorem.

Theorem 11. Let $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ be a MULTI-BUDGETED CUT instance, let $Z \subseteq \bigcup_{i=1}^{\ell} E_i$ be a set of deletable edges, and denote $k = \sum_{i=1}^{\ell} k_i$. Then one can in $2^{\mathcal{O}(k^2 \ell \log(k\ell))}(|V(G)| + |E(G)|)$ time enumerate a family of minimal Z-respecting budget-respecting XY-cuts of size $2^{\mathcal{O}(k^2 \ell \log(k\ell))}$ that contains all important ones.

Theorem 3 follows from Theorem 11 via an analogous arguments as in [23]. In this section we focus on the proof of Theorem 11.

Proof of Theorem 11. Consider the recursive algorithm presented in Figure 6.4. The recursive procedure **ImportantCut** takes as an input a MULTI-BUDGETED CUT instance $I = (G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ and a set $Z \subseteq \bigcup_{i=1}^{\ell} E_i$, with the goal to enumerate all important Z-respecting budget-respecting XY-cuts. Note that the procedure may output some more Z-respecting budget-respecting XY-cuts; we need only to ensure that

- 1. it outputs all important ones,
- 2. it outputs $2^{\mathcal{O}(k^2\ell \log(k\ell))}$ cuts, and
- 3. it runs within the desired time.

The procedure first invokes the algorithm of Theorem 9 on (G, X, Y, k, Z), where $k = \sum_{i=1}^{\ell} k_i$. If the call returned that there is no Z-respecting XY-cut of size at most k, we can return an empty set. Otherwise, let $(P_j)_{j=1}^{\lambda}$, B, and C be the computed objects. We perform a branching step, with each branch labelled with a tuple $(A_1, A_2, \ldots, A_{\ell})$ where $A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$ for every $i \in [\ell]$. A branch $(A_1, A_2, \ldots, A_{\ell})$ is supposed to capture important cuts C_0 with $\{j|C_0 \cap B \cap E(P_j) \cap E_i \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$; that is, for every $i \in [\ell]$ and $j \in [\lambda]$ we guess if C_0 contains a *bottleneck* edge of color i on path P_j . All this information (i.e., paths P_j , the set B, the cut C, and the sets A_i) are passed to an auxiliary procedure **Enum**.

The procedure **Enum** shrinks the set Z according to sets A_i . More formally, for every $i \in [\ell]$ and $j \in [\lambda] \setminus A_i$ we delete from Z all edges from $B \cap E_i \cap E(P_j)$, obtaining a set $\widehat{Z} \subseteq Z$. At this point, we check if the reduction of the set Z to \widehat{Z} increased the size of minimum Z-respecting XY-cut by invoking Theorem 9 on $(G, X, Y, k, \widehat{Z})$ and obtaining objects $(\widehat{P}_j)_{j=1}^{\widehat{\lambda}}, \widehat{B}, \widehat{C}$ or a negative answer. If the size of the minimum cut increased, that is, $\widehat{\lambda} > \lambda$, we recurse with the original procedure **ImportantCut**. Otherwise, we add one cut to S, namely \widehat{C} . Furthermore, we try to shrink one of the sets A_i by one and recurse; that is, for every $i \in [\ell]$ and every $j \in A_i$, we recurse with the procedure **Enum** on sets $A'_{i'}$ where $A'_i = A_i \setminus \{j\}$ and $A'_{i'} = A_{i'}$ for every $i' \in [\ell] \setminus \{i\}$.

Let us first analyse the size of the search tree. A call to **ImportantCut** invokes at most $2^{\lambda\ell} \leq 2^{k\ell}$ calls to **Enum**. Each call to **Enum** either falls back to **ImportantCut** if $\hat{\lambda} > \lambda$ or branches into $\sum_{i=1}^{\ell} |A_i| \leq k\ell$ recursive calls to itself. In each recursive call, the sum $\sum_{i=1}^{\ell} |A_i|$ decreases by one. Consequently, the initial call to **Enum** results in at most $(k\ell)^{k\ell}$ recursive calls, each potentially falling back to **ImportantCut**. Since each recursive call to **ImportantCut** uses strictly larger value of λ , which cannot grow larger than k, the total size of the recursive call to **Enum** adds at most one set to S, while each recursive call to **ImportantCut** and **Enum** runs in time $\mathcal{O}(2^{k\ell} \cdot k \cdot (|V(G)| + |E(G)|))$. The promised size of the family S and the running time bound follows. It remains to show correctness, that is, that every important Z-respecting budget-respecting XY-cut is contained in S returned by a call to **ImportantCut** (I, Z).

We prove by induction on the size of the recursion tree that (1) every call to **Important-** $\operatorname{Cut}(I, Z)$ enumerates all important Z-respecting budget-respecting XY-cuts, and (2) every call to $\operatorname{Enum}(I, (P_j)_{j=1}^{\lambda}, B, C, Z, (A_i)_{i=1}^{\ell})$ enumerates all important Z-respecting budget-respecting XY-cuts C_0 with the property that $\{j | E_i \cap E(P_j) \cap B \cap C_0 \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$.

The inductive step for a call **ImportantCut**(I, Z) is straightforward. Let us fix an arbitrary important Z-respecting budget-respecting XY-cut C_0 . Since C_0 is budget-respecting, C_0 is a Z-respecting cut of size at most k, and thus the initial call to Theorem 9 cannot return NO. Consider the tuple $(A_1, A_2, \ldots, A_\ell)$ where for every $i \in [\ell], \{j | E(P_j) \cap E_i \cap B \cap C_0\} = A_i$. Since C_0 is budget-respecting and the paths P_j do not share an edge of Z, we have that $|A_i| \leq k_i$

ImportantCut(I, Z)Input: A MULTI-BUDGETED CUT instance $I = (G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ and a set $Z \subseteq \bigcup_{i=1}^{\ell} E_i$. Output: a family S of minimal Z-respecting budget-respecting XY-cuts that contains all important ones. 1. $\mathcal{S} := \emptyset;$ 2. apply the algorithm of Theorem 9 to (G, X, Y, k, Z) with $k = \sum_{i=1}^{\ell} k_i$, obtaining either objects $(P_i)_{i=1}^{\lambda}$, B, and C, or an answer NO; 3. if an answer NO is obtained, then return S; 4. for each $(A_1, ..., A_\ell)$ such that for every i in $[\ell], A_i \subseteq [\lambda]$ and $|A_i| \leq k_i$ do 4.1 $\mathcal{S} := \mathcal{S} \cup \mathbf{Enum}(I, (P_j)_{j=1}^{\lambda}, B, C, Z, (A_i)_{i=1}^{\ell})$ 5. return S**Enum** $(I, Z, (P_j)_{j=1}^{\lambda}, B, C, (A_i)_{i=1}^{\ell})$ Input: A MULTI-BUDGETED CUT instance $I = (G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$, a set $Z \subseteq \bigcup_{i=1}^{\ell} E_i$, a family $(P_j)_{j=1}^{\lambda}$ of paths from X to Y such that every edge of Z appears on at most one path P_j , a set B consisting of all edges that participate in some minimum Z-respecting XY-cut, a minimum Zrespecting XY-cut C closest to Y, and sets $A_i \subseteq [\lambda]$ of size at most k_i for every $i \in [\ell]$ Output: a family S of minimal Z-respecting budget-respecting XY-cuts that contains all cuts C_0 that are important Z-respecting budget respecting XY-cuts and satisfy $\{j|E(P_j) \cap B \cap C_0 \cap E_i \neq j\}$ \emptyset $\subseteq A_i$ for every $i \in [\ell]$. a. $\widehat{Z} := Z;$ b. for each $i \in [\ell]$ do for each $j \in [\lambda] \setminus A_i$ do $\widehat{Z} := \widehat{Z} \setminus (B \cap E_i \cap E(P_j));$ c. apply the algorithm of Theorem 9 to $(G, X, Y, k, \widehat{Z})$, obtaining either objects $(\widehat{P}_i)_{i=1}^{\widehat{\lambda}}, \widehat{B}$, and \widehat{C} or an answer NO; d. if $\widehat{\lambda}$ exists and $\widehat{\lambda} > \lambda$, then d.1 $\mathcal{S} := \mathcal{S} \cup \mathbf{ImportantCut}(I, \widehat{Z});$ e. else if $\hat{\lambda}$ exists and equals λ , then e.1 $\mathcal{S} := \mathcal{S} \cup \{\widehat{C}\};$ for each $i \in [\ell]$ do e.2for each $j \in A_i$ do $A'_i := A_i \setminus \{j\}$ and $A'_{i'} := A_{i'}$ for every $i' \in [\ell] \setminus \{i\}$ $\mathcal{S} := \mathcal{S} \cup \mathbf{Enum}(I, \widehat{Z}, (P_j)_{j=1}^{\lambda}, \widehat{B}, \widehat{C}, (A'_i)_{i=1}^{\ell}).$ f. return S

Figure 4.2: FPT algorithm for enumerating important multi-budgeted Z-respecting XY-cuts
for every $i \in [\ell]$ and the algorithm considers this tuple in one of the branches. Then, from the inductive hypothesis, the corresponding call to **Enum** returns a set containing C_0 .

Consider now a call to $\operatorname{Enum}(I, (P_j)_{j=1}^{\lambda}, B, C, Z, (A_i)_{i=1}^{\ell})$ and an important Z-respecting budget-respecting XY-cuts C_0 with the property that $\{j|E_i \cap E(P_j) \cap B \cap C_0 \neq \emptyset\} \subseteq A_i$ for every $i \in [\ell]$. By the construction of \widehat{Z} and the above assumption, C_0 is \widehat{Z} -respecting. In particular, the call to the algorithm of Theorem 9 cannot return NO. Hence, in the case when $\widehat{\lambda} > \lambda$, C_0 is enumerated by the recursive call to **ImportantCut** and we are done. Assume then $\widehat{\lambda} = \lambda$.

For $i \in [\ell]$, let $\widehat{A}_i = \{j | E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\}$. Since $\widehat{Z} \subseteq Z$ but the sizes of minimum Z-respecting and \widehat{Z} -respecting XY-cuts are the same, we have $\widehat{B} \subseteq B$. Consequently, $\widehat{A}_i \subseteq A_i$ for every $i \in [\ell]$.

Assume there exists $i \in [\ell]$ such that $\widehat{A}_i \subsetneq A_i$ and let $j \in A_i \setminus \widehat{A}_i$. Consider then the branch (i, j) of the **Enum** procedure, that is, the recursive call with $A'_i = A_i \setminus \{j\}$ and $A'_{i'} = A_{i'}$ for $i' \in [\ell] \setminus \{i\}$. Observe that we have $\{j|E_{i'} \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\} \subseteq A'_{i'}$ for every $i' \in [\ell]$ and, by the inductive hypothesis, the corresponding call to **Enum** enumerates C_0 . Hence, we are left only with the case $\widehat{A}_i = A_i$, that is, $A_i = \{j|E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\}$ for every $i \in [\ell]$.

We claim that in this case $C_0 = \widehat{C}$. Assume otherwise. Since $|\widehat{C}| = \widehat{\lambda} = \lambda$ and $\widehat{Z} \subseteq Z$, \widehat{C} contains exactly one edge on every path P_j . Also, $\widehat{C} \subseteq \widehat{B}$ by the definition of the set \widehat{B} . Since \widehat{C} is the minimum \widehat{Z} -respecting XY-cut that is closest to Y, $\widehat{C} = \{e_1, e_2, \ldots, e_{\lambda}\}$ where e_j is the last (closest to Y) edge of \widehat{B} on the path P_j for every $j \in [\lambda]$.

Let R_0 and \widehat{R} be the set of vertices reachable from X in $G - C_0$ and $G - \widehat{C}$, respectively. Let D be a minimal XY-cut contained in $\delta^+(R_0 \cup \widehat{R})$. (Note that $\delta^+(R_0 \cup \widehat{R})$ is an XY-cut because $X \subseteq R_0 \cup \widehat{R}$ and $Y \cap (R_0 \cup \widehat{R}) = \emptyset$.) Then since $D \subseteq C_0 \cup \widehat{C} \subseteq Z$, D is Z-respecting. By definition, every vertex reachable from X in $G - R_0$ is also reachable from X in G - D.

We claim that D is budget-respecting and, furthermore, dominates C_0 . Fix a color $i \in [\ell]$; our goal is to prove that $|D \cap E_i| \leq |C_0 \cap E_i|$. To this end, we charge every edge of color i in $D \setminus C_0$ to a distinct edge of color i in $C_0 \setminus D$. Since $D \subseteq C_0 \cup \widehat{C}$, we have that $D \setminus C_0 \subseteq \widehat{C}$, that is, an edge of $D \setminus C_0$ of color i is an edge e_j for some $j \in [\lambda]$ with $e_j \in E_i$ and $e_j \in D \setminus C_0$.

Recall that we are working in the case $A_i = \{j | E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset\}$. Since $e_j \in \widehat{C} \subseteq \widehat{Z}$, we have that $j \in A_i$. Hence, there exists $e'_j \in E_i \cap E(P_j) \cap \widehat{B} \cap C_0$. By the definition of \widehat{C} , e_j is the last (closest to Y) edge of \widehat{B} on P_j . Since $e_j \notin C_0$, $e'_j \neq e_j$ and e'_j lies on the subpath of P_j between X and the tail of e_j . This entire subpath is contained in \widehat{R} and, hence, $e'_j \notin D$.

We charge e_j to e'_j . Since $e'_j \in E(P_j) \cap E_i \cap \widehat{B} \cap (C_0 \setminus D)$, for distinct j, the edges e'_j are distinct as the paths P_j do not share an edge belonging to Z and $\widehat{B} \subseteq \widehat{Z} \subseteq Z$. Consequently, $|D \cap E_i| \leq |C_0 \cap E_i|$. This finishes the proof that D dominates C_0 .

Since C_0 is important, we have $D = C_0$. In particular, $\widehat{R} \subseteq R_0$. On the other hand, for every $j \in [\lambda]$ we have that $e_j \in \widehat{C} \subseteq \widehat{Z} \subseteq Z \subseteq \bigcup_{i=1}^{\ell} E_i$. In particular, there exists $i \in [\ell]$ such that $e_j \in E_i$ and $j \in A_i$. Hence, we also have $E_i \cap E(P_j) \cap \widehat{B} \cap C_0 \neq \emptyset$. But the entire subpath of P_j from X to the tail of e_j lies in $\widehat{R} \subseteq R_0$, while e_j is the last edge of \widehat{B} on P_j . Hence, $e_j \in C_0$. Since the choice of j is arbitrary, $\widehat{C} \subseteq C_0$. Since \widehat{C} is an XY-cut and C_0 is minimal, $\widehat{C} = C_0$ as claimed.

This finishes the proof of Theorem 11.

Next we show the applications of multi-budgeted important separators to the problems MULTI-BUDGETED SKEW MULTICUT and MULTI-BUDGETED DIRECTED FEEDBACK ARC SET. We start by observing a direct corollary of the maximality criterium in the definition of important budget-respecting separators. **Lemma 6.** Given an instance $(G, X, Y, \ell, (E_i, k_i)_{i=1}^{\ell})$ of MULTI-BUDGETED CUT, for every minimal budget-respecting XY-cut C there exists an important budget-respecting XY-cut C' that dominates C.

Similar to the pushing lemma for SKEW EDGE MULTICUT [23], we propose a pushing lemma for the multi-budgeted variant.

Lemma 7. Every yes-instance $I = (G, \ell, (E_i, k_i)_{i=1}^{\ell}, (s_i, t_i)_{i=1}^{q})$ of SKEW EDGE MULTICUT admits a solution that contains an important budget-respecting XY-cut for $X = \{s_q\}$ and $Y = \{t_1, t_2, \ldots, t_q\}$.

Proof. Let C be a solution to I. Let $X = \{s_q\}$, $Y = \{t_1, ..., t_q\}$, and R be the set of vertices reachable from s_q in G - C. Since C is a solution, $\delta^+(R) \subseteq C$ is a budget-respecting XY-cut; let $D \subseteq \delta^+(R)$ be a minimal one. By Lemma 6, there exists an important budget-respecting XY-cut D^* dominating D. Let R^* be the set of vertices reachable from s_q in $G - D^*$. We claim that $C^* := (C \setminus D) \cup D^*$ is a solution to I as well.

Suppose for contradiction that there is a directed path P from s_i to t_j for some $i \ge j$ in $G - C^*$. If P contains one vertex of R^* , it contradicts that D^* is an XY-cut because P must contain one edge of D^* . Thus P is disjoint from R^* . Since $R \subseteq R^*$, P is disjoint from R, and hence P is disjoint from D. Since P is not cut by $C^* = (C \setminus D) \cup D^*$, P is not cut by $C \setminus D$. It follows that P is not cut by $C = (C \setminus D) \cup D$, contradicting that C is a solution.

To complete the proof, note that for every $i \leq [\ell]$ we have $|D^* \cap E_i| \leq |D \cap E_i|$ since D' dominates D, and hence $|C^* \cap E_i| \leq |C \cap E_i|$. Consequently, C^* is budget-respecting.

Lemma 7 yields the following branching strategy.

Theorem 12. There is an FPT algorithm for MULTI-BUDGETED SKEW EDGE MULTICUT running in time $2^{\mathcal{O}(k^3 \ell \log(k\ell))} \cdot (|V(G)| + |E(G)|)$.

Proof. If $k_i < 0$ for some $i \in [\ell]$, then we can answer NO. Otherwise, if q = 0, then we can answer YES. Otherwise, perform a depth-first search from s_q . If no terminal t_i has been reached, delete the visited vertices together with t_q , decrease q by one and restart the algorithm. Since this operation can be performed in time linear in the size of the deleted part of the graph, in total it takes linear time.

Otherwise, proceed as follows. By Lemma 7 if the input instance is a yes-instance, there is a solution C^* which contains an important budget-respecting $s_q\{t_1, t_2, \ldots, t_q\}$ -cut. By Theorem 11, we can enumerate in time $2^{\mathcal{O}(k^2 \ell \log(k\ell))}(|V(G)| + |E(G)|)$ a set of minimal budgetrespecting XY-cuts S of size $2^{\mathcal{O}(k^2 \ell \log(k\ell))}$ that contains all important ones. We invoke this enumeration, and branch on the choice of important budget-respecting $s_q\{t_1, t_2, \ldots, t_q\}$ -cut contained in the sought solution. In a branch where a cut D is chosen, we delete D from the graph and decrease each budget k_i by $|D \cap E_i|$. Since at least one terminal t_i is reachable from s_q , in every branch the cut D is nonempty and thus $k = \sum_{i=1}^{\ell} k_i$ decreases by at least one. Consequently, the depth of the recursion is bounded by k. The running time bound follows.

We now use the algorithm of Theorem 12 to give an algorithm for MULTI-BUDGETED DI-RECTED FEEDBACK ARC SET, completing the proof of Theorem 3.

Theorem 13. MULTI-BUDGETED DIRECTED FEEDBACK ARC SET can be solved in time $2^{\mathcal{O}(k^3 \ell \log(k\ell))}(|V(G)| + |E(G)|).$

Proof. Let $I = (G, \ell, (E_i, k_i)_{i=1}^{\ell})$ be an input instance. We start by applying the algorithm of Chen et al. [23] for the classic DIRECTED FEEDBACK ARC SET on G with parameter $k = \sum_{i=1}^{\ell}$.

If the call returned that there is no solution, we can safetely return NO. Otherwise, let W be the set of tails of the arcs of the returned solution; clearly $|W| \leq k$ and G - W is acyclic.

Suppose I is a yes-instance and there is a solution S. Then G-S is a directed acyclic graph, admiting a topological ordering of V(G). This ordering indices a permutation of the vertices in W. In our algorithm, we branch on every permutation of the vertices in W, ensuring that at least one of the permutation is the same as the permutation induced by the topological ordering of G-S. Let $w_1, ..., w_{|W|}$ be an arbitrary permutation of the vertices in W. We construct a graph G'as follows. For each $i \in [|W|]$, we replace every vertex w_i with two vertices s_i, t_i , every edge (w_i, a) with (s_i, a) of the same color and every edge (b, w_i) with (b, t_i) of the same color. Then we add a directed edge (t_i, s_i) for each $i \in [|W|]$ with no color. In this manner, we construct a MULTI-BUDGETED SKEW EDGE MULTICUT instance $I' = (G', \ell, (E'_i, k_i)_{i=1}^{\ell}, (s_i, t_i)_{i=1}^{|W|})$ corresponding to the permutation $w_1, ..., w_{|W|}$.

We claim that the input instance I of MULTI-BUDGETED DIRECTED FEEDBACK ARC SET is a yes-instance if and only if there exists one permutation $w_1, ..., w_{|W|}$ of W such that the corresponding MULTI-BUDGETED SKEW EDGE MULTICUT instance I' is a yes-instance. For the "only if direction", let S be a solution to I. We have a topological ordering of V(G), inducing an ordering $w_1, ..., w_{|W|}$ on W. For this ordering, let I' be the corresponding instance of MULTI-BUDGETED SKEW EDGE MULTICUT. According to the way we construct G', every edge in Shas a corresponding edge in G'. Let S' be the set of the corresponding edges of edges in S. We claim that S' is a solution for I'. Obviously S' is budget-respecting. Suppose for contradiction that there is a directed path P from s_i to t_j for some $i \ge j$ in G' - S'. If i = j, then the corresponding edges of P form a directed cycle passing through w_i in G - S, a contradiction. Suppose that i > j. If P goes through some edge in $\{(t_i, s_i)|i \in [|W|]\}$, then there must be a subpath of P' from $s_{i'}$ to $t_{j'}$ such that i' > j' and P' contains no edges in $\{(t_i, s_i)|i \in [|W|]\}$. Then the corresponding edges of P' is a directed path from w_i to w_j , contradicting that w_i is later than w_j in the topological ordering of V(G) after removing S.

For the "if direction", suppose that S' is a solution for I' and $w_1, ..., w_{|W|}$ is the corresponding ordering of W. Let S be the set of edges consisting of the corresponding edges of S'. We claim that S is the solution for I. Obviously S is budget-respecting. Suppose that there is a cycle Qin G - S. Since W is a feedback vertex set for G, Q must go through at least one vertex in W. Suppose that Q goes through a vertex in W, namely w_i . Then we can find a path from s_i to t_i in G' - S', contradicting that S' is a solution to I'.

This finishes the proof of the lemma and of the whole Theorem 3.

Chapter 5

Two Disjoint Shortest Paths Problem with transition restrictions

In this chapter, we present polynomial-time algorithms for vertex-disjoint and edge-disjoint cases of 2-DSPP WITH TRANSITION RESTRICTIONS when every directed cycle has positive length. Recall that in a directed graph G, a transition is an ordered pair of adjacent edges such that the head of the first edge is equal to the tail of the second edge. A transition system T is a set of transitions in G. We say that a path P is T-compatible if every two consecutive edges of Pform a transition of T. Given a directed graph G = (V, E), a length function $w : E \to \mathbb{R}_{\geq 0}$, two pairs of vertices $(s_1, t_1), (s_2, t_2)$ in G and a prescribed transition system T, the 2-DSPP WITH TRANSITION RESTRICTIONS asks to find two disjoint (vertex-disjoint or edge-disjoint) paths P_1 and P_2 in G such that P_i is a shortest path (even without transition restrictions) from s_i to t_i and P_i is also T-compatible for i = 1, 2.

Organization of this chapter. In Section 5.1, we present a polynomial-time algorithm for edge-disjoint case of 2-DSPP WITH TRANSITION RESTRICTIONS when every directed cycle has positive length. In Section 5.2, we present a polynomial-time algorithm for vertex-disjoint case of 2-DSPP WITH TRANSITION RESTRICTIONS when every directed cycle has positive length.

Additional notions for this chapter. We define E_i to be the set of edges that appear in some shortest path (without transition restrictions) from s_i to t_i for i = 1, 2. By this definition, an s_i - t_i path is a shortest T-compatible s_i - t_i path if and only if it consists of edges of E_i and is also T-compatible for i = 1, 2. Thus the edge-disjoint (vertex-disjoint) 2-DSPP WITH TRANSITION RESTRICTIONS is equivalent to finding two edge-disjoint (vertex-disjoint) T-compatible paths P_1 and P_2 such that P_i is from s_i to t_i , $E(P_i) \subseteq E_i$ and P_i satisfies the transition restrictions for i = 1, 2. Each set E_i can be computed in polynomial time using the method from the paper of Bérczi and Kobayashi [9]. First, we compute the distance $d_i(v)$ from s_i to v for i = 1, 2, using Dijkstra's algorithm. Let $\mathcal{E}_i = \{uv \mid d_i(v) - d_i(u) = w(uv)\}$. Then $E_i = \{uv \in \mathcal{E}_i \mid$ there exists a path from v to t_i in $\mathcal{E}_i\}$.

For a set F of directed edges, let \overline{F} be the set of edges obtained by reversing all edges of F, that is, $\overline{F} = \{vu \mid uv \in F\}$. For a directed edge e = uv, let $\overline{e} = vu$ denote the edge obtained by reversing e.

5.1 Edge-disjoint case of 2DSPP

We show that the edge-disjoint case of 2-DSPP WITH TRANSITION RESTRICTIONS can be solved in polynomial time. We use the method of Bérczi and Kobayashi [9], which reduces the problem of EDGE DISJOINT 2-DSPP to finding a path in a graph \mathcal{G} constructed from the input graph G. Based on that, we just need to delete edges of \mathcal{G} which correspond to forbidden transitions of G with respect to T and it suffices to find the path in the remaining subgraph of \mathcal{G} .

We repeat the procedure of Bérczi and Kobayashi [9] briefly here for consistency. Let G be a graph (without transition systems T) such that the length of every dicycle in G is positive. First, we compute E_i for i = 1, 2. Then we create four new vertices s'_1, s'_2, t'_1, t'_2 , create four edges $s'_1s_1, s'_2s_2, t_1t'_1, t_2t'_2$ of length 0 respectively, and add $s'_is_i, t_it'_i$ to E_i for i = 1, 2. Let $E_0 = E_1 \cap E_2, E_1^* = E_1 \setminus E_0, E_2^* = E_2 \setminus E_0$. We remove all edges of $E(G) \setminus (E_1 \cup E_2)$, contract all edges of E_0 and reverse all edges of E_2^* . Finally we get a new graph $G^* = (V^*, E^* = E_1^* \cup \overline{E_2^*})$. Let $V_0 \subseteq V$ be the set of vertices that are newly created after contracting E_0 . For $v \in V_0$, we use G_v to denote the subgraph of $G - (E(G) \setminus (E_1 \cup E_2))$ induced by the vertices corresponding to v before contracting. For an edge $e \in E^*$, let $f(e) \in E(G)$ be the edge corresponding to ebefore the contracting and reversing operations.

The following two lemmas show that G_v is acyclic for every $v \in V_0$ and G^* is acyclic.

Lemma 8. [9] The edge set E_i forms no dicycle in G for i = 1, 2.

Lemma 9. [9] In the graph G, suppose that C is a dicycle in $E_1 \cup \overline{E_2}$. Then $E_1 \cap E(C) \subseteq E_2$ and $E_2 \cap \overline{E(C)} \subseteq E_1$.

Then we define a new digraph \mathcal{G} whose vertex set is $W = E_1^* \times \overline{E_2^*}$. There is a directed edge from (e_1, e_2) to (e'_1, e'_2) if one of three cases holds.

- (i) $e_1 = e'_1$ and $\operatorname{head}_{G^*}(e_2) = \operatorname{tail}_{G^*}(e'_2) = v$. There is no path from $\operatorname{head}_{G^*}(e_1)$ to v in G^* . Moreover, if $v \in V_0$, then G_v contains a path from $\operatorname{tail}_G(e'_2)$ to $\operatorname{head}_G(e_2)$.
- (ii) $e_2 = e'_2$ and $\text{head}_{G^*}(e_1) = \text{tail}_{G^*}(e'_1) = v$. There is no path from $\text{head}_{G^*}(e_2)$ to v in G^* . Moreover, if $v \in V_0$, then G_v contains a path from $\text{head}_G(e_1)$ to $\text{tail}_G(e'_1)$.
- (iii) head_{G*}(e_2) = tail_{G*}(e'_2) = head_{G*}(e_1) = tail_{G*}(e'_1) = v. If $v \in V_0$, then G_v contains two edge-disjoint paths from head_G(e_1) to tail_G(e'_1) and from tail_G(e'_2) to head_G(e_2) respectively.

Finally the following lemma reduces the edge-disjoint version of 2-DSPP to finding a path in \mathcal{G} from (s'_1s_1, t'_2t_2) to $(t_1t'_1, s_2s'_2)$. Note that $s_i, t_i \in V(G)$ might be the endpoints of edges of E_0 for i = 1, 2. In this case, although we might contract the edges incident to $s_i, t_i \in V(G)$ and replace these vertices with new vertices. Therefore, we slightly abuse the notation and use s_i and t_i to denote the vertex adjacent to s'_i and t'_i respectively in G^* for i = 1, 2, for the sake of simplicity.

Lemma 10. [9] There is a directed path in \mathcal{G} from (s'_1s_1, t'_2t_2) to $(t_1t'_1, s_2s'_2)$ if and only if G has two edge-disjoint paths P_1 and P_2 such that P_i is from s'_i to t'_i and $P_i \subseteq E_i$ for i = 1, 2.

To solve the edge-disjoint version of DIRECTED TWO DISJOINT SHORTEST PATHS PROBLEM (2-DSPP) WITH TRANSITION RESTRICTIONS, we will show that it suffices to delete the edges in \mathcal{G} which correspond to forbidden transitions of G and find a path in the remaining graph of \mathcal{G} from (s'_1s_1, t'_2t_2) to $(t_1t'_1, s_2s'_2)$. For every edge in \mathcal{G} , we check whether it corresponds to forbidden transitions according to the following three cases and delete the edge if it corresponds to forbidden transitions. Suppose the edge is from some vertex $(e_1, e_2) \in W$ to another vertex $(e'_1, e'_2) \in W$.

- The edge is of type (i), i.e., $e_1 = e'_1$ and $\operatorname{head}_{G^*}(e_2) = \operatorname{tail}_{G^*}(e'_2) = v$. If $v \in V_0$, let G^s be the subgraph of G consisting of all edges of G_v together with $f(e_2)$ and $f(e'_2)$. In this case, if there is no T-compatible paths in G^s from $\operatorname{tail}_G(f(e'_2))$ to $\operatorname{head}_G(f(e_2))$, then remove the edge from \mathcal{G} . If $v \notin V_0$ and $\{\overline{e'_2}, \overline{e_2}\} \notin T_G(v)$, then remove the edge from \mathcal{G} .
- The edge is of type (ii), i.e., $e_2 = e'_2$ and $\operatorname{head}_{G^*}(e_1) = \operatorname{tail}_{G^*}(e'_1) = v$. If $v \in V_0$, let G^s be the subgraph of G consisting of all edges of G_v together with $f(e_1)$ and $f(e'_1)$. In this case, if $v \in V_0$ and there is no T-compatible path in G^s from $\operatorname{tail}_G(f(e_1))$ to $\operatorname{head}_G(f(e'_1))$, then remove the edge from \mathcal{G} . If $v \notin V_0$ and $\{e_1, e'_1\} \notin T_G(v)$, then remove the edge from \mathcal{G} .
- The edge is of type (iii), i.e., head_{G*}(e_2) = tail_{G*}(e'_2) = head_{G*}(e_1) = tail_{G*}(e'_1) = v. If $v \in V_0$, let G^s be the subgraph of G consisting of all edges of G_v together with $f(e_1), f(e'_1), f(e_2)$ and $f(e'_2)$. In this case, if G^s does not contain two T-compatible edge-disjoint paths such that one path is from tail_G($f(e_1)$) to head_G($f(e'_1)$) and the other path is from tail_G($f(e'_2)$) to head_G($f(e_2)$), then remove the edge from \mathcal{G} . If $v \notin V_0$ and $\{e_1, e'_1\} \notin T_G(v)$ or if $v \notin V_0$ and $\{\overline{e'_2}, \overline{e_2}\} \notin T_G(v)$, then remove the edge from \mathcal{G} .

We need to check whether there exists a T-compatible path between two given vertices in a (direced) forbidden-transition graph. Szeider shows a dichotomy of NP-complete and lineartime solvable for the problem of finding a T-compatible path between two given vertices of an (undirected) graph [119]. In contrast, the following lemma shows that in a directed acyclic graph, we can find a T-compatible path between two given vertices in polynomial time.

Lemma 11. In a directed acyclic graph G = (V, E) with transition system T_G , we can compute if there is a directed T-compatible path P from s to t for $s, t \in V(G)$ in polynomial time.

Proof. We construct a directed graph \tilde{G} as follows. First create two vertices s_0, t_0 . Then for every edge $e \in E(G)$, create a vertex v_e . For any two edges $e, e' \in E(G)$, create an edge $v_e v_{e'}$ if $ee' \in E(T_G(v))$ for some $v \in V(G)$. Finally, create edges $s_0 v_e$ for every $e \in E(G)$ such that $tail_G(e) = s$ and create edges $v_{e'}t_0$ for every $e' \in E(G)$ such that $head_G(e') = t$. We claim that we can find a directed path P' from s_0 to t_0 in \tilde{G} if and only if there is a directed *T*-compatible path *P* from *s* to *t* in *G*. For the "if" direction, suppose that there is such a path $P = e_1, e_2, \ldots, e_\ell$ in *G*, where e_1, \ldots, e_ℓ are the consecutive edges of *P*. Then we can obviously get the path $P' = s_0 v_{e_1}, v_{e_1} v_{e_2}, \ldots, v_{e_\ell} t_0$ by the definition of \tilde{G} . For the "only if" direction, suppose that there is a directed path $P' = s_0 v_{e_{i_1}}, v_{e_{i_1}} v_{e_{i_2}}, \ldots, v_{e_{i_\ell}} t_0$ in \tilde{G} . Then $P = e_{i_1}, e_{i_2}, \ldots, e_{i_\ell}$ is a directed *T*-compatible walk from *s* to *t* in *G*. Since *G* is acyclic, *P* is also a path. This completes the proof of the claim. We can build the graph \tilde{G} in $O(|E|^2)$ -time and find an $s_0 t_0$ path in \tilde{G} using DFS in $O(|E|^2)$ time. Thus the lemma holds. □

For $v \in V_0$, by Lemma 8, there is no dicycle in G_v . Moreover, observe that we cannot have a vertex in $V(G) \setminus V(G_v)$ adjacent to more than one edge from $E(G^s) \setminus E(G_v)$, so G^s is also acyclic. So we can decide whether or not to remove the edges of type (i) or (ii) from \mathcal{G} in polynomial time according to lemma 11. For the edges of type (iii), we need to compute if there are two edge-disjoint *T*-compatible paths in a directed acyclic graph. We show that it can be done in polynomial time and the algorithm is an adaption of the algorithm of finding two vertex-disjoint paths in DAG by Perl and Shiloach [111].

Lemma 12. In a directed acyclic graph G = (V, E) with transition system T_G , we can solve the edge-disjoint version of 2-DSPP WITH TRANSITION RESTRICTIONS in polynomial time.

Proof. First we modify the graph G as follows. We create four vertices s'_1, s'_2, t'_1, t'_2 and update V(G) as $V(G) \leftarrow V(G) \cup \{s'_1, s'_2, t'_1, t'_2\}$. We create four edges $\{s'_1s_1, s'_2s_2, t_1t'_1, t_2t'_2\}$ and update

E(G) as $E(G) \leftarrow E(G) \cup \{s'_1s_1, s'_2s_2, t_1t'_1, t_2t'_2\}$. Also, for i = 1, 2, we update $T_G(s_i)$ as

$$T_G(s_i) \leftarrow T_G(s_i) \cup \{\{e, e'\} \mid e = s'_i s_i \text{ and } \operatorname{tail}_G(e') = s_i\},\$$

and we update $T_G(t_i)$ as

$$T_G(t_i) \leftarrow T_G(t_i) \cup \{\{e, e'\} \mid e' = t_i t'_i \text{ and } \operatorname{head}_G(e) = t_i\}.$$

For every vertex $v \in V(G)$, define the level $\ell(v)$ as the length of a longest directed path in G starting from v. Since G is acyclic, this can be computed by repeatedly removing a vertex of G. Then we create a graph \tilde{G} as follows. Let the vertex set of \tilde{G} be $V(\tilde{G}) = \{(e_1, e_2) \mid e_1, e_2 \in \mathcal{C}\}$ E(G) and $e_1 \neq e_2$. For every $(e_1, e_2), (e'_1, e'_2) \in V(\tilde{G})$, create an edge from (e_1, e_2) to (e'_1, e'_2) if one of the following cases holds:

- (1) $e_1 = e'_1, \ \ell(\operatorname{head}_G(e_2)) \ge \ell(\operatorname{head}_G(e_1)), \ \{e_2, e'_2\} \in T_G(\operatorname{head}_G(e_2)).$
- $\begin{array}{l} (2) \quad e_2 = e_2', \ \ell(\operatorname{head}_G(e_1)) \ge \ell(\operatorname{head}_G(e_2)), \ \{e_1, e_1'\} \in T_G(\operatorname{head}_G(e_1)). \\ (3) \quad e_1 = e_1' = t_1 t_1', \ \ell(\operatorname{head}_G(e_2)) < \ell(t_1'), \ \{e_2, e_2'\} \in T_G(\operatorname{head}_G(e_2)). \\ (4) \quad e_2 = e_2' = t_2 t_2', \ \ell(\operatorname{head}_G(e_1)) < \ell(t_2'), \ \{e_1, e_1'\} \in T_G(\operatorname{head}_G(e_1)). \end{array}$

We claim that there are two T-compatible edge-disjoint paths P_1 and P_2 in G such that P_i is from s'_i to t'_i for i = 1, 2 if and only if there is a path P from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in \tilde{G} .

("only if" direction): Let $P_1 = e_1^0, e_1^1, \ldots, e_1^{p+1}$ and $e_1^0 = s_1's_1, e_1^{p+1} = t_1t_1'$. Let $P_2 = e_2^0, e_2^1, \ldots, e_2^{q+1}$ and $e_2^0 = s_2's_2, e_2^{q+1} = t_2t_2'$. For any $i \in \{0, 1, \ldots, p+1\}, j \in \{0, 1, \ldots, q+1\}$ such that $(i, j) \neq (p+1, q+1)$, one of the following four cases must hold.

- $i \leq p$ and $j \leq q$, $\ell(\text{head}_G(e_1^i)) \leq \ell(\text{head}_G(e_2^j))$ and there is an edge in \tilde{G} from (e_1^i, e_2^j) to $(e_1^i, e_2^{j+1}).$
- $i \leq p$ and $j \leq q$, $\ell(\text{head}_G(e_1^i)) \geq \ell(\text{head}_G(e_2^j))$ and there is an edge in \tilde{G} from (e_1^i, e_2^j) to
- i = p + 1 and $j \le q$, $\ell(\text{head}_G(e_2^j)) < \ell(t_1')$ and there is an edge in \tilde{G} from (e_1^{p+1}, e_2^j) to (e_1^{p+1}, e_2^{j+1}) .
- j = q + 1 and $i \le p$, $\ell(\text{head}_G(e_1^i)) < \ell(t_2')$ and there is an edge in \tilde{G} from (e_1^i, e_2^{q+1}) to (e_1^{i+1}, e_2^{q+1}) .

As a result, there is a path P from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in \tilde{G} . This finishes the proof for "only if" direction.

("if" direction): Suppose that there exists a path P from $(s_1's_1, s_2's_2)$ to (t_1t_1', t_2t_2') in G. Let $P = (e_1^0, e_2^0), (e_1^1, e_2^1), \dots, (e_1^r, e_2^r)$ such that $(s_1's_1, s_2's_2) = (e_1^0, e_2^0)$ and $(e_1^r, e_2^r) = (t_1t_1', t_2t_2')$. We construct two edge-disjoint T-compatible paths P_1, P_2 as follows. First we initialize $P_1 =$ $e_1^0, P_2 = e_2^0$. Then for $i = 0, \ldots, r - 1$, we update P_1 and P_2 according to the following cases:

- Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (1). Then $P_2 \leftarrow P_2 \cdot e_2^{i+1}$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (2). Then $P_1 \leftarrow P_1 \cdot e_1^{i+1}$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (3). Then $P_2 \leftarrow P_2 \cdot e_2^{i+1}$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (4). Then $P_1 \leftarrow P_1 \cdot e_1^{i+1}$.

By the definition of edges of \tilde{G} , we get that P_1 and P_2 are two *T*-compatible edge-disjoint paths in G such that P_i is from s'_i to t'_i for i = 1, 2. We can construct a graph \tilde{G} in $O(|E|^3)$ time and find a path from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in $O(|E|^3)$ time. Thus the lemma holds.

Thus we can also decide whether or not to remove an edge of type (iii) from \mathcal{G} in polynomial time and let $\hat{\mathcal{G}}$ be the remaining subgraph of \mathcal{G} . The following lemma shows that we can

reduce edge-disjoint version of 2-DSPP WITH TRANSITION RESTRICTIONS to finding a path from (s'_1s_1, t'_2t_2) to $(t_1t'_1, s_2s'_2)$ in $\hat{\mathcal{G}}$.

Lemma 13. There is a directed path in $\hat{\mathcal{G}}$ from (s'_1s_1, t'_2t_2) to $(t_1t'_1, s_2s'_2)$ if and only if G has two edge-disjoint T-compatible paths P_1 and P_2 such that P_i is from s'_i to t'_i and $P_i \subseteq E_i$ for i = 1, 2.

Proof. ("if" direction) Suppose that G has two edge-disjoint T-compatible paths P_1 and P_2 such that P_i is from s'_i to t'_i and $P_i \subseteq E_i$ for i = 1, 2. $E(P_1) \setminus E_0$ forms a directed path P_1^* in G^* from s'_1 to t'_1 . $\overline{E(P_2)} \setminus E_0$ forms a directed path P_2^* in G^* from t'_2 to s'_2 . Let $P_1^* = e_1^0, e_1^1, \dots, e_1^{p+1}$ and $e_1^0 = s'_1 s_1, e_1^{p+1} = t_1 t'_1$. Let $P_2^* = e_2^0, e_2^1, \dots, e_2^{q+1}$ and $e_2^0 = t'_2 t_2, e_2^{q+1} = s_2 s'_2$. It follows that $e_1^i \in E_1^*$ for $i = 0, 1, \dots, p+1$ and $e_2^j \in E_2^*$ for $j = 0, 1, \dots, q+1$. By the proof of Lemma 10 (interested readers could refer to the proof of Lemma 8 in [9]), there is a directed path P in \mathcal{G} from $(s'_1 s_1, t'_2 t_2)$ to $(t_1 t'_1, s_2 s'_2)$ such that every edge of P is of one of the three types: (i) from (e_1^i, e_2^j) to (e_1^i, e_2^{j+1}) ($i \in \{0, \dots, p+1\}, j \in \{0, \dots, q\}$); (ii) from (e_1^i, e_2^j) to (e_1^{i+1}, e_2^j) ($i \in \{0, \dots, p\}, j \in \{0, \dots, q+1\}$); (iii) from (e_1^i, e_2^j) to (e_1^{i+1}, e_2^j) ($i \in \{0, \dots, p\}, j \in \{0, \dots, q+1\}$); Compatible, by the rules we construct $\hat{\mathcal{G}}$, we can see that all edges of P in \mathcal{G} remains in $\hat{\mathcal{G}}$. This completes the proof for "if direction".

("only if" direction) Suppose that there is a directed path P from $(e_1^0, e_2^0) = (s_1's_1, t_2't_2)$ to $(e_1^r, e_2^r) = (t_1t_1', s_2s_2')$ in $\hat{\mathcal{G}}$ that goes through $(e_1^0, e_2^0), (e_1^1, e_2^1), \ldots, (e_1^r, e_2^r)$ consecutively. Since $\hat{\mathcal{G}}$ is a subgraph of \mathcal{G} , by Lemma 10, there exists two edge-disjoint paths P_1 and P_2 in \mathcal{G} such that P_i is from s_i' to t_i' and $P_i \subseteq E_i$ for i = 1, 2 in \mathcal{G} . Moreover, again from the proof of Lemma 10, it follows that $e_1^i \in E(P_1)$ and $\overline{e_2^i} \in E(P_2)$. By the rule we construct $\hat{\mathcal{G}}$, for an edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) $(i \in \{0, \ldots, r-1\})$, there is a T-compatible subpath of P_1 from $tail_{\mathcal{G}}(f(e_1^i))$ to $head_{\mathcal{G}}(f(e_1^{i+1}))$ if $e_1^i \neq e_1^{i+1}$ or there is a T-compatible subpath of P_2 from $tail_{\mathcal{G}}(f(e_2^{i+1}))$ to head_ $\mathcal{G}(f(e_2^i))$ if $e_2^i \neq e_2^{i+1}$. It follows that P_1 and P_2 are also T-compatible. This finishes the proof for "only if" direction.

Since $\hat{\mathcal{G}}$ is a subgraph of \mathcal{G} and \mathcal{G} contains at most $|E|^2$ vertices, we can detect a path in $\hat{\mathcal{G}}$ in polynomial time. Thus Lemma 13 shows that we can solve edge-disjoint version of 2-DSPP WITH TRANSITION RESTRICTIONS in polynomial time assuming that every cycle in the input graph has positive length.

5.2 Vertex-disjoint case of 2DSPP

When computing vertex-disjoint version of 2-DSPP in the paper of Bérczi and Kobayashi [9], they create a new digraph G_2 as follows: for every vertex $v \in V$ create two vertices v^+ and v^- . Create an edge v^-v^+ with $w(v^-v^+) = 0$. Create an edge u^+v^- if there is an edge uv in G and let $w(u^+v^-) = w(uv)$. Thus VERTEX-DISJOINT 2-DSPP in G is reduced to edge-disjoint variant of 2-DSPP in G_2 . However, this method does not work in the forbidden-transitions setting because part of the information of transitions will be lost after creating the new graph G_2 .

In order to keep the information of transitions, we first modify G as follows. We compute the set E_1 and E_2 of G. Remove all edges of $E(G) \setminus (E_1 \cup E_2)$ from E(G) and all isolated vertices from V(G). When removing the edges or vertices we update the transition system accordingly. Then create four new vertices s'_1, s'_2, t'_1, t'_2 and four edges $s'_1s_1, s'_2s_2, t_1t'_1, t_2t'_2$ all with length 0. Add s'_is_i and $t_it'_i$ to E_i for i = 1, 2. Thus a shortest path from s_i to t_i corresponds to a shortest path from s'_i to t'_i starting with the edge s'_is_i and ending with the edge $t_it'_i$. We update $T_G(s_i)$ by adding $\{\{e, e'\} \mid e = s'_is_i$ and $tail_G(e') = s_i\}$ to it for i = 1, 2. Let $T_G(t_i) = \{\{e, e'\} \mid head_G(e) = t_i$ and $e' = t_it'_i\}$ for i = 1, 2. Then we create a graph G' as follows. For every vertex $v \in V(G) \setminus \{s'_1, s'_2, t'_1, t'_2\}$, create two vertices v^+ and v^- . We also create four vertices s'_1, s'_2, t'_1, t'_2 in G' and create four edges $s'_1s_1^-, s'_2s_2^-, t_1^+t'_1, t_2^+t'_2$ in G' all with length 0. For every vertex $v \in V(G) \setminus \{s'_1, s'_2, t'_1, t'_2\}$, let $in_1(v), \ldots, in_{r_v}(v)$ be the incoming edges of v. Then create r_v parallel edges $e_1(v), \ldots, e_{r_v}(v)$ with $tail_{G'}(e_j(v)) = v^-$ and $head_{G'}(e_j(v)) = v^+$ in G' for $j = 1, \ldots, r_v$ such that each of the edges is of length 0. If there is an edge $uv = in_p(v)$ in G for some $p \in [r_v]$ and $u, v \notin$ $\{s'_1, s'_2, t'_1, t'_2\}$, create an edge $in_p(v^-) = u^+v^-$ in G' and let $w(u^+v^-) = w(uv)$. Next, we define the transition system for G' as follows. $T_{G'}(v^-) = \{\{in_j(v^-), e_j(v)\} \mid j \in [r_v]\}$. For every $e, e' \in (E_1 \cup E_2) \setminus \{t_1t'_1, t_2t'_2\} \subseteq E(G)$ such that $e = uv = in_p(v), e' = vw$ (let $\hat{e} = v^+w^-$), if $\{e, e'\} \in T_G(v)$, then $\{e_p(v), \hat{e}\} \in T_{G'}(v^+)$. In particular, let $e_i = t_i^+t'_i$ for i = 1, 2. If $e = ut_i = in_q(t_i) \in E(G)$ for some $q \in [r_{t_i}]$, then $\{e_q(t_i), e_i\} \in T_{G'}(t_i^+)$.

We also need to compute the set of edges E'_i that exist in some shortest path (without transitions) from s'_i to t'_i for i = 1, 2. By this definition, obviously $s'_i s_i^-, s_i^- s_i^+, t_i^- t_i^+, t_i^+ t'_i \in E'_i$ for i = 1, 2.

Lemma 14. For $u, v \in V(G) \setminus \{s'_1, s'_2, t'_1, t'_2\}$, $uv \in E_i$ if and only if $u^+v^- \in E'_i$ for i = 1, 2. Moreover, if some incoming edge of v^- belongs to E'_i , then all of the parallel edges v^-v^+ belong to E_i for i = 1, 2.

Proof. Suppose that $P_1 = s_1, w, ..., u, v, ..., t_1$ is a shortest path from s_1 to t_1 in G. We claim that $P'_1 = s'_1, s^-_1, s^+_1, w^-, w^+, ..., u^-, u^+, v^-, v^+, ..., t^-_1, t^+_1, t'_1$ is a shortest path from s'_1 to t'_1 in G'. For contradiction, suppose the claim is not true. Then we can find a path $P'_0 = s'_1, s^-_1, s^+_1, w^-_1, w^+_1, ..., w^-_\ell, w^+_\ell, t^-_1, t^+_1, t'_1$ in G' such that $w(P'_0) < w(P'_1) = w(P_1)$. Then there is a path $P_0 = s_1, w_1, ..., w_\ell, t_1$ in G such that $w(P_0) = w(P'_0) < w(P_1)$, contradicting that P_1 is a shortest path from s_1 to t_1 .

Suppose that $P'_1 = s'_1, s_1^-, s_1^+, w^-, w^+, \dots, u^-, u^+, v^-, v^+, \dots, t_1^-, t_1^+, t_1'$ is a shortest path from s'_1 to t'_1 in G'. We claim that $P_1 = s_1, w, \dots, u, v, \dots, t_1$ is a shortest path from s_1 to t_1 in G. For contradiction, suppose that the claim is not true. Then there exists a path $P_0 = s_1w_1\dots w_\ell t_1$ in G such that $w(P_0) < w(P_1) = w(P'_1)$. Thus there is a path $P'_0 = s'_1, s_1^-, s_1^+, w_1^-, w_1^+, \dots, w_\ell^-, w_\ell^+, t_1^-, t_1^+, t_1'$ in G' such that $w(P'_0) = w(P_0) < w(P'_1)$, contradicting that P'_1 is a shortest path from s'_1 to t'_1 in G'.

Similarly we can show that $P_2 = s_2, w, ..., u, v, ..., t_2$ is a shortest path from s_2 to t_2 in G if and only if $P'_2 = s'_2, s_2^-, s_2^+, w^-, w^+, ..., u^-, u^+, v^-, v^+, ..., t_2^-, t_2^+, t'_2$ is a shortest path from s'_2 to t'_2 in G'. It follows that for $u, v \in V(G) \setminus \{s'_1, s'_2, t'_1, t'_2\}, uv \in E_i$ if and only if $u^+v^- \in E'_i$ for i = 1, 2.

For i = 1, 2, as $w(v^-v^+) = 0$, we have that $d_i(v^+) = d_i(v^-) + w(v^-v^+)$. Since some ingoing edge of v^- belongs to E'_i , there is a $v^-t'_i$ path in E'_i . It follows that there is also a $v^+t'_i$ path in E'_i . By the definition of E'_i , all of the parallel edges v^-v^+ belong to E'_i .

It's not hard to verify that Lemma 8 and Lemma 9 also apply to G', but we will also state them here for clarity.

Lemma 15. [9] The edge set E'_i forms no dicycle in G' for i = 1, 2.

Lemma 16. [9] In the graph G', suppose that C is a dicycle in $E'_1 \cup \overline{E'_2}$. Then $E'_1 \cap E(C) \subseteq E'_2$ and $E'_2 \cap \overline{E(C)} \subseteq E'_1$.

Let $E'_0 = E'_1 \cap E'_2$, $E^*_1 = E'_1 \setminus E'_0$, $E^*_2 = E'_2 \setminus E'_0$. We contract all edges of E'_0 and get a graph G'' = (V'', E''). For an edge $e \in E''$, let $f(e) \in E(G')$ denote the edge corresponding to e before the contracting operations. We need to compute the new transition system of G'' as follows. Let $V'_0 \subseteq V''$ be the set of vertices that are newly created after contracting E'_0 . For $v \in V'_0$, we use

 G'_v to denote the subgraph of $G' - (E(G') \setminus (E'_1 \cup E'_2))$ induced by the vertices corresponding to v before contracting. For every $u \in V(G'') \setminus V'_0$, if $f(e)f(e') \in T_{G'}(u)$ then $\{e, e'\} \in T_{G''}(u)$. Let $v \in V'_0$ and head $G''(e) = \operatorname{tail}_{G''}(e') = v$. If there is a T-compatible path in the subgraph of G' consisting of all edges of G'_v together with f(e) and f(e') from $\operatorname{tail}_{G'}(f(e))$ to head G'(f(e')), then $\{e, e'\} \in T_{G''}(v)$. By Lemma 15, there is no dicycle in G'_v . Moreover, the subgraph of G'consisting of all edges of G'_v together with f(e) and f(e') is also acyclic. So we can compute $T_{G''}(v)$ for every $v \in V'_0$ in polynomial time according to Lemma 11. Since $E_1^* \cap E_2^* = \emptyset$, then we can reverse all edges of E_2^* (the lengths of edges unchanged) with E_1^* unchanged. We get a new graph $G^* = (V^*, E^*)$, such that $V^* = V''$ and $E^* = E_1^* \cup \overline{E_2^*}$.

Then we also need to compute the new transition systems of G^* . If $e, g \in E_1^*$ and $\{e, g\} \in T_{G''}(v)$ for some $v \in V''$, then $\{e, g\} \in T_{G^*}(v)$. If $e, g \in E_2^*$ and $\{e, g\} \in T_{G''}(v)$ for some $v \in V''$, then $\{\bar{g}, \bar{e}\} \in T_{G^*}(v)$. Here we use $\bar{e}, \bar{g} \in \overline{E_2^*}$ to denote the reverse of e, g respectively.

Claim 4. After reversing the edges of E_2^* , there is no dicycle in G^* .

Proof. Suppose for contradiction that there is a dicycle C in G^* . By Lemma 15, $E(C) \not\subseteq E_1^*, E(C) \not\subseteq \overline{E_2^*}$. It follows that $E(C) \cap E_1^* \neq \emptyset$ and $E(C) \cap \overline{E_2^*} \neq \emptyset$. Then by Lemma 16, E(C) should have been contracted in G'', contradicting that C is a dicycle in G^* .

We define a new digraph \mathcal{G} as follows. Let $W = E_1^* \times \overline{E_2^*}$ be its vertex set. For $(e_1, e_2), (e_1', e_2') \in W$, there is a directed edge from (e_1, e_2) to (e_1', e_2') if one of three cases hold.

- (i) $e_1 = e'_1$, head $G^*(e_2) = \text{tail}_{G^*}(e'_2) = v$ and $\{e_2, e'_2\} \in T_{G^*}(v)$. There is no path from head $G^*(e_1)$ to v in G^* .
- (ii) $e_2 = e'_2$, head_{G*} $(e_1) = \text{tail}_{G^*}(e'_1) = v$ and $\{e_1, e'_1\} \in T_{G^*}(v)$. There is no path from head_{G*} (e_2) to v in G^* .
- (iii) head_{G*}(e_2) = tail_{G*}(e'_2) = head_{G*}(e_1) = tail_{G*}(e'_1) = v and $\{e_1, e'_1\}, \{e_2, e'_2\} \in T_{G*}(v)$. Furthermore, if $v \in V_0$, let G^s be the subgraph of G' consisting of all edges of G'_v together with $f(e_1), f(e'_1), f(\overline{e_2})$ and $f(\overline{e'_2})$. Then G^s contains two T-compatible vertex-disjoint paths such that one path is from tail_{G'}($f(e_1)$) to head_{G'}($f(e'_1)$) and the other path is from tail_{G'}($f(\overline{e'_2})$) to head_{G'}($f(\overline{e_2})$).

In the third case above, we claim that v must belong to V'_0 . Suppose for contradiction that $v \notin V'_0$. Clearly, $v \notin \{s'_1, s'_2, t'_1, t'_2\}$, as in must be both, head and tail of some edges. So there are two remaining cases. The first case is that $v = u^-$ for some $u \in V(G)$. Then all outgoing edges of u^- in G'' are parallel edges, that is, head $_{G''}(\overline{e_2}) = \text{head}_{G''}(e'_1)$. Then e'_1 and e_2 form a cycle in G^* , contradicting that G^* is acyclic. The second case is that $v = u^+$ for some $u \in V(G)$. Then all ingoing edges of u^+ in G'' are parallel edges, that is, $\text{tail}_{G''}(\overline{e'_2}) = \text{tail}_{G''}(e_1)$. Then e_1 and e_2 form a cycle in G^* , contradicting that G^* is acyclic. The second case is that $v = u^+$ for some $u \in V(G)$. Then all ingoing edges of u^+ in G'' are parallel edges, that is, $\text{tail}_{G''}(\overline{e'_2}) = \text{tail}_{G''}(e_1)$. Then e_1 and e'_2 form a cycle in G^* , contradicting that G^* is acyclic. Thus v must belong to V'_0 . Then we need to solve the vertex-disjoint version of 2-DSPP WITH TRANSITION RESTRICTIONS in the acyclic graph $G'_v \cup \{e_1, e'_1, \overline{e_2}, \overline{e'_2}\}$. The following lemma shows that we can do it in polynomial time. The algorithm is an adaption of the algorithm of finding two vertex-disjoint paths in DAG given by Perl and Shiloach [111].

Lemma 17. In a directed acyclic graph G = (V, E) with transition system T_G , we can solve the vertex-disjoint version of 2-DSPP WITH TRANSITION RESTRICTIONS in polynomial time.

Proof. First we modify the graph G as follows. We create four vertices s'_1, s'_2, t'_1, t'_2 and update V(G) as $V(G) \leftarrow V(G) \cup \{s'_1, s'_2, t'_1, t'_2\}$. We create four edges $\{s'_1s_1, s'_2s_2, t_1t'_1, t_2t'_2\}$ and update E(G) as $E(G) \leftarrow E(G) \cup \{s'_1s_1, s'_2s_2, t_1t'_1, t_2t'_2\}$. Also, for i = 1, 2, we update $T_G(s_i)$ as

$$T_G(s_i) \leftarrow T_G(s_i) \cup \{\{e, e'\} \mid e = s'_i s_i \text{ and } \operatorname{tail}_G(e') = s_i\},\$$

and we update $T_G(t_i)$ as

$$T_G(t_i) \leftarrow T_G(t_i) \cup \{\{e, e'\} \mid e' = t_i t'_i \text{ and } \operatorname{head}_G(e) = t_i\}.$$

For every vertex $v \in V(G)$, define the level $\ell(v)$ as the length of a longest directed path in G starting from v. This can be computed by repeatedly removing a vertex of G. Then we create a graph G as follows. Let the vertex set of G be $V(G) = \{(e_1, e_2) \mid e_1, e_2 \in E(G) \text{ and } e_1 \neq e_2\}.$ For every $(e_1, e_2), (e'_1, e'_2) \in V(\tilde{G})$, create an edge from (e_1, e_2) to (e'_1, e'_2) if one of the following cases holds:

- (1) $e_1 = e'_1, \ \ell(\text{head}_G(e_2)) \ge \ell(\text{head}_G(e_1)), \ \{e_2, e'_2\} \in T_G(\text{head}_G(e_2)), \ \text{head}_G(e'_2) \neq \text{tail}_G(e_1)$ and head_G $(e'_2) \neq$ head_G (e_1) .
- (2) $e_2 = e'_2, \ \ell(\text{head}_G(e_1)) \ge \ell(\text{head}_G(e_2)), \ \{e_1, e'_1\} \in T_G(\text{head}_G(e_1)), \ \text{head}_G(e'_1) \neq \text{tail}_G(e_2)$ and head_G(e'_1) \neq head_G(e_2). (3) $e_1 = e'_1 = t_1t'_1$, $\ell(\text{head}_G(e_2)) < \ell(t'_1)$, $\{e_2, e'_2\} \in T_G(\text{head}_G(e_2))$. (4) $e_2 = e'_2 = t_2t'_2$, $\ell(\text{head}_G(e_1)) < \ell(t'_2)$, $\{e_1, e'_1\} \in T_G(\text{head}_G(e_1))$.

We claim that there are two T-compatible vertex-disjoint paths P_1 and P_2 in G such that P_i is

from s'_i to t'_i for i = 1, 2 if and only if there is a path P from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in \tilde{G} . ("only if" direction): Let $P_1 = e_1^0, e_1^1, \dots, e_1^{p+1}$ and $e_1^0 = s'_1s_1, e_1^{p+1} = t_1t'_1$. Let $P_2 = e_2^0, e_2^1, \dots, e_2^{q+1}$ and $e_2^0 = s'_2s_2, e_2^{q+1} = t_2t'_2$. For any $i \in \{0, 1, \dots, p+1\}, j \in \{0, 1, \dots, q+1\}$, such that $(i, j) \neq (p + 1, q + 1)$, one of the following four cases must hold.

- $i \leq p$ and $j \leq q$, $\ell(\text{head}_G(e_1^i)) \leq \ell(\text{head}_G(e_2^j))$, then there is an edge in \tilde{G} from (e_1^i, e_2^j) to $(e_1^i, e_2^{j+1}).$
- $i \leq p$ and $j \leq q$, $\ell(\text{head}_G(e_1^i)) \geq \ell(\text{head}_G(e_2^j))$, then there is an edge in \tilde{G} from (e_1^i, e_2^j) to $(e_1^{i+1}, e_2^j).$
- i = p + 1 and $j \leq q$, $\ell(\text{head}_G(e_2^j)) < \ell(t_1')$, then there is an edge in \tilde{G} from (e_1^{p+1}, e_2^j) to $(e_1^{p+1}, e_2^{j+1}).$
- j = q + 1 and $i \le p$, $\ell(\text{head}_G(e_1^i)) < \ell(t_2')$, then there is an edge in \tilde{G} from (e_1^i, e_2^{q+1}) to (e_1^{i+1}, e_2^{q+1}) .

As a result, there is a path P from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in \tilde{G} . This finishes the proof for "only if" direction.

("if" direction): Suppose that there exists a path P from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in \tilde{G} . Let $P = (e_1^0, e_2^0), (e_1^1, e_2^1), \dots, (e_1^r, e_2^r)$, such that $(e_1^0, e_2^0) = (s_1's_1, s_2's_2)$ and $(e_1^r, e_2^r) = (t_1t_1', t_2t_2')$. We construct two vertex-disjoint T-compatible paths P_1, P_2 as follows. First we initialize $P_1 =$ $e_1^0, P_2 = e_2^0$. Then for $i = 0, \ldots, r - 1$, we update P_1 and P_2 according to the following cases:

- Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (1). Then $P_2 \leftarrow P_2 \cdot e_2^{i+1}$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (2). Then $P_1 \leftarrow P_1 \cdot e_1^{i+1}$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (3). Then $P_2 \leftarrow P_2 \cdot e_2^{i+1}$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (4). Then $P_1 \leftarrow P_1 \cdot e_1^{i+1}$.

By the definition of edges of G, we get that P_1 and P_2 are two T-compatible vertex-disjoint paths in G such that P_i is from s'_i to t'_i for i = 1, 2. We can construct a graph \tilde{G} in $O(|E|^3)$ time and find a path from (s'_1s_1, s'_2s_2) to $(t_1t'_1, t_2t'_2)$ in $O(|E|^3)$ time. Thus the lemma holds.

By the results above, we can construct \mathcal{G} in polynomial time. Now we show that we can solve the vertex-disjoint version of 2-DSPP with transition restrictions in G by finding a path in \mathcal{G} from $(s'_1s_1^-, t'_2t_2^+)$ to $(t_1^+t'_1, s_2^-s'_2)$. Note that $s_i^-, t_i^+ \in V(G')$ might be the endpoints of edges of E'_0 for i = 1, 2. In this case, although we might contract the edges incident to

 $s_i^-, t_i^+ \in V(G')$ and replace these vertices with new vertices, we slightly abuse s_i^-, t_i^+ to denote the vertex adjacent to s'_i, t'_i respectively in G^* for i = 1, 2 for the sake of simplicity.

Lemma 18. There is a directed path in \mathcal{G} from $(s_1's_1^-, t_2't_2^+)$ to $(t_1^+t_1', s_2^-s_2')$ if and only if G' has two vertex-disjoint T-compatible paths P_1 and P_2 such that P_i is from s'_i to t'_i and $P_i \subseteq E'_i$ for i = 1, 2.

Proof. ("if" direction) Suppose that G' has two vertex-disjoint T-compatible paths P_1 and P_2 such that P_i is from s'_i to t'_i and $P_i \subseteq E'_i$ for i = 1, 2. Recall that we contract the edges of E'_0 in G'and reverse the edges of E_2^* in G'' to get G^* . So by the definition of transition systems of G'' and $\underline{G^*}$, the set $E(P_1) \setminus E'_0$ forms a directed T-compatible path P_1^* in G^* from s'_1 to t'_1 , and the set $\overline{E(P_2) \setminus E'_0}$ forms a directed *T*-compatible path P_2^* in G^* from t'_2 to s'_2 . Let $P_1^* = e_1^0, e_1^1, \dots, e_1^{p+1}$ and $e_1^0 = s'_1 s_1^-, e_1^{p+1} = t_1^+ t'_1$. Let $P_2^* = e_2^0, e_2^1, \dots, e_2^{q+1}$ and $e_2^0 = t'_2 t_2^+, e_2^{q+1} = s_2^- s'_2$. It follows that $e_1^i \in E_1^*$ for $i = 0, 1, \dots, p+1$ and $e_2^j \in E_2^*$ for $j = 0, 1, \dots, q+1$. Since G^* is acyclic, for any i = 0, 1, ..., p + 1 and for any j = 0, 1, ..., q + 1, at least one of the following three cases holds.

- (1) There is no directed path from $\text{head}_{G^*}(e_1^i)$ to $\text{head}_{G^*}(e_2^j)$ in G^* .
- (2) There is no directed path from $\text{head}_{G^*}(e_2^j)$ to $\text{head}_{G^*}(e_1^i)$ in G^* .
- (3) head_{G*}(e_1^i) = head_{G*}(e_2^j).

By the definition of \mathcal{G} , the following statements hold.

- If (1) holds and $j \neq q+1$, then \mathcal{G} has an edge from (e_1^i, e_2^j) to (e_1^i, e_2^{j+1}) .
- If (2) holds and $i \neq p+1$, then \mathcal{G} has an edge from (e_1^i, e_2^j) to (e_1^{i+1}, e_2^j) .
- If (3) holds, then \mathcal{G} has an edge from (e_1^i, e_2^j) to (e_1^{i+1}, e_2^{j+1}) .

We can see that if i = p+1 then (1) holds and if j = q+1 then (2) holds. As a result, there is an edge from (e_1^i, e_2^j) to (e_1^{i+1}, e_2^{j+1}) , (e_1^i, e_2^{j+1}) or (e_1^{i+1}, e_2^{j+1}) in \mathcal{G} if $(i, j) \neq (p+1, q+1)$. It follows that starting from (e_1^i, e_2^j) with i = 0, j = 0, we can find a directed path ending at (e_1^{p+1}, e_2^{q+1}) through increasing i by 1, increasing j by 1 or increasing both i and j by 1 iteratively. This concludes the proof for "if direction".

("only if" direction) Suppose that there is a directed path from $(e_1^0, e_2^0) = (s_1's_1^-, t_2't_2^+)$ to $(e_1^r, e_2^r) = (t_1^+ t_1', s_2^- s_2')$ in \mathcal{G} that goes through $(e_1^0, e_2^0), (e_1^1, e_2^1), \dots, (e_1^r, e_2^r)$ consecutively. We construct two T-compatible paths P_1, P_2 in G' as follows. First we initialize $P_1 = e_1^0, P_2 = e_2^0$. Then for i = 0, ..., r - 1, we update P_1 and P_2 according to the following three cases:

- Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (i), namely $e_1^i = e_1^{i+1}$, head_{G*} $(e_2^i) = \text{tail}_{G^*}(e_2^{i+1}) = v$ and $\{e_2^i, e_2^{i+1}\} \in T_{G^*}(v)$. There is no path from head_{G*} (e_1^i) to v in $\tilde{G^*}$. If $v \in V'_0$, let Q be the \tilde{T} -compatible path in the subgraph of G' consisting of all edges of G'_v together with $f(\overline{e_2^i})$ and $f(\overline{e_2^{i+1}})$ from $\operatorname{tail}_{G'}(f(\overline{e_2^{i+1}}))$ to $\operatorname{head}_{G'}(f(\overline{e_2^i}))$. Then $P_2 \leftarrow f(\overline{e_2^{i+1}}) \cdot Q \setminus \{f(\overline{e_2^i}), f(\overline{e_2^{i+1}})\} \cdot P_2$. Otherwise, if $v \notin V'_0$, $P_2 \leftarrow f(\overline{e_2^{i+1}}) \cdot P_2$. • Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (ii), namely $e_2^i = e_2^{i+1}$, $\operatorname{head}_{G^*}(e_1^i) = \operatorname{tail}_{G^*}(e_1^{i+1}) = v$ and $\{e_1^i, e_1^{i+1}\} \in T_{G^*}(v)$. There is no path from $\operatorname{head}_{G^*}(e_2^i)$
- to v in G^* . If $v \in V'_0$, let Q be the T-compatible path in the subgraph of G' consisting of all edges of G'_v together with $f(e_1^i)$ and $f(e_1^{i+1})$ from $\operatorname{tail}_{G'}(f(e_1^i))$ to $\operatorname{head}_{G'}(f(e_1^{i+1}))$. Then $P_1 \leftarrow P_1 \cdot Q \setminus \{f(e_1^i), f(e_1^{i+1})\} \cdot f(e_1^{i+1})$. Otherwise, if $v \notin V'_0$, $P_1 \leftarrow P_1 \cdot f(e_1^{i+1})$. Suppose that the edge from (e_1^i, e_2^i) to (e_1^{i+1}, e_2^{i+1}) is of type (iii), namely $\operatorname{head}_{G^*}(e_1^i) = \operatorname{tail}_{G^*}(e_1^{i+1}) = \operatorname{head}_{G^*}(e_2^i) = \operatorname{tail}_{G^*}(e_1^{i+1}) = v$ and $\{e_1^i, e_1^{i+1}\} \in T_{G^*}(v), \{e_2^i, e_2^{i+1}\} \in T_{G^*}(v)$.
- $T_{G^*}(v)$. If $v \in V'_0$, let G^s be the subgraph of G' consisting of all edges of G'_v together with $f(e_1), f(e'_1), f(\overline{e_2})$ and $f(\overline{e'_2})$. There are two T-compatible vertex-disjoint paths

in G^s , namely Q_1 from $\operatorname{tail}_{G'}(f(e_1^i))$ to $\operatorname{head}_{G'}(f(e_1^{i+1}))$ and Q_2 from $\operatorname{tail}_{G'}(f(\overline{e_2^{i+1}}))$ to $\operatorname{head}_{G'}(f(\overline{e_2^i}))$. Then $P_1 \leftarrow P_1 \cdot Q_1 \setminus \{f(e_1^i), f(e_1^{i+1})\} \cdot f(e_1^{i+1})$ and $P_2 \leftarrow f(\overline{e_2^{i+1}}) \cdot Q_2 \setminus \{f(\overline{e_2^i}), f(\overline{e_2^{i+1}})\} \cdot P_2$. Otherwise, if $v \notin V'_0$, then $P_1 \leftarrow P_1 \cdot f(e_1^{i+1})$ and $P_2 \leftarrow f(\overline{e_2^{i+1}}) \cdot P_2$.

As a result, we construct two vertex-disjoint *T*-compatible paths P_1 and P_2 such that P_i is from s'_i to t'_i and $P_i \subseteq E'_i$ for i = 1, 2. This finishes the proof for "only if" direction.

Since \mathcal{G} contains $O(|E|^3)$ edges, we can detect a path in \mathcal{G} in polynomial time. Thus Lemma 18 shows that the vertex version of 2-DSPP WITH TRANSITION RESTRICTIONS can be solved in polynomial time assuming that every cycle in the input graph has positive length.

Chapter 6

Cluster Editing parameterized above modification-disjoint P_3 -packings

In this chapter, we show the NP-hardness of CLUSTER EDITING ABOVE MODIFICATION-DISJOINT P_3 PACKING (CEAMP) when $\ell = 0$ and give an XP algorithm for CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_3 PACKING (CEATMP). Recall that CEAMP is a variant of CLUSTER EDITING. In CEAMP, we are given a graph G = (V, E), a packing \mathcal{H} of modification-disjoint induced P_3 s (no pair of P_3 s in \mathcal{H} share an edge or non-edge) and an integer ℓ . The task is to decide whether G can be transformed into a union of vertex-disjoint cliques by at most $\ell + |\mathcal{H}|$ modifications (edge deletions or insertions). CEATMP is a variant of CEAMP where every vertex of the input graph belongs to at most two P_3 s of the packing \mathcal{H} .

Organization of this chapter. In Section 6.1, we present an NP-hardness reduction from 3-SAT to CEAMP when $\ell = 0$. In Section 4.2, we give a polynomial-time algorithm for CEATMP when $\ell = 0$ and show an XP algorithm for CEATMP.

6.1 NP-hardness for tight modification-disjoint packings

In this section, we prove Theorem 5 by showing a reduction from the NP-hard problem of deciding satisfiability of 3-CNF formulas. Given a 3-CNF formula Φ , we construct a graph G = (V, E) with a modification-disjoint packing \mathcal{H} of induced P_3s such that Φ has a satisfying assignment if and only if G has a cluster editing set S which consists of exactly one vertex pair of each P_3 in \mathcal{H} . In other words, the CEAMP instance $(G, \mathcal{H}, 0)$ is a yes-instance. We assume that every clause of Φ has exactly 3 literals of pair-wise different variables as we can preprocess the formula to achieve this in polynomial time otherwise. Similarly, we can assume that every variable of Φ appears at least twice. In the following, we let m denote the number of clauses in Φ , denote the clauses of Φ by $\Gamma_0, \ldots, \Gamma_{m-1}$, let n be the number of variables, and denote the variables of Φ by x_0, \ldots, x_{n-1} . Furthermore, we let m_i denote the number of clauses that contain the variable x_i , $i = 0, \ldots, n-1$.

6.1.1 Construction

Before giving the hardness proof, it is instructive to determine some easy and difficult cases when solving CEAMP with $\ell = 0$. This will give us an intuition about the underlying combinatorial problem that we need to solve.

Let $(G, \mathcal{H}, 0)$ be an instance of CEAMP. It is helpful to consider the subgraph G_{fix} of G that contains only those edges of G that are not contained in any P_3 in \mathcal{H} . Suppose that $(G, \mathcal{H}, 0)$ has



Figure 6.1: Four proto-clusters A through E and two P_{3s} in the underlying graph and in the P_{3} -packing that connects A to C via B and C to E via D, respectively. The dashed edge between B and D means that there is a dividing non-edge between B and D.

a solution S and let G_{sol} be the associated cluster graph. Observe that each connected component of G_{fix} is part of a single cluster in G_{sol} . Let us hence call the connected components of G_{fix} *proto-clusters*. Our task in finding G_{sol} is indeed to find a vertex partition \mathcal{P} which is coarser than the vertex partition given by the proto-clusters, and satisfies certain further conditions. The additional conditions herein are given by the P_{3s} in G and also by the non-edges of G which are not contained in any P_3 in \mathcal{H} —let us call such non-edges *dividing*. A dividing non-edge between two proto-clusters implies that these proto-clusters cannot be together in a cluster in G_{sol} . Hence, we are searching for a vertex partition \mathcal{P} as above subject to the constraints that certain proto-cluster pairs end up in different parts.

The constraints on \mathcal{P} given by P_{35} in G can be distinguished based on the intersection of the P_{35} with the proto-clusters. We only want to highlight two situations that are most relevant for the hardness construction. The first situation is when a P_3 , name it P, intersects with three proto-clusters D_1 , D_2 , and D_3 , each in exactly one vertex and with center vertex in D_2 . The corresponding constraint on \mathcal{P} is that either D_1 and D_2 are merged or D_2 and D_3 are merged into one cluster. We can satisfy such constraints easily, in the absence of further constraints, by merging all proto-clusters into one large cluster. However, together with the constraints from dividing non-edges a difficult picture emerges. Consider Fig. 6.1. Proto-clusters B and D cannot be merged into one cluster because of a dividing non-edge. However, there is a path in G from B to D via vertices of C. Hence, either B and C are in different clusters in G_{sol} or C and Dare. If B and C are in different clusters, then since we have only budget one for the P_3 involving A, B, and C, it follows that A and B are merged into one cluster in G_{sol} . It is not hard to imagine that such behavior can be very non-local and in fact two different generalizations of this behavior form the basis for the variable and clause gadget in our hardness reduction.

The second case is when there is a P_3 in G and also in the packing \mathcal{H} that has an edge contained in one proto-cluster A and the remaining vertex in a different proto-cluster B. Call this $P_3 P$. Intuitively, regardless of whether A and B are merged into one cluster in G_{sol} , P can be edited without excess cost over \mathcal{H} to accommodate this choice. In our hardness reduction, a main difficulty will be to pad subconstructions with P_{3s} in the packing \mathcal{H} , so that we are able to find a solution with zero excess edits. For this we will heavily use P_{3s} of the form that we just described.

6.1.2 Construction

Our basic building blocks will be proto-clusters. A proto-cluster is a subgraph that is connected through edges that are not contained in any P_3 in the constructed packing \mathcal{H} . The proto-clusters then have to be joined into larger clusters in a way that represents a satisfying assignment to Φ . The variable gadget basically consists of an even-length cycle of proto-clusters, connected by P_{3s} so that either odd or even pairs of proto-clusters on the cycle have to be merged. These two options represent a truth assignment. The construction of the variable gadget is more involved than a simple cycle of proto-clusters, however, because of the connection to the clause gadgets: We need to ensure that all vertex pairs between certain proto-clusters of a variable and clause gadget are covered by P_{3} s in \mathcal{H} , so to be able to merge these clusters in the completeness proof. The way in which we cover these vertex pairs imposes some constraints on the construction of the variable gadgets, making the gadgets more complicated.

Variable Gadget

As mentioned, a variable will be represented by a cycle of proto-clusters such that any solution needs to merge either each odd or each even pair of consecutive proto-clusters. These two options represent the truth value assigned to the variable. In order to enable both associated solutions with zero edits above the packing lower bound, we build an associated packing of P_{3} s such that all vertex pairs between consecutive proto-clusters are covered by a P_{3} in the packing. Since we later on need to connect the variable gadgets to the clause gadgets, each proto-cluster will contain five vertices, giving us enough attachment points for later.

Recall that m_i denotes the number of clauses that contain the variable x_i , i = 0, 1, ..., n-1. For each variable x_i , i = 0, 1, ..., n-1, we create $4m_i$ vertex-disjoint cliques with 5 vertices each, namely $K_0^i, ..., K_{4m_i-1}^i$. In each K_j^i , $j = 0, 1, ..., 4m_i - 1$, the vertices are $v_{j,0}^i, ..., v_{j,4}^i$. For each $j = 0, 2, ..., 4m_i - 2$, we create P_{3s} connecting K_j^i, K_{j+1}^i and K_{j+2}^i as we explain below (here we identify K_0^i as $K_{4m_i}^i$).

Throughout the construction, the cliques we have just introduced will remain proto-clusters, that is, they contain a spanning tree of edges that are not covered by P_{3} s in the packing \mathcal{H} . We now add pairwise modification-disjoint P_{3} s so as to cover all edges between the cliques K_{j}^{i} we have just introduced. Recall that \mathbb{F}_{5} is the finite field of the integers modulo 5. We take three consecutive cliques and add P_{3} s with one vertex in each of the three cliques. To do this without overlapping two P_{3} s, we think about the cliques' vertices as elements of \mathbb{F}_{5} and add a P_{3} for each possible arithmetic progression. That is, in each added P_{3} the difference of the first two elements of the P_{3} is equal to the difference of the second two elements. In this way, each vertex pair is contained in a single P_{3} since the third element is uniquely defined by the arithmetic progression.

Formally, for every triple of elements $p, q, r \in \mathbb{F}_5$ satisfying the equality q - p = r - q over \mathbb{F}_5 , we add to the graph the edges $v_{j,p}^i v_{j+1,q}^i$ and $v_{j+1,q}^i v_{j+2,r}^i$ and we add to the packing \mathcal{H} the P_3 given by $v_{j,p}^i v_{j+1,q}^i v_{j+2,r}^i$. Note that in this manner the clique K_{j+1}^i becomes fully adjacent to K_j^i and to K_{j+2}^i while K_{j+1}^i stays anti-adjacent to all other cliques $K_{j'}^i$.

Observe that the P_{3s} given by $v_{j,p}^{i}v_{j+1,q}^{i}v_{j+2,r}^{i}$ for $j = 0, 2, \ldots, 4m_{i} - 2$ such that q - p = r - q are pairwise modification-disjoint: For each $j = 0, 2, \ldots, 4m_{i} - 2$, an arbitrary edge just introduced between K_{j}^{i} and K_{j+1}^{i} has the form $\{v_{j,p}^{i}, v_{j+1,q}^{i}\}$ for some $p, q \in \mathbb{F}_{5}$. It belongs to the unique P_{3} given by $v_{j,p}^{i}v_{j+1,q}^{i}v_{j+2,r}^{i}$, where r = 2q - p. Similarly, an arbitrary edge $\{v_{j+1,q}^{i}, v_{j+2,r}^{i}\}$ for $q, r \in \mathbb{F}_{5}$ belongs to the unique P_{3} given by $v_{j,2q-r}^{i}v_{j+1,q}^{i}v_{j+2,r}^{i}$ and an arbitrary non-edge $\{v_{j,p}^{i}, v_{j+2,r}^{i}\}$ for $p, r \in \mathbb{F}_{5}$ belongs to the unique P_{3} given by $v_{j,p}^{i}v_{j+1,(p+r)\cdot 2^{-1}}v_{j+2,r}^{i}$, where 2^{-1} is the multiplicative inverse of 2 over \mathbb{F}_{5} , that is, $2^{-1} = 3$.

After this construction, we set the modification-disjoint packing of the variable gadget to be

$$\mathcal{H}_{\mathsf{var}} = \{ P_3 \text{ given by } v_{j,p}^i v_{j+1,q}^j v_{j+2,r}^i \mid i = 0, \dots, n-1; j = 0, 2, \dots, 4m_i - 2; p, q, r \in \mathbb{F}_5; \text{ and } q-p = r-q \}.$$

This finishes the first stage of the construction. Notice that the cliques K_j^i form a cyclic structure. Intuitively, every second pair of cliques needs to be merged into one cluster by any solution due to the P_{3s} we have introduced, and we will see that the two resulting solutions are in fact



Figure 6.2: Skeleton of a clause gadget $\Gamma_d = (x_a \vee \neg x_b \vee \neg x_c)$. The white circles represent cliques. The blue dotted lines inside Q_d^2 and Q_d^3 indicate that Q_d^1 , Q_d^2 , Q_d^3 and Q_d^4 are in one connected component. A pair of incident brown thick lines indicates a set of four transferring P_{3s} used to connect a clause gadget to a variable gadget. The cycles made from cliques and gray thick lines represent variable gadgets, where a dashed gray line indicates an omitted part of the cycle. The cycle for variable x_a is shown completely, where we assume that $m_a = 3$, that is, variable x_a is in three clauses. Labels **T** and **F** on thick gray edges indicate the pairs of cliques that shall be merged into one cluster if the variable is to be set to **true** or **false**, respectively.

the only ones. The truth values of the variable are then represented as follows. For every variable x_i , i = 0, ..., n - 1, if K_j^i and K_{j+1}^i are merged for $j = 0, ..., 4m_i - 2$, then this represents the situation that we assign false to the variable x_i . If K_{j+1}^i and K_{j+2}^i are merged for $j = 0, ..., 4m_i - 2$, then this represents variable x_i being true. We will make minor modifications to the variable gadgets and \mathcal{H}_{var} in the following section, so as to transmit the choice of truth value to the clause gadgets.

Skeleton of the Clause Gadget

In order to introduce the construction of the clause gadget, we first give a description of the skeleton of the clause gadget. The skeleton is a subgraph of the final construction that allows us to prove the soundness. The final construction is given in the succeeding sections. We give a picture of the skeleton in Fig. 6.2. The basic idea is as follows: A clause Γ_d is represented by four proto-clusters (cliques), Q_d^i , $i = 1, \ldots, 4$, as in Fig. 6.2. The proto-clusters are connected by a path P containing vertices of Q_d^1 , Q_d^2 , Q_d^3 , and Q_d^4 in that order. However, between Q_d^1 and Q_d^4 there is a dividing non-edge, a non-edge that is not contained in any P_3 in the packing, meaning that every solution has to cut the path P by deleting all edges between Q_d^1 and Q_d^2 , or between Q_d^2 and Q_d^3 , or between Q_d^3 and Q_d^4 . We use this three-way choice to force the solution to select a variable that satisfies the clause Γ_d .

Main Gadget Formally, for each variable x_i , i = 0, 1, ..., n - 1, we fix an arbitrary ordering of the clauses that contain x_i . If a clause Γ_j contains x_i , let $\pi(i, j) \in \{0, ..., m_i - 1\}$ denote the position of the clause Γ_j in this ordering. Let initially $\mathcal{H}_{tra} = \emptyset$. For each clause Γ_d (d = 0, ..., m - 1) proceed as follows. We first introduce four cliques Q_d^1, Q_d^2, Q_d^3 and Q_d^4 . Let Γ_d contain the variables x_a, x_b and x_c . We introduce the cliques T_d^a, T_d^b and T_d^c , called *transferring cliques*. All of the cliques introduced are pairwise vertex disjoint and can be of different sizes. We will give the exact sizes later.

Next, we introduce the following P_{3s} :

- Introduce two P_{3s} , P_d^1 and P_d^2 , both connecting T_d^a and Q_d^2 via Q_d^1 , such that P_d^1 and P_d^2 share the same vertex in Q_d^1 .
- Introduce two P_{3} s, P_{d}^{3} and P_{d}^{4} , both connecting T_{d}^{b} and Q_{d}^{2} via Q_{d}^{3} , such that P_{d}^{3} and P_{d}^{4} share the same vertex in Q_{d}^{3} .
- Introduce two P_{3s} , P_d^5 and P_d^6 , both connecting T_d^c and Q_d^3 via Q_d^4 , such that P_d^5 and P_d^6 share the same vertex in Q_d^4 .

All the $P_{3s} P_d^i$ are pairwise vertex-disjoint except for the pairs sharing the center (as explicitly mentioned in the description). We add each P_d^i for i = 1, ..., 6 to \mathcal{H}_{tra} . We call the P_{3s} of \mathcal{H}_{tra} transferring P_{3s} .

Connection to the Variable Gadgets Next we connect the transferring cliques T_a^a , T_b^a , and T_d^c to the variable gadgets of x_a , x_b , and x_c , respectively. To avoid additional notation, we only explain the procedure for T_d^a and x_a , the other pairs are connected analogously. We connect T_d^a to the variable gadget of x_a by a set of four modification-disjoint P_{3s} as shown in Fig. 6.3 and explained formally below. The centers of these P_{3s} are in $K_{4\pi(a,d)+1}^a$. For each of these four P_{3s} , exactly one endpoint is an arbitrary distinct vertex in T_d^a which is different from the endpoints of the P_{3s} connecting T_d^a to Q_d^1 ; we denote these endpoints as w_1, w_2, w_3, w_4 . The other endpoint is in $K_{4\pi(a,d)+2}^a$ if x_a appears positively in Γ_d and the other endpoint is in $K_{4\pi(a,d)+2}^a$ or the precise centers and endpoints in the cliques $K_{4\pi(a,d)+2}^a$ or $K_{4\pi(a,d)}^a$ are specified below. Since these newly introduced P_{3s} use edges that belong to some P_{3s} in \mathcal{H}_{var} that were introduced while constructing the variable gadgets, we will remove such P_{3s} in the variable gadget from \mathcal{H}_{var} , remove their corresponding edges from the graph, and add some new P_{3s} to \mathcal{H}_{var} as described below. As a result, the clique $K_{4\pi(a,d)+1}^a$ may no longer be fully adjacent to $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+2}^a$. We will however maintain the invariant that each vertex pair between $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+2}^a$ is covered by a P_3 in the packing and that all the P_{3s} of \mathcal{H}_{var} are pairwise modification-disjoint.

Formally, if x_a appears positively in Γ_d , we denote:

$$\begin{aligned} v_1 &= v^a_{4\pi(a,d)+1,0} & v_2 &= v^a_{4\pi(a,d)+1,1} \\ v_3 &= v^a_{4\pi(a,d)+2,1} & v_4 &= v^a_{4\pi(a,d)+2,2} \\ v_5 &= v^a_{4\pi(a,d),0} & v_6 &= v^a_{4\pi(a,d),1} \\ v_7 &= v^a_{4\pi(a,d),3} & v_8 &= v^a_{4\pi(a,d),4}. \end{aligned}$$

If x_a appears negatively in Γ_d , we swap the roles of $K^a_{4\pi(a,d)}$ and $K^a_{4\pi(a,d)+2}$, that is:

$$\begin{aligned} v_1 &= v^a_{4\pi(a,d)+1,0} & v_2 &= v^a_{4\pi(a,d)+1,1} \\ v_3 &= v^a_{4\pi(a,d),1} & v_4 &= v^a_{4\pi(a,d),2} \\ v_5 &= v^a_{4\pi(a,d)+2,0} & v_6 &= v^a_{4\pi(a,d)+2,1} \\ v_7 &= v^a_{4\pi(a,d)+2,3} & v_8 &= v^a_{4\pi(a,d)+2,4}. \end{aligned}$$



Figure 6.3: Connection of a clause gadget with a variable gadget for a variable x_a which appears positively in the clause. White ellipses represent cliques. The vertices in the cliques in the variable gadget are ordered from top to bottom according to the elements of \mathbb{F}_5 which they represent. For example, the topmost vertex in $K^a_{4\pi(a,d)}$ is $v^a_{4\pi(a,d),0}$ (corresponding to $0 \in \mathbb{F}_5$) and the bottom-most is $v_{4\pi(a,d),4}^a$ (corresponding to $4 \in \mathbb{F}_5$). The gray lines adjacent to cliques in the variable gadget represent some of the P_{3s} that were introduced into the variable gadgets in the beginning. (Some gray lines are super-seeded by edges of other colors.) The P_{3} s represented by the gray lines have the associated arithmetic progression "+0", that is, q - p = r - q = 0in the definition of the P_{3s} . The P_{3s} for the remaining arithmetic progressions are omitted for clarity. In colors red, black, green, and blue we show the P_{3s} that connect the transferring clique T_d^a with the variable gadget of variable x_a . Herein, dotted lines are non-edges and solid lines are edges. Note that these connecting P_{3s} supplant some of the edges of previously present P_{3s} in the variable gadget—the previously present P_3 s are then removed. For example the green P_3 replaces the edge v_2v_3 of the P_3 given by $v_6v_2v_3$ that was previously present. To maintain that each vertex pair between consecutive cliques in the variable gadget is covered by some P_3 in the packing, we add the brown P_{3s} .

As shown in Fig. 6.3, we remove P_{3s} given by $v_8v_1v_3, v_7v_1v_4, v_6v_2v_3, v_5v_2v_4$ from \mathcal{H}_{var} and we remove their corresponding edges from the graph. Then we add the P_{3s} given by $v_5v_6v_2$ and $v_1v_7v_8$ to the graph and to \mathcal{H}_{var} . Finally, we connect T_d^a via $K_{4\pi(a,d)+1}^a$ by adding the P_{3s} given by $w_1v_1v_3, w_2v_2v_4, w_3v_2v_3$, and $w_4v_1v_4$ to the graph and to \mathcal{H}_{tra} . Note that, indeed, each vertex pair between $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+1}^a$ remains covered by a P_3 in the packing after replacing all P_{3s} . This finishes the construction of the skeleton of the clause gadgets.

The intuitive idea behind the connection to the variable gadget and how it is used in the soundness proof is as follows. Recall from above that we need to delete at least one of three sets of edges in the solution, namely the edges between Q_d^1 and Q_d^2 , the edges between Q_d^2 and Q_d^3 , or the edges between Q_d^3 and Q_d^4 . Assume that the edges between Q_d^1 and Q_d^2 are deleted and the variable x_a appears positively in the clause Γ_d as in Fig. 6.2. Because of the constraints imposed by the $P_{3s} P_d^1$ and P_d^2 , cliques T_d^a and Q_d^1 have to be merged in the final cluster graph. Since $K_{4\pi(a,d)+1}^a$ cannot be merged with Q_d^1 (there are no edges between Q_d^1 and $K_{4\pi(a,d)+1}^a$, and no P_{3s} connecting Q_d^1 and $K_{4\pi(a,d)+1}^a$), we have to separate T_d^a from $K_{4\pi(a,d)+1}^a$. Then, the P_{3s} connecting T_d^a with $K_{4\pi(a,d)+2}^a$ force $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+2}^a$ to merge. This means x_a is true and it satisfies the clause Γ_d .

The P_{3} s added so far are indeed sufficient to conduct a soundness proof of the above reduction: They ensure that there exists a satisfying assignment to the input formula provided that there exists an appropriate cluster editing set. However, the completeness is much more difficult: We need to add some more "padding" P_{3} s to the packing (and edges to the graph between the cliques that can be potentially merged) to ensure that a satisfying assignment can always be translated into a cluster-editing set. In other words, if two cliques have the potential to be merged or separated, because of the constraint that $\ell = 0$, every edited edge or non-edge between the vertices of the two cliques must belong to exactly one P_{3} in the packing \mathcal{H} . The goal of the next two sections is to develop a methodology of padding such cliques with P_{3} s in the packing. The padding will rely on the special structure of P_{3} s that we have established above in the clause gadget and connection between clause and variable gadget.

Merging Model of the Clause Gadget

In the sections above, we have defined all proto-clusters of the final constructed graph: As we will see in the correctness proof, each clique will be a proto-cluster in the end. Thus, all solutions will construct a cluster graph whose clusters represent a coarser partition than the partition given by the proto-clusters, or cliques. What remains is to ensure that the proto-clusters indeed can be merged as required to construct a solution from a satisfying assignment to Φ in the completeness proof. To do this, we pad the proto-clusters with P_{3s} (in the graph and packing \mathcal{H}). To simplify this task we now divide the set of proto-clusters into five levels L_0, \ldots, L_4 . Then, we will go through the levels in increasing order and add padding P_{3s} from proto-clusters the current level to the proto-clusters of all lower levels if necessary.

There are two issues that we need to deal with when introducing the padding P_{3s} . For the padding, we will use a number-theoretic tool that we introduce in the following part which has the limitation that, when padding a proto-cluster D with P_{3s} to some sequence D_1, \ldots, D_s of proto-clusters of lower level, we need to increase the number of vertices in D to be roughly $2 \cdot \sum_{i=1}^{s} |D_i|$. Hence, first, we need to make sure that the number of levels is constant since the number of size increases of proto-clusters compounds exponentially with the number of levels. Second, we aim for the property that each vertex is only in a constant number of P_{3s} in \mathcal{H} and thus, we need to ensure that the number s of lower-level proto-clusters and their size is constant.

To achieve the above goals, we introduce an auxiliary undirected graph H, the merging



Figure 6.4: Merging model of a clause $\Gamma_d = (x_a \vee \neg x_b \vee \neg x_c)$. The number $i \in \{0, 1, 2, 3, 4\}$ beside a vertex v denotes that $v \in L_i$. The placement of vertices corresponds to the placement of the cliques in Fig. 6.2. For example, the two vertices on level 1 on the top correspond to Q_d^1 and Q_d^4 . We assume that $m_a = 3$.

model, which will further guide the padding process. The merging model has as vertices the cliques that were introduced before and an edge between two cliques if we want it to be possible that they are merged by a solution. Formally,

$$V(H) := \{ K_j^i \mid i = 0, 1, \dots, n-1 \text{ and } j = 0, 1, \dots, 4m_i - 1 \} \cup \{ Q_d^1, Q_d^2, Q_d^3, Q_d^4 \mid d = 0, 1, \dots, m-1 \} \cup \{ T_s^a \mid \text{variable } x_a \text{ occurs in clause } \Gamma_s \},$$

and the edge set, E(H), is defined as follows. See also Fig. 6.4. First, it shall be possible to merge the cliques in the variable gadget in a cyclic fashion,¹ that is, we add

$$\{\{K_i^i, K_{i+1}^i\} \mid i = 0, 1, \dots, n-1 \text{ and } j = 0, 1, \dots, 4m_i - 1\}$$

to E(H). Second, it shall be possible to merge transferring cliques of clause gadget to any of the relevant cliques of the associated variable gadget, that is, we add to E(H) the set

$$\{\{T_d^i, K_{4\pi(i,d)}^i\}, \{T_d^i, K_{4\pi(i,d)+1}^i\}, \{T_d^i, K_{4\pi(i,d)+2}^i\} \mid \text{variable } x_i \text{ occurs in clause } \Gamma_d\}.$$

Third, it shall be possible to merge subsets of $\{Q_d^1, Q_d^2, Q_d^3, Q_d^4\}$, and hence we add to E(H) the set

$$\{\{Q_d^1, Q_d^2\}, \{Q_d^1, Q_d^3\}, \{Q_d^2, Q_d^3\}, \{Q_d^2, Q_d^4\}, \{Q_d^3, Q_d^4\} \mid d = 0, 1, \dots, m-1\}.$$

¹Indeed, we have already ensured that this is possible. The edges introduced in the first step purely serve to reinforce the intuition of the merging model.

Finally, it shall be possible to merge the transferring cliques to subsets of $\{Q_d^1, Q_d^2, Q_d^3, Q_d^4\}$. Hence, we add to E(H) the set

 $\{\{T_d^i, Q_d^k\} \mid \text{if variable } x_i \text{ occurs in } \Gamma_d \text{ and } T_d^i \text{ is adjacent in } G \text{ to } Q_d^k \text{ with } k \in \{1, 4\}\} \cup$ $\{\{T_d^i, Q_d^3\}, \{T_d^i, Q_d^4\} \mid \text{if variable } x_i \text{ occurs in } \Gamma_d \text{ and } T_d^i \text{ is adjacent in } G \text{ to } Q_d^3\}.$

Note that this construction is slightly asymmetric (see Fig. 6.4). This finishes the definition of the merging model H.

Now we define the levels L_0 to L_4 such that orienting the edges in H from higher to lower level gives an acyclic orientation when ignoring the edges in level L_0 .

- L_0 contains all cliques in variable gadgets.
- L_1 contains Q_d^1 and Q_d^4 for each d = 0, ..., m 1. L_2 contains Q_d^3 for each d = 0, ..., m 1. L_3 contains Q_d^2 for each d = 0, ..., m 1.

- L₄ contains all transferring cliques.

Observe that, apart from edges in L_0 , all edges in H are between vertices of different levels and, indeed, ignoring edges in L_0 , there are no cycles in G when orienting edges from higher to lower level. In the following section, we will look at each clique R in levels L_1 and higher and add P_{3s} to the packing \mathcal{H} so as to cover all vertex pairs containing a vertex of R and a out-neighbor of R in H.

Implementation of the Clause Gadget

In this section, we first introduce a number-theoretical construction (see Lemma 19) that serves as a basic building block for "padding" P_{3s} in the packing. Then we use this construction to perform the actual padding of P_{3s} .

The abstract process of padding P_{3s} works as follows. It takes as input a clique R in H (represented by W in the below Lemma 19), and a set of cliques that are out-neighbors of R in H (represented by V). Furthermore, it receives a set of vertex pairs between R and its out-neighbors that have previously been covered (represented by F). The goal is then to find a packing of P_{3s} that cover all vertex pairs *except* the previously covered pairs. The previously covered vertex pairs have some special structure that we carefully selected so as to make covering of all remaining vertex pairs possible in a general way: The construction so far was carried out in such a way that the connected components induced by previously covered vertex pairs are P_{3} s or C_{8} s.

In Lemma 19 we will indeed pack triangles instead of P_{35} because this is more convenient in the proof. We will replace the triangles by P_{38} afterwards: Recall the intuition that P_{38} in the packing \mathcal{H} which have exactly one endpoint in one clique T and their remaining two vertices in another clique R can accommodate both merging R and T or separating R and T without excess edits. Hence, we will replace the triangles by such P_{3s} . Recall that we aim for each clique to be a proto-cluster in the final construction, that is, each clique contains a spanning tree of edges which are not contained in P_{3s} in \mathcal{H} . Since putting the above kind of P_{3s} into the packing \mathcal{H} allows in principle to delete edges within R, we need to ensure that R remains a proto-cluster. We achieve this via the connectedness property in Lemma 19.

Number-Theoretic Padding Tool.

Lemma 19. Let p be a prime number with $p \ge 2$. Let B = (V, W, E) be a complete bipartite graph such that |V| = p and |W| = 2p. Let $F \subseteq E$ be a set of edges such that each connected



Figure 6.5: Left: The labels of a C_8 in $(V \cup W, F)$. Right: The triangles in τ_F^2 covering a C_8 .

component of $(V \cup W, F)$ is a either a singleton, a P_3 with a center in V, or a C_8 . Then there exists an edge-disjoint triangle packing τ in $(V \cup W, (E \setminus F) \cup {W \choose 2})$ which covers $E \setminus F$ such that every triangle in τ contains exactly one vertex of V and the graph $(W, {W \choose 2} \setminus E(\bigcup \tau))$ is connected. Moreover, each vertex $v \in V \cup W$ is in at most p triangles of τ , it is in at most p-1 triangles if v is in a connected component of $(V \cup W, F)$ that is a P_3 , and in at most p-2triangles if v is in connected component of $(V \cup W, F)$ that is a C_8 .

Proof. First, we divide W into two parts W_1 and W_2 of equal sizes such that if two vertices $w, w' \in W$ are connected to the same vertex $v \in V$ by edges in F, then w and w' are in different parts. Note that this is easy for a connected component of $(V \cup W, F)$ if it is a P_3 . For a connected component of $(V \cup W, F)$ which is a C_8 , this is also doable as shown in Fig. 6.5, where $w_i, w_{i+1}, w_{i+2}, w_{i+3}$ belong to $W_1, w'_i, w'_{i+1}, w'_{i+2}, w'_{i+3}$ belong to W_2 , and $v_i, v_{i+1}, v_{i+2}, v_{i+3}$ belong to V.

We now label the vertices by elements from the finite field \mathbb{F}_p of size p (recall that \mathbb{F}_p consists of the elements $\{0, 1, \ldots, p-1\}$ with addition and multiplication modulo p). To each vertex $v \in V$, each vertex $w \in W_1$, and each vertex $w' \in W_2$, we will assign a unique label v_i, w_j , and w'_k , respectively, with $i, j, k \in \mathbb{F}_p$. In other words, we construct three bijections that map \mathbb{F}_p to V, W_1 , and W_2 , respectively.

First, we label the vertices from the connected components of $(V \cup W, F)$ (and some singleton vertices) by going through the connected components one-by-one. For each yet-unlabeled connected component of $(V \cup W, F)$ that is a P_3 given by wvw' such that $v \in V, w \in W_1, w' \in W_2$, we label vertex w as w_j , vertex v as v_j and vertex w' as w'_j for the smallest j from \mathbb{F}_p which is not yet used in the labeling of vertices of V. For each yet-unlabeled connected component Cin $(V \cup W, F)$ that is a C_8 we proceed as follows. By the way we have divided vertices from Winto W_1 and W_2 , we can assign, to each such connected component C, four vertices which have degree zero in $(V \cup W, F)$: two in W_1 and two in W_2 ; see also Fig. 6.5. We thus label the vertices in C and the four degree-zero vertices assigned to C as in Fig. 6.5, for the smallest integer ifrom \mathbb{F}_p such that i, i + 1, i + 2 and i + 3 are not used in the labeling of vertices of V.

Second, we label the remaining unlabeled vertices that are not in the connected components of $(V \cup W, F)$. For an unlabeled vertex $w \in W_1$, label it as w_k for an arbitrary integer k from \mathbb{F}_p which is not used in the labeling of vertices in W_1 . Similarly, for an unlabeled vertex $v \in V$, we label it as v_h for an arbitrary integer h from \mathbb{F}_p which is not used in the labeling of vertices in V and for an unlabeled vertex $w' \in W_2$, we label it as w'_s for an arbitrary integer s from \mathbb{F}_p which is not used in the labeling of vertices in W_2 . After the labeling, the vertices in V, W_1 and W_2 are $v_1, \ldots, v_{p-1}, w_1, \ldots, w_{p-1}$ and w'_1, \ldots, w'_{p-1} , respectively. We now proceed to constructing the packing τ . First, let

 $\tau_{\mathsf{all}} = \{uvw \mid uvw \text{ is a triangle in } (V \cup W, E \cup \binom{W}{2}) \text{ such that } u \in V, v \in W_1, w \in W_2\}, \text{ and } \tau_{\mathsf{cover}} = \{v_iw_jw'_k \in \tau_{\mathsf{all}} \mid i, j, k \in \mathbb{F}_p \text{ and } j-i=k-j \text{ over } \mathbb{F}_p\}.$

In the following, for any triangle packing τ , by $E(\tau)$ we will denote the union of the edge sets of the triangles in τ .

We claim that the triangles in τ_{cover} are edge-disjoint and cover all edges of E. Consider an arbitrary edge $v_i w_j \in E$ between V and W_1 for $i, j \in \mathbb{F}_p$. According to the definition of τ_{cover} , each triangle $v_i w_j w'_x \in \tau_{\text{cover}}$ that covers edge $v_i w_j$ satisfies x = 2j - i (over \mathbb{F}_p). Since \mathbb{F}_p is a field, there is thus exactly one such triangle. Similarly, each edge $v_h w'_k \in E$ between V and W_1 for some $h, k \in \mathbb{F}_p$ is covered by the unique triangle $v_h w_{(h+k)\cdot 2^{-1}} w'_k \in \tau_{\text{cover}}$. Finally, each edge $w_s w'_t$ between W_1 and W_2 is covered by the unique triangle $v_{2s-t} w_s w'_t \in \tau_{\text{cover}}$. Thus the claim holds.

Let

$$\tau_F^1 = \{ v_h w_h w'_h \in \tau_{\mathsf{all}} \mid \text{vertices } w_h, v_h, w'_h \text{ induce a } P_3 \text{ in } (V \cup W, F) \}, \text{ and}$$

$$\tau_F^2 = \{ v_h w_{h+1} w'_{h+2}, v_{h+1} w_{h+1} w'_{h+1}, v_{h+2} w_{h+2} w'_{h+2}, v_{h+3} w_{h+2} w'_{h+1} \in \tau_{\mathsf{all}} \mid vertices v_h, w'_{h+2}, v_{h+2}, w_{h+2}, v_{h+3}, w'_{h+1}, v_{h+1}, w_{h+1} \text{ induce a } C_8 \text{ in } (V \cup W, F) \}.$$

Observe that $\tau_F^1, \tau_F^2 \subseteq \tau_{\text{cover}}$. For example, if we put $v_{h+3}w_{h+2}w'_{h+1} = v_iw_jw'_k$, then it follows that j-i=p-1=k-j over \mathbb{F}_p , that is, $v_{h+3}w_{h+2}w'_{h+1}$ satisfies the conditions in the definition of τ_{cover} . Moreover, $\tau_F^1 \cup \tau_F^2$ covers all edges of F. Furthermore, each edge in the edge set $E(\tau_F^1 \cup \tau_F^2)$ of $\tau_F^1 \cup \tau_F^2$ is either in F or between W_1 and W_2 . (See also Fig. 6.5.) Thus, $E \setminus F$ has an empty intersection with $E(\tau_F^1 \cup \tau_F^2)$. Let $\tau = \tau_{\text{cover}} \setminus (\tau_F^1 \cup \tau_F^2)$. It follows that τ covers all edges of $E \setminus F$. It remains only to show that τ satisfies the connectedness condition. Since τ_{cover} does not cover any edge of $\binom{W_1}{2}$ or $\binom{W_2}{2}$, it follows that $(W_1, \binom{W_1}{2} \setminus E(\tau))$ and $(W_2, \binom{W_2}{2} \setminus E(\tau))$ are cliques. Now observe that $\tau_F^1 \cup \tau_F^2$ contains at most |V| = p edges of $\binom{W}{2}$, while $W_1 \times W_2$ is of size $p^2 > p$. Thus in the graph $(W, \binom{W}{2} \setminus E(\tau))$ there is at least one edge $\{w_1, w_2\}$ such that $w_1 \in W_1$ and $w_2 \in W_2$. As a result, $(W, \binom{W}{2} \setminus E(\tau))$ is connected. Finally, observe that each vertex $v \in V \cup W$ is in at most p triangles in τ_{cover} . If v is in a P_3 of $(V \cup W, F)$, then at least two of the triangles in τ_{cover} that contain v are removed to obtain τ . This concludes the proof.

The following corollary is slightly easier to apply than Lemma 19.

Corollary 2. Let p be a prime and let B = (V, W, E) be a complete bipartite graph with $|V| \leq p, |W| = 2p$. Let $F \subseteq E$ be a nonempty set of edges such that every connected component of $(V \cup W, F)$ is a either a P_3 with a center in V or a C_8 . Then there exists an edge-disjoint triangle packing τ in $(V \cup W, E \setminus F \cup {W \choose 2})$ which covers $E \setminus F$ such that every triangle in τ contains exactly one vertex of V and $(W, {W \choose 2} \setminus E(\tau))$ is connected. Each vertex $v \in V \cup W$ is in at most p triangles of τ , at most p - 1 if v is in a connected component of $(V \cup W, F)$ that is a P_3 , and at most p - 2 if v is in connected component of $(V \cup W, F)$ that is a C_8 .

Proof. Add extra p - |V| dummy vertices to V, obtaining a complete bipartite graph B' = (V', W, E), apply Lemma 19 to B', p, and F, obtaining a packing τ' , and return a sub-packing $\tau \subseteq \tau'$ containing only triangles with vertices in B. Since every triangle in τ' contains exactly one vertex of V', τ satisfies all the required properties. \Box

Concluding the Construction. Equipped with Lemma 19 and Corollary 2, we can finish the construction of the clause gadgets and indeed the whole instance $(G, \mathcal{H}, 0)$ of CEAMP. We now specify the exact size of each clique introduced above and add padding P_{3s} to G and \mathcal{H} so as to cover all vertex pairs between cliques that are adjacent in the merging model H. Put initially the set \mathcal{H}_{pad} of padding P_{3s} to be $\mathcal{H}_{pad} = \emptyset$. We start with levels 0 and 1. We do not change the sizes of any clique on these two levels. That is, as shown in the variable gadget, there are five vertices in every clique of level 0, and there is one vertex in every clique of level 1. Note that no cliques of levels 0 and 1 are adjacent in the merging model H, that is, no two of them need to be merged in the solution. Hence, it is not necessary to add padding P_{3s} within these levels.

Now we turn each level $i, i \geq 2$, in order of increasing i. For each clique Q of level i, we apply Corollary 2 in the following scenario. Let V be the union of all cliques of levels j < i that are out-neighbors of Q in the merging model H. Let p be the smallest prime with $p \geq |V|$ and $2p \geq |Q|$. Introduce 2p - |Q| new vertices, put them into Q, and make Q a clique. Put W = Q and let $E = \{\{u, v\} \mid u \in V, v \in W\}$. Let F be the set of vertex pairs that each contain one vertex of W and one of V and that each are contained in the transferring P_{3} s (the P_{3} s in \mathcal{H}_{tra}) between W and V. Note that \mathcal{H}_{tra} contains each edge that has been introduced into G so far and that is between two cliques of which one is of level at least two.

We claim that Corollary 2 is applicable to p, graph B = (V, W, E), and F. To see this, we need to show that each connected component in $(V \cup W, F)$ is either a P_3 with center in V or a C_8 . Indeed, if Q is not a transferring clique, that is, $Q = Q_d^j$ for some $d \in \{0, 1, \ldots, m-1\}$ and $j \in \{1, 2, 3, 4\}$, then each connected component in $(V \cup W, F)$ consists of two edges of two different transferring P_{38} with the same center in V, as claimed (see also Fig. 6.4). If Q is a transferring clique, then each connected component of $(V \cup W, F)$ consists either of two edges of two different transferring P_{38} with the same center in some $Q_d^j \subseteq V$ for some $j \in \{1, 3, 4\}$, or of some vertex pairs of transferring P_{38} between Q and the cliques of a variable gadget. In the first case, the claim clearly holds. In the second case, observe that the edges and non-edges between V and W in the transferring P_{38} are each incident with one of w_1, w_2, w_3, w_4 and one of v_1, v_2, v_3, v_4 as defined when connecting variable and clause gadgets. These edges and nonedges indeed induce a C_8 given by $v_1w_1v_3w_3v_2w_2v_4w_4v_1$ (see also Fig. 6.3). Thus, Corollary 2 applies.

Corollary 2 gives us an edge-disjoint triangle packing τ in $(V \cup W, E \setminus F \cup {W \choose 2})$ which covers all edges of $E \setminus F$ such that $(W, {W \choose 2} \setminus E(\tau))$ is connected. Every triangle $vw_1w_2 \in \tau$ has one vertex $v \in V$ and two vertices $w_1, w_2 \in W$. For every triangle $vw_1w_2 \in \tau$, we add a P_3 to G by using exactly two edges of the triangle in G; more precisely, we put $\{v, w_1\}, \{w_1, w_2\} \in E(G), vw_2 \notin$ E(G), and then add the P_3 of G given by vw_1w_2 into \mathcal{H}_{pad} . Finally, let $\mathcal{H} = \mathcal{H}_{var} \cup \mathcal{H}_{tra} \cup \mathcal{H}_{pad}$. Note that \mathcal{H} is a modification-disjoint packing of P_3 s: This is by construction for $\mathcal{H}_{var} \cup \mathcal{H}_{tra}$ and, by Corollary 2, no P_3 in \mathcal{H}_{pad} shares a vertex pair with any P_3 in $\mathcal{H}_{var} \cup \mathcal{H}_{tra}$. This concludes the construction of the CEAMP instance $(G, \mathcal{H}, 0)$.

To see that the construction takes polynomial time and to see that indeed each vertex is in some constant number of P_{3} s in \mathcal{H} , let us now derive the precise sizes of each clique in the construction. Recall that the cliques on level 0 are exactly those in the variable gadgets, and these have exactly five vertices each. The cliques on level 1 are Q_d^1 and Q_d^4 for $d \in \{0, 1, \ldots, m-1\}$, and they have 1 vertex each. On level 2 we have the cliques Q_d^3 , $d \in \{0, 1, \ldots, m-1\}$, and since the only out-neighbor in H of Q_d^3 is Q_d^4 , our procedure sets p = 2 and thus Q_d^3 has 4 vertices. On level 3 there are the cliques Q_d^2 , $d \in \{0, 1, \ldots, m-1\}$, and we set p = 7 as $|Q_d^1 \cup Q_d^3 \cup Q_d^4| = 6$. Thus clique Q_d^2 has 14 vertices. For the clique T_d^a , we set p = 17 as $|Q_d^1 \cup K_{4\pi(a,d)}^a \cup K_{4\pi(a,d)+1}^a \cup K_{4\pi(a,d)+2}^a| = 16$. So the clique T_d^a has $2 \cdot 17 = 34$ vertices. Similarly, T_d^c has 34 vertices as well. For the clique T_d^b , we set p = 23, as $|Q_d^3 \cup Q_d^4 \cup K_{4\pi(b,d)}^b \cup K_{4\pi(b,d)+1}^b \cup K_{4\pi(b,d)+2}^b| = 20$. Thus T_d^b is a clique of size $2 \cdot 23 = 46$. By the bounds on the number of triangles in the packing, each vertex is in at most 23 P_{3} s of \mathcal{H} . It also follows that construction takes overall polynomial time.

6.1.3Correctness

We now prove the correctness of the reduction given in Section 6.1.2

Completeness

Now we show how to translate a satisfying assignment of Φ into a cluster editing set of size $|\mathcal{H}|$ for the constructed instance.

Lemma 20. If the input formula Φ is satisfiable, then the constructed instance $(G, \mathcal{H}, \ell = 0)$ is a yes-instance.

Proof. Assume that there is a satisfying assignment α of the formula Φ . Recall that n is the number of variables of Φ and m is the number of clauses of Φ . Instead of building the solution directly, we build a partition \mathcal{P} of V(G) into clusters. Then, we argue that the number of edges between clusters and the number of non-edges inside clusters is at most $|\mathcal{H}|$. Thus, the partition \mathcal{P} will induce a solution with the required number of edge edits.

The basic building blocks of our vertex partition \mathcal{P} are the cliques in V(H). We will never separate such a clique during building \mathcal{P} , that is, \mathcal{P} corresponds to a partition of V(H). We build \mathcal{P} by taking initially $\mathcal{P} = V(H)$ and then successively merging clusters in \mathcal{P} , which means to take the clusters out of \mathcal{P} and replace them by their union. Since there are no non-edges inside any of the cliques in V(H), below it suffices to consider edges and non-edges between pairs of cliques in V(H) to determine the number of edits in the solution corresponding to \mathcal{P} .

We start with the variable gadgets. Consider each variable x_i , i = 0, 1, ..., n-1. Call a pair of cliques K_{i}^{i} , K_{i+1}^{i} in x_{i} 's variable gadget even if j is even and odd otherwise (indices are taken modulo $4m_i$). If $\alpha(x_i) = \text{true}$, then merge each odd pair. If $\alpha(x_i) = \text{false}$, then merge each even pair. We will not merge any further pair of cliques contained in variable gadgets.

Now consider each clause Γ_d , $d = 0, \ldots, m - 1$, in some arbitrary order. Let x_a , x_b , and x_c be the variables in Γ_d . We use the same notation as when defining the clause gadgets. See Fig. 6.2 for the skeleton of the clause gadget of Γ_d , up to variables appearing positively instead of negatively or vice versa. We choose an arbitrary variable that satisfies Γ_d . The basic idea is to cut (that is, to not merge) the transferring clique and the cliques in the satisfying variable's gadget, cutting some edges of the transferring P_{3s} . This is will induce at most one edit for each transferring P_3 since the remaining edge in a transferring P_3 will be part of a cluster in \mathcal{P} . Then we cut from the clause gadget all transferring cliques belonging to variables that have not been chosen. Since we do not spend edits inside of transferring P_{3s} in this way, this allows us to merge the transferring cliques to the variable gadgets regardless of whether the variable was set to true or false.

Formally, we perform the following merges in \mathcal{P} .

If we have chosen x_a from the variables satisfying the clause Γ_d :

- Merge T^a_d with Q¹_d.
 Merge the cliques Q²_d, Q³_d and Q⁴_d.

If we have chosen x_b :

- Merge the cliques Q¹_d, Q²_d.
 Merge the cliques T^b_d, Q³_d, and Q⁴_d.

If we have chosen x_c :

- Merge T^c_d with Q⁴_d.
 Merge the cliques Q¹_d, Q²_d and Q³_d.

Finally, let $\beta \in \{a, b, c\}$ be the index of the chosen variable that satisfies Γ_d . For both $\gamma \in$ $\{a, b, c\} \setminus \{\beta\}$ do the following. If $\alpha(x_{\gamma}) =$ true, then merge T_d^{γ} with the cluster of \mathcal{P} consisting of $K^{\gamma}_{4\pi(\gamma,d)+1}$ and $K^{\gamma}_{4\pi(\gamma,d)+2}$. If $\alpha(x_{\gamma}) = \mathsf{false}$, then merge T^{γ}_d with the cluster of \mathcal{P} consisting of $K_{4\pi(\gamma,d)+1}^{\gamma}$ and $K_{4\pi(\gamma,d)}^{\gamma}$. This concludes the definition of the vertex partition \mathcal{P} . Let us denote the corresponding cluster editing set by S. That is, S contains all edges in G between clusters of \mathcal{P} and all non-edges within clusters of \mathcal{P} .

We claim that (c1) each edit in S is contained in a P_3 of \mathcal{H} and (c2) every P_3 of \mathcal{H} is edited at most once by S. Note that the claim implies that S is a solution to $(G, \mathcal{H}, 0)$. We first prove part (c1) of the claim. Note that each edit in S is between two cliques in V(H). There are three types of edits in \mathcal{H} : within a variable gadget, between a clause and a variable gadget, and within a clause gadget.

Consider first the edits contained in the variable gadget of an arbitrary variable x_i . Observe that each such edit is contained in an odd or an even pair of x's gadget. Such an edit is contained in a P_3 in \mathcal{H} , because, by construction of the variable gadgets, all edges and non-edges between the cliques of an odd or an even pair are covered by P_{3s} in \mathcal{H} .

For the edits in S which are not contained in variable gadgets, observe that between each pair of cliques in a single level L_s , s > 0, there are no edges in G. Whenever we merge two or more clusters during the construction of \mathcal{P} , we either merge a clique on level L_4 to two cliques on level L_0 or we merge cliques on pairwise different levels. Hence, each edit $e \in S$ which is not in a variable gadget is between two cliques on different levels. Moreover, observe that the cliques containing the endpoints of e are adjacent in V(H). Thus, by the way we have defined \mathcal{H}_{pad} via Corollary 2, there is a P_3 in \mathcal{H}_{pad} containing e. We have thus shown that claim (c1) holds.

For part (c2) of the claim, we first observe the following. Each P_3 in \mathcal{H} that intersects only two cliques in V(H) contains at most one edit of S. Let P be such a P_3 and let D_1 , D_2 be the two cliques in V(H) that intersect P. Note that \mathcal{H}_{tra} does not contain P_{3s} that intersect only two cliques in V(H) and thus either $P \in \mathcal{H}_{var}$ or $P \in \mathcal{H}_{pad}$. In both cases, there is exactly one edge and one non-edge of P between D_1 and D_2 : This is clear if $P \in \mathcal{H}_{pad}$. If $P \in \mathcal{H}_{var}$ then P was introduced when connecting a clause gadget to a variable gadget. In the notation used there, either $P = v_5 v_6 v_2$ or $P = v_1 v_7 v_8$, both of which have the required form. Thus, as D_1 and D_2 are either merged or not in \mathcal{P} , there is at most one edit in P.

To prove (c2) it remains to consider P_3 s in \mathcal{H} that intersect three cliques in V(H). Let P be such a P_3 . Note that $P \notin \mathcal{H}_{pad}$. If $P \in \mathcal{H}_{var}$, then it connects K_i^i to K_{i+2}^i via K_{i+1}^i for some even j and some variable index $i \in \{0, 1, \dots, n-1\}$. Since we merge either all odd or all even pairs in x_i 's variable gadget to obtain \mathcal{P} , indeed exactly one edge of P is edited, as claimed. If $P \in \mathcal{H}_{tra}$, then we distinguish two cases.

First, P does not contain a vertex of some variable-gadget clique. Then, P connects some clique Q_d^s to some transferring clique T_d^δ via $Q_d^{s'}$. According to the construction of \mathcal{P} , either T_d^δ and $Q_d^{s'}$ are in different clusters of \mathcal{P} and $Q_d^{s'}$ and Q_d^s are merged, or T_d^δ and $Q_d^{s'}$ are merged and Q_d^s and $Q_d^{s'}$ are in different clusters of \mathcal{P} . In both cases, there is at most one edit of S in P.

Second, P contains a vertex of some variable-gadget clique. Then, by construction of G and \mathcal{H} , path P indeed contains two vertices of two variable-gadget cliques, say K_i^i and K_{i+1}^i and one vertex of a transferring clique, say T_d^i . Assume that variable x_i appears positively in clause Γ_d , the other case is analogous. Then the center of P is K_j^i and moreover j is odd. If x_i was not chosen among the variables satisfying clause Γ_d when constructing \mathcal{P} , then T_d^i and K_j^i is in the same cluster Q of \mathcal{P} . Furthermore K_{j+1}^i is either in a cluster different from Q or also in Q. In both cases, there is at most one edit from S in \mathcal{P} . If x_i was chosen among the the variables satisfying clause Γ_d when constructing \mathcal{P} , then T_d^i is in a cluster in \mathcal{P} which is different from the one(s) containing K_j^i and K_{j+1}^i . However, since x_i satisfies Γ_d , we have $\alpha(x_i) = \text{true}$ and thus K_j^i and K_{j+1}^i are merged (recall that j is odd).

Thus, indeed, the claim holds, finishing the proof.

Soundness

Before we show how to translate a cluster editing set of size $|\mathcal{H}|$ for the constructed instance into a satisfying assignment of Φ , we make some structural observations.

Recall the definition of a proto-cluster, a connected component of the subgraph of G whose edge set contains precisely those edges of G which are not contained in any P_3 in \mathcal{H} .

Lemma 21. V(H) is precisely the set of proto-clusters of G and \mathcal{H} .

Proof. By construction, all edges in G between two cliques in V(H) are in a P_3 in \mathcal{H} . Thus each proto-cluster is contained in some clique in V(H). We claim that each clique $C \in V(H)$ contains a spanning tree of edges which are not contained in a P_3 in \mathcal{H} . If $C \in L_1$, then this is clear; such a C contains only a single vertex and a trivial spanning tree. If $C \in L_0$, then there are only two P_3 s in \mathcal{H} that contain edges of C: The one given by $v_5v_6v_2$ and the one given by $v_1v_7v_8$ as shown in Fig. 6.3 when connecting variable and clause gadgets. Since |C| = 5, indeed C contains the required spanning tree. If $C \in L_i$ for $i \geq 2$, then by the connectedness property of Corollary 2, C has the required spanning tree.

Observe that each solution S to $(G, \mathcal{H}, 0)$ cannot remove any edge from G which is not contained in a P_3 in \mathcal{H} . Thus, since V(H) is a vertex partition of G, each solution S generates a cluster graph $G \triangle S$ whose clusters induce a coarser vertex partition than V(H). This leads to the following.

Observation 4. For each solution S to $(G, \mathcal{H}, 0)$, each cluster in $G \triangle S$ is a disjoint union of cliques in V(H).

In the following it will also be useful to define the notion of a *dividing non-edge*, which is a non-edge which is not contained in any P_3 in \mathcal{H} . Using this and the above structural observations, we are now ready to prove the soundness of the construction.

Lemma 22. If the constructed instance $(G, \mathcal{H}, \ell = 0)$ is a yes-instance, then the formula Φ is satisfiable.

Proof. Suppose that there exists a set of vertex pairs $S \subseteq \binom{V}{2}$ so that $G\Delta S$ is a union of vertexdisjoint cliques and $|S| - |\mathcal{H}| = 0$. In other words, there exists a solution that transforms G into a cluster graph G' by editing exactly one edge or non-edge of every P_3 of \mathcal{H} . We will show that there exists a satisfying assignment $\alpha: \{x_0, x_1, \ldots, x_{n-1}\} \to \{\text{true, false}\}$ for the formula Φ .

By Observation 4, the set of clusters in G' induces a partition of the cliques in V(H). Call two cliques in V(H) merged if they are in the same cluster in G' and divided otherwise.

We aim to define the assignment α . For this we need the following observation on the solution. Consider variable x_i and the cliques K_j^i , $j = 0, 1, \ldots, 4m_i - 1$, in x_i 's variable gadget. Call a pair K_j^i , K_{j+1}^i even if j is even (where j + 1 is taken modulo $4m_i$) and call this pair odd otherwise. We claim that either (i) each even pair is merged and each odd pair is divided, or (ii) each odd pair is merged and each even pair is divided (and not both). Note that, for each even

j, pair K_j^i , K_{j+1}^i is merged or pair K_{j+1}^i , K_{j+2}^i is merged, because there is a P_3 in G containing vertices in these cliques with center in K_{j+1}^i . To show the claim, it is thus enough to show that not both an odd pair and an even pair is merged.

For the sake of contradiction, suppose that an odd pair is merged and an even pair is merged. Then, there exists an index $j \in \{0, 1, \ldots, 4m_i - 1\}$ and a cluster C in G' such that $K_j^i, K_{j+1}^i, K_{j+2}^i \subseteq C$, where here and below the indices are taken modulo $4m_i$. Observe that there are no edges between K_j^i and K_{j+2}^i in G. If j is odd, then all of these non-edges are dividing, that is, they are not contained in any P_3 in \mathcal{H} . All of these non-edges are thus in S. This is a contradiction to the fact that S contains at most $|\mathcal{H}|$ vertex pairs. Thus, j is even.

We now show that for each $k \in \mathbb{N} \cup \{0\}$, pair K_{j+1+2k}^i , K_{j+2+2k}^i is merged by induction on k. Clearly, for k = 0, this holds by supposition. If k > 0 then, by construction, there are dividing non-edges between K_{j+2k-1}^i and K_{j+2k+1}^i . Combining this with the fact that $K_{j+1+2(k-1)}^i = K_{j+2k-1}^i$ and $K_{j+2+2(k-1)}^i = K_{j+2k}^i$ are merged by inductive assumption, it follows that K_{j+2k+2}^i and K_{j+2k+1}^i are divided. Since there is a P_3 in G connecting K_{j+2k}^i , K_{j+2k+1}^i , and K_{j+2k+2}^i with center in K_{j+2k+1}^i , it follows that K_{j+2k+1}^i , K_{j+2k+2}^i are merged, as required.

It now follows in particular that K_{j-1}^i and K_j^i are merged (recall that indices are taken modulo $4m_i$). Since by assumption also K_j^i and K_{j+1}^i are merged, we have that $K_{j'}^i$, $K_{j'+1}^i$, and $K_{j'+2}^i$ are contained in the same cluster in G' for some odd j'. As already argued, this leads to a contradiction. Thus the claim holds.

We define the assignment α as follows. For each variable x_i , $i = 0, 1, \ldots, n-1$, if in G' all even pairs K_{2j}^i , K_{2j+1}^i , $j = 0, 1, \ldots, m_i - 1$, are merged, then $\alpha(x_i) = \mathsf{false}$. Otherwise $\alpha(x_i) = \mathsf{true}$.

We now show that α satisfies Φ . Consider an arbitrary clause Γ_d of Φ containing the three variables x_a , x_b , and x_c . We use the same notation as when defining the clause gadget and its connection to the variable gadget. Since there are dividing non-edges between cliques Q_d^1 and Q_d^4 , cliques Q_d^1 and Q_d^4 must end up in different clusters in G'. In other words, Q_d^1 and Q_d^4 are divided. Observe that there is a path in G consisting of vertices in Q_d^1 , Q_d^2 , Q_d^3 , and Q_d^4 in this sequence. Since each of these four cliques is a proto-cluster (see Lemma 21), in order to divide Q_d^1 and Q_d^2 are deleted. In other words, Q_d^1 and Q_d^2 are divided. (ii) Q_d^2 and Q_d^2 are deleted. In other words, Q_d^1 and Q_d^2 are divided. (iii) Q_d^2 and Q_d^4 are divided. We now show that case (i), (ii), and (iii) imply that variable x_a , x_b , and x_c , respectively, is set by α so as to satisfy Γ_d . We only give the proof showing that case (i) implies that x_a is set accordingly. The other cases are analogous.

Assume that case (i) holds. Then, by the constraints imposed by the two transferring $P_{3s} P_d^1$ and P_d^2 , cliques T_d^a and Q_d^1 are merged. Since there are dividing non-edges between $K_{4\pi(a,d)+1}^a$ and Q_d^1 , it follows that $K_{4\pi(a,d)+1}^a$ and Q_d^1 are divided. Consider the case that x_a appears positively in Γ_d . Then, when connecting the variable gadget of x_a to the clause gadget of Γ_d we have introduced into $G \ P_3$ connecting T_d^a , $K_{4\pi(a,d)+1}^a$, and $K_{4\pi(a,d)+2}^a$ with center in $K_{4\pi(a,d)+1}^a$ (for example, the P_3 given by $w_1v_1v_3$). Since T_d^a and $K_{4\pi(a,d)+1}^a$ are divided, thus $K_{4\pi(a,d)+1}^a$ and $K_{4\pi(a,d)+2}^a$ are merged. There is thus at least one odd pair in x_a 's variable gadget that is merged and thus $\alpha(x_a) = \text{true}$. The case where x_a appears negatively in Γ_d is similar: We have introduced into $G \ P_3$ given by $w_1v_1v_3$). It follows that $K_{4\pi(a,d)+1}^a$, and $K_{4\pi(a,d)}^a$ are merged, showing that at least one even pair is merged in x_a 's variable gadget. Thus, $\alpha(x_a) = \text{false}$.

Thus each clause Γ_d is satisfied, finishing the proof.

Algorithm 1: Solve CEATMP. **Input**: An instance (G, \mathcal{H}, ℓ) of CEATMP. **Output**: Whether (G, \mathcal{H}, ℓ) is a yes-instance. 1 foreach $\ell_a = 0, 1, \ldots, \ell$ do foreach $\ell_b = 0, 1, \ldots, \ell - \ell_a$ do $\mathbf{2}$ **foreach** set S_a of ℓ_a vertex pairs $\{u, v\} \in \binom{V(G)}{2}$ such that 3 $\forall P \in \mathcal{H} \colon |\{u, v\} \cap V(P)| \leq 1$ do $G_a \leftarrow G \triangle S_A$ 4 foreach set \mathcal{H}_b of ℓ_b distinct P_3s in \mathcal{H} do $\mathbf{5}$ **foreach** set S_b containing for each $P \in \mathcal{H}_b$ at least two vertex pairs in 6 V(P) do if $|S_a| + |S_b| \le |\mathcal{H}_b| + \ell$ then 7 $G_b \leftarrow G_a \triangle S_B$ 8 $\mathcal{H}' \leftarrow \mathcal{H} \setminus \mathcal{H}_b$ 9 if G_b has a cluster-editing set with $|\mathcal{H}'|$ edits then /* Using $\mathbf{10}$ Theorem 15 */ accept and halt 11 12 reject

6.2 XP-algorithm for 2-restricted packings

In this section, we study CEAMP in the special setting where every vertex is incident with at most two P_{3s} of the packing \mathcal{H} . More precisely, we consider the following variant of CEAMP.

Cluster Editing above two-restricted modification-disjoint P_3 packing (CEATMP)

Input: A graph G = (V, E), a packing \mathcal{H} of modification-disjoint induced P_{3} s of G such that every vertex $v \in V(G)$ is incident with at most 2 P_{3} s of \mathcal{H} , and a nonnegative integer ℓ . **Question:** Is there a cluster editing set, i.e. a set of vertex pairs $S \subseteq {V \choose 2}$ so that $G \triangle S$ is a union of disjoint cliques, with $|S| - |\mathcal{H}| \leq \ell$?

We give a polynomial-time algorithm to solve CEATMP when ℓ is a fixed constant, in contrast with the NP-hardness of the general version of CEAMP when $\ell = 0$.

Theorem 14 (Restated). CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_3 PACKING parameterized by the number ℓ of excess edits is in XP. It can be solved in $O(n^{2\ell+O(1)})$ time, where n is the number of vertices in the input graph.

The main tool in proving Theorem 14 is a polynomial-time algorithm for the case where $\ell = 0$:

Theorem 15 (Restated). CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_3 PACKING can be solved in polynomial time when $\ell = 0$, that is, when no excess edits are allowed.

The proof of Theorem 15 will be given in 6.2.2. With this tool in hand, we can show Theorem 14.

Proof of Theorem 14. Let (G, \mathcal{H}, ℓ) be an instance of CEATMP. The algorithm is given in Theorem 1. Essentially, it guesses (by trying all possibilities) the number, ℓ_a , of excess edits that are not contained in any P_3 in \mathcal{H} and guesses the concrete edits to be made (Lines 1-4). Then it guesses the P_{35} in \mathcal{H} that harbor the remaining excess edits and it guesses how these P_{35} are resolved (Lines 5-9). Then it checks whether the remaining instance has a cluster-editing set without excess edits over the remaining P_3 packing \mathcal{H}' using the algorithm from Theorem 15.

For the running time, observe that there are at most $n^{2\ell_a}$ choices for S_a . Since each vertex is in at most two P_3 s in \mathcal{H} and each P_3 covers exactly three vertices, we have $3|\mathcal{H}| \leq 2n$ and thus there are in total at most $n P_3$ s in \mathcal{H} . Thus, there are $O(n^{\ell_b})$ choices for \mathcal{H}_b . Since there are four possibilities to select a set of at least two vertex-pairs in the vertex set of a P_3 , there are $O(4^{\ell_b})$ possibilities for S_b in Line 6. Hence, overall the running time is $O(4^{\ell_b}n^{2\ell_a+\ell_b+O(1)}) \leq O(n^{2\ell+O(1)})$.

It remains to prove the correctness. If the algorithm accepts, then there is a cluster-editing set S_0 for G_b with $|\mathcal{H}'|$ edits. Since S_0 is contained in the vertex sets of the P_{3s} in \mathcal{H}' , set S_0 is disjoint from S_a and S_b . Thus, $G \triangle S^*$ is a cluster graph where $S^* = S_a \cup S_b \cup S_0$. Moreover, $|S^*| \leq |\mathcal{H}'| + |\mathcal{H}_b| + \ell = |\mathcal{H}| + \ell$, and thus, (G, \mathcal{H}, ℓ) is a yes-instance.

Conversely, if (G, \mathcal{H}, ℓ) is a yes-instance, then there is a cluster-editing set S^* of G with $|S^*| \leq |\mathcal{H}| + \ell$. Let S_a^* be the subset of S^* that contains precisely those edits in S^* that are not contained in P_3 s of \mathcal{H} . In one of the iterations of Algorithm 1, $\ell_a = |S_a^*|$ and $S_a = S_a^*$. Now let \mathcal{H}_b^* be the subset of \mathcal{H} that contains precisely those P_3 s P such that S^* contains at least two edits in V(P). Observe that $|\mathcal{H}_b^*| \leq \ell - \ell_a$. Thus, in one of the iterations of Algorithm 1, we have $\ell_b = |\mathcal{H}_b^*|$ and $\mathcal{H}_b = \mathcal{H}_b^*$. Moreover, in one of the iterations $S_b = S_b^*$, where S_b^* is the subset of S^* that contains precisely those edits that are contained in the P_3 s in \mathcal{H}_b . Let $S_0^* = S^* \setminus (S_a^* \cup S_b^*)$. Since each edit in S_0^* is contained in a unique P_3 in $\mathcal{H} \setminus \mathcal{H}_b^*$, we have $|S_a| + |S_b| = |S_a^*| + |S_b^*| \leq |\mathcal{H}_b^*| + \ell = |\mathcal{H}_b| + \ell$. Thus, in that iteration the algorithm proceeds to the if-condition in Line 10. Again since each edit in S_0^* is contained in a unique P_3 in $\mathcal{H} \setminus \mathcal{H}_b^*$, this set witnesses that $(G_b, \mathcal{H}', 0)$ is a yes-instance and thus the algorithm accepts. Hence, the algorithm is correct.

6.2.1 Reduction Rules for 2-restricted P₃-packings

In this section we give reduction rules for CEATMP when $\ell = 0$. Again, we use the term *proto-clusters* to denote the connected components of the graph obtained by removing the edges of all packed P_{3s} . We say a proto-cluster C is *isolated* from a proto-cluster D if there are no edges between C and D. Since $\ell = 0$ and the P_{3s} of \mathcal{H} are modification-disjoint, we have the following observation.

Observation 5. A solution S to an instance of CEATMP must edit exactly one edge or nonedge of every P_3 of \mathcal{H} , and no non-packed edges or non-packed non-edges can be edited by S.

The intuition behind the polynomial-time algorithm is, with the constraint that every vertex $v \in V(G)$ is incident with at most two packed P_{3} s, we cannot freely merge or separate two large proto-clusters without excess edits as in the NP-hardness proof of Section 6.1, since the triangles formed by the packed P_{3} s cannot cover every vertex pair between two large proto-clusters. Thus we can separate the large proto-clusters and deal with them separately. After some reduction rules, we show that the instance can be reduced to an instance of 2-SAT, which is well-known to be polynomial-time solvable.

We classify the P_{3s} of \mathcal{H} into four types. For an induced $P_{3} xyz \in \mathcal{H}$, if x, y belong to one proto-cluster and z belongs to another proto-cluster, or symmetrically y, z belong to one proto-cluster and x belongs to another proto-cluster, then xyz is a type- αP_{3} ; if x, z belong to one proto-cluster and y belongs to another proto-cluster, then xyz is a type- βP_3 ; if x, y, z belong to three distinct proto-clusters respectively, then xyz is a type- γP_3 ; if x, y, z belong to one proto-cluster then xyz is a type- δP_3 . Note that none of the reduction rules increases or decreases the parameter ℓ . By removing the corresponding packed P_3 from \mathcal{H} , we mean that if a packed P_3 is edited by the reduction rules, then remove it from \mathcal{H} .

Reduction Rule 1. For any proto-cluster C, if there are two vertices $u, v \in V(C)$ such that uv is a non-packed non-edge, i.e. uv is not covered by any P_3 of \mathcal{H} , then return NO.

Lemma 23. Reduction Rule 1 is safe.

Proof. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP satisfying the condition of Reduction Rule 1, suppose for contradiction that there is a solution S to this instance. Since u, v belong to the same proto-cluster, there is a non-packed path P from u to v. By Observation 5, $uv \notin S$ and none of the edges of P is edited by S. Thus $G \triangle S$ is not a cluster graph, contradicting that the instance has a solution. This completes the proof for the lemma. \Box

The second reduction rule handles type- β and type- δP_{3s} .

Reduction Rule 2. If there is a type- β or type- δP_3 $xyz \in \mathcal{H}$, insert an edge to xz and remove xyz from \mathcal{H} .

Lemma 24. Reduction Rule 2 is safe.

Proof. Suppose that the given instance of CEATMP is $(G, \mathcal{H}, \ell = 0)$ such that there exists a type- $\beta P_3 xyz$ in G. After inserting an edge to xz and removing xyz from \mathcal{H} , we get an instance $(G', \mathcal{H}', \ell = 0)$. We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. On one hand, suppose that $(G', \mathcal{H}', \ell = 0)$ is a YES-instance and S' is a cluster editing set of G' such that $|S'| = |\mathcal{H}'|$. Obviously, $S' \cup \{xz\}$ is a cluster editing set for G and $|S' \cup \{xz\}| = |\mathcal{H}|$. On the other hand, suppose that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance and $S \setminus \{xz\}$ is the solution for $(G', \mathcal{H}', \ell = 0)$. For contradiction, suppose this is not true. Then either $xy \in S$ or $yz \in S$ holds. Without loss of generality we assume that $xy \in S$. Suppose that after deleting xy from G and removing xyz from \mathcal{H} , we get an instance $(G'', \mathcal{H}', \ell = 0)$. Since x, z belong to one proto-cluster of G, there is a non-packed path P from x to z in G. Thus x, z belong to one proto-cluster. Since xyz is removed from \mathcal{H}, xz becomes a non-packed non-edge. By Reduction Rule 1, $(G'', \mathcal{H}'', \ell = 0)$ is a NO-instance, contradicting that S is a solution to $(G, \mathcal{H}, \ell = 0)$.

A very similar analysis applies to the case that $xyz \in \mathcal{H}$ is a type- δP_3 . This completes the proof for the lemma.

After applying Reduction Rules 1 and 2 exhaustively, if the algorithm did not return NO, then there is no type- β or type- δ P_{3} s in the instance. Next reduction rule applies to the case in which there is both a non-packed non-edge and a packed edge between two proto-clusters.

Reduction Rule 3. For any two proto-clusters A and B, if there is a non-packed non-edge uv such that $u \in V(A)$ and $v \in V(B)$, and there is a packed edge xy such that $x \in V(A)$ and $y \in V(B)$, then delete xy and remove the corresponding packed P_3 from \mathcal{H} .

Lemma 25. Reduction Rule 3 is safe.

Proof. Suppose that we have applied Reduction Rules 1 and 2 exhaustively. Since the algorithm did not return NO, there are no type- βP_3 s in the instance. Thus xy is covered by either a type- γP_3 or a type- αP_3 .

Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP satisfying the condition of Reduction Rule 3 with xy covered by a type- $\gamma P_3 xyz$ (without loss of generality, we do not analyse the symmetrical case of yxz'), we get an instance $(G', \mathcal{H}', \ell = 0)$ of CEATMP after deleting xy and removing xyz from \mathcal{H} . We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. For completeness, assume that $(G', \mathcal{H}', \ell = 0)$ is a YES-instance and S' is a cluster editing set of size $|\mathcal{H}'|$ for G'. Then obviously $S' \cup \{xy\}$ is a solution to $(G, \mathcal{H}, \ell = 0)$. For soundness, assume that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance and S is a cluster editing set of size $|\mathcal{H}|$ for G. We claim that $xy \in S$. Suppose for contradiction that $xy \notin S$. Then xy becomes a non-packed edge in $G \triangle S$. Since $u, x \in V(A)$ and $v, y \in V(B)$, there is a non-packed path P_A from u to x and a non-packed path P_B from v to y in G. By Observation 5, the edges of P_A and P_B are not edited by S and $uv \notin S$. Thus there is a non-packed path from u to v. Since uv is a non-packed non-edge in $G \triangle S$, $G \triangle S$ is not a cluster graph, contradicting that S is a solution to $(G, \mathcal{H}, \ell = 0)$.

A very similar analysis applies to the case in which xy is covered by a type- $\alpha P_3 xyz$ (or the symmetrical case of yxz'). This concludes the proof for the lemma.

Reduction Rule 4. If there is a connected component C in the graph of size at most 6, then do brute force on C to check if there is a cluster editing set F for C such that |F| is equal to the number of packed P_{3s} incident with a vertex of C. If there is such a cluster editing set F, then perform the operations of F to C and remove the corresponding packed P_{3s} from \mathcal{H} . Otherwise, if there is no such cluster editing set F, return NO.

Lemma 26. Reduction Rule 4 is safe.

Proof. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP such that there is a connected component \mathcal{C} in the graph of size at most 6, suppose that there is a cluster editing set F for \mathcal{C} satisfying the condition of Reduction Rule 4. After performing the operations of F, we get an instance $(G', \mathcal{H}', \ell = 0)$ of CEATMP. We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. On one hand, assume that $(G', \mathcal{H}', \ell = 0)$ has a solution S'. Obviously, $S' \cup F$ is a cluster editing set for G and $|S' \cup F| = |\mathcal{H}|$. On the other hand, assume that $(G, \mathcal{H}, \ell = 0)$ has a solution S. By Observation 5, no vertex pair between $V(\mathcal{C})$ and $V(G) \setminus V(\mathcal{C})$ is edited by S. Let $S_1 \subseteq S$ be the set of vertex pairs which are edges or non-edges of \mathcal{C} . Then $S \setminus S_1$ is a solution to $(G', \mathcal{H}', \ell = 0)$.

Suppose that there is no such cluster editing set F for C. We claim that $(G, \mathcal{H}, \ell = 0)$ is a NO-instance. For contradiction, assume that $(G, \mathcal{H}, \ell = 0)$ has a solution S. Let $S_1 \subseteq S$ be the set of vertex pairs which are edges or non-edges of C. Then S_1 is a cluster editing set for Cand $|S_1|$ is equal to the number of packed P_{3S} incident with a vertex of C by Observation 5, a contradiction. Thus $(G, \mathcal{H}, \ell = 0)$ is a NO-instance.

The component C is of size at most 6 so we can do brute force in polynomial time. This completes the proof for the lemma.

Reduction Rule 5. If there is a proto-cluster C which is an isolated clique, then remove C from the graph.

Lemma 27. Reduction Rule 5 is safe.

Proof. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP such that there is a proto-cluster C which is an isolated clique, we remove C from G and get an instance $(G', \mathcal{H}, \ell = 0)$. We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}, \ell = 0)$ is a YES-instance. On one hand, assume that $(G', \mathcal{H}, \ell = 0)$ is a YES-instance. Then obviously $(G, \mathcal{H}, \ell = 0)$ is a YES-instance. On the other hand, assume that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance and S is a solution. Since C is an isolated clique, by Observation 5, no edges of C or non-edges between V(C) and $V(G) \setminus V(C)$ are edited by S. Thus S is also a solution to $(G', \mathcal{H}, \ell = 0)$. This completes the proof for the lemma.

Lemma 28. After applying Reduction Rules 1 - 5 exhaustively, if the algorithm did not return NO, then there is no proto-cluster of size at least 5.

Proof. Suppose for contradiction that there is a proto-cluster C of size at least 5. If C is a proto-cluster which is isolated from other proto-clusters, then C must be a clique since otherwise Reduction Rule 1 or Reduction Rule 2 can be applied, a contradiction. Then Reduction Rule 5 can be applied and C will be removed from the graph. Thus C is not an isolated proto-cluster. Let D be a proto-cluster such that there is an edge uv between C and D, say $u \in V(C)$ and $v \in V(D)$. If uv is covered by a type- βP_3 , then Reduction Rule 2 can be applied, a contradiction. Thus we assume that uv is covered by a type- α or a type- γP_3 . Since v is incident with at most two packed P_{3s} , there must be one vertex $w \in V(C)$ such that wv is a non-packed non-edge. Then Reduction Rule 3 can be applied, a contradiction. As a result, there is no proto-cluster of size at least 5. This completes the proof for the lemma.

Next we focus on proto-clusters of size 4.

Lemma 29. After applying Reduction Rules 1 - 3 exhaustively, if there is a proto-cluster C of size 4 which is not an isolated clique. Then there must be a proto-cluster D of size 1 such that the vertex pairs between C and D are covered by two type- α P₃s. In addition, $V(C) \cup V(D)$ forms a connected component in the graph.

Proof. After applying Reduction Rules 1 - 3 exhaustively, let C be a proto-cluster of size 4 and $V(C) = \{v_1, v_2, v_3, v_4\}$. Let w be a vertex such that there is an edge between w and V(C). If the vertex pairs between V(C) and w are not covered by two type- αP_{3s} , then Reduction Rule 2 or 3 can be applied, a contradiction. Without loss of generality, suppose that v_1v_2 and v_3v_4 are covered by these two type- αP_{3s} . Assume for contradiction that there is another vertex u such that u and (without loss of generality) v_1 are adjacent, and uv_1 is a packed edge (u can be either in the same proto-cluster with w or in a different proto-cluster from w). Since we have applied Reduction Rule 2 exhaustively, there are no type- βP_3 s in the graph. Thus uv_1 must be covered by a type- α or a type- γ P₃. We claim that there must be a non-packed non-edge from u to a vertex of C. For contradiction, suppose this is not true. Then either v_1v_4, v_2v_3 are covered by two type- αP_{3s} respectively, or v_1v_3, v_2v_4 are covered by two type- αP_{3s} respectively. In both cases, v_1, v_2, v_3 and v_4 are not in one proto-cluster anymore since after removing the packed edges, v_1, v_2, v_3 and v_4 are not in one connected component, a contradiction. Thus there must be a non-packed non-edge between V(C) and u. Since uv_1 is a packed edge, Reduction Rule 3 can be applied, a contradiction. Thus there are no edges between V(C) and any other vertices except w. Suppose that w belongs to a clique of size at least two. Then Reduction Rule 3 can be applied, a contradiction. Thus w belongs to a proto-cluster of size one and let this proto-cluster be D. Since w is already incident with two packed P_{3s} , w is isolated from any other proto-clusters except C. Obviously, $V(C) \cup V(D)$ forms a connected component in the graph. This completes the proof for the lemma.

Lemma 30. After applying Reduction Rules 1 - 5 exhaustively, there is no proto-cluster of size 4.

Proof. Suppose for contradiction that there is a proto-cluster C of size at least 4. If C is an isolated proto-cluster, C must be a clique since otherwise Reduction Rule 1 or 2 can be applied, a contradiction. Then Reduction Rule 5 can be applied and C will be removed from the graph.

Thus C is not an isolated proto-cluster. By Lemma 29, there is a proto-cluster D of size 1 such that $V(C) \cup V(D)$ forms a connected component of size 5 in the graph. Then Reduction Rule 4 can be applied, a contradiction. As a result, there is no proto-cluster of size at least 4. This completes the proof for the lemma.

Next we focus on proto-clusters of size 3.

Lemma 31. After applying Reduction Rules 1 - 5 exhaustively, if there is a proto-cluster C of size 3, then there must be a proto-cluster B of size 1 and a proto-cluster A of size 1, such that the vertex pairs between C and B are covered by a type- α P₃ and a type- γ P₃, and the type- γ P₃ connects C and A via B. In addition, C is isolated from any other proto-clusters except B, and B is isolated from any other proto-clusters except A and C.

Proof. After applying Reduction Rules 1 - 5 exhaustively, let C be a proto-cluster of size 3. If Cis isolated from other proto-clusters, then C must be a clique since otherwise Reduction Rule 1 can be applied, a contradiction. Then Reduction Rule 5 can be applied, a contradiction. Thus we assume that C is not an isolated proto-cluster. Let the three vertices of C be u_1, u_2 and u_3 . Let v be a vertex such that there is an edge between v and V(C). If the vertex pairs between V(C) and v are not covered by a type- αP_3 and a type- γP_3 , then Reduction Rule 2 or 3 can be applied, a contradiction. Without loss of generality, suppose that u_1, u_3 and v belong to a type- αP_3 . Assume for contradiction that there is another vertex w such that w is adjacent to some vertex of V(C) (w can either belong to the same proto-cluster as v or belong to a different proto-cluster from v). If the vertex pairs between V(C) and w are not covered by a type- αP_3 and a type- γP_3 , then Reduction Rule 2 or 3 can be applied, a contradiction. If the vertex pairs between V(C) and u are covered by a type- αP_3 and a type- γP_3 , say u_1, u_2 and w belong to the type- αP_3 , then u_1, u_2 and u_3 are not in one proto-cluster, a contradiction. It follows that there is no vertex adjacent to one of the vertices of V(C) except v. Let B be the proto-cluster to which v belongs. Assume that |B| > 1 and there is another vertex y belonging to B. Then y is not adjacent to any vertex of V(C). Thus Reduction Rule 3 can be applied, a contradiction. It follows that |B| = 1 and C is isolated from any other proto-clusters except B. We have assumed that u_1, u_3 and v belong to a type- αP_3 . Let u_2vx be a type- γP_3 and x belongs to a proto-cluster A. We claim that |A| = 1. Suppose for contradiction that |A| > 1 and there is another vertex $z \in V(A)$. Then vz must be a non-packed non-edge since v is already incident with two packed P_{3s} . Thus Reduction Rule 3 can be applied, a contradiction. It follows that |A| = 1. This concludes the proof for the lemma.

Reduction Rule 6. After applying Reduction Rules 1 - 5 exhaustively, if there is a proto-cluster C of size 3, a proto-cluster B of size 1 and a proto-cluster A of size 1 such that C is not isolated from B, and a type- γ P_3 connects A and C via B, then delete the packed edge between A and B, insert an edge to the packed non-edge between C and B, and remove the corresponding P_3 s from \mathcal{H} .

Lemma 32. Reduction Rule 6 is safe.

Proof. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP satisfying the condition of Reduction Rule 6, let u_1, u_2 and u_3 be the three vertices of C, let v be the vertex of B and w be the vertex of A. Without loss of generality, let u_1u_3v and u_2vw be two packed P_{3s} . After applying Reduction Rule 6, we get an instance $(G', \mathcal{H}', \ell = 0)$ of CEATMP. We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. For completeness, suppose that $(G', \mathcal{H}', \ell = 0)$ is a YES-instance and S' is a cluster editing set of G' such that $|S'| = |\mathcal{H}'|$. Obviously $S = S' \cup \{u_1v, vw\}$ is a solution to $(G, \mathcal{H}, \ell = 0)$. For soundness, suppose that


Figure 6.6: An example of forming a clique of size 6 in $G \triangle S$. The black edges are non-packed edges. The vertex pairs of the same color which is not black belong to the same packed P_3 and the dashed edges represent non-edges. The same rule of notation applies to the following pictures.

 $(G, \mathcal{H}, \ell = 0)$ is a YES-instance and S is a cluster editing set of G such that $|S| = |\mathcal{H}|$. If $\{u_1v, vw\} \subseteq S$, then $S' = S \setminus \{u_1v, vw\}$ is a solution to $(G', \mathcal{H}', \ell = 0)$. If $vw \notin S$, then either $u_2w \in S$ or $u_2v \in S$. First, we assume that $u_2w \in S$, and after inserting u_2w and removing u_2vw from \mathcal{H} we get an instance $(G'', \mathcal{H}', \ell = 0)$ of CEATMP. Then u_3u_2w is a non-packed path in G'' and u_3w is a non-packed non-edge. Thus Reduction Rule 1 can be applied and the algorithm returns NO, contradicting that S is a solution to $(G, \mathcal{H}, \ell = 0)$. Second, we assume that $u_2v \in S$. After deleting u_2v and removing u_2vw from \mathcal{H} , u_2v becomes a non-packed non-edge. Thus Reduction Rule 3 can be applied. It's not hard to see that $u_3v \in S$ since otherwise S is not a solution to $(G, \mathcal{H}, \ell = 0)$. By Lemma 31, C is isolated from any other proto-clusters except B, and B is isolated from any other proto-clusters except A and C. It follows that in $G \bigtriangleup S$, u_1, u_2 and u_3 form a clique of size 3 while v and w form a clique of size 2. Besides, $V(G) \setminus \{u_1, u_2, u_3, v, w\}$ form a cluster graph in $G \bigtriangleup S$. Let $\widehat{S} = S \setminus \{u_2v, u_3v\} \cup \{vw, u_1v\}$. Obviously $G \bigtriangleup \widehat{S}$ is also a cluster graph and $|\widehat{S}| = |\mathcal{H}|$. Thus \widehat{S} is also a solution to $(G, \mathcal{H}, \ell = 0)$.

Corollary 3. After applying Reduction Rules 1 - 6 exhaustively, there are no isolated cliques in the instance and every proto-cluster of the instance is of size at most 2. Moreover, every packed P_3 is a type- γP_3 .

After applying Reduction Rules 1 - 6 exhaustively, we have an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP. Suppose that S is a solution to $(G, \mathcal{H}, \ell = 0)$. Now we consider the cluster graph $G \triangle S$.

Lemma 33. After applying Reduction Rules 1 - 6 exhaustively, we have an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP. Suppose that S is a solution to $(G, \mathcal{H}, \ell = 0)$. Then there is no clique of size larger than 6 in $G \triangle S$.

Proof. Suppose for contradiction that A is a clique of size at least 7 in $G \triangle S$ and let u be a vertex in A. Then there are at least 6 vertex pairs between $\{u\}$ and $V(A) \setminus \{u\}$, which are either non-packed edges or covered by packed P_{3s} . Since u is incident with at most 2 packed P_{3s} , at most 4 vertex pairs between $\{u\}$ and $V(A) \setminus \{u\}$ are covered by a packed P_{3} . Thus at least 2 vertex pairs between $\{u\}$ and $V(A) \setminus \{u\}$ are non-packed edges. By Corollary 3, every proto-cluster in G is of size at most 2, a contradiction. This completes the proof for the lemma.

Lemma 34. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP such that the size of every protocluster in G is at most 2 and S is a solution to $(G, \mathcal{H}, \ell = 0)$, suppose that A is a clique of size 6 in $G \triangle S$. Then the vertices of V(A) belong to three proto-clusters C_1, C_2 and C_3 of size two in G. In addition, every vertex pair between C_1 and C_2 , between C_1 and C_3 , between C_2 and C_3 is covered by some P_3 of \mathcal{H} . $V(C_1) \cup V(C_2) \cup V(C_3)$ forms a connected component \mathcal{C} in G.

Proof. Suppose for contradiction that $u \in V(A)$ belongs to a proto-cluster of size one in G. Then there are 5 vertex pairs between $\{u\}$ and $V(A) \setminus \{u\}$, which are covered by packed P_{3s} . Since u belongs to at most 2 packed P_{3s} , at most 4 vertex pairs between $\{u\}$ and $V(A) \setminus \{u\}$ are covered by a packed P_3 , a contradiction. Next we show that the vertices of V(A) belong to three proto-clusters C_1, C_2 and C_3 of size two in G as shown in Fig 6.6. We see that for every vertex $v \in V(A)$, four of the vertex pairs between $\{v\}$ and $V(A) \setminus \{v\}$ are covered by packed P_{3s} and the other one is a non-packed edge. Since every vertex $v \in V(A)$ already belongs to two packed P_{3s} , the proto-cluster C_i is isolated from any other proto-clusters in $G \setminus \{C_1, C_2, C_3\}$ for i = 1, 2, 3. Without loss of generality, suppose that xyz is a P_3 such that $x \in V(C_1), y \in V(C_2)$ and $z \in V(C_3)$. Thus $V(C_1) \cup V(C_2) \cup V(C_3)$ forms a connected component. This completes the proof for the lemma.

Lemma 35. After applying Reduction Rules 1 - 6 exhaustively, we have an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP. Suppose that S is a solution to $(G, \mathcal{H}, \ell = 0)$. Then there is no clique of size 6 in $G \triangle S$.

Proof. Suppose for contradiction that A is a clique of size 6 in $G \triangle S$. According to Lemma 34, V(A) induces a connected component of size 6 in the input graph. Then Reduction Rule 4 and Reduction Rule 5 can be applied, a contradiction. This completes the proof for the lemma. \Box

Lemma 36. After applying Reduction Rules 1 - 3 exhaustively, let $(G, \mathcal{H}, \ell = 0)$ be an instance of CEATMP such that the size of every proto-cluster in G is at most 2 and S is a solution to $(G, \mathcal{H}, \ell = 0)$. Suppose that A is a clique of size 5 in $G \triangle S$. Then the vertices of V(A) belong to three proto-clusters C_1, C_2 and C_3 (or C_2, C_3 and C_4) in G such that $|C_1| = |C_4| = 1$ and $|C_2| = |C_3| = 2$. Every vertex pair between C_i and C_j $(i, j \in \{1, 2, 3, 4\}, i \neq j)$ is covered by a packed P_3 except that the vertex pair between C_1 and C_4 is a non-packed non-edge. In addition, $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ forms a connected component C in G.

Proof. Suppose for contradiction that at least three vertices of V(A) belong to proto-clusters of size one in G, say $u, v, w \in V(A)$ belong to three distinct proto-clusters of size one respectively and two vertices of V(A), say $x, y \in V(A)$ belong to a proto-cluster of size two or belong to two distinct proto-clusters of size one respectively. It follows that every vertex pair of $\binom{V(A)}{2}$ is either a non-packed edge or covered by some P_3 of \mathcal{H} . Without loss of generality, assume that u, v, xbelong to a packed P_3 and w, v, y belong to another packed P_3 . Now uw, uy must be covered by one packed P_3 , since u is already incident with one packed P_3 . However, the packed P_3 covering uw, wy is not modification-disjoint with the packed P_3 covering vw, vy, a contradiction. Next we show that V(A) belong to three proto-clusters C_1, C_2 and C_3 (or C_2, C_3 and C_4) in G such that $|C_1| = |C_4| = 1$ and $|C_2| = |C_3| = 2$ as shown in Case (1) and Case (2) of Fig. 6.7. Let $V(C_1) = \{x\}, V(C_2) = \{u_1, u_2\}$ and $V(C_3) = \{v_1, v_2\}$. Without loss of generality, let x, u_1, v_1 belong to a packed P_3 and x, u_2, v_2 belong to another packed P_3 . Then u_1v_2 and u_2v_1 must be covered by packed P_{3s} since otherwise Reduction Rule 3 can be applied. Assume that there are two vertices y_1, y_2 such that y_1, u_1, v_2 belong to one packed P_3 and y_2, u_2, v_1 belong to another packed P_3 . Then y_1u_2, y_1v_1 are non-packed non-edges since u_2, v_1 are already incident with two packed P_{3s} respectively. Thus Reduction Rule 3 can be applied, a contradiction. It follows that there is a vertex y such that $yu_2v_1, yu_1v_2 \in \mathcal{H}$. Let C_4 be the proto-cluster to which y belongs. If $|C_4| > 1$, then there must be a non-packed non-edge between C_4 and C_2 and a non-packed non-edge between C_4 and C_3 . Thus Reduction Rule 3 can be applied, a contradiction. Thus

 $|C_4| = 1$. Since u_1, u_2, v_1, v_2, x, y are all incident with two packed P_3 s, the subgraph induced by $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ is isolated from the other parts of the graph. Obviously, $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ forms a connected component C in G. This completes the proof for the lemma.



Figure 6.7: Some examples of Lemma 36. In Case (1), C_1 is separated from C_2 and C_3 , and C_2, C_3, C_4 are merged into a clique of size 5 in $G \triangle S$. In Case (2), C_4 is separated from C_2 and C_3 , and C_1, C_2, C_3 are merged into a clique of size 5 in $G \triangle S$. In Case (3), C_1, C_2 are merged into a clique of size 3 and C_3, C_4 are merged into a clique of size 3 such that these two cliques of size 3 are separated from each other. In Case (4), the instance is a NO-instance. Case (3) and Case (4) are not touched by Lemma 36 but they can be handled by Reduction Rule 4 and 5.

Lemma 37. After applying Reduction Rules 1 - 6 exhaustively, we have an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP. Suppose that S is a solution to $(G, \mathcal{H}, \ell = 0)$. Then there is no clique of size 5 in $G \triangle S$.

Proof. Suppose for contradiction that A is a clique of size 5 in $G \triangle S$. According to Lemma 36, V(A) belong to a connected component of size 6 in the input graph. Then Reduction Rule 4 and Reduction Rule 5 can be applied, a contradiction. This completes the proof for the lemma. \Box

Lemma 38. After applying Reduction Rules 1 - 6 exhaustively, let $(G, \mathcal{H}, \ell = 0)$ be an instance of CEATMP such that the size of every proto-cluster in G is at most 2 and S is a solution to $(G, \mathcal{H}, \ell = 0)$. Suppose that A is a clique of size 4 in $G \triangle S$ and $V(A) = \{x, y, z_1, z_2\}$. Then three vertices of V(A), say x, y, z_2 belong to one packed P_3 in G, and one vertex of x, y, z_2 , say z_2 , with z_1 forms a proto-cluster C_1 of size two in G while x and y form a proto-cluster C_2 of size one and a proto-cluster C_3 of size one in G respectively. Moreover, there are two vertices u and v such that x, u, z_1 belong to a packed P_3 in G, y, v, z_1 belong to another packed P_3 in G. u and v form a proto-cluster C_4 of size one and a proto-cluster C_5 of size one in G respectively.

Proof. For contradiction, suppose that V(A) does not belong to one proto-cluster of size two and two proto-clusters of size one in G. Then there are two cases: (i) two vertices of V(A), say x_1, x_2 , belong to a proto-cluster C_2 of size two and the other two vertices of V(A), say y_1, y_2 , belong to a proto-cluster C_3 of size two. (ii) four vertices x_1, x_2, y_1, y_2 of V(A) belong to four distinct proto-clusters C_1, C_2, C_3 and C_4 respectively.

Case (i): Since C_2 and C_3 need to be fully-covered to form a clique of size four, without loss of generality, assume that there is a vertex u such that u, x_1 and y_1 belong to a packed P_3 . Suppose that there is another vertex u' such that u', x_2 and y_2 belong to a packed P_3 . Since



Figure 6.8: An example of forming a clique of size 4 in $G \triangle S$. Vertices z_1, z_2 form a proto-cluster of size 2 and each vertex of u, v, x, y belongs to a proto-cluster of size 1.

neither u, x_2, y_1 nor u, x_1, y_2 could belong to a packed P_3 , one of the vertex pairs ux_2 and uy_2 must be a non-packed non-edge and Reduction Rule 3 can be applied, a contradiction. Thus u, x_2 and y_2 belong to a packed P_3 . Similarly, we can show that there is another vertex v such that v, x_1, y_2 belong to a packed P_3 and v, x_2, y_1 belong to a packed P_3 . It follows that each vertex of $\{x_1, x_2, y_1, y_2, u, v\}$ is incident with two packed P_3 s. Assume that u and v belong to two different proto-clusters, say C_1 and C_4 respectively. If $|C_1| > 1$ or $|C_4| > 1$, then Reduction Rule 3 can be applied. Thus $|C_1| = |C_4| = 1$. It follows that $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ induces a connected component and Reduction Rule 4 can be applied, a contradiction. Assume that u and v belong to one proto-cluster, say C_1 . If $|C_1| > 2$, then Reduction Rule 3 can be applied. Thus $|C_1| = 2$ and $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ induces a connected component. It follows that Reduction Rule 4 can be applied, a contradiction Rule 3 can be applied. Thus $|C_1| = 2$ and $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ induces a connected component. It follows that Reduction Rule 4 can be applied, a contradiction Rule 3 can be applied. Thus $|C_1| = 2$ and $V(C_1) \cup V(C_2) \cup V(C_3) \cup V(C_4)$ induces a connected component. It follows that Reduction Rule 4 can be applied, a contradiction. Therefore Case (i) does not exist.

Case (ii): Since C_1, C_2, C_3 and C_4 need to be pairwise fully-covered to form a clique of size four, without loss of generality, assume that x_1, x_2, y_1 belong to a packed P_3 . Since x_1y_2 needs to be covered by a packed P_3 , there is another vertex y_3 such that x_1, y_2, y_3 belong to a packed P_3 . The vertex pairs x_2y_2 and y_1y_2 cannot be covered by one packed P_3 since x_2y_1 is already covered by a packed P_3 . Thus x_2y_2 and y_1y_2 need to be covered by two distinct P_3 s respectively. However, y_2 is incident with 3 packed P_3 s, a contradiction. Therefore Case (ii) does not exist.

Next we show that the statement of this lemma is true as in Fig 6.8. Suppose that A is a clique of size 4 in $G \triangle S$. Let $V(A) = \{x, y, z_1, z_2\}$. By the analysis above, we get that two vertices of A belong to a proto-cluster of size two and the other two vertices of A belong to two distinct proto-clusters of size one respectively. Without loss of generality, assume that z_1, z_2 form a proto-cluster C_1 of size two in G while x and y form a proto-cluster C_2 of size one and a proto-cluster C_3 of size one in G respectively. Since there are three vertex pairs, i.e. $\{xy, xz_1, xz_2\}$ between x and $V(A) \setminus \{x\}$, two of the three vertex pairs are covered by one packed P_3 . Without loss of generality, let x, y, z_2 belong to a packed P_3 . Thus there is another vertex usuch that x, u, z_1 belong to a packed P_3 in G. Also yz_1 needs to be covered by a packed P_3 , so there is another vertex v such that y, v, z_1 belong to a packed P_3 . Suppose that u or v belongs to one proto-cluster of size at least two. If uz_1 or vz_1 is a packed edge, Reduction Rule 3 can be applied since either uz_2 or vz_2 must be a non-packed non-edge. If uz_1 and vz_1 are two packed non-edges, ux and vy must be two packed edges. Since x and y are incident with two packed P_{3s} respectively, uy and vx are two non-packed non-edges. Thus Reduction Rule 3 can be applied, a contradiction. It follows that u and v must belong to two distinct proto-clusters. Assume that there is a vertex u' such that u' and u belong to one proto-cluster of size at least two. Since x, z_1 are already incident with two packed P_{3S} respectively, u'x and z_1x must be non-packed

non-edges. Then Reduction Rule 3 can be applied since ux or uz_1 is a packed edge. It follows that u belongs to a proto-cluster of size one, say C_4 . Similarly, we can show that v belongs to a proto-cluster of size one, say C_5 . This completes the proof for the lemma.



Figure 6.9: Examples of Reduction Rule 7. Vertices z_1, z_2 form a proto-cluster of size 2 and each of the other vertices belongs to a proto-cluster of size 1.

Reduction Rule 7. After applying Reduction Rules 1 - 6 exhaustively, let C_1, C_2, C_3, C_4 and C_5 be five proto-clusters such that $V(C_1) = \{z_1, z_2\}, V(C_2) = \{x\}, V(C_3) = \{y\}, V(C_4) = \{u\}, V(C_5) = \{v\}$ and x, y, z_2 belong to a packed P_3, x, u, z_1 belong to a packed P_3, y, v, z_1 belong to a packed P_3 . We list the following reduction approaches under different conditions.

If uz_2 and vz_2 are non-packed non-edges and uz_1, vz_1 are packed non-edges, then

(1) Delete the edges vy and ux, insert an edge to the packed non-edge of the P_3 which covers xy and remove the corresponding packed P_{3s} from \mathcal{H} .

Otherwise, if there is a vertex w such that u, w, z_2 belong to a packed P_3 and wz_2, vz_1 is a packed non-edge, then we do reductions according to the following cases:

- (2) xz_1 and xz_2 are packed non-edges. Return NO.
- (3) xz_1 is a packed non-edge and xz_2 is a packed edge. Delete the edge uw, delete the edges between y and $\{x, z_1, z_2\}$, and add an edge to the non-edge xz_1 . Remove the corresponding packed P_{3s} from \mathcal{H} .
- (4) xz_1 is a packed edge and xz_2 is a packed non-edge. Delete the edge vy, delete the edges between u and $\{y, z_1, z_2\}$, and add an edge to the non-edge xz_2 . Remove the corresponding packed P_{3s} from \mathcal{H} .
- (5) xz_1 and xz_2 are packed edges. Replace the subgraph induced by $\{u, v, w, x, y, z_1, z_2\}$ with two P_{3s} vab and bcw. Remove the four packed P_{3s} incident with one vertex of $\{u, x, y, z_1, z_2\}$ from \mathcal{H} , and add vab and bcw to \mathcal{H} .

Otherwise, if there is a vertex w' such that v, w', z_2 belong to a packed P_3 and $uz_1, w'z_2$ are packed non-edges, then we do reductions according to the following cases:

(6) yz_1 and yz_2 are packed non-edges. Return NO.

- (7) yz_1 is a packed non-edge and yz_2 is a packed edge. Delete the edge vw', delete the edges between x and $\{y, z_1, z_2\}$, and add an edge to the non-edge yz_1 . Remove the corresponding packed P_{3s} from \mathcal{H} .
- (8) yz_1 is a packed edge and yz_2 is a packed non-edge. Delete the edge ux, delete the edges between v and $\{y, z_1, z_2\}$, and add an edge to the non-edge yz_2 . Remove the corresponding packed P_{3s} from \mathcal{H} .
- (9) yz_1 and yz_2 are packed edges. Replace the subgraph induced by $\{u, v, w', x, y, z_1, z_2\}$ with two $P_{3s} w'ab$ and bcu. Remove the four packed P_{3s} incident with one vertex of $\{v, x, y, z_1, z_2\}$ from \mathcal{H} , and add w'ab and bcu to \mathcal{H} .

Lemma 39. Reduction Rule 7 is safe.

Proof. Suppose that $(G, \mathcal{H}, \ell = 0)$ is an instance of CEATMP satisfying the condition of Item (1) of Reduction Rule 7. First, we claim that vz_1 and uz_1 must be packed non-edges. Suppose for contradiction that uz_1 or vz_1 is a packed edge. Since vz_2 and uz_2 are non-packed non-edges as in the assumption, Reduction Rule 3 can be applied, a contradiction. Let F be the set of vertex pairs edited by Item (1) and we can check that F contains exactly one vertex pair of each of the packed P_{3s} incident with one of the vertices of $\{x, y, z_1, z_2\}$. After applying the operations of Item (1) we get an instance $(G' = G \triangle F, \mathcal{H}', \ell = 0)$. We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. On one hand, assume that $(G', \mathcal{H}', \ell = 0)$ has a solution S'. Then $S = S' \cup F$ is a solution to $(G, \mathcal{H}, \ell = 0)$. On the other hand, assume that $(G, \mathcal{H}, \ell = 0)$ has a solution S. Then $G \triangle S$ is a cluster graph. We claim that z_2 is not incident with any other packed P_{3s} except the one covering xy. Suppose that there is another vertex z_3 such that $z_3 z_2$ is a packed edge, then $z_3 z_1$ is a non-packed non-edge since z_1 is already incident with two packed P_{3s} . Then Reduction Rule 3 can be applied, a contradiction. Thus z_2 is not incident with any other packed P_{3s} except the one covering xy. Let $S_1 \subseteq S$ be the set of vertex pairs which are packed edges or packed non-edges of the subgraph of Ginduced by $\{u, v, x, y, z_1, z_2\}$. We claim that $\widehat{S} = S \setminus S_1 \cup F$ is also a solution to $(G, \mathcal{H}, \ell = 0)$. Since S is a solution to $(G, \mathcal{H}, \ell = 0)$, S_1 must contain exactly one vertex pair of each of the packed P_{3s} incident with one of the vertices of $\{x, y, z_1, z_2\}$. Since $F \cap (S \setminus S_1) = \emptyset$, $S_1 \subseteq S$ and $|F| = |S_1|$, we get that $|\widehat{S}| = |S|$. Since $G \triangle S$ is a cluster graph, the subgraph of $G \triangle S$ induced by $V(G) \setminus \{x, y, z_1, z_2\}$ is also a cluster graph G. It follows that $G \triangle \widehat{S}$ is a cluster graph consisting of two isolated parts, i.e. G and the clique of size four formed by $\{x, y, z_1, z_2\}$. It follows that $S \setminus S_1$ is a solution to $(G', \mathcal{H}', \ell = 0)$. As a result, Item (1) is safe.

For the proof of the correctness of Items (2) - (5), we claim that vz_1 and wz_2 are packed non-edges. Suppose for contradiction that vz_1 or wz_2 is a packed edge. Since z_1, z_2 are already incident with two packed P_{3s} , vz_2 and wz_1 must be non-packed non-edges. Then Reduction Rule 3 can be applied, a contradiction. Thus vz_1 and wz_2 are packed non-edges. In a similar way, for the proof of the correctness of Items (6) - (9), we can prove that uz_1 and $w'z_2$ are packed non-edges.

For contradiction, suppose that an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP satisfying the condition of Item (2) of Reduction Rule 7 has a solution S. Since vx is a non-packed non-edge, at least one packed edge of vy, xy belongs to S since otherwise Reduction Rule 1 can be applied. Suppose that $xy \in S$. Then yz_2 becomes a non-packed edge and $yz_1 \notin S$ since otherwise Reduction Rule 1 can be applied. If $vz_1 \in S$, then Reduction Rule 1 can be applied. Thus $vy \in S$ and the subgraph induced by $\{y, z_1, z_2\}$ is a proto-cluster now. Since uy is a non-packed non-edge and uz_1 is a packed edge, Reduction Rule 3 can be applied, which deletes uz_1 and makes ux become a non-packed edge. Now u, x form a proto-cluster of size two. Again by Reduction Rule 3, uz_2 is deleted and uw becomes a non-packed edge. Then Reduction Rule 1 can be applied to the proto-cluster formed by u, w, x, a contradiction. Suppose that $vy \in S$. Then yz_1 becomes a non-packed edge, so $yz_2 \notin S$ since otherwise Reduction Rule 1 can be applied. If $xz_2 \in S$, then $xz_1 \in S$ since otherwise Reduction Rule 1 can be applied. Then uz_1 and ux become non-packed edges. Since uy is a non-packed non-edge, Reduction Rule 1 can be applied to the proto-cluster formed by u, x, y, z_1, z_2 . Thus $xz_2 \notin S$ and $xy \in S$. Then by Reduction Rule 3 uz_1 is deleted and ux becomes a non-packed edge. Now u, x form a proto-cluster of size two. Again by Reduction Rule 3, uz_2 is deleted and uw becomes a non-packed edge. Then Reduction Rule 1 can be applied to the proto-cluster formed by u, w, x, a contradiction. As a result, $(G, \mathcal{H}, \ell = 0)$ is a NO-instance and Item (2) is safe. Given an instance $(\widehat{G}, \widehat{\mathcal{H}}, \ell = 0)$ of CEATMP satisfying the condition of Item (6) of Reduction Rule 7, we can check that the subgraph of \widehat{G} induced by $\{u, v, w', x, y, z_1, z_2\}$ is isomorphic to the subgraph of G induced by $\{u, v, w, x, y, z_1, z_2\}$. Thus by similar analysis, we can prove that Item (6) is safe.

Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP satisfying the condition of Item (3) of Reduction Rule 7, let F be the set of vertex pairs edited by Item (3) and we can check that Fcontains exactly one vertex pair of each of the packed P_{3} s incident with one of the vertices of $\{u, x, y, z_1, z_2\}$. After applying the operations of Item (3) we get an instance $(G' = G \triangle F, \mathcal{H}', \ell =$ 0). We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. On one hand, assume that $(G', \mathcal{H}', \ell = 0)$ has a solution S'. Then $S = S' \cup F$ is a solution to $(G, \mathcal{H}, \ell = 0)$. On the other hand, assume that $(G, \mathcal{H}, \ell = 0)$ has a solution S. We claim that $F \subseteq S$. Suppose for contradiction that $xz_1 \notin S$. Then there are two cases: (1) $ux \in S$ and (2) $uz_1 \in S$. Case (1): If $ux \in S$, then uz_1 becomes a non-packed edge and xz_1 becomes a non-packed non-edge. It follows that $uz_2 \notin S$ and $xz_2 \in S$ since otherwise Reduction Rule 1 can be applied. Since wz_1 is a non-packed non-edge, $wz_2 \notin S$. Thus $uw \in S$. Since u, z_1 and z_2 form a proto-cluster of size 3 now, by Reduction Rule 3, $xz_2 \in S$. If xy is a non-edge and yz_2 is an edge, yz_2 becomes a non-packed edge after xz_2 is deleted. Since uy is a non-packed non-edge, Reduction Rule 1 can be applied, a contradiction. If yz_2 is a non-edge and yx is an edge, then y and x form a cluster of size two. Since yz_1 is a packed edge and uy is a non-packed non-edge, Reduction Rule 3 can be applied and $yz_1 \in S$. Thus vy becomes a non-packed edge. Since vx is a non-packed non-edge, Reduction Rule 1 can be applied, a contradiction. Case (2): If $uz_1 \in S$, then $uz_2 \in S$ since otherwise Reduction Rule 1 can be applied. Thus ux and uwbecome non-packed edges after uz_1 and uz_2 are deleted. Since xw is a non-packed non-edge, Reduction Rule 1 can be applied, a contradiction. It follows that $xz_1 \in S$ and $xz_2, uz_2 \notin F$. Thus $uw \in S$ since otherwise Reduction Rule 1 can be applied. Now u, x, z_1, z_2 belong to one proto-cluster. Then by Reduction Rule 3, $yz_1 \in S$. u, x, z_1, z_2 are in one proto-cluster. Since uy is a non-packed non-edge, if xy is an edge, $xy \in S$ since otherwise Reduction Rule 1 can be applied. Then by Reduction Rule 3, $yz_1 \in S$. Similarly, If yz_2 is an edge, $yz_2 \in S$ since otherwise Reduction Rule 1 can be applied. Then by Reduction Rule 3, $yz_1 \in S$. As a result, $F \subseteq S$. We claim that $\widehat{S} = S \setminus F$ is a solution to $(G', \mathcal{H}', \ell = 0)$. Since u, x, y, z_1, z_2 are already incident with two packed P_{3s} , $\{u, x, y, z_1, z_2\}$ are isolated from $V(G) \setminus \{u, v, w, x, y, z_1, z_2\}$ in G. It follows that in $G \triangle S$, v, y belong to a clique of size two, u, x, z_1, z_2 belong to a clique of size four and $V(G) \setminus \{u, v, x, y, z_1, z_2\}$ induces a cluster graph such that there are no edges between $\{u, v, x, y, z_1, z_2\}$ and $V(G) \setminus \{u, v, x, y, z_1, z_2\}$. Thus $G' \triangle \widehat{S}$ is a cluster graph and $|\widehat{S}| = |\mathcal{H}'|$. As a result, Item (3) is safe. By very similar approaches, we can prove that Items (4), (7) and (8) are safe.

Let $(G, \mathcal{H}, \ell = 0)$ be an instance of CEATMP satisfying the condition of Item (5) of Reduction Rule 7. Since xz_1, xz_2 are packed edges, there are two packed edges between y and $\{x, z_1, z_2\}$ and let the set of the two packed edges be W_y . Also, there are two packed edges between uand $\{x, z_1, z_2\}$ and let the set of the two packed edges be W_u . After applying the operations of Item (5) we get an instance $(G', \mathcal{H}', \ell = 0)$. We claim that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. For completeness, suppose that $(G', \mathcal{H}', \ell = 0)$ has a solution S'. Thus $G' \triangle S'$ is a cluster graph. There are four possible cases: (1) $\{ab, cw\} \subseteq S'$. Since $G' \triangle S'$ is a cluster graph, the subgraph of $G' \triangle S'$ induced by $V(G) \setminus \{b, c\}$ is also a cluster graph. Let $S = S' \setminus \{ab, cw\} \cup W_y \cup \{ux, uz_1, uz_2\} \setminus W_u \cup \{uw\}$. Then $G \triangle S$ is also a cluster graph and $|S| = |\mathcal{H}|$. Thus $(G, \mathcal{H}, \ell = 0)$ is a YES-instance. (2) $\{va, bc\} \subseteq S'$. We can show that $(G, \mathcal{H}, \ell = 0)$ has a solution in a very similar way to that of Case (1). (3) $\{vb, bc\} \subseteq S'$. Since vertices a, b and c are not adjacent to any vertex of $V(G') \setminus \{a, b, c, v, w\}$ in $G', \{v, a, b\}$ induces a triangle which is a connected component and $\{c, w\}$ induces a clique of size two which is also a connected component in $G' \triangle S'$. Let $S = S' \setminus \{vb, bc\} \cup W_u \cup \{yx, yz_1, yz_2\} \setminus W_u \cup \{vy\}$. It follows that $G \triangle S$ is a cluster graph and $|S| = |\mathcal{H}|$. Thus $(G, \mathcal{H}, \ell = 0)$ is a YES-instance. (4) $\{ab, bw\} \subseteq S'$. We can show that $(G, \mathcal{H}, \ell = 0)$ has a solution in a very similar way to that of Case (3). This completes the proof for completeness. For soundness, suppose that $(G, \mathcal{H}, \ell = 0)$ has a solution S. We can check that there are only two possible cases: (1) $F_1 = W_u \cup \{yx, yz_1, yz_2\} \setminus W_y \cup \{vy\} \subseteq S$. Since vertices u, x, y, z_1, z_2 are not adjacent to any vertex of $V(G) \setminus \{u, v, w, x, y, z_1, z_2\}$ in $G, \{x, y, z_1, z_2\}$ induces a clique of size four which is a connected component and $\{u, w\}$ induces a clique of size two which is also a connected component in $G \triangle S$. Let $S' = S \setminus F_1 \cup \{va, bc\}$. It follows that $G' \triangle S'$ is a cluster graph and $|S'| = |\mathcal{H}'|$. Thus $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. (2) $F_2 = W_y \cup \{ux, uz_1, uz_2\} \setminus W_u \cup \{uw\} \subseteq S$. Since vertices u, x, y, z_1, z_2 are not adjacent to any vertex of $V(G) \setminus \{u, v, w, x, y, z_1, z_2\}$ in $G, \{u, x, z_1, z_2\}$ induces a clique of size four which is a connected component and $\{v, y\}$ induces a clique of size two which is also a connected component in $G \triangle S$. Let $S' = S \setminus F_2 \cup \{ab, cw\}$. It follows that $G' \triangle S'$ is a cluster graph and $|S'| = |\mathcal{H}'|$. Thus $(G', \mathcal{H}', \ell = 0)$ is a YES-instance. As a result, Item (5) is safe. In a very similar way, we can prove that Item (9) is safe.

As a result, Reduction Rule 7 is safe. This completes the proof for the lemma.

After applying Reduction Rule 7, Reduction Rule 5 can be applied to remove the isolated cliques.

Lemma 40. After applying Reduction Rules 1 - 7 exhaustively, let $(G, \mathcal{H}, \ell = 0)$ be an instance of CEATMP which has a solution S. Then there is no clique of size at least 4 in $G \triangle S$.

Proof. By Lemma 33, 35 and 37, there is no clique of size at least 5 in $G \triangle S$. Suppose for contradiction that A is a clique of size 4 in $G \triangle S$. Let $V(A) = \{x, y, z_1, z_2\}$. Then by Lemma 38, three vertices of V(A), say x, y, z_2 belong to one packed P_3 in G, and one vertex of x, y, z_2 , say z_2 , with z_1 forms a proto-cluster C_1 of size two in G while x and y form a proto-cluster C_2 of size one and a proto-cluster C_3 of size one in G respectively. Moreover, there are two vertices uand v such that x, u, z_1 belong to a packed P_3 in G, y, v, z_1 belong to another packed P_3 in G, and u and v form a proto-cluster C_4 of size one and C_5 of size one in G respectively. There are six cases: (1) uz_2 and vz_2 are non-packed non-edges. Then Item (1) of Reduction Rule 7 can be applied. (2) uz_2 is a packed edge and vz_2 is a non-packed non-edge. Then one of Items (2) - (5) can be applied. (3) uz_2 is a packed non-edge and vz_2 is a non-packed non-edge. Then Reduction Rule 3 can be applied. (4) vz_2 is a packed edge and uz_2 is a non-packed non-edge. Then one of Items (6) - (9) can be applied. (5) vz_2 is a packed non-edge and uz_2 is a non-packed non-edge. Then Reduction Rule 3 can be applied. (6) u, v, z_2 belong to a packed P_3 . Since every vertex of $\{u, v, x, y, z_1, z_2\}$ is already incident with two packed P_{3s} , the subgraph induced by $\{u, v, x, y, z_1, z_2\}$ is isolated from other parts of the graph and it's not hard to see that it is a connected component. Thus Reduction Rule 4 can be applied. It follows that there is no clique of size 4 in $G \triangle S$. This completes the proof for the lemma.

6.2.2 Polynomial-time Reduction to 2-SAT Problem

First, we introduce a new problem called CLUSTER DELETION ABOVE MODIFICATION-DISJOINT P_3 PACKING. The formal definition is as follows:

CLUSTER DELETION ABOVE MODIFICATION-DISJOINT P_3 PACKING (CDAMP) **Input:** A graph G = (V, E), a modification-disjoint packing \mathcal{H} of induced P_{38} of G, and a nonnegative integer ℓ . **Question:** Is there a cluster deletion set, i.e. a set of edges $S \subseteq E$ so that $G' = (V, E \setminus S)$ is a union of disjoint cliques, with $|S| - |\mathcal{H}| \leq \ell$?

Note that in the definition of CDAMP, the P_{3s} of \mathcal{H} are still modification-disjoint although the solution to the problem contains only edge deletions.

Lemma 41. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP, after applying Reduction Rules 1 - 7 exhaustively, we get an instance $(G', \mathcal{H}', \ell = 0)$ of CEATMP. Then $(G, \mathcal{H}, \ell = 0)$ is a YES-instance of CEATMP if and only if $(G', \mathcal{H}', \ell = 0)$ is a YES-instance of CDAMP.

Proof. (Completeness) Assume that $(G', \mathcal{H}', \ell = 0)$ is a YES-instance of CDAMP and S' is a cluster deletion set of size $|\mathcal{H}'|$. Obviously S' is also a cluster editing set of G'. Thus $(G', \mathcal{H}', \ell = 0)$ is a YES-instance of CEATMP. It follows that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance of CEATMP.

(Soundness) Assume that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance of CEATMP. Then $(G', \mathcal{H}', \ell = 0)$ is a YES-instance of CEATMP and let S' be its solution. By Lemma 40, there is no clique of size at least four in $G' \triangle S'$. By Observation 5, every non-edge of S' is a packed non-edge. Let $uw \in S'$ be a non-edge of G' which is covered by a P_3 uvw of \mathcal{H}' . Then in $G' \triangle S'$, $\{u, v, w\}$ must induces a triangle which is a connected component. It follows that $S' \setminus \{uw\} \cup \{uv\}$ is also a solution to $(G', \mathcal{H}', \ell = 0)$. Let $S_1 \subseteq S'$ be the set of non-edges of S'. Then there is a set S_2 of packed edges of G' such that $S' \setminus S_1 \cup S_2$ is a cluster deletion set for G' of size $|\mathcal{H}'|$. Thus $(G', \mathcal{H}', \ell = 0)$ is a YES-instance of CDAMP. This completes the proof for the lemma.

Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP, after applying Reduction Rules 1 - 7 exhaustively, we get an instance $(G', \mathcal{H}', \ell = 0)$ of CDAMP. Let $E_c \subseteq E(G')$ be the set of edges covered by some P_3 of \mathcal{H}' and let $\lambda = 2|\mathcal{H}'|$. We fix an arbitrary ordering of the edges of E_c and label these edges by $e_0, e_1, ..., e_{\lambda-1}$ according to this ordering. We construct an instance of 2-SAT with λ variables $x_0, x_1, ..., x_{\lambda-1}$ as follows. First initialize the 2-SAT formula $\Phi =$ true. For each induced $P_3 xyz \in \mathcal{H}'$, if $e_i = xy, e_j = yz$, then update $\Phi \leftarrow \Phi \land (x_i \lor x_j) \land (\neg x_i \lor \neg x_j)$. For each induced $P_3 uvw$ in G' such that uv and vw belong to two distinct P_3 s of \mathcal{H}' respectively, if $uv = e_p$ and $vw = e_q$, then update $\Phi \leftarrow \Phi \land (x_p \lor x_q)$. This completes the construction of the 2-SAT instance.

Lemma 42. Given an instance $(G, \mathcal{H}, \ell = 0)$ of CEATMP, after applying Reduction Rules 1 - 7 exhaustively, we get an instance $(G', \mathcal{H}', \ell = 0)$ of CDAMP. We construct a 2-SAT formula Φ as described above. Then $(G, \mathcal{H}, \ell = 0)$ is a YES-instance if and only if Φ is satisfiable.

Proof. (Completeness) Assume that Φ is satisfiable and let α be a satisfying assignment to Φ . Let $S' = \{e_i \mid \alpha(x_i) = \mathsf{true}\}$. We will show that S' is a cluster deletion set for G' of size $|\mathcal{H}'|$. First we claim that for every induced $P_3 xyz \in \mathcal{H}'$, exactly one edge of xy and yz belongs to S'. Assume that $e_i = xy$ and $e_j = yz$ for some $i, j \in \{0, ..., \lambda - 1\}$. Since $(x_i \lor x_j)$ and $(\neg x_i \lor \neg x_j)$ are two clauses of Φ and α is a satisfying assignment to Φ , either $x_i = \mathsf{false}, x_j = \mathsf{true}$ or $x_i = \mathsf{true}, x_j = \mathsf{false}$ holds. Thus the claim is true and $|S'| = |\mathcal{H}'|$. Then for every induced P_3 uvw in G' such that uv and vw belong to two distinct P_3 s of \mathcal{H}' , let $uv = e_p$ and $vw = e_q$ for some $p, q \in \{0, ..., \lambda - 1\}$. By the construction, $(x_p \lor x_q)$ is a clause of Φ so it is satisfied by α . Thus at least one edge of uvw belongs to S'.

We claim that there is no induced $P_3 xyz$ in $G' \triangle S'$ such that both xy and yz are packed edges in G'. Suppose for contradiction that there is an induced $P_3 xyz$ in $G' \triangle S'$ such that both xy and yz are packed edges in G'. Then xy and yz must be covered by two distinct packed P_{3s} since otherwise xy or yz belongs to S' by the analysis of last paragraph. Besides, we claim that xz must be a packed edge covered by another packed P_3 in G', i.e. xy, yz and xz are covered by three distinct packed P_{3s} in G' since otherwise xyz is an induced P_3 in G' and xy or yz belongs to S' by the analysis of last paragraph. If xz is a non-packed edge in G', then xz is an edge in $G' \triangle S'$ since xz cannot be deleted by the solution, a contradiction. Suppose that xz is a packed edge and without loss of generality, xz is covered by $uxz \in \mathcal{H}'$. Then $ux, xy, yz \notin S'$. Since y is already incident with two packed P_{3s} , uy is either a non-packed non-edge in G' or a non-packed edge in G'. If uy is a non-packed non-edge in G', then uxy is an induced P_3 in G'. Let $ux = e_i$ and $xy = e_j$, then the clause $(x_i \lor x_j)$ of Φ is not satisfied, a contradiction. If uy is a non-packed edge, then Reduction Rule 3, 4 or 7 can be applied, a contradiction.

By corollary 3, there is no proto-cluster of size at least three in G'. Thus there is no induced P_3 abc in $G' \triangle S'$ such that ab and bc are non-packed edges in G'.

Finally, we claim that there is no induced P_3 in $G' \triangle S'$ such that one edge of this P_3 is a non-packed edge in G' and the other edge is a packed edge in G'. Suppose for contradiction that there is such a $P_3 uvw$ in $G' \triangle S'$ such that uv is a non-packed edge and vw is a packed edge in G'. Then there is another vertex x such that v, w, x belong to a packed P_3 in G'. Since Reduction Rule 3 cannot be applied to $(G', \mathcal{H}', \ell = 0)$, then uw must be covered by a packed P_3 in G', i.e. there is a vertex y such that u, w, y belong to a packed P_3 in G'. Suppose that wuy is a packed P_3 in G'. Since Reduction Rule 3 cannot be applied, vy must be covered by a packed P_3 , i.e. there is a vertex z such that v, y, z belong to a packed P_3 in G', then Reduction Rule 3, 4 or 7 can be applied, a contradiction. It follows that $uwy \in \mathcal{H}'$. Similarly, suppose that wvx is a packed P_3 in G'. Since Reduction Rule 3 cannot be applied, ux must be covered by a packed P_3 , i.e. there is a vertex z' such that u, x, z' belong to a packed P_3 , then Reduction Rule 3, 4 or 7 can be applied, a contradiction. It follows that that $vwx \in \mathcal{H}'$. Since uvw is an induced P_3 in $G' \triangle S'$, $uw, wx \in S$. If vy is a non-edge in G', then vwy is an induced P_3 in G'. Assume that $vw = e_p$ and $wy = e_q$. Then the assignment α cannot satisfy $(x_p \lor x_q)$ which is a clause of Φ , contradicting that α is a satisfying assignment to Φ . Otherwise, suppose that vy is a packed edge in G'. If there is a vertex a such that $vya \in \mathcal{H}'$, then Reduction Rule 4 or 7 can be applied. If there is a vertex a' such that $yva' \in \mathcal{H}'$, then either ua' is a non-packed non-edge, which triggers Reduction Rule 3, or ua' is covered by a packed P_3 , which triggers Reduction Rule 3, 4 or 7. It follows that there is no induced P_3 in $G' \triangle S'$ such that one edge of this P_3 is a non-packed edge in G' and the other edge of this P_3 is a packed edge in G'. As a result, S' is a solution to the instance $(G', \mathcal{H}', \ell = 0)$ of CDAMP. By Lemma 41, $(G, \mathcal{H}, \ell = 0)$ is a YES-instance.

(Soundness) Assume that $(G, \mathcal{H}, \ell = 0)$ is a YES-instance. By Lemma 41, $(G', \mathcal{H}', \ell = 0)$ is a YES-instance of CDAMP and let S' be a cluster deletion set for G' of size $|\mathcal{H}'|$. Let α be an assignment to Φ such that $\alpha(x_i) = \mathsf{true}$ if and only if $e_i \in S'$ for $i = 0, ..., \lambda - 1$. We claim that α is a satisfying assignment to Φ . Since $|S'| = |\mathcal{H}'|$ and the P_3 s of \mathcal{H}' are modification-disjoint, S' contains exactly one edge of every packed P_3 of \mathcal{H}' . It follows that for every $P_3 xyz \in \mathcal{H}'$ $(xy = e_i, yz = e_j)$, the two clauses $(x_i \vee x_j)$ and $(\neg x_i \vee \neg x_j)$ are satisfied. Since S' is a solution to $(G', \mathcal{H}', \ell = 0)$, there is no induced P_3 in $G' \triangle S'$. Thus for every induced $P_3 uvw$ in G' such that uv and vw belong to two distinct packed P_{3s} $(uv = e_p, vw = e_q)$ respectively, at least one edge of $\{uv, vw\}$ belongs to S' and the clause $(x_p \vee x_q)$ is satisfied. As a result, α is a satisfying assignment to Φ . This concludes the proof for the lemma. \Box **Theorem 16** (Restated). CLUSTER EDITING ABOVE TWO-RESTRICTED MODIFICATION-DISJOINT P_3 PACKING can be solved in polynomial time when $\ell = 0$, that is, when no excess edits are allowed.

Proof. By Lemma 42, given an instance of $(G, \mathcal{H}, \ell = 0)$ of CEATMP, after applying Reduction Rules 1 - 7 exhaustively, we reduce it to an equivalent instance of 2-SAT in polynomial time. Then we can decide the 2-SAT instance by invoking the algorithm for 2-SAT. It is well-known that 2-SAT can be solved in polynomial time. This completes the proof for the theorem. \Box

Chapter 7

Hardness of Metric Dimension in Graphs of Constant Treewidth

In this chapter, we show that METRIC DIMENSION is NP-hard on graphs of treewidth at most 24. Recall that the METRIC DIMENSION problem asks for a minimum-sized resolving set in a given (unweighted, undirected) graph G. Here, a set $S \subseteq V(G)$ is resolving if for any two distinct vertices $u, v \in V(G)$, there exists a vertex $w \in S$ such that $\operatorname{dist}_G(u, w) \neq \operatorname{dist}_G(v, w)$.

Organization of this chapter. In Section 7.1, we present an NP-hardness reduction from 3-DIMENSIONAL MATCHING to *n*-MULTICOLORED RESOLVING SET. In Section 4.2, we present a polynomial-time reduction from an instance of *n*-MULTICOLORED RESOLVING SET constructed in Section 7.1 to an instance of METRIC DIMENSION. Finally, we prove that METRIC DIMENSION is NP-hard on graphs of treewidth at most 24.

Additional notions for this chapter. We define the length of a path P to be the number of edges on the path and denote it by |P|. For two vertices $u, v \in V(G)$, let P(u, v) be a path from u to v in G. Note that P(u, v) and P(v, u) denote the same path. We denote the neighbor of u on P(u, v) by $N_u(u, v)$ (or $N_u(v, u)$). Similarly, if there is a path which is named as, for example, $P^h(i, j, x)$ such that u is one endpoint of $P^h(i, j, x)$, we denote the neighbor of u on $P^h(i, j, x)$ by $N_u^h(i, j, x)$. In an undirected graph G, sometimes we abuse the notation in the sense that for a path P, we use P to denote the path or the vertex set of the path just for simplicity. The meaning should be clear in the context. For two vertices $u, v \in V(G)$, we define the distance between u and v to be the length of any shortest path from u to v, denoted by $\text{dist}_G(u, v)$. Note that we use dist(u, v) to denote the distance between u and v nostly if the graph is clear in the context. For a path P of even length with two endpoints u and v, let w be the vertex on P such that the length of the subpath of P from u to w equals the length of the subpath of P from w to v. Then we call w the *middle vertex* of P and denote it by mid(P).

7.1 Reduction from 3-Dimensional Matching to Multicolored Resolving Set

Bonnet and Purohit introduced k-MULTICOLORED RESOLVING SET as an intermediate problem in order to show the W[1]-hardness of METRIC DIMENSION parameterized by treewidth [17]. *k*-MULTICOLORED RESOLVING SET **Input:** An undirected graph G = (V, E), an integer k, a set $\chi = \{X_1, ..., X_k\}$ where $X_1, ..., X_k$ are disjoint subsets of V(G) and a set $\mathcal{P} = \{\{x_1, y_1\}, ..., \{x_h, y_h\}\}$ where $\{x_1, y_1\}, ..., \{x_h, y_h\}$ are vertex pairs of G. **Question:** Is there a set of k vertices S such that (i) $|S \cap X_i| = 1$ for every i = 1, ..., k, and (ii) for every $\ell \in \{1, ..., h\}$, there exists a vertex $v \in S$ such that $\operatorname{dist}(v, x_\ell) \neq \operatorname{dist}(v, y_\ell)$.

We show that this problem is NP-hard on graphs of constant treewidth. We make a reduction from 3-DIMENSIONAL MATCHING, which is well-known to be NP-hard [76].

3-DIMENSIONAL MATCHING

Input: the universe $U = \{1, 2, 3\} \times [n]$ and a set $\mathcal{F} = \{A_1, ..., A_m\}$ of tuples such that for every $j \in [m]$, the tuple $A_j = \{(1, x), (2, y), (3, z)\}$ where $(1, x), (2, y), (3, z) \in U$. **Question:** are there *n* tuples $A_{j_1}, ..., A_{j_n}$ such that $\bigcup_{h=1}^n A_{j_h} = U$.

Given an instance (U, \mathcal{F}) of 3-DIMENSIONAL MATCHING with the universe $U = \{1, 2, 3\} \times [n]$ and a set \mathcal{F} of m tuples $A_1, ..., A_m \subseteq U$, we construct an instance $(G, n, \chi, \mathcal{P})$ of n-MULTICOLORED RESOLVING SET as follows. First, we create m vertices $s_i^1, ..., s_i^m$ as X_i for each $i \in [n]$. Let $\chi = \{X_1, ..., X_n\}$ and $X = \bigcup_{i=1}^n X_i$. Then we create n vertex pairs $\{u_r^1, v_r^1\}, ..., \{u_r^n, v_r^n\}$ for each $r \in \{1, 2, 3\}$ and let $\mathcal{P}_r = \{\{u_r^i, v_r^i\} | i = 1, ..., n\}$. We create 3 vertices a_r, b_r, c_r and let $W_r = \{a_r, b_r, c_r\}$ for each $r \in \{1, 2, 3\}$. Let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$ and $W = W_1 \cup W_2 \cup W_3$. Finally, let M = 40(n + 1). For each tuple $A_j = \{(1, x), (2, y), (3, z)\}$ $(j \in [m], x, y, z \in [n])$ of the given instance and each integer $i \in [n]$, we link s_i^j to a_1, b_1, c_1 with three paths $P(s_i^j, a_1), P(s_i^j, b_1), P(s_i^j, c_1)$ of lengths $\frac{M}{2} + 10x, \frac{M}{2} + 5x + 1$ and $\frac{M}{2} - 10x$ respectively, link s_i^j to a_2, b_2, c_2 with three paths $P(s_i^j, a_2), P(s_i^j, b_2), P(s_i^j, c_2)$ of lengths $\frac{M}{2} + 10y, \frac{M}{2} + 5y + 1$ and $\frac{M}{2} - 10y$ respectively, and link s_i^j to a_3, b_3, c_3 with three paths $P(s_i^j, a_3), P(s_i^j, b_3), P(s_i^j, c_3)$ of lengths $\frac{M}{2} + 10z, \frac{M}{2} + 5z + 1$ and $\frac{M}{2} - 10z$ respectively. For every vertex pair $\{u_r^p, v_r^p\}$ $(p \in [n], r \in \{1, 2, 3\})$, we link u_r^p to a_r, b_r, c_r with three paths $P(u_r^p, a_r), P(u_r^p, b_r), P(u_r^p, c_r)$ of lengths $\frac{M}{2} - 10p, \frac{M}{2} - 5p - 1$ and $\frac{M}{2} - 10p, \frac{M}{2} - 5p - 1$ and $\frac{M}{2} - 10p, \frac{M}{2} - 5p - 2$ and $\frac{M}{2} + 10p$ respectively. This finishes the construction.

Lemma 43. For an arbitrary vertex pair $\{u_r^x, v_r^x\} \in \mathcal{P}$ $(r \in \{1, 2, 3\}, x \in [n]), \{u_r^x, v_r^x\}$ is resolved by s_i^j $(i \in [n], j \in [m])$ if and only if $(r, x) \in A_j$.

Proof. On one hand, suppose that $(r, x) \in A_j$. For an arbitrary $i \in [n]$, the three paths from s_i^j to u_r^x via a_r, b_r and c_r have lengths M, M + 1 and M respectively. The three paths from s_i^j to v_r^x via a_r, b_r and c_r have lengths M, M - 1 and M respectively. Note that there could be other paths from s_i^j to v_r^x or u_r^x that go repeatedly between vertices in X and vertices in W. However, the lengths of such paths are at least M - 20n + M - 10n > M. As a result, the shortest paths from s_i^j to u_r^x and v_r^x are of lengths M and M - 1 respectively. Thus $\{u_r^x, v_r^x\}$ is resolved by s_i^j .

On the other hand, for an arbitrary tuple $A_i = \{(1, p_1), (2, p_2), (3, p_3)\}$, the paths from the vertex s_i^j $(i \in [n])$ to u_r^x $(r \in \{1, 2, 3\})$ via a_r, b_r and c_r have lengths $M + 10(p_r - x), M + 5(p_r - x) + 1$ and $M - 10(p_r - x)$ respectively. The paths from the vertex s_i^j $(i \in [n])$ to v_r^x $(r \in \{1, 2, 3\})$ via a_r, b_r and c_r have lengths $M + 10(p_r - x), M + 5(p_r - x) - 1$ and $M - 10(p_r - x)$ respectively. Note that the paths from s_i^j to u_r^x $(or v_r^x)$ that go repeatedly between the vertices

in X and the vertices in W have lengths at least M - 20n + M - 10n > M + 10n. They are not the shortest paths from s_i^j to u_r^x (or v_r^x). If $p_r < x$, the shortest paths from s_i^j to u_r^x and v_r^x both have lengths $M + 10(p_r - x)$. If $p_r > x$, the shortest paths from s_i^j to u_r^x and v_r^x both have lengths $M - 10(p_r - x)$. If $p_r = x$, the shortest paths from s_i^j to u_r^x and v_r^x have lengths M and M - 1 respectively. As a result, if $\{u_r^x, v_r^x\}$ is resolved by s_i^j , then $p_r = x$. According to the construction, $(r, x) \in A_j$.

Lemma 44. The constructed instance $(G, n, \chi, \mathcal{P})$ of n-MULTICOLORED RESOLVING SET is a yes-instance if and only if the given instance (U, \mathcal{F}) of 3-DIMENSIONAL MATCHING is a yes-instance.

Proof. (\Leftarrow) For an arbitrary tuple $A_i = \{(1, x), (2, y), (3, z)\}$, according to Lemma 43, pairs $\{u_1^x, v_1^x\}, \{u_2^y, v_2^y\}$ and $\{u_3^z, v_3^z\}$ are all resolved by s_i^j for every $i \in [n]$. Suppose that the given instance of 3-DIMENSIONAL MATCHING is a yes-instance, that is, there exists A_{j_1}, \ldots, A_{j_n} satisfying that $\bigcup_{h=1}^n A_{j_h} = U$. It follows that $S = \{s_h^{j_h} : h \in [n]\}$ is a solution for the constructed instance of *n*-MULTICOLORED RESOLVING SET.

(⇒) Let $S = \{s_h^{j_h} : h \in [n]\}$ be a solution for the constructed instance of *n*-MULTICOLORED RESOLVING SET. For an arbitrary pair $\{u_r^x, v_r^x\}$, since it is resolved by some $s_{h'}^{j_{h'}} \in S$, according to Lemma 43, $(r, x) \in A_{j_{h'}}$. As a result, $\{A_{j_h} : h \in [n]\}$ is a solution for the instance of 3-DIMENSIONAL MATCHING.

It is well-known that the treewidth of a graph is bounded by the size of a minimum feedback vertex set of the graph. We can easily observe that W is a feedback vertex set of size 9 for G. It follows that the treewidth of G is at most 10. Then we have the following lemma.

Lemma 45. k-Multicolored Resolving Set is NP-hard even on graphs of treewidth at most 10.

7.2 Reduction from Multicolored Resolving Set to Metric Dimension

In this section, we create in polynomial time an instance (G', k) of METRIC DIMENSION which is equivalent to the instance $(G, n, \chi, \mathcal{P})$ of *n*-MULTICOLORED RESOLVING SET we created in last section. Roughly speaking, the reduction consists in adding gadgets on base of the constructed instance $(G, n, \chi, \mathcal{P})$ to solve the following two issues: (1) the solution for METRIC DIMENSION could contain vertices not in any set of χ or more than one vertex from some set of χ , which could spoil the desired reduction; (2) we did not make sure that every pair of distinct vertices are resolved by the solution in an instance of *n*-MULTICOLORED RESOLVING SET. We find that similar strategies to those in [17] can be used to solve these two issues. More specifically, we solve the first issue by adding *forced set gadgets*. One such gadget contains two pairs of vertices such that they are only resolved simultaneously by a vertex of X_i (where it is attached). We solve the second issue by adding *forced vertex gadgets*. One such gadget contains a pair of pendant neighboring vertices (false twins). Obviously one vertex of the false twins has to be chosen in the solution. Thus the chosen vertices (*forced vertices*) are designed to resolve the remaining unresolved vertex pairs. Besides, we need to add a number of extra paths and set appropriate budget k to make sure that the reduction works as described above.

7.2.1 Construction of the forced set gadgets

Let $(G, n, \chi, \mathcal{P})$ be an instance of *n*-MULTICOLORED RESOLVING SET that we created in last section. For every $X_i \in \chi$ $(i \in [n])$, we add two pairs of isolated vertices $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$.

Then we add two vertices π_i^1 and π_i^2 such that p_i^1, q_i^1 are adjacent to π_i^1, p_i^2, q_i^2 are adjacent to π_i^2 . The vertex triples p_i^1, q_i^1, π_i^1 and p_i^2, q_i^2, π_i^2 $(i \in [n])$ form a forced set gadget. Then we create a path $P(s_i^j, p_i^1)$ of length 20(n+1) from s_i^j to p_i^1 and create a path $P(s_i^j, p_i^2)$ of length 20(n+1) from s_i^j to p_i^2 for each $i \in [n], j \in [m]$. In order to make sure that a vertex can resolve p_i^1, q_i^1 and p_i^2, q_i^2 simultaneously if and only if it belongs to X_i , we need to create 4 paths of length 20(n+1) from π_i^1 to $N_{s_i^j}(s_i^j, p_i^2)$ respectively for each $i \in [n], j \in [m]$ and $r \in \{1, 2, 3\}$. For simplicity, we name the four paths as $P^1(i, j, a_r)$, $P^1(i, j, b_r)$, $P^1(i, j, c_r)$ and $P^1(i, j, p_i^2)$ respectively. Symmetrically, we need to create 4 paths of length 20(n+1) from π_i^2 to $N_{s_i^j}(s_i^j, c_r)$ and $P^2(i, j, p_i^1)$ respectively. Let $\Pi^h(i, j, r) = \{P^h(i, j, a_r), P^h(i, j, b_r), P^h(i, j, c_r), P^h(i, j, p_i^{3-h})\}$ for $i \in [n], j \in [m], r \in \{1, 2, 3\}$.

This completes the construction of the first phase.

7.2.2 Construction of the forced vertex gadgets

A forced vertex gadget consists of a triangle, namely three vertices such that each vertex is adjacent to the other two vertices. Two vertices of the triangle are false twins whose degrees are exactly 2 and we call the other vertex in the triangle the *connecting vertex* of the gadget. When we say that we add a forced vertex gadget F to a vertex v, we mean that we create a forced vertex gadget F such that v is identified with the connecting vertex of F. For each $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$, we add a forced vertex gadget $F^h(i, j, a_r)$ to $N^h_{\pi_h}(i, j, a_r)$, $F^h(i, j, b_r)$ to $N^h_{\pi_h}(i, j, c_r)$ to $N^h_{\pi_h}(i, j, c_r)$ and $F^h(i, j, p_i^{3-h})$ to $N^h_{\pi_h}(i, j, p_i^{3-h})$.

In order to make sure that $f^h(i, j, b_r)$ for $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$ does not resolve any vertex pair of \mathcal{P} , we create a path $P(\pi_i^h, a_r)$ and a path $P(\pi_i^h, c_r)$ both of length 10(n+1) for $i \in [n], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$.

For each $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$, we add a forced vertex gadget $F(\pi_i^h, a_r)$ to $N_{a_r}(\pi_i^h, a_r)$ and a forced vertex gadget $F(\pi_i^h, c_r)$ to $N_{c_r}(\pi_i^h, c_r)$. For each $i \in [n], j \in [m], r \in \{1, 2, 3\}$, we add a forced vertex gadget $F(s_i^j, a_r)$ to $N_{a_r}(s_i^j, a_r)$ and a forced vertex gadget $F(s_i^j, c_r)$ to $N_{c_r}(s_i^j, a_r)$ and a forced vertex gadget $F(s_i^j, c_r)$ to $N_{c_r}(s_i^j, c_r)$.

Let $\operatorname{mid}(P^h(i, j, p_i^{3-h}))$ be the middle vertex of $P^h(i, j, p_i^{3-h})$ for $i \in [n], j \in [m], h \in \{1, 2\}$. In order to make sure that $f^h(i, j, p_i^{3-h})$ does not resolve the vertex pair $\{p_i^{3-h}, q_i^{3-h}\}$, create a path $P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ from q_i^h to $\operatorname{mid}(P^{3-h}(i, j, p_i^h))$ of length $|P^{3-h}(i, j, p_i^h)|/2 + |P(s_i^j, p_i^h)| - 1$. Then add a forced vertex gadget $F^{mid}(i, j, h)$ to $\operatorname{mid}(P^h(i, j, p_i^{3-h}))$.

For $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$, add a forced vertex gadget $F^{ecc}(i, j, h, r)$ to the vertex $x \in P^h(i, j, a_r)$ such that $dist(\pi_i^h, x) = 10(n+1) + 1$.

For each $i \in [n], r \in \{1, 2, 3\}$, create two forced vertex gadgets $F^1(u_r^i, v_r^i)$ and $F^2(u_r^i, v_r^i)$ for the vertex pair $\{u_r^i, v_r^i\} \in \mathcal{P}_r$. Then create an edge from the connecting vertex of $F^1(u_r^i, v_r^i)$ to u_r^i , to v_r^i , to $N_{u_r^i}(a_r, u_r^i)$ and to $N_{u_r^i}(c_r, u_r^i)$ respectively for $i \in [n], r \in \{1, 2, 3\}$. Create an edge from the connecting vertex of $F^2(u_r^i, v_r^i)$ to u_r^i , to v_r^i , to the vertex x such that $x \in P(a_r, u_r^i)$ and dist $(x, u_r^i) = 2$, and to the vertex y such that $y \in P(c_r, u_r^i)$ and dist $(y, u_r^i) = 2$. This completes the construction of the second phase.

Finally, let G' be the graph constructed by above two phases and set k = 34nm + 19n. This finishes constructing the instance (G', k) of METRIC DIMENSION. Figure 7.1 shows a part of G'.



Figure 7.1: An example showing a part of G'. Triangles represent corresponding forced vertex gadgets. For clarity, some forced vertex gadgets do not appear on the figure. Dotted or dashed lines are used in order for cleanness of the figure.

7.2.3 Soundness of the reduction

First, we define the vertex sets to be used in the following parts. For every $i \in [n], r \in \{1, 2, 3\}, h \in \{1, 2\}$, let

$$U_{i}^{h} = \bigcup_{j \in [m]} P(s_{i}^{j}, p_{i}^{h}),$$

$$H_{i,r} = \bigcup_{j \in [m]} P(s_{i}^{j}, a_{r}) \cup P(s_{i}^{j}, b_{r}) \cup P(s_{i}^{j}, c_{r}),$$

$$S_{i}^{h} = \bigcup_{r \in \{1, 2, 3\}} P(\pi_{i}^{h}, a_{r}) \cup P(\pi_{i}^{h}, c_{r}),$$

$$L_{i}^{h} = \bigcup_{j \in [m]} P(q_{i}^{h}, \operatorname{mid}(P^{3-h}(i, j, p_{i}^{h}))),$$

 $R_r = \bigcup_{i \in [n]} P(a_r, u_r^i) \cup P(a_r, v_r^i) \cup P(b_r, u_r^i) \cup P(b_r, v_r^i) \cup P(c_r, u_r^i) \cup P(c_r, v_r^i), \text{ and }$

$$\Pi^{h}(i,j,r) = P^{h}(i,j,a_{r}) \cup P^{h}(i,j,b_{r}) \cup P^{h}(i,j,c_{r}) \cup P^{h}(i,j,p_{i}^{3-h}).$$

For every $i \in [n]$, let

$$U_{i} = \bigcup_{h \in \{1,2\}} U_{i}^{h} \qquad H_{i} = \bigcup_{r \in \{1,2,3\}} H_{i,r} \qquad S_{i} = \bigcup_{h \in \{1,2\}} S_{i}^{h}$$
$$L_{i} = \bigcup_{h \in \{1,2\}} L_{i}^{h} \qquad \Pi_{i} = \bigcup_{j \in [m], r \in \{1,2,3\}, h \in \{1,2\}} \Pi^{h}(i,j,r).$$

Let \mathcal{F} be the union of all forced vertex gadgets, i.e. $\mathcal{F} = \bigcup_{i \in [n], j \in [m], r \in \{1,2,3\}, h \in \{1,2\}} (F(s_i^j, a_r) \cup F(s_i^j, c_r) \cup F(\pi_i^h, a_r) \cup F(\pi_i^h, c_r) \cup F^h(u_r^i, v_r^i) \cup F^h(i, j, a_r) \cup F^h(i, j, b_r) \cup F^h(i, j, c_r) \cup F^h(i, j, p_i^{3-h}) \cup F^{mid}(i, j, h) \cup F^{ecc}(i, j, h, r)).$

Next we introduce a lemma about forced set gadgets and this lemma is important for the correctness of the reduction.

Lemma 46. The following three statements are true for the instance (G', k).

- (a) The vertex s_i^j for $i \in [n], j \in [m]$ resolves both pairs $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$. Moreover, s_i^j does not resolve any vertex pair $\{p_{i'}^h, q_{i'}^h\}$ such that $i' \in [n], h \in \{1, 2\}$ and $i' \neq i$.
- (b) The vertices of any forced vertex gadget do not resolve any vertex pair of $\{\{p_i^h, q_i^h\} \mid i \in [n], h \in \{1, 2\}\}$.
- (c) Any vertex $v \in V(G') \setminus (X_i \cup \mathcal{F})$ resolves at most one vertex pair of $\{\{p_i^h, q_i^h\} \mid i \in [n], h \in \{1, 2\}\}$.

Proof. By the construction of G', $\operatorname{dist}(s_i^j, q_i^h) = |P(s_i^j, p_i^h)| + 2 = 20(n+1) + 2 > \operatorname{dist}(s_i^j, p_i^h)$ for $i \in [n], j \in [m]$ and $h \in \{1, 2\}$. Thus any vertex of X_i resolves both pairs $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$ for $i \in [n]$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from s_i^j to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $c_{r'}$ and $\pi_{i'}^{h'}$ with some integer $r' \in \{1, 2, 3\}$. Thus a vertex $s \in X_i$ resolves exactly two vertex pairs of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1, 2\}\}$.

First we claim that vertices of \mathcal{F} do not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1, 2\}$. For any vertex $v \in F^h(u_r^i, v_r^i)$ for $i \in [n], r \in \{1, 2, 3\}, h \in \{1, 2\}$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through a_r and $\pi_{i'}^{h'}$. Thus v does not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1, 2\}$. For any vertex $v \in F^{mid}(i, j, h) \cup F^{ecc}(i, j, h, r)$ for $i \in [n], j \in [m], h \in \{1, 2\}, r \in \{1, 2, 3\}$, we can see that $\operatorname{dist}(v, p_{i'}^{h'}) = \operatorname{dist}(v, q_{i'}^{h'})$ with i' = i.

There is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through π_i^h , a_r and $\pi_{i'}^{h'}$ with $i' \neq i$. Thus v does not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1, 2\}$. For any vertex $v \in \mathcal{F} \setminus \bigcup_{i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}} (\mathcal{F}^h(u_r^i, v_r^i) \cup \mathcal{F}^{ecc}(i, j, h, r) \cup \mathcal{F}^{mid}(i, j, h))$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$ with i' = i. There is a shortest path from v to $p_{i'}^{h'}$ or $q_i^{h'}$ or $q_{i'}^{h'}$ so and $\pi_{i'}^{h'}$ with $i' \neq i$. Thus v does not resolve any pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$. As a result, vertices of \mathcal{F} do not resolve any vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ for $i' \in [n], h' \in \{1, 2\}$.

Then we show that any vertex $v \in V(G') \setminus (X_i \cup \mathcal{F})$ resolves at most one pair of $\{p_i^1, q_i^1\}$ and $\{p_i^2, q_i^2\}$.

For a vertex $v \in U_i^h \setminus X_i$ for $i \in [n], h \in \{1, 2\}$, $\operatorname{dist}(v, p_i^h) = \operatorname{dist}(v, q_i^h) - 2 < \operatorname{dist}(v, q_i^h)$. $\operatorname{dist}(v, q_i^{3-h}) = \operatorname{dist}(v, N_{s_i^j}(s_i^j, p_i^h)) + |P^{3-h}(i, j, p_i^h)| + 1 = \operatorname{dist}(v, p_i^{3-h})$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus $v \in U_i^h \setminus X_i$ for $i \in [n], h \in \{1, 2\}$ resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1, 2\}\}$.

Let $P(\operatorname{mid}(P^{3-h}(i,j,p_i^h)), N_{s_i^j}(s_i^j,p_i^h))$ be the subpath of $P^{3-h}(i,j,p_i^h)$ from $\operatorname{mid}(P^{3-h}(i,j,p_i^h))$ to $N_{s_i^j}(s_i^j,p_i^h)$. Let $\Lambda_i^h = (\bigcup_{j\in[m]} P(\operatorname{mid}(P^{3-h}(i,j,p_i^h)), N_{s_i^j}(s_i^j,p_i^h))) \setminus {\operatorname{mid}(P^{3-h}(i,j,p_i^h))} \mid j \in [m]$ }. For a vertex $v \in \Lambda_i^h$ for $i \in [n], h \in \{1,2\}$, $\operatorname{dist}(v,p_i^h) = \operatorname{dist}(v,q_i^h) - 2 < \operatorname{dist}(v,q_i^h)$. $\operatorname{dist}(v,q_i^{3-h}) = \operatorname{dist}(v,\pi_i^{3-h}) + 1 = \operatorname{dist}(v,p_i^{3-h})$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus $v \in \Lambda_i^h$ for $i \in [n], h \in \{1,2\}$ resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1,2\}\}$.

resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\}: i \in [n], h \in \{1, 2\}\}$. For a vertex $v \in L_i^h \setminus \{\operatorname{mid}(P^h(i, j, p_i^{3-h})) \mid j \in [m]\}$ for $i \in [n], h \in \{1, 2\}$, $\operatorname{dist}(v, q_i^h) = \operatorname{dist}(v, p_i^h) - 2 < \operatorname{dist}(v, p_i^h)$. There is a shortest path from v to p_i^{3-h} or q_i^{3-h} going through π_i^{3-h} . For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to p_i^{3-h} or q_i^{3-h} going through π_i^{3-h} . For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus $v \in L_i^h \setminus \{\operatorname{mid}(P^h(i, j, p_i^{3-h})) \mid j \in [m]\}$ for $i \in [n], h \in \{1, 2\}$ resolves exactly one vertex pair of $\{\{p_i^h, q_i^h\}: i \in [n], h \in \{1, 2\}\}$.

For a vertex $v \in \Pi_i \cup S_i \cup H_i \setminus (X_i \cup \Lambda_i^1 \cup \Lambda_i^2)$ for $i \in [n]$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$ with $i = i', h' \in \{1, 2\}$. For a vertex pair $\{p_{i'}^{h'}, q_{i'}^{h'}\}$ such that $i' \neq i$, there is a shortest path from v to $p_{i'}^{h'}$ or $q_{i'}^{h'}$ going through $\pi_{i'}^{h'}$. Thus v does not resolve any vertex pair of $\{\{p_i^h, q_i^h\} : i \in [n], h \in \{1, 2\}\}$.

For a vertex $v \in R_r$ for $r \in \{1, 2, 3\}$, there is a shortest path from v to p_i^h or q_i^h for $i \in [n], h \in \{1, 2\}$ going through a_r and π_i^h . Thus v does not resolve any vertex pair of $\{\{p_i^h, q_i^h\}: i \in [n], h \in \{1, 2\}\}$. This completes the proof for the lemma.

Lemma 47. The forced vertices do not resolve any vertex pair $\{u_r^i, v_r^i\} \in \mathcal{P}$ for $r \in \{1, 2, 3\}$ and $i \in [n]$.

Proof. We fix arbitrary integers $i \in [n], j \in [m], r \in \{1, 2, 3\}, h \in \{1, 2\}$. For the forced vertex $f^h(i, j, a_r), \operatorname{dist}(f^h(i, j, a_r), u_{r'}^{i'}) = 2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = 2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, v_{r'}^{i'})| = \operatorname{dist}(f^h(i, j, a_r), v_{r'}^{i'}) \text{ for } i' \in [n], r' \in \{1, 2, 3\}$. Thus $f^h(i, j, a_r)$ does not resolve any vertex pair of \mathcal{P} . Similarly, the forced vertices $f^h(i, j, b_r), f^h(i, j, c_r)$ and $f^h(i, j, p_i^{3-h})$ do not resolve any vertex pair of \mathcal{P} . For the forced vertex $f^{mid}(i, j, h), \operatorname{dist}(f^{mid}(i, j, h), u_{r'}^{i'}) = \operatorname{dist}(f^{mid}(i, j, h), v_{r'}^{i'}) = |P^h(i, j, p_i^{3-h})|/2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})|$. Thus $f^{mid}(i, j, h, r), u_{r'}^{i'}) = \operatorname{dist}(f^{ecc}(i, j, h, r), v_{r'}^{i'}) = 10(n+1) + 1 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})|$. Thus $f^{ecc}(i, j, h, r)$ does not resolve any vertex pair of \mathcal{P} .

We fix arbitrary integers $i \in [n], j \in [m], r \in \{1, 2, 3\}$. For the forced vertex $f(s_i^j, a_r)$, $\operatorname{dist}(f(s_i^j, a_r), u_r^{i'}) = 2 + |P(a_r, u_r^{i'})| = 2 + |P(a_r, v_r^{i'})| = \operatorname{dist}(f(s_i^j, a_r), v_r^{i'})$ for $i' \in [n]$. For the forced vertex $f(s_i^j, c_r)$, $\operatorname{dist}(f(s_i^j, c_r), u_r^{i'}) = 2 + |P(c_r, u_r^{i'})| = 2 + |P(c_r, v_r^{i'})| = \operatorname{dist}(f(s_i^j, c_r), v_r^{i'})$ for $i' \in [n]$. Thus $f(s_i^j, a_r)$ and $f(s_i^j, c_r)$ do not resolve any vertex pair of \mathcal{P}_r . Similarly, $f(\pi_i^h, a_r)$ and $f(\pi_i^h, c_r)$ for $i \in [n], h \in \{1, 2\}, r \in \{1, 2, 3\}$ do not resolve any vertex pair of \mathcal{P}_r . For vertex pairs of $\mathcal{P}_{r'}$ with $r' \in \{1, 2, 3\}$ and $r' \neq r$, $\operatorname{dist}(f(s_i^j, a_r), u_{r'}^{i'}) = 2 + |P(a_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, v_r^{i'})| = \operatorname{dist}(f(s_i^j, a_r), v_{r'}^{i'})$ for $i' \in [n]$. $\operatorname{dist}(f(s_i^j, c_r), u_{r'}^{i'}) = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^1)| + |P(a_{r'}, \pi_i^1)| + |P(a_{r'}, u_r^{i'})| = 2 + |P(c_r, \pi_i^{i'})| = 2$

We fix arbitrary integers $i \in [n], r \in \{1, 2, 3\}$. For the forced vertex $f^1(u_r^i, v_r^i)$ or $f^2(u_r^i, v_r^i)$, obviously it does not resolve the vertex pair $\{u_r^i, v_r^i\}$. For a vertex pair $\{u_r^i, v_r^i'\}$ with $i' \in [n]$ and $i' \neq i$, $\operatorname{dist}(f^1(u_r^i, v_r^i), u_r^{i'}) = 2 + |P(a_r, u_r^i)| - 1 + |P(a_r, u_r^i)| = 2 + |P(a_r, u_r^i)| - 1 + |P(a_r, v_r^i)| = 2 + |P(a_r, u_r^i)| - 1 + |P(a_r, v_r^i)| = 4 \operatorname{dist}(f^1(u_r^i, v_r^i), v_r^{i'})$. For a vertex pair $\{u_{r'}^{i'}, v_{r'}^{i'}\}$ with $i' \in [n]$ and $r' \in \{1, 2, 3\}$ and $r' \neq r$, $\operatorname{dist}(f^1(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 1 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = 4 \operatorname{dist}(f^1(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 1 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = 4 \operatorname{dist}(f^1(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(a_r, u_r^i)| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(a_r, u_r^i)| - 2 + |P(a_r, u_r^i)| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, u_{r'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, u_{r'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r'}^{i'}) = 2 + |P(a_r, u_r^i)| - 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(\pi_i^1, u_{r'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r''}) = 2 + |P(u_r, u_r^i)| - 2 + |P(\pi_i^1, u_{r'})| = 4 \operatorname{dist}(f^2(u_r^i, v_r^i), u_{r''}) = 2 + |P(\pi_i^1, v_r^$

Lemma 48. If G' has a resolving set of size at most 34nm + 19n, then $(G, n, \chi, \mathcal{P})$ is a yesinstance.

Proof. Suppose that S is a resolving set for G' of size at most 34nm + 19n. Let $\hat{S} = S \cap X$. We claim that \hat{S} is solution for $(G, n, \chi, \mathcal{P})$. Note that for the false twins $\{u, u'\}$ of a forced vertex gadget, no vertex resolves the vertex pair $\{u, u'\}$ except u (or u'). It follows that S contains 34nm + 18n forced vertices since there are 34nm + 18n forced vertex gadgets in G'. Since X has no intersection with the vertex set of all forced vertex gadgets, $|\hat{S}| \leq n$. By Lemma 46, we get that $|\hat{S} \cap X_i| = 1$ for each $i \in [n]$. Thus $|\hat{S}| = n$. By Lemma 47 and the assumption that S is a resolving set for G', \hat{S} resolves every pair $\{u_r^i, v_r^i\}$ in G' for $r \in \{1, 2, 3\}$ and $i \in [n]$. We can check that the distance between s_i^j and $u_r^{i'}$ in G' (and the distance between s_i^j and $v_r^{i'}$ in G') for $i \in [n], j \in [m], i' \in [n], r \in \{1, 2, 3\}$ is the same as that in G. Thus \hat{S} is a solution for $(G, n, \chi, \mathcal{P})$.

7.2.4 Treewidth bound of the graph

Since the completeness proof takes a large amount of space, before proceeding to that, we first show that G' is of constant treewidth. In fact, we will prove a slightly stronger statement that G' is of constant pathwidth by giving a search strategy with a constant number of searchers.

Lemma 49. The pathwidth of G' is at most 24.

Proof. We give a search strategy with 25 searchers. First, we put 9 searchers on $\bigcup_{r \in \{1,2,3\}} \{a_r, b_r, c_r\}$. The 9 searchers remain there until the end of the whole search process. The search process consists of two phases. We search the "left" part of G' in the first phase and the "right" part of G' in the second phase.

The first phase of the search process consists of n rounds. At the beginning of the *i*-th round $(i \in [n])$, we put 6 searchers on $\bigcup_{h \in \{1,2\}} \{p_i^h, q_i^h, \pi_i^h\}$. Here when we say that we clean a path, there are already two searchers guarding at the endpoints (or the neighbor of the endpoints) of this path and we use 3 extra searchers x, y, z such that x, y move alternately from one end of the path to the other end to clean the edges of the path. When a searcher, say x arrives at the connecting point of a forced vertex gadget, we put y, z on the false twins of this forced vertex gadget to clean the edges of the after removing y, z, put y ahead of x

to continue the alternating process if x does not reach the endpoint of this path. Then for each $j \in [m]$, we

- put 5 vertices on $N_{G'}(s_i^j)$.
- put 2 vertices on $\operatorname{mid}(P^{h}(i, j, p_{i}^{3-h}))$ for $h \in \{1, 2\}$.
- use 3 extra searchers to clean the paths $P(s_i^j, p_i^h)$ for $h \in \{1, 2\}$, the paths $P(s_i^j, a_r)$, $P(s_i^j, b_r), P(s_i^j, c_r)$ for $r \in \{1, 2, 3\}$, the paths $P^h(i, j, a_r), P^h(i, j, b_r), P^h(i, j, c_r), P^h(i, j, p_i^{3-h})$ for $h \in \{1, 2\}, r \in \{1, 2, 3\}$ successively (including all forced vertex gadgets attached to the vertices on these paths).
- remove the above 10 searchers that are still on the graph.

At the end of the *i*-th round, we remove the 6 searchers on $\bigcup_{h \in \{1,2\}} \{p_i^h, q_i^h, \pi_i^h\}$.

The second phase of the search process consists of 3 rounds. During the r-th round $(r \in \{1, 2, 3\})$, we operate as follows. For each $i \in [n]$, we

- put 4 searchers on u_r^i , v_r^i and the connecting point of $F^h(u_r^i, v_r^i)$ for $h \in \{1, 2\}$.
- use 2 extra searchers to clean the paths $P(a_r, u_r^i)$, $P(b_r, u_r^i)$, $P(c_r, u_r^i)$, $P(a_r, v_r^i)$, $P(b_r, v_r^i)$ and $P(c_r, v_r^i)$ (including the forced vertex gadgets $F^h(u_r^i, v_r^i)$ for $h \in \{1, 2\}$ and the edges between $F^h(u_r^i, v_r^i)$ and the paths listed above).
- remove the above 6 searchers that are still on the graph.

This completes the description of the the search strategy.

As a result, the node search number of G' is at most 25. It follows that the pathwidth of G' is bounded by 24.

7.2.5 Completeness of the reduction

For every forced vertex gadget of G', we choose a vertex from the false twins arbitrarily as a forced vertex and let the set of all chosen forced vertices be F. In this section, we show that if $(G, n, \chi, \mathcal{P})$ has a solution S, then $S' = S \cup F$ is a resolving set of size at most 34nm + 19n for G'. Formally, we will prove the following lemma.

Lemma 50. If $(G, n, \chi, \mathcal{P})$ is a yes-instance, then G' has a resolving set of size at most 34nm + 19n.

To prove the above lemma, we need to show that every pair of distinct vertices of G' is resolved by some vertex of S'. First of all, We have the following claim.

Claim 5. Every vertex pair $\{u_r^{i'}, v_r^{i'}\}$ in G' for $r \in \{1, 2, 3\}, i' \in [n]$ is resolved by S'.

Proof. Since $(G, n, \chi, \mathcal{P})$ is a yes-instance, $\operatorname{dist}_G(s_i^j, u_r^{i'}) = \operatorname{dist}_{G'}(s_i^j, u_r^{i'})$, $\operatorname{dist}_G(s_i^j, v_r^{i'}) = \operatorname{dist}_{G'}(s_i^j, v_r^{i'})$ for $i, i' \in [n], j \in [m], r \in \{1, 2, 3\}$, every vertex pair $\{u_r^{i'}, v_r^{i'}\}$ in G' for $r \in \{1, 2, 3\}$ and $i' \in [n]$ is resolved by some vertex of $S \subset S'$.

Suppose that $V(G') = V_1 \cup V_2 \cup ... \cup V_t$. Our general method is to show that for each $i \in [t]$, every internal vertex pair of V_i is resolved by S' and every vertex pair of $V_{i'} \times V_i$ for each i' < i is resolved by S'. Note that when we mention the vertex pairs of $V_{i'} \times V_i$, we ignore the vertex pairs with two identical vertices by default as it's meaningless in our problem. In the following parts, we give a series of lemmas to show that Lemma 50 is true. Let $U = \bigcup_{i \in [n]} U_i, \Pi = \bigcup_{i \in [n]} \Pi_i, H = \bigcup_{i \in [n]} H_i, S = \bigcup_{i \in [n]} S_i, L = \bigcup_{i \in [n]} L_i$ and $R = \bigcup_{r \in \{1,2,3\}} R_r$. Table 7.1 shows the indexes of the corresponding lemmas.

Lemma 51. Every pair of distinct vertices $x, y \in \bigcup_{i \in [n], h \in \{1,2\}} U_i^h$ is resolved by S'.

	U	Π	S	L	H	R
U	51	57	59	58	60	61
П		54	64	63	62	65
S			53	67	69	70
L				55	66	68
Η					52	71
R						56

Table 7.1: Indexes of the lemmas for the completeness of the reduction.

Proof. First, we show that every pair of distinct vertices $x, y \in \bigcup_{h \in \{1,2\}} U_i^h$ for $i \in [n]$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m]$ and $h \in \{1,2\}$ such that $j' \neq j$.

Suppose that $x, x' \in P(s_i^j, p_i^h)$. For a vertex $x \in P(s_i^j, p_i^h)$, let $P(s_i^j, x)$ be the subpath of $P(s_i^j, p_i^h)$ from s_i^j to x and $|P(s_i^j, x)| = \ell_x$. Since $\operatorname{dist}(f^h(i, j', a_1), x) = 3 + 20(n+1) - \ell_x$, $f^h(i, j', a_1)$ resolves every pair $\{x, x'\}$ such that $\ell_x \neq \ell_{x'}$.

Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(s_i^{j'}, p_i^h)$ such that $j' \in [m]$ and $j' \neq j$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the second paragraph. dist $(f^{mid}(i, j, 3-h), x) = 10(n+1) + \ell_x$ if $\ell_x \geq 1$ and dist $(f^{mid}(i, j, 3-h), x) = 10(n+1) + 2$ if $\ell_x = 1$. dist $(f^{mid}(i, j, 3-h), y) = \min(2+|P^{3-h}(i, j, p_i^h)|/2+|P(s_i^{j'}, p_i^{3-h})| + \ell_y, |P^{3-h}(i, j, p_i^h)|/2+|P(s_i^{j'}, p_i^h)| + |P(s_i^{j'}, p_i^h)| - \ell_y) = \min(2+30(n+1)+\ell_y, 50(n+1)-\ell_y) \geq 30(n+1) \geq \text{dist}(f^{mid}(i, j, 3-h), x)$ and the equalities hold if and only if $x = y = p_i^h$. Thus every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, 3-h)$.

Suppose that $x \in P(s_i^j, p_i^h) \setminus \{s_i^j\}$ and $y \in P(s_i^j, p_i^{3-h}) \setminus \{s_i^j\}$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the second paragraph. Then $\operatorname{dist}(f^h(i, j', a_1), x) = 3 + 20(n+1) - \ell_x \leq 2+20(n+1)$ and $\operatorname{dist}(f^h(i, j', a_1), y) = \min(1+|P^h(i, j, p_i^{3-h})|+\ell_y, 3+|P(\pi_i^h, c_1)|+|P(\pi_i^{3-h}, c_1)|+|P(s_i^j, p_i^{3-h})|-\ell_y) = \min(1+20(n+1)+\ell_y, 3+40(n+1)-\ell_y) \geq 2+20(n+1)$. We see from the two equations that $\operatorname{dist}(f^h(i, j, a_1), x) \neq \operatorname{dist}(f^h(i, j, a_1), y)$ unless $\ell_x = \ell_y = 1$. We can check that $f^h(i, j, p_i^{3-h})$ resolves the pair $\{x, y\}$ with $\ell_x = \ell_y = 1$. Thus every pair $\{x, y\}$ is resolved by $f^h(i, j', a_1)$ or $f^h(i, j, p_i^{3-h})$.

Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(s_i^{j'}, p_i^{3-h})$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the second paragraph. Then $\operatorname{dist}(f^{mid}(i, j, h), x) = \min(2 + |P^h(i, j, p_i^{3-h})|/2 + \ell_x, 2 + |P^h(i, j, p_i^{3-h})|/2 + |P(s_i^j, p_i^h)| - \ell_x) = \min(2 + 10(n+1) + \ell_x, 2 + 30(n+1) - \ell_x) \leq 2 + 20(n+1)$. $\operatorname{dist}(f^{mid}(i, j, h), y) = \min(|P^h(i, j, p_i^{3-h})|/2 + |P(s_i^{j'}, p_i^{3-h})| + |P(s_i^{j'}, p_i^{3-h})| - \ell_y, 2 + |P^h(i, j, p_i^{3-h})|/2 + |P(s_i^{j'}, p_i^{h})| + \ell_y) = \min(50(n+1) - \ell_y, 2 + 30(n+1) + \ell_y) \geq 2 + 30(n+1)$. Thus every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, h)$.

Finally we show that every pair of distinct vertices $\{x, y\} \in \bigcup_{h \in \{1,2\}} U_i^h \times \bigcup_{h' \in \{1,2\}} U_{i'}^h$ with $i, i' \in [n]$ and $i \neq i'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m]$ and $h, h' \in \{1,2\}$ such that $i \neq i'$. Let $x \in P(s_i^j, p_i^h)$ and $y \in P(s_{i'}^{j'}, p_{i'}^{h'})$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the second paragraph. Then as we show in last paragraph, dist $(f^{mid}(i, j, h), x) = \min(2+10(n+1)+\ell_x, 2+30(n+1)-\ell_x) \leq 2+20(n+1)$. dist $(f^{mid}(i, j, h), s_{i'}^{j'}) = \min_{r \in \{1,2,3\}}(1+|P^h(i, j, p_i^{3-h})|/2+|P(\pi_i^h, c_r)|+|P(c_r, s_{i'}^{j'})|)$. dist $(f^{mid}(i, j, h), y) = \min(\operatorname{dist}(f^{mid}(i, j, h), s_{i'}^{j'}) + \ell_y, 2+|P^h(i, j, p_i^{3-h})|/2+|P(\pi_i^h, c_1)|+|P(\pi_{i'}^{h'}, c_1)|+|P(s_{i'}^{j'}, p_{i'}^{h'})|-\ell_y) > 1+30(n+1) > \operatorname{dist}(f^{mid}(i, j, h), x)$. Thus every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, h)$. This completes the proof for the lemma.

Lemma 52. Every pair of distinct vertices $x, y \in \bigcup_{i \in [n], r \in \{1,2,3\}} H_{i,r}$ is resolved by S'.

Proof. First we show that every vertex pair of $H_{i,r} \times H_{i',r}$ such that $i', i \in [n]$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j \in [m]$ and $r \in \{1, 2, 3\}$.

Given two distinct vertices $x_1, x_2 \in P(s_i^j, a_r)$, we can verify that $f(\pi_i^1, a_r)$ resolves the pair $\{x_1, x_2\}$. Similarly, two distinct vertices $x'_1, x'_2 \in P(s^j_i, c_r)$ are distinguished by $f(\pi^1_i, c_r)$. Suppose that $x \in P(s_i^j, a_r) \setminus \{s_i^j\}, y \in P(s_i^j, b_r) \setminus \{s_i^j\}$ and $z \in P(s_i^j, c_r) \setminus \{s_i^j\}$. Let $P(s_i^j, x)$, $P(s_i^j, y)$ and $P(s_i^j, z)$ be the subpath of $P(s_i^j, a_r)$, $P(s_i^j, b_r)$ and $P(s_i^j, c_r)$ respectively. Let $|P(s_i^j, x)| = \ell_x, |P(s_i^j, y)| = \ell_y$ and $|P(s_i^j, z)| = \ell_z$. Similarly, for a vertex $y' \in P(s_i^j, b_r)$, we define $|P(s_i^j, y')| = \ell_{y'}$. Since dist $(f^{mid}(i, j, 1), y) = 2 + |P^1(i, j, p_i^2)|/2 + \ell_y$ and dist $(f^{mid}(i, j, 1), y') = \ell_{y'}$. $2+|P^1(i,j,p_i^2)|/2+\ell_{y'}$, two distinct vertices y, y' are distinguished by $f^{mid}(i,j,1)$. dist $(f(\pi_i^1,a_r),x)$ $=2+|P(s_i^j,a_r)|-\ell_x. \operatorname{dist}(f(\pi_i^1,a_r),b_r)=\min_{j'\in[m]}(2+|P(s_i^{j'},a_r)|+|P(s_i^{j'},b_r)|)=2+20(n+1)+10(n+1)$ $10+20(n+1)+5. \operatorname{dist}(f(\pi_i^1, a_r), y) = \min (2+|P(s_i^j, a_r)| + \ell_y, \operatorname{dist}(f(\pi_i^1, a_r), b_r) + |P(s_i^j, b_r)| - \ell_y).$ For a vertex pair $\{x, y\}$ such that $dist(f(\pi_i^1, a_r), y) = 2 + |P(s_i^j, a_r)| + \ell_y$, obviously the pair is resolved by $f(\pi_i^1, a_r)$. For a vertex pair $\{x, y\}$ such that $\operatorname{dist}(f(\pi_i^1, a_r), y) = \operatorname{dist}(f(\pi_i^1, a_r), b_r) +$ $|P(s_i^j, b_r)| - \ell_y$, dist $(f(\pi_i^1, a_r), y) \ge 40(n+1) + 17 > \text{dist}(f(\pi_i^1, a_r), x)$. Thus every pair $\{x, y\}$ is resolved by $f(\pi_i^1, a_r)$. Similarly, every pair $\{y, z\}$ is resolved by $f(\pi_i^1, c_r)$. For a vertex pair $\{x, z\}, \operatorname{dist}(f(\pi_i^1, c_r), z) = 2 + |P(s_i^j, c_r)| - \ell_z < 20(n+1) \text{ if } z \neq c_r \text{ and } \operatorname{dist}(f(\pi_i^1, c_r), c_r) = 2.$ $\operatorname{dist}(f(\pi_i^1, c_r), x) = \min (2 + |P(s_i^j, c_r)| + \ell_x, |P(\pi_i^1, c_r)| + |P(\pi_i^1, a_r)| + |P(s_i^j, a_r)| - \ell_x) \ge 2 + \ell_x$ $|P(s_i^j, c_r)| > \operatorname{dist}(f(\pi_i^1, c_r), z)$. Thus every pair $\{x, z\}$ is resolved by $f(\pi_i^1, c_r)$.

Let $i' \in [n], j' \in [m]$ be integers such that $i' \neq i$ or $j' \neq j$. Suppose that $x \in P(s_i^j, a_r) \setminus \{a_r\}$, $y \in P(s_i^j, b_r) \setminus \{b_r\}$ and $z \in P(s_i^j, c_r) \setminus \{c_r\}$. Suppose that $x' \in P(s_{i'}^{j'}, a_r) \setminus \{a_r\}, y' \in P(s_{i'}^{j'}, a_r) \setminus \{a_r\}$ $P(s_{i'}^{j'}, b_r) \setminus \{b_r\}$ and $z' \in P(s_{i'}^{j'}, c_r) \setminus \{c_r\}$. We define $\ell_x, \ell_y, \ell_z, \ell_{x'}, \ell_{y'}$ and $\ell_{z'}$ in a similar way to that of ℓ_x in the second paragraph. For a pair $\{x, x'\}$, since $dist(f(\pi_i^1, a_r), x) =$ $2 + |P(s_i^j, a_r)| - \ell_x$ and dist $(f(\pi_i^1, a_r), x') = 2 + |P(s_{i'}^{j'}, a_r)| - \ell_{x'}, f(\pi_i^1, a_r)$ resolves every pair $\{x, x'\}$ such that $|P(s_i^j, a_r)| - \ell_x \neq |P(s_{i'}^{j'}, a_r)| - \ell_{x'}$. Since dist $(f(s_i^j, a_r), x) = |P(s_i^j, a_r)| - \ell_x$ and dist $(f(s_i^j, a_r), x') = 2 + |P(s_{i'}^{j'}, a_r)| - \ell_{x'}, f(s_i^j, a_r)$ resolves every pair $\{x, x'\}$ such that $|P(s_i^j, a_r)| - \ell_x = |P(s_{i'}^{j'}, a_r)| - \ell_{x'}$. As a result, every pair $\{x, x'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. Similarly, every pair $\{z, z'\}$ is resolved by $f(\pi_i^1, c_r)$ or $f(s_i^j, c_r)$. For a pair $\{y, y'\}$, there are two cases. Case 1: i = i' and $j \neq j'$. $dist(f^{mid}(i, j, 1), y) = 2 + |P^1(i, j, p_i^2)|/2 + \ell_y$. $\operatorname{dist}(f^{mid}(i,j,1),y') = \min \ (2 + |P^1(i,j,p_i^2)|/2 + |P(s_i^j,b_r)| + |P(s_{i'}^{j'},b_r)| - \ell_{y'}, 1 + |P^1(i,j,p_i^2)|/2 + |$ $|P^1(i', j', b_r)| + \ell_{y'} - 1$ if $y' \neq s_{i'}^{j'}$ and $\operatorname{dist}(f^{mid}(i, j, 1), s_{i'}^{j'}) = 2 + 30(n+1)$. If a pair $\{y, y'\}$ satisfies that $\operatorname{dist}(f^{mid}(i, j, 1), y') = 2 + |P^1(i, j, p_i^2)|/2 + |P(s_i^j, b_r)| + |P(s_{i'}^{j'}, b_r)| - \ell_{y'}$, obviously $f^{mid}(i, j, 1)$ resolves this pair. Thus if a pair $\{y, y'\}$ is not resolved by $f^{mid}(i, j, 1)$, then we have $dist(f^{mid}(i, j, 1), y) = 2 + 10(n+1) + \ell_y = dist(f^{mid}(i, j, 1), y') = 30(n+1) + \ell_{y'}, \text{ i.e. } \ell_y - \ell_{y'} = 20(n+1) - 2.$ For this pair, $dist(f^{mid}(i', j', 1), y') = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i', j', 1), y) = 2 + 10(n$ min $(2 + 10(n+1) + |P(s_{i'}^{j'}, b_r)| + |P(s_i^{j}, b_r)| - \ell_y, 30(n+1) + \ell_y)$. Thus in this case, every pair $\{y, y'\}$ is resolved by $f^{mid}(i, j, 1)$ or $f^{mid}(i', j', 1)$. Case 2: $i \neq i'$. dist $(f^{mid}(i, j, 1), y) =$ $2 + |P^{1}(i, j, p_{i}^{2})|/2 + \ell_{y}. \operatorname{dist}(f^{mid}(i, j, 1), s_{i'}^{j'}) = 1 + |P^{1}(i, j, p_{i}^{2})|/2 + \min_{r' \in \{1, 2, 3\}}(|P(\pi_{i}^{1}, c_{r'})| + \ell_{y})|/2 + \ell_{y}.$ $|P(s_{i'}^{j'}, c_{r'})|)$. dist $(f^{mid}(i, j, 1), y') = \min (2 + |P^1(i, j, p_i^2)|/2 + |P(s_i^j, b_r)| + |P(s_{i'}^{j'}, b_r)| - \ell_{y'},$ $dist(f^{mid}(i, j, 1), s_{i'}^{j'}) + \ell_{y'})$. If a pair $\{y, y'\}$ satisfies that $dist(f^{mid}(i, j, 1), y') = 2 + |P^1(i, j, p_i^2)|/2 + |P^1(i, j, p_i^2)|$ $|P(s_i^j, b_r)| + |P(s_{i'}^{j'}, b_r)| - \ell_{y'}$, obviously $f^{mid}(i, j, 1)$ resolves this pair. Thus if a pair $\{y, y'\}$ is not resolved by $f^{mid}(i, j, 1)$, then we have $\operatorname{dist}(f^{mid}(i, j, 1), y) = 2 + 10(n+1) + \ell_y = 2$ $dist(f^{mid}(i,j,1),y') = dist(f^{mid}(i,j,1),s_{i'}^{j'}) + \ell_{y'}, \text{ i.e. } \ell_y - \ell_{y'} = dist(\pi_i^1, s_{i'}^{j'}) - 1. \text{ For this pair, } dist(f^{mid}(i',j',1),y') = 2 + 10(n+1) + \ell_{y'} < dist(f^{mid}(i',j',1),y) = \min(2 + 10(n+1)) + \ell_{y'} < dist(f^{mid}(i',j',1),y) = \min(2 + 10(n+1)) + \ell_{y'} < dist(f^{mid}(i',j',1),y) = 0$ 1) + $|P(s_{i'}^{j'}, b_r)| + |P(s_i^j, b_r)| - \ell_y$, dist $(f^{mid}(i', j', 1), s_i^j) + \ell_y)$. Thus in this case, every pair $\{y, y'\}$ is resolved by $f^{mid}(i, j, 1)$ or $f^{mid}(i', j', 1)$. It follows that every pair $\{y, y'\}$ is resolved by $f^{mid}(i, j, 1)$ or $f^{mid}(i', j', 1)$. For a pair $\{x, y'\}$, there are two cases. Case 1: $|P(s_i^j, a_r)| > 1$ $20(n+1) + 10 \cdot 1 = \min_{\alpha \in [m]} |P(s_i^{\alpha}, a_r)|$. Then $dist(f(\pi_i^1, a_r), y') = \min (2 + |P(s_{i'}^{j'}, a_r)| + 10 \cdot 1)$

$$\begin{split} \ell_{y'}, 2 + 20(n+1) + 10 \cdot 1 + 20(n+1) + 5 \cdot 1 + 1 + |P(s_{i'}^{j'}, b_r)| - \ell_{y'} &= \operatorname{dist}(f(s_i^j, a_r), y') &= \operatorname{dist}(f(s_i^j, a_r), x) = 2 + |P(s_i^j, a_r)| - \ell_x &= \operatorname{dist}(f(s_i^j, a_r), x) + 2. \text{ In this case, } \{x, y'\} \text{ is resolved by } f(\pi_i^1, a_r) &= 0 + |P(s_i^j, a_r)| = 20(n+1) + 10 \cdot 1 = \min_{p \in [m]} |P(s_i^p, a_r)|. \text{ Then } \\ \operatorname{dist}(f(s_i^j, a_r), x) &\leq 20(n+1) + 10 < \operatorname{dist}(f(s_i^j, a_r), y'). \text{ Thus in this case, } \{x, y'\} \text{ is resolved } \\ \operatorname{by} f(s_i^j, a_r). & \text{It follows that every pair } \{x, y'\} \text{ is resolved by } f(\pi_i^1, a_r) &= f(s_i^j, a_r). \text{ For a pair } \\ \{x, z'\}, & \operatorname{dist}(f(\pi_i^1, a_r), x) = 2 + |P(s_i^j, a_r)| - \ell_x = \operatorname{dist}(f(s_i^j, a_r), x) + 2, \text{ and } \operatorname{dist}(f(\pi_i^1, a_r), z') = \\ \min(2 + |P(s_{i'}^{j'}, a_r)| + \ell_{z'}, |P(\pi_i^1, a_r)| + |P(\pi_i^1, c_r)| + |P(s_{i'}^{j'}, c_r)| - \ell_{z'}) \leq \operatorname{dist}(f(s_i^j, a_r), z') = \\ \min(2 + |P(s_{i'}^{j'}, a_r)| + \ell_{z'}, 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, c_r)| + |P(s_{i'}^{j'}, c_r)| - \ell_{z'}). \text{ It follows that every } \\ \operatorname{pair} \{x, z'\} \text{ is resolved by } f(\pi_i^1, a_r) \text{ or } f(s_i^j, a_r). \text{ For a pair } \{y, z'\}, \text{ there are two cases. Case 1: } \\ |P(s_{i'}^{j'}, c_r)| + |P(s_{i'}^{j'}, b_r)| > 20(n+1) - 10n + 20(n+1) + 5n+1 = \min_{\alpha \in [m]} (|P(s_i^\alpha, c_r)| + |P(s_{i'}^\alpha, b_r)|). \\ \text{Then } \operatorname{dist}(f(\pi_i^1, c_r), y) = \min(2 + |P(s_i^j, c_r)| + \ell_y, 3 + 40(n+1) - 5n + |P(s_i^j, b_r)| - \ell_y) = \\ \operatorname{dist}(f(s_{i'}^{j'}, c_r), y) \text{ and } \operatorname{dist}(f(\pi_i^1, c_r), z') = 2 + |P(s_{i'}^{j'}, c_r)| - \ell_{z'} = \operatorname{dist}(f(s_{i'}^{j'}, c_r), z') + 2. \text{ In this } \\ \operatorname{case, } \{y, z'\} \text{ is resolved by } f(\pi_i^1, c_r) \text{ or } f(s_{i'}^{j'}, c_r). \text{ Case 2: } |P(s_{i'}^{j'}, c_r)| + |P(s_{i'}^{j'}, b_r)| = 20(n+1) - 10n + 20(n+1) + 5n + 1. \text{ Then } \operatorname{dist}(f(s_{i'}^{j'}, c_r), y) \geq 2 + 20(n+1) - 10n > \operatorname{dist}(f(s_{i'}^{j'}, c_r), z'). \text{ It follows that every pair } \{y, z'\} \text{ is resolved by } f(\pi_i^1, c_r) \text{ or } f(s_{i'}^{j'}, c_r) \text{ or } f(s_{i'}^{j'}, c_r). \text{ This completes the proof to show that every vertex pair of H_{i,r} \times H$$

Next we show that every vertex pair of $H_{i,r} \times H_{i',r'}$ such that $i, i' \in [n], r, r' \in \{1, 2, 3\}$ and $r \neq r'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m]$ and $r, r' \in \{1, 2, 3\}$ such that $r \neq r'$. Suppose that $x \in P(s_i^j, a_r), y \in P(s_i^j, b_r), z \in P(s_i^j, c_r), x' \in P(s_{i'}^{j'}, a_{r'}),$ $y' \in P(s_{i'}^{j'}, b_{r'})$ and $z' \in P(s_{i'}^{j'}, c_{r'})$. We define $\ell_x, \ell_y, \ell_z, \ell_{x'}, \ell_{y'}$ and $\ell_{z'}$ in a similar way to that of ℓ_x in the second paragraph. For a vertex pair $\{x, x'\}$, there are two cases. Case 1: i = i' and j = j'. Then dist $(f(\pi_i^1, a_r), x) = 2 + |P(s_i^j, a_r)| - \ell_x$, dist $(f(\pi_i^1, a_r), x') = \min(2 + |P(s_{i'}^{j'}, a_r)| + \ell_x$ $\ell_{x'}, |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(s_{i'}^{j'}, a_r)| - \ell_{x'}).$ dist $(f(s_i^j, a_r), x) = |P(s_i^j, a_r)| - \ell_x$ when $x \neq a_r$ and dist $(f(s_i^j, a_r), a_r) = 1$. dist $(f(s_i^j, a_r), x') = \min(|P(s_i^j, a_r)| + \ell_{x'}, 2 + |P(\pi_i^1, a_r)| + \ell_{x'}, 2 + \ell_$ $|P(\pi_i^1, a_{r'})| + |P(s_i^j, a_{r'})| - \ell_{x'})$. Thus for the vertex pair $\{x, x'\}$ which is not resolved by $f(s_i^j, a_r)$, i.e., dist $(f(s_i^j, a_r), x) = dist(f(s_i^j, a_r), x') = 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(s_i^j, a_{r'})| - \ell_{x'}$, it satisfies that dist $(f(\pi_i^1, a_r), x) > \text{dist}(f(s_i^j, a_r), x)$ and dist $(f(\pi_i^1, a_r), x') < \text{dist}(f(s_i^j, a_r), x')$. Thus in this case, every pair every pair $\{x, x'\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^1, a_r)$. Case 2: $i \neq i'$ or $j \neq j'$. $dist(f(s_i^j, a_r), x) = |P(s_i^j, a_r)| - \ell_x$ when $x \neq a_r$ and $dist(f(s_i^j, a_r), a_r) = 1$. $dist(f(s_i^j, a_r), x') = 1$ min $(2 + |P(s_{i'}^{j'}, a_r)| + \ell_{x'}, 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, a_{r'})| + |P(s_{i'}^{j'}, a_{r'})| - \ell_{x'})$. For the vertex pair $\{x, x'\}$ which is not resolved by $f(s_i^j, a_r)$, i.e., $\operatorname{dist}(f(s_i^j, a_r), x) = \operatorname{dist}(f(s_i^j, a_r), x')$, it satisfies that dist $(f(\pi_i^1, a_r), x) > \text{dist}(f(s_i^j, a_r), x)$ and dist $(f(\pi_i^1, a_r), x') \leq \text{dist}(f(s_i^j, a_r), x')$. Thus in this case, every pair every pair $\{x, x'\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^1, a_r)$. It follows that every pair $\{x, x'\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^1, a_r)$. For a vertex pair $\{z, z'\}$, similarly it is resolved by $f(s_i^j, c_r)$ or $f(\pi_i^1, c_r)$. For a vertex pair $\{y, y'\}$, let $\{u_r^{i_r}, v_r^{i_r}\}$ be the vertex pair resolved by s_i^j , i.e. $|P(s_i^j, b_r)| = 20(n+1) + 5i_r + 1$. Then dist $(f^1(u_r^{i_r}, v_r^{i_r}), y) = 40(n+1) + 1 - \ell_y = \text{dist}(f^2(u_r^{i_r}, v_r^{i_r}), y)$ when $y \neq s_i^j$. We observe that there is a shortest path from $f^h(u_r^{i_r}, v_r^{i_r})$ $(h \in \{1, 2\})$ to y' which either goes through one vertex of $\{a_r, c_r\}$, then goes through $s_{i'}^{j'}$, finally reaches y' or goes through one vertex of $\{a_r, c_r\}$, then goes through some vertex $s_{i''}^{j''}$ $(i'' \in [n], j'' \in [m])$, then goes through $b_{r'}$, finally reaches y'. Thus we get that $\operatorname{dist}(f^1(u_r^{i_r}, v_r^{i_r}), y') = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), y') + 1$. Thus every vertex pair $\{y, y'\}$ such that $y \neq s_i^j$ is resolved by $f^1(u_r^{i_r}, v_r^{i_r})$ or $f^2(u_r^{i_r}, v_r^{i_r})$. For a pair $\{s_i^j, y'\}$, obviously dist $(f^{mid}(i, j, 1), s_i^j) < dist(f^{mid}(i, j, 1), y')$. It follows that every vertex pair $\{y, y'\}$ is resolved by $f^1(u_r^{i_r}, v_r^{i_r})$, $f^2(u_r^{i_r}, v_r^{i_r})$ or $f^{mid}(i, j, 1)$. For a vertex pair $\{x, y'\}$, there are two cases. Case 1: $i \neq i'$ or $j \neq j'$. dist $(f(\pi_i^1, a_r), x) = 2 + |P(s_i^j, a_r)| - \ell_x$.

 $\operatorname{dist}(f(\pi_i^1, a_r), b_{r'}) = 2 + \min_{\alpha \in [m]}(|P(s_i^\alpha, a_r)| + |P(s_i^\alpha, b_{r'})|).$ Then $\operatorname{dist}(f(\pi_i^1, a_r), y') = \min(2 + i)$ $|P(s_{i'}^{j'}, a_r)| + \ell_{y'}, \operatorname{dist}(f(\pi_i^1, a_r), b_{r'}) + |P(s_{i'}^{j'}, b_{r'})| - \ell_{y'}). \quad \operatorname{dist}(f(s_i^j, a_r), x) = |P(s_i^j, a_r)| - \ell_x < 0$ $dist(f(\pi_i^1, a_r), x)$ if $x \neq a_r$ and $dist(f(s_i^j, a_r), a_r) = 2$. $dist(f(s_i^j, a_r), y') = dist(f(\pi_i^1, a_r), y')$. Thus in this case, every pair $\{x, y'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. Case 2: i = i' and j = j'. dist $(f(s_i^j, a_r), y') = \min(|P(s_i^j, a_r)| + \ell_{y'}, \operatorname{dist}(f(\pi_i^1, a_r), b_{r'}) + |P(s_{i'}^{j'}, b_{r'})| - \ell_{y'})$. If $dist(f(s_i^j, a_r), y') = |P(s_i^j, a_r)| + \ell_{y'}$, then $\{x, y'\}$ is obviously resolved by $f(s_i^j, a_r)$. Otherwise, for a vertex pair $\{x, y'\}$ which is not resolved by $f(s_i^j, a_r)$, dist $(f(s_i^j, a_r), x) = \text{dist}(f(s_i^j, a_r), y') =$ $\operatorname{dist}(f(\pi_i^1, a_r), y') < \operatorname{dist}(f(\pi_i^1, a_r), x)$. Thus in this case, every pair $\{x, y'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. It follows that every pair $\{x, y'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. For a vertex pair $\{y, z'\}$, similarly we can show that every pair $\{y, z'\}$ is resolved by $f(\pi_i^1, c_r)$ or $f(s_i^j, c_r)$. For a vertex pair $\{x, z'\}$, there are two cases. Case 1: $i \neq i'$ or $j \neq j'$. dist $(f(\pi_i^1, a_r), x) =$ $\ell_{z'}, |P(\pi_i^1, a_r)| + |P(\pi_i^1, c_{r'})| + |P(s_{i'}^{j'}, c_{r'})| - \ell_{z'}). \operatorname{dist}(f(s_i^j, a_r), z') = \min(2 + |P(s_{i'}^{j'}, a_r)| + \ell_{z'}, 2 + \ell_{z'})$ $|P(\pi_i^1, a_r)| + |P(\pi_i^1, c_{r'})| + |P(s_{i'}^{j'}, c_{r'})| - \ell_{z'} \ge \operatorname{dist}(f(\pi_i^1, a_r), z')$. Thus in this case, every pair $\{x, z'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. Case 2: i = i' and j = j'. dist $(f(s_i^j, a_r), z') =$ $\min\left(|P(s_i^j, a_r)| + \ell_{z'}, 2 + |P(\pi_i^1, a_r)| + |P(\pi_i^1, c_{r'})| + |P(s_{i'}^{j'}, c_{r'})| - \ell_{z'}\right).$ If dist $(f(s_i^j, a_r), z') =$ $|P(s_i^j, a_r)| + \ell_{z'}$ Then $\{x, z'\}$ $(x \neq a_r)$ is obviously resolved by $f(s_i^j, a_r)$. Otherwise, for a vertex pair $\{x, z'\}$ which is not resolved by $f(s_i^j, a_r)$, $\operatorname{dist}(f(\pi_i^1, a_r), x) > \operatorname{dist}(f(s_i^j, a_r), x) =$ $\operatorname{dist}(f(s_i^j, a_r), z') > \operatorname{dist}(f(\pi_i^1, a_r), z')$. Thus every pair $\{x, z'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. It follows that every pair $\{x, z'\}$ is resolved by $f(\pi_i^1, a_r)$ or $f(s_i^j, a_r)$. This completes the proof for the lemma.

Lemma 53. Every pair of distinct vertices $x, y \in \bigcup_{i \in [n], h \in \{1,2\}} S_i^h$ is resolved by S'.

Proof. Let $x \in P(\pi_i^h, a_r)$ for arbitrary integers $i \in [n], h \in \{1, 2\}, r \in \{1, 2, 3\}$. Let $x' \in P(\pi_{i'}^{h'}, a_r)$ for arbitrary integers $i' \in [n], h' \in \{1, 2\}$. We fix an arbitrary integer $j \in [m]$. Let $P(x, a_r)$ be the subpath of $P(\pi_i^h, a_r)$ and $\ell_x = |P(x, a_r)|$. Let $P(x', a_r)$ be the subpath of $P(\pi_{i'}^h, a_r)$ and $\ell_{x'} = |P(x', a_r)|$. For a vertex pair $\{x, x'\}$, dist $(f(s_i^j, a_r), x) = 2 + \ell_x$ and dist $(f(s_i^j, a_r), x') = 2 + \ell_{x'}$. Then every vertex pair $\{x, x'\}$ such that $\ell_x \neq \ell_{x'}$ is resolved by $f(s_i^j, a_r)$. Since dist $(f(\pi_i^h, a_r), x) = \ell_x$ if $x \neq a_r$ and dist $(f(\pi_i^h, a_r), x') = 2 + \ell_{x'}$ if $i \neq i'$ or $h \neq h'$, every vertex pair $\{x, x'\}$ such that $\ell_x = \ell_{x'}$ is resolved by $f(\pi_i^h, a_r)$. It follows that every vertex pair $\{x, x'\}$ is resolved by $f(\pi_i^h, a_r)$ or $f(s_i^j, a_r)$. Let $y \in P(\pi_i^h, c_r)$ for arbitrary integers $i \in [n], h \in \{1, 2\}, r \in \{1, 2, 3\}$. Let $y' \in P(\pi_{i'}^{h'}, c_r)$ for arbitrary integers $i' \in [n], h' \in \{1, 2\}$.

Let $x_1 \in P(\pi_i^h, a_r)$ for arbitrary integers $i \in [n], h \in \{1, 2\}, r \in \{1, 2, 3\}$. Let $x_2 \in P(\pi_{i'}^{h'}, a_{r'})$ for arbitrary integers $i' \in [n], h' \in \{1, 2\}, r' \in \{1, 2, 3\}$. We fix an arbitrary integer $j \in [m]$. We define ℓ_{x_1} and ℓ_{x_2} in a similar way to that of ℓ_x in the first paragraph. For a vertex pair $\{x_1, x_2\}$ such that $r \neq r'$, $\operatorname{dist}(f(s_i^j, a_r), x_1) = 2 + \ell_{x_1}$ and $\operatorname{dist}(f(s_i^j, a_r), x_2) = 2 + |P(\pi_{i'}^{h'}, a_r)| + |P(\pi_{i'}^{h'}, a_{r'})| - \ell_{x_2} = 2 + 20(n + 1) - \ell_{x_2} > \operatorname{dist}(f(s_i^j, a_r), x_1)$ unless $x_1 = \pi_i^h, x_2 = \pi_{i'}^{h'}$ and $\pi_i^h \neq \pi_{i'}^{h'}$. The vertex pair $\{\pi_i^h, \pi_{i'}^{h'}\}$ is obviously resolved by $f^h(i, j, a_r)$. Thus every vertex pair $\{x_1, x_2\}$ such that $r \neq r'$ is resolved by $f(s_i^j, a_r)$ or $f^h(i, j, a_r)$. Let $y_1 \in P(\pi_i^h, c_r)$ for arbitrary integers $i \in [n], h \in \{1, 2\}, r \in \{1, 2, 3\}$. Let $y_2 \in P(\pi_{i'}^{h'}, c_{r'})$ for arbitrary integers $i' \in [n], h' \in \{1, 2\}, r' \in \{1, 2, 3\}$ such that $r \neq r'$. We define ℓ_{y_1} and ℓ_{y_2} in a similar way to that of ℓ_x in last paragraph. Similarly, we can show that every vertex pair $\{y_1, y_2\}$ such that $r \neq r'$ is resolved by $f(s_i^j, a_r)$. For a pair $\{x_1, y_2\}$, $\operatorname{dist}(f(s_i^j, a_r), x_1) = 2 + \ell_{x_1}$ and $\operatorname{dist}(f(s_i^j, a_r), y_2) = 2 + |P(\pi_{i'}^{h'}, a_r)| + |P(\pi_{i'}^{h'}, c_{r'})| - \ell_{y_2} = 2 + 20(n + 1) - \ell_{y_2} > \operatorname{dist}(f(s_i^j, a_r), x_1)$ unless $x_1 = \pi_i^h, y_2 = \pi_{i'}^{h'}$ and $\pi_i^h \neq \pi_{i'}^{h'}$. The vertex pair $\{\pi_i^h, \pi_{i'}^{h'}\}$ is obviously resolved by resolved by resolved by resolved by resolved paragraph. Similarly, we can show that every vertex pair $\{y_1, y_2\}$ such that $r \neq r'$ is resolved by $f(s_i^j, c_r)$ or $f^h(i, j, a_r)$. For a pair $\{x_1, y_2\}$, $\operatorname{dist}(f(s_i^j, a_r), x_1) = 2 + \ell_{x_1}$ and $\operatorname{dist}(f(s_i^j, a_r), y_2) = 2 + |P(\pi_{i'}^{h'}, a_r)| + |P(\pi_{i'}^{h'}, c_{r'})| - \ell_{y_2} = 2 + 20(n + 1) - \ell_{y_2} > \operatorname{dist}(f(s_i^j, a_r), x_1)$ unless $x_1 = \pi_i^h, y_2 = \pi_{i'}^{h'}$ and $\pi_i^h \neq \pi_$

 $f^h(i, j, a_r)$. Thus every vertex pair $\{x_1, y_2\}$ is resolved by $f(s_i^j, a_r)$ or $f^h(i, j, a_r)$. This completes the proof for the lemma.

Lemma 54. Every pair of distinct vertices $x, y \in \bigcup_{i \in [n], h \in \{1,2\}, j \in [m], r \in \{1,2,3\}} \Pi^h(i, j, r)$ is resolved by S'.

Proof. Let $x_1, x_2 \in P^h(i, j, a_r)$ be two distinct vertices for arbitrary integers $i \in [n], j \in [m], h \in \{1, 2\}, r \in \{1, 2, 3\}$. Let $j' \in [m]$ be an integer such that $j \neq j'$. Obviously the pair $\{x_1, x_2\}$ is resolved by $f^h(i, j', a_r)$. Similarly, the vertex pair $\{y_1, y_2\}$ for two distinct vertices $y_1, y_2 \in P^h(i, j, b_r)$, the vertex pair $\{z_1, z_2\}$ for two distinct vertices $z_1, z_2 \in P^h(i, j, c_r)$, and the vertex pair $\{w_1, w_2\}$ for two distinct vertices $w_1, w_2 \in P^h(i, j, a_r)$. Let $x \in P^h(i, j, a_r), y \in P^h(i, j, b_r), z \in P^h(i, j, c_r)$ and $w \in P^h(i, j, p_i^{3-h})$ for arbitrary induction.

tegers $i \in [n], j \in [m], h \in \{1, 2\}, r \in \{1, 2, 3\}$. Let $x' \in P^{h'}(i', j', a_{r'}), y' \in P^{h'}(i', j', b_{r'}), z' \in P^{h'}(i', j', c_{r'})$ and $w' \in P^{h'}(i', j', p_{i'}^{3-h'})$ for arbitrary integers $i' \in [n], j' \in [m], h' \in \{1, 2\}, r' \in P^{h'}(i', j', c_{r'})$ $\{1,2,3\}$. We define $\ell_x = \operatorname{dist}(x,\pi_i^h)$. In a similar way, we define $\ell_y, \ell_z, \ell_w, \ell_{x'}, \ell_{y'}, \ell_{z'}, \ell_{w'}$. For a pair $\{x, y\}$, dist $(f^h(i, j, c_r), x) = 2 + \ell_x$ and dist $(f^h(i, j, c_r), y) = 2 + \ell_y$. Thus $f^h(i, j, c_r)$ resolves every pair $\{x, y\}$ such that $\ell_x \neq \ell_y$. Since dist $(f^h(i, j, a_r), x) = \ell_x$ if $x \neq \pi_i^h$, dist $(f^h(i, j, a_r), \pi_i^h) = \ell_x$ 2 and dist $(f^h(i, j, c_r), y) = 2 + \ell_y$, $f^h(i, j, a_r)$ resolves every pair $\{x, y\}$ such that $\ell_x = \ell_y$. Thus every pair $\{x, y\}$ is resolved by $f^h(i, j, a_r)$ or $f^h(i, j, c_r)$. In a similar way, we can show that two distinct vertices from $\bigcup_{i \in [m], r \in \{1,2,3\}} \Pi^h(i,j,r)$ are distinguished by S'. For a pair $\{x, y'\}$ with i = i' and $h \neq h'$, $\operatorname{dist}(f^h(i, j, c_r), x) = 2 + \ell_x < \operatorname{dist}(f^h(i, j, c_r), y') = 2$ $\min (2 + |P(\pi_i^h, a_r)| + |P(\pi_i^{h'}, a_r)| + \ell_{y'}, 2 + |P(\pi_i^h(i, j', b_{r'}))| + |P(\pi_i^{h'}(i, j', b_{r'})| - \ell_{y'}).$ Thus every pair $\{x, y'\}$ with i = i' and $h \neq h'$ is resolved by $f^h(i, j, c_r)$. Similarly we can show that every vertex pair of $\bigcup_{j \in [m], r \in \{1,2,3\}} \Pi^{h}(i,j,r) \times \bigcup_{j \in [m], r \in \{1,2,3\}} \Pi^{3-h}(i,j,r)$ is resolved by S'. For a pair $\{x, y'\}$ with $i \neq i'$, dist $(f^h(i, j, c_r), x) = 2 + \ell_x$. dist $(f^h(i, j, c_r), s_{i'}^{j'}) = \min_{d \in \{1, 2, 3\}} (2 + \ell_x)$ $|P(\pi_i^h, c_d)| + |P(s_{i'}^{j'}, c_d)|).$ Thus dist $(f^h(i, j, c_r), y') = \min(2 + |P(\pi_i^h, a_r)| + |P(\pi_i^{h'}, a_r)| + |P(\pi_i^{h'},$ $\ell_{y'}, \operatorname{dist}(f^{h}(i, j, c_{r}), s_{i'}^{j'}) + 1 + |P^{h'}(i', j', b_{r'})| - \ell_{y'}). \text{ We get } \operatorname{dist}(f^{h}(i, j, c_{r}), x) < \operatorname{dist}(f^{h}(i, j, c_{r}), y') \text{ unless } \ell_{x} = 20(n+1) \text{ and } \ell_{y'} = 0. \text{ If } \ell_{x} = 20(n+1) \text{ and } \ell_{y'} = 0, \operatorname{dist}(f^{h}(i, j, a_{r}), y') = 2 + 20(n+1) \text{ and } \ell_{y'} = 0.$ 1) > dist $(f^h(i, j, a_r), x) = 20(n+1)$. Thus every pair $\{x, y'\}$ with $i \neq i'$ is resolved by $f^h(i, j, c_r)$ or $f^h(i, j, a_r)$. Similarly we can show that every vertex pair of $\bigcup_{j \in [m], r \in \{1, 2, 3\}} \Pi^h(i, j, r) \times$ $\bigcup_{j' \in [m], r' \in \{1,2,3\}} \prod^{h'}(i', j', r')$ with $i \neq i'$ is resolved by S'. This completes the proof for the lemma.

Lemma 55. Every pair of distinct vertices $x, y \in \bigcup_{i \in [n], h \in \{1,2\}} L_i^h$ is resolved by S'.

 $\begin{array}{l} Proof. \mbox{ First we show that every vertex pair of } L_i^h \times L_i^h \mbox{ is resolved by } S' \mbox{ for } i \in [n], h \in \{1,2\}. \mbox{ We fix arbitrary integers } i \in [n], j \in [m] \mbox{ and } h \in \{1,2\}. \mbox{ For a vertex } x \in P(q_i^h, \min(P^{3-h}(i,j,p_i^h))), \mbox{ let } P(q_i^h, x) \mbox{ be the subpath of } P(q_i^h, \min(P^{3-h}(i,j,p_i^h))) \mbox{ from } q_i^h \mbox{ to } x \mbox{ and let } |P(q_i^h,x)| = \ell_x. \mbox{ For two distinct vertices } x_1, x_2 \in P(q_i^h, \min(P^{3-h}(i,j,p_i^h))), \mbox{ dist}(f^{mid}(i,j,3-h),x_1) = 1 + |P(q_i^h, \min(P^{3-h}(i,j,p_i^h)))| - \ell_{x_1} = 30(n+1) - \ell_{x_1} \mbox{ and } \operatorname{dist}(f^{mid}(i,j,3-h),x_2) = 30(n+1) - \ell_{x_2}. \mbox{ Thus } f^{mid}(i,j,3-h) \mbox{ resolves every pair } \{x_1,x_2\}. \mbox{ Let } x \in P(q_i^h, \min(P^{3-h}(i,j,p_i^h))) \mbox{ and } x' \in P(q_i^h, \min(P^{3-h}(i,j',p_i^h))) \mbox{ with some integer } j' \neq j. \mbox{ dist}(f^h(i,j,a_1),x) = 3 + \ell_x \mbox{ and } \operatorname{dist}(f^{hid}(i,j,a_1),x') = 3 + \ell_{x'}. \mbox{ Thus } f^h(i,j,a_1) \mbox{ resolves every pair } \{x,x'\} \mbox{ such that } \ell_x = \ell_{x'}, \mbox{ dist}(f^{mid}(i,j,3-h),x) = 30(n+1) - \ell_x \mbox{ and } \operatorname{dist}(f^{mid}(i,j,3-h),x') = \min (1 + |P(q_i^h, \min(P^{3-h}(i,j,p_i^h)))| + \ell_{x'}, 1 + |P^{3-h}(i,j,p_i^h)|/2 + |P^{3-h}(i,j',p_i^h)|/2 + |P(q_i^h, \min(P^{3-h}(i,j',p_i^h)))| - \ell_{x'}) \mbox{ min } (30(n+1) + \ell_{x'}, 50(n+1) - \ell_{x'}). \mbox{ Thus } \operatorname{dist}(f^{mid}(i,j,3-h),x') \mbox{ dist}(f^{mid}(i,j,3-h),x') \mbox{ and } f^{mid}(i,j,3-h) \mbox{ resolves this pair. It follows that every pair } \{x,x'\} \mbox{ is resolved by } f^{mid}(i,j,3-h) \mbox{ or } f^h(i,j,a_1). \mbox{ and } f^{mid}(i,j,3-h) \mbox{ or } f^h(i,j,a_1). \mbox{ for } f^h(i,j,a_1). \mbox{ for } f^h(i,j,a_1). \mbox{ for } f^h(i,j,a_1). \mbox{ for } f^h(i,j,a_1). \mbox{ dist}(f^{mid}(i,j,3-h),x') \mbox{ for } f^h(i,j,3-h) \mbox{ or } f^h(i,j,a_1). \mbox{ for } f^h(i,j,a_1). \mb$

Next we show that every vertex pair of $L_i^h \times L_i^{3-h}$ is resolved by S' for $i \in [n], h \in \{1, 2\}$. We fix arbitrary integers $i \in [n], j \in [m]$ and $h \in \{1, 2\}$. Let $x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ and $y \in [m]$

 $P(q_i^{3-h}, \operatorname{mid}(P^h(i, j, p_i^{3-h}))))$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in last paragraph. For a pair $\{x, y\}$, dist $(f^{mid}(i, j, 3-h), x) = 30(n+1) - \ell_x$ and dist $(f^{mid}(i, j, 3-h), y) = \min(2+1)$ $|P^{3-h}(i,j,p_i^{h})|/2 + \ell_y, 3 + |P^{3-h}(i,j,p_i^{h})|/2 + |P^{h}(i,j,p_i^{3-h})|/2 + |P(q_i^{3-h},\operatorname{mid}(P^{h}(i,j,p_i^{3-h})))| - |P^{3-h}(i,j,p_i^{3-h})|/2 + |P^{3-h}(i,j,p_i^{3-h})$ ℓ_y). For a pair $\{x, y\}$ which is not resolved by $f^{mid}(i, j, 3 - h)$, there are two cases. Case 1: $\operatorname{dist}(f^{mid}(i, j, 3-h), y) = 2 + |P^{3-h}(i, j, p_i^h)|/2 + \ell_y = 2 + 10(n+1) + \ell_y \le 2 + 50(n+1) - \ell_y$ when $\ell_{y} \leq 20(n+1)$. We have $dist(f^{mid}(i,j,3-h),x) = 30(n+1) - \ell_{x} = dist(f^{mid}(i,j,3-h),x) = 30(n+1) - \ell_{x} = dist(f^{mid}(i,j,3-h),x) = dist($ $h(y) = 2 + 10(n+1) + \ell_y$, i.e. $\ell_x + \ell_y = 20(n+1) - 2$. Case 2: dist $(f^{mid}(i, j, 3-h), y) = 0$ $2 + 50(n+1) - \ell_y$ when $\ell_y > 20(n+1)$. We have $dist(f^{mid}(i, j, 3-h), x) = 30(n+1) - \ell_x = 100(n+1)$ $dist(f^{mid}(i, j, 3-h), y) = 2 + 50(n+1) - \ell_y, i.e. \ \ell_y - \ell_x = 20(n+1) + 2. \ dist(f^{3-h}(i, j, a_1), y) = 3 + \ell_y. \ dist(f^{3-h}(i, j, a_1), x) = 3 + |P(\pi_i^{3-h}, a_1)| + |P(\pi_i^{h}, a_1)| + \ell_x \text{ when } \ell_x \le 10(n+1) \text{ and } \ell_x \le 10(n+1)$ $\operatorname{dist}(f^{3-h}(i,j,a_1),x) = 2 + |P^{3-h}(i,j,p_i^h)|/2 + |P(q_i^h,\operatorname{mid}(P^{3-h}(i,j,p_i^h)))| - \ell_x \text{ when } \ell_x > 10(n+1)$ 1). For a pair $\{x, y\}$ which is not resolved by $f^{3-h}(i, j, a_1)$, there are two cases. Case 1: dist $(f^{3-h}(i, j, a_1), y)$ = dist $(f^{3-h}(i, j, a_1), x)$ when $\ell_x \le 10(n+1)$, i.e. $\ell_y - \ell_x = 3 + 20(n+1)$. Case 2: dist $(f^{3-h}(i, j, a_1), y) = dist(f^{3-h}(i, j, a_1), x)$ when $\ell_x > 10(n+1)$, i.e. $\ell_y + \ell_x = 10(n+1)$ 40(n+1) - 2. Thus we see that if a pair $\{x, y\}$ is not resolved by $f^{mid}(i, j, 3 - h)$, then it is resolved by $f^{3-h}(i, j, a_1)$. It follows that every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, 3-h)$ and $f^{3-h}(i,j,a_1)$. Let $x' \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j',p_i^h)))$ with some integer $j' \in [m]$ and $j' \neq j$. Then dist $(f^{mid}(i, j', 3-h), x') = 30(n+1) - \ell_{x'}$ and dist $(f^{mid}(i, j', 3-h), y) = 2 + |P^{3-h}(i, j', p_i^h)|/2 + |P^{3-h}(i, j', p_i^h)$ $\ell_y = 2 + 10(n+1) + \ell_y$. For a pair $\{x', y\}$ which is not resolved by $f^{mid}(i, j', 3-h)$, it satisfies that $30(n+1) - \ell_{x'} = 2 + 10(n+1) + \ell_y$, i.e. $\ell_{x'} + \ell_y = 20(n+1) - 2$. Similar to vertex x, for vertex x', $\operatorname{dist}(f^{3-h}(i, j, a_1), x') = 3 + |P(\pi_i^{3-h}, a_1)| + |P(\pi_i^h, a_1)| + \ell_{x'}$ when $\ell_{x'} \le 10(n+1)$ and $\operatorname{dist}(f^{3-h}(i, j, a_1), x') = 2 + |P^{3-h}(i, j', p_i^h)|/2 + |P(q_i^h, \operatorname{mid}(P^{3-h}(i, j', p_i^h)))| - \ell_{x'}$ when $\ell_{x'} > 10(n+1)$. Thus for a pair $\{x', y\}$ which is not resolved by $f^{3-h}(i, j, a_1)$, there are two cases. Case 1: dist $(f^{3-h}(i, j, a_1), y) = dist(f^{3-h}(i, j, a_1), x')$ when $\ell_{x'} \leq 10(n+1)$, i.e. $\ell_y - \ell_{x'} = 3 + 20(n+1)$. Case 2: dist $(f^{3-h}(i, j, a_1), y) = dist(f^{3-h}(i, j, a_1), x')$ when $\ell_{x'} > 10(n+1)$, i.e. $\ell_y + \ell_{x'} = 40(n+1) - 2$. Thus we see that if a pair $\{x', y\}$ is not resolved by $f^{mid}(i, j', 3-h)$, then it is resolved by $f^{3-h}(i, j, a_1)$. It follows that every pair $\{x', y\}$ is resolved by $f^{mid}(i, j', 3-h)$ and $f^{3-h}(i, j, a_1)$.

Finally we show that every vertex pair of $L_i^h \times L_{i'}^{h'}$ is resolved by S' for $i, i' \in [n], h, h' \in \{1, 2\}$ and $i \neq i'$. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h, h' \in \{1, 2\}$ such that $i \neq i'$. Let $x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ and $y \in P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the first paragraph. For a pair $\{x, y\}$, $\operatorname{dist}(f^{mid}(i, j, 3 - h), x) = 30(n+1) - \ell_x$ and $\operatorname{dist}(f^{mid}(i, j, 3-h), y) = \min(2+|P^{3-h}(i, j, p_i^h)|/2+|P(\pi_i^{3-h}, a_1)|+|P(\pi_{i'}^{3-h'}, a_1)|+|\ell_y, 1+|P^{3-h'}(i, j, p_i^h)|/2+|P(\pi_i^{3-h}, a_1)|+|P(\pi_{i'}^{3-h'}, a_1)|+|P^{3-h'}(i', j', p_{i'}^{h'})|+|P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))|-\ell_y) \geq 1+30(n+1) > \operatorname{dist}(f^{mid}(i, j, 3-h), x)$. It follows that every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, 3-h)$. This completes the proof for the lemma. \Box

Lemma 56. Every pair of distinct vertices $x, y \in \bigcup_{r \in \{1,2,3\}} R_r$ is resolved by S'.

Proof. Let's fix an arbitrary integer $r \in \{1, 2, 3\}$.

First, we show that every pair of distinct vertices of $\bigcup_{i \in [n]} (P(a_r, u_r^i) \cup P(a_r, v_r^i))$ is resolved by S'. Let's fix an arbitrary integer $i \in [n]$. Let $x_u \in P(a_r, u_r^i)$. Let $P(a_r, x_u)$ be the subpath of $P(a_r, u_r^i)$ from a_r to x_u and let $\ell_{x_u} = |P(a_r, x_u)|$. Since dist $(f(\pi_1^1, a_r), x_u) = 2 + \ell_{x_u}$, obviously two distinct vertices of $P(a_r, u_r^i)$ are distinguished by $f(\pi_1^1, a_r)$. Let $x_v \in P(a_r, v_r^i)$. Let $P(a_r, x_v)$ be the subpath of $P(a_r, v_r^i)$ from a_r to x_v and let $\ell_{x_v} = |P(a_r, x_v)|$. Since dist $(f(\pi_1^1, a_r), x_v) =$ $2 + \ell_{x_v}$, obviously two distinct vertices of $P(a_r, v_r^i)$ are distinguished by $f(\pi_1^1, a_r)$. For the pair $\{x_u, x_v\}$, if $\ell_{x_u} \neq \ell_{x_v}$, then it is resolved by $f(\pi_1^1, a_r)$. Otherwise, if $\ell_{x_u} = \ell_{x_v} < |P(a_r, u_r^i)|$, then dist $(f^1(u_r^i, v_r^i), x_u) = 1 + |P(a_r, u_r^i)| - \ell_{x_u} < \text{dist}(f^1(u_r^i, v_r^i), x_v) = 2 + |P(a_r, v_r^i)| - \ell_{x_v}$. By Claim 5, the vertex pair $\{u_r^i, v_r^i\}$ is resolved by S'. Thus every pair $\{x_u, x_v\}$ is resolved by S'. Let $x'_u \in P(a_r, u_r^i)$ and $x'_v \in P(a_r, v_r^i)$ for some integer $i' \in [n]$ such that $i' \neq i$. We define $\ell_{x'_u}$ and $\ell_{x'_v}$ in a similar way to that of ℓ_{x_u} and ℓ_{x_v} . For a pair $\{x_u, x'_u\}$, dist $(f^1(u^i_r, v^i_r), x_u) = 1 + |P(a_r, u^i_r)| - \ell_{x_u} < \text{dist}(f^1(u^i_r, v^i_r), x'_u) = 1 + |P(a_r, u^i_r)| + \ell_{x'_u}$. Thus every pair $\{x_u, x'_u\}$ is resolved by $f^1(u^i_r, v^i_r)$. Similarly, we can show that every pair $\{x_u, x'_v\}$, $\{x_v, x'_v\}$ is resolved by $f^1(u^i_r, v^i_r)$.

Then we show that every pair of distinct vertices of $\bigcup_{i \in [n]} (P(c_r, u_r^i) \cup P(c_r, v_r^i))$ is resolved by S'. Let's fix arbitrary integers $i, i' \in [n]$ such that $i \neq i'$. Let $z_u \in P(c_r, u_r^i), z_v \in P(c_r, v_r^i)$, $z'_u \in P(c_r, u_r^{i'})$ and $z'_v \in P(c_r, v_r^{i'})$. We define $\ell_{z_u}, \ell_{z_v}, \ell_{z'_u}$ and $\ell_{z'_v}$ in a similar way to that of ℓ_{x_u} and ℓ_{z_v} in last paragraph. Since dist $(f(\pi_1^1, c_r), z_u) = 2 + \ell_{z_u}$, obviously two distinct vertices of $P(c_r, u_r^i)$ are distinguished by $f(\pi_1^1, c_r)$. Since $\operatorname{dist}(f(\pi_1^1, c_r), z_v) = 2 + \ell_{z_v}$, two distinct vertices of $P(c_r, v_r^i)$ are distinguished by $f(\pi_1^1, c_r)$. For a pair $\{z_u, z_v\}$, if $\ell_{z_u} \neq \ell_{z_v}$, then it is resolved by $f(\pi_1^1, c_r)$. Otherwise, if $\ell_{z_u} = \ell_{z_v} < |P(c_r, u_r^i)|$, then $dist(f^1(u_r^i, v_r^i), z_u) = 1 + |P(c_r, u_r^i)| - \ell_{z_u} < 1 + |P(c_r, u_r^i)$ $\operatorname{dist}(f^1(u_r^i, v_r^i), z_v) = 2 + |P(c_r, v_r^i)| - \ell_{z_v}$. Thus every pair $\{z_u, z_v\}$ is resolved by S'. For a pair $\{z_u, z'_u\}$, if $\ell_{z_u} \neq \ell_{z'_u}$, then it is resolved by $f(\pi_1^1, c_r)$. Otherwise if $\ell_{z_u} = \ell_{z'_u}$, then $\operatorname{dist}(f^{1}(u_{r}^{i}, v_{r}^{i}), z_{u}^{\prime}) = \min\left(1 + |P(c_{r}, u_{r}^{i})| + \ell_{z_{u}^{\prime}}, 1 + |P(a_{r}, u_{r}^{i})| + |P(a_{r}, u_{r}^{i^{\prime}})| + |P(c_{r}, u_{r}^{i^{\prime}})| - \ell_{z_{u}^{\prime}} - 2\right)$ if $|P(c_r, u_r^{i'})| - \ell_{z'_u} \ge 2$. dist $(f^1(u_r^i, v_r^i), z'_u) = 1 + |P(a_r, u_r^i)| + |P(a_r, u_r^{i'})| + |P(c_r, u_r^{i'})| - \ell_{z'_u}$ if $|P(c_r, u_r^{i'})| - \ell_{z'_u} < 2$. If dist $(f^1(u_r^i, v_r^i), z'_u) = 1 + |P(c_r, u_r^i)| + \ell_{z'_u}$, then $\{z_u, z'_u\}$ is resolved by $f^{1}(u_{r}^{i}, v_{r}^{i})$. Otherwise, if dist $(f^{1}(u_{r}^{i}, v_{r}^{i}), z_{u}^{\prime}) = 1 + |P(a_{r}, u_{r}^{i})| + |P(a_{r}, u_{r}^{i^{\prime}})| + |P(c_{r}, u_{r}^{i^{\prime}})| - \ell_{z_{u}^{\prime}}$ suppose that there is a pair $\{z_u, z'_u\}$ which is not resolved by $f^1(u_r^i, v_r^i)$. We get that i = 2(n+1), a contradiction. If dist $(f^1(u_r^i, v_r^i), z'_u) = 1 + |P(a_r, u_r^i)| + |P(a_r, u_r^{i'})| + |P(c_r, u_r^{i'})| - \ell_{z'_u} - 2$, suppose that there is a pair $\{z_u, z'_u\}$ which is not resolved by $f^1(u_r^i, v_r^i)$. We get that 20i = 40(n+1) - 2, a contradiction. Thus every pair $\{z_u, z'_u\}$ is resolved by $f^1(u_r^i, v_r^i)$ or $f(\pi_1^1, c_r)$. Similarly, we can show that every pair $\{z_u, z'_v\}$ and $\{z_v, z'_v\}$ is resolved by $f^1(u_r^i, v_r^i)$ or $f(\pi_1^1, c_r)$.

Next we show that every pair of distinct vertices of $\bigcup_{i \in [n]} (P(b_r, u_i^i) \cup P(b_r, v_r^i))$ is resolved by S'. Let's fix an arbitrary integer $i, i' \in [n]$ such that $i' \neq i$ Let $y_u \in P(b_r, u_r^i), y_v \in P(b_r, v_r^i), y'_u \in P(b_r, v_r^i)$ and $y'_v \in P(b_r, v_r^i)$. We define $\ell_{y_u}, \ell_{y_v}, \ell_{y'_u}$ and $\ell_{y'_v}$ in a similar way to that of ℓ_{x_u} and ℓ_{x_v} in the second paragraph. Since $(G, n, \chi, \mathcal{P})$ is a YES-instance, by Claim 5, the pair $\{u_i^r, v_i^r\}$ is resolved by some vertex of S, say s_η^τ . Since dist $(s_\eta^\tau, y_u) = |P(s_\eta^\tau, b_r)| + \ell_{y_u} =$ $20(n+1) + 5i + 1 + \ell_{y_u}$, every vertex pair of $P(b_r, u_i^i)$ is resolved by s_η^τ . Since dist $(s_\eta^\tau, y_v) =$ $|P(s_\eta^\tau, b_r)| + \ell_{y_v} = 20(n+1) + 5i + 1 + \ell_{y_v}$, every vertex pair of $P(b_r, v_i^i)$ is resolved by s_η^τ . For a pair $\{y_u, y_v\}$, if $\ell_{y_u} \neq \ell_{y_v}$, then it is resolved by s_η^τ . For a pair $\{y_u, y_v\}$ such that $\ell_{y_u} = \ell_{y_v}$, dist $(f^1(u_r^i, v_r^i), y_u) = 2 + |P(b_r, u_i^i)| - \ell_{y_u} = 1 + 20(n+1) - 5i - \ell_{y_u} > \text{dist}(f^1(u_r^i, v_r^i), y_v) =$ $2 + |P(b_r, v_i^r)| - \ell_{y_v} = 20(n+1) - 5i - \ell_{y_v}$. Thus every pair $\{y_u, y_v\}$ is resolved by s_η^τ or $f^1(u_r^i, v_r^i)$. For a pair $\{y_u, y'_u\}$ such that $y_u \neq b_r$ and $y'_u \neq b_r$, dist $(f^1(u_r^i, v_r^i), y_u) = 1 + 20(n+1) - 5i - \ell_{y_u} \le$ 20(n+1) - 5i. dist $(f^1(u_r^i, v_r^i), y'_u) = \min (2 + |P(b_r, v_r^i)| + \ell_{y'_u}, 1 + |P(a_r, u_r^i)| + |P(a_r, u_r^i)| + |P(b_r, u_r^i)| - \ell_{y'_u}) > 20(n+1) - 5i$. Thus every pair $\{y_u, y'_u\}$ such that $y_u \neq b_r$ and $y'_u \neq b_r$ is resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{y_u, y'_v\}$ and $\{y_v, y'_v\}$ is resolved by $f^1(u_r^i, v_r^i)$.

Then we show that every pair of distinct vertices of R_r is resolved by S'. Let's fix arbitrary integers $i, i' \in [n]$ such that $i' \neq i$. Let $x_u \in P(a_r, u_r^i), x_v \in P(a_r, v_r^i), y_u \in P(b_r, u_r^i), y_v \in P(b_r, v_r^i), z_u \in P(c_r, u_r^i)$ and $z_v \in P(c_r, v_r^i)$. Let $x'_u \in P(a_r, u_r^i'), x'_v \in P(a_r, v_r^i), y'_u \in P(b_r, u_r^i), y'_v \in P(b_r, v_r^i), z'_u \in P(c_r, u_r^i)$ and $z'_v \in P(c_r, v_r^i)$. We define $\ell_{x_u}, \ell_{x_v}, \ell_{y_u}, \ell_{y_v}, \ell_{z_u}, \ell_{z_v}, \ell_{x'_u}, \ell_{x'_v}, \ell_{y'_u}, \ell_{y'_v}, \ell_{z'_u}$ and $\ell_{z'_v}$ in a similar way to that of ℓ_{x_u} and ℓ_{x_v} in the second paragraph. For a pair $\{x_u, y_u\}$, dist $(f(\pi_1^1, a_r), x_u) = 2 + \ell_{x_u} < \text{dist}(f(\pi_1^1, a_r), y_u) = 2 + |P(a_r, u_r^i)| + |P(b_r, u_r^i)| - \ell_{y_u}$. Thus every pair $\{x_u, y_u\}$ is resolved by $f(\pi_1^1, a_r)$. Similarly, every pair $\{x_u, y_v\}, \{x_v, y_v\}$ and $\{x_v, y_u\}$ are resolved by $f(\pi_1^1, a_r), z_u) = 2 + |P(a_r, u_r^i)| - \ell_{z_u} - 2, |P(\pi_1^1, a_r)| + |P(\pi_1^1, c_r)| + \ell_{z_u})$ if $|P(c_r, u_r^i)| - \ell_{z_u} \geq 2$. dist $(f(\pi_1^1, a_r), x_u) = 2 + |P(a_r, u_r^i)| + |P(c_r, u_r^i)| - \ell_{z_u}$ if $|P(c_r, u_r^i)| - \ell_{z_u}$ if $|P(c_r, u_r^i)| - \ell_{z_u}$ is resolved by $f(\pi_1^1, a_r), x_u) = 2 + \ell_{x_u}$. Thus every pair $\{x_u, z_u\}$ are resolved by $f(\pi_1^1, a_r), x_u) = 2 + |P(a_r, u_r^i)| - \ell_{z_u}$ if $|P(c_r, u_r^i)| - \ell_{z_u} \geq 2$. dist $(f(\pi_1^1, a_r), x_u) = 2 + |P(a_r, u_r^i)| - \ell_{z_u}$ dist $(f(\pi_1^1, a_r), x_u) = 2 + |P(a_r, u_r^i)| - \ell_{z_u}$ if $|P(c_r, u_r^i)| - \ell_{z_u}$ dist $(f(\pi_1^1, a_r), z_u)$ are resolved by $f(\pi_1^1, a_r), x_u) = 2 + \ell_{x_u}$ dist $(f(\pi_1^1, a_r), z_u) = 2 + |P(a_r, u_r^i)| + |P(c_r, u_r^i)| - \ell_{z_u}$ if $|P(c_r, u_r^i)| - \ell_{z_u}$ if $|P(c_r, u_r^i)| - \ell_{z_u}$ dist $(f(\pi_1^1, a_r), x_u) = 2 + \ell_{x_u}$ dist $(f(\pi_1^1, a_r), z_u)$. Thus every pair $\{x_u, z_u\}$ is resolved by $f(\pi_1^1, a_r), x_u) = 2 + \ell_{x_u}$ dist $(f(\pi_1^1, a_r), z_u)$. Thus every pair $\{x_u, z_u\}$ is resolved by $f(\pi_1^1, a_r), x_u$ are re-

solved by $f(\pi_1^1, a_r)$. For a pair $\{y_u, z_u\}$, $\operatorname{dist}(f(\pi_1^1, c_r), z_u) = 2 + \ell_{z_u}$. $\operatorname{dist}(f(\pi_1^1, c_r), b_r) = \min_{j \in [m]}(|P(c_r, s_j^1)| + |P(s_j^1, b_r)|) = 35n + 41$. $\operatorname{dist}(f(\pi_1^1, c_r), y_u) = \min(2 + |P(c_r, u_r^i)| + |P(b_r, u_r^i)| - \ell_{y_u}, \operatorname{dist}(f(\pi_1^1, c_r), b_r) + \ell_{y_u}) > \operatorname{dist}(f(\pi_1^1, c_r), z_u)$. Thus every pair $\{y_u, z_u\}$ is resolved by $f(\pi_1^1, a_r)$. Similarly, every pair $\{y_u, z_v\}$, $\{y_v, z_v\}$ and $\{y_v, z_u\}$ are resolved by $f(\pi_1^1, c_r)$. For a pair $\{x_u, y_u'\}$, $\operatorname{dist}(f^1(u_r^i, v_r^i), x_u) = 1 + |P(a_r, u_r^i)| - \ell_{x_u}$ if $x_u \neq u_r^i$ and $\operatorname{dist}(f^1(u_r^i, v_r^i), u_r^i) = 2$. $\operatorname{dist}(f^1(u_r^i, v_r^i), y_u') = \min(1 + |P(a_r, u_r^i)| + |P(a_r, u_r^i')| + |P(b_r, u_r^i')| - \ell_{y_u'}, 2 + |P(b_r, v_r^i)| + \ell_{y_u'}) > \operatorname{dist}(f^1(u_r^i, v_r^i), x_u)$. Thus every pair $\{x_u, y_u'\}$ is resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{x_u, y_u'\}$, $\{x_v, y_u'\}$, $\{x_v, y_u'\}$, $\{x_v, y_u'\}$, $\{x_u, z_u'\}$, $\{x_v, z_v'\}$, and $\{x_v, z_u'\}$ are resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{x_u, v_u'\}$, $\{x_v, y_u'\}$, $\{x_v, y_u'\}$, $\{x_v, y_u'\}$, $\{x_v, z_v'\}$, and $\{x_v, z_u'\}$ are resolved by $f^1(u_r^i, v_r^i)$. For a pair $\{y_u, z_u'\}$, $\operatorname{dist}(f^1(u_r^i, v_r^i), y_u) = 2 + |P(b_r, u_r^i)| - \ell_{y_u}$ if $y_u \neq b_r$ and $\operatorname{dist}(f^1(u_r^i, v_r^i), y_u) = 1 + |P(b_r, u_r^i)| + |P(c_r, u_r^i)| - \ell_{y_u}$ if $y_u \neq b_r$ and $\operatorname{dist}(f^1(u_r^i, v_r^i), y_u) = 1 + |P(b_r, u_r^i)| + |P(c_r, u_r^i)| - \ell_{z_u'} \geq 2$. $\operatorname{dist}(f^1(u_r^i, v_r^i), z_u') = \min(1 + |P(a_r, u_r^i)| + |P(a_r, u_r^i)| + |P(c_r, u_r^i)| + |P(c_r, u_r^i)| - \ell_{z_u'} + |P(c_r, u_r^i)| - \ell_{z_u'} < 2$. It follows that dist $(f^1(u_r^i, v_r^i), z_u') > 20(n + 1) > \operatorname{dist}(f^1(u_r^i, v_r^i), y_u)$. Thus every pair $\{y_u, z_u'\}$ is resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{y_u, z_v'\}$, $\{y_v, z_v'\}$ and $\{y_v, z_u'\}$ are resolved by $f^1(u_r^i, v_r^i)$. As a result, every pair of distinct vertices of R_r is resolved by S'.

Finally, we show that every vertex pair of $R_r \times R_{r'}$ with $r' \in \{1, 2, 3\}$ and $r' \neq r$ is resolved by S'. Let's fix arbitrary integers $i, i' \in [n]$ and $r' \in \{1, 2, 3\}$ such that $r' \neq r$. Let $x_u \in$ $P(a_r, u_r^i), x_v \in P(a_r, v_r^i), y_u \in P(b_r, u_r^i), y_v \in P(b_r, v_r^i), z_u \in P(c_r, u_r^i) \text{ and } z_v \in P(c_r, v_r^i).$ Let $x'_{u} \in P(a_{r'}, u^{i'}_{r'}), x'_{v} \in P(a_{r'}, v^{i'}_{r'}), y'_{u} \in P(b_{r'}, u^{i'}_{r'}), y'_{v} \in P(b_{r'}, v^{i'}_{r'}), z'_{u} \in P(c_{r'}, u^{i'}_{r'}) \text{ and } z'_{v} \in P(c_{r'}, u^{i'}_{r'})$ $P(c_{r'}, v_{r'}^{i'})$. We define $\ell_{x_u}, \ell_{x_v}, \ell_{y_u}, \ell_{y_v}, \ell_{z_u}, \ell_{z_v}, \ell_{x'_u}, \ell_{x'_v}, \ell_{y'_u}, \ell_{y'_v}, \ell_{z'_u}$ and $\ell_{z'_v}$ in a similar way to that of ℓ_{x_u} and ℓ_{x_v} in the second paragraph. For a pair $\{x_u, x'_u\}$, $\operatorname{dist}(f(s_1^1, a_r), x_u) = 2 + \ell_{x_u} < 20(n + 1)$ 1) $< \operatorname{dist}(f(s_1^1, a_r), x'_u) = 2 + |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + \ell_{x'_u}$. Thus every pair $\{x_u, x'_u\}$ is resolved by $f(s_1^1, a_r)$. Similarly, every pair $\{x_u, x'_v\}$, $\{x_v, x'_u\}$ and $\{x_v, x'_v\}$ are resolved by $f(s_1^1, a_r)$. For a pair $\{x_u, z'_u\}$, dist $(f(s_1^1, a_r), x_u) = 2 + \ell_{x_u} < 20(n+1)$. dist $(f(s_1^1, a_r), z'_u) = \min(2 + |P(\pi_1^1, a_r)| + \ell_{x_u})$ $|P(\pi_1^1, c_{r'})| + \ell_{z'_u}, 2 + |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| - 2 + |P(c_{r'}, u_{r'}^{i'})| - \ell_{z'_u}) \text{ if } |P(c_{r'}, u_{r'}^{i'})| - \ell_{z'_u} |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| - 2 + |P(\pi_1^1, a_{r'})| - \ell_{z'_u} |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| - 2 + |P(\pi_1^1, a_{r'})| - \ell_{z'_u} |P(\pi_1^1, a_{r'})| - \ell_{z'_u} |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| - 2 + |P(\pi_1^1, a_{r'})| - \ell_{z'_u} |P(\pi_1^1, a_{r'})| - \ell_{z'_u} |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| + |P(\pi_1^1, a_{r'})| - 2 + |P(\pi_1^1, a_{r'})| - \ell_{z'_u} |P($ $\ell_{z'_u} \ge 2. \operatorname{dist}(\tilde{f(s_1^1, a_r)}, z'_u) = \min (2 + |P(\pi_1^1, a_r)| + |P(\pi_1^1, c_{r'})| + \ell_{z'_u}, 2 + |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + |P(\pi_1^$ $|P(a_{r'}, u_{r'}^{i'})| + |P(c_{r'}, u_{r'}^{i'})| - \ell_{z'_u}) \text{ if } |P(c_{r'}, u_{r'}^{i'})| - \ell_{z'_u} < 2. \text{ It follows that } \operatorname{dist}(f(s_1^1, a_r), z'_u) > 20(n + \ell_{z'_u})$ 1). Thus every pair $\{x_u, z'_u\}$ is resolved by $f(s_1^1, a_r)$. Similarly, every pair $\{x_u, z'_v\}, \{x_v, z'_u\}$ and $\{x_v, z'_v\}$ are resolved by $f(s_1^1, a_r)$. For a pair $\{x_u, y'_u\}$ such that $y'_u \neq b_{r'}$, dist $(f(\pi_1^1, a_r), x_u) = 2 +$ $\ell_{x_u} < 20(n+1). \operatorname{dist}(f(\pi_1^1, a_r), b_{r'}) = 2 + \min_{j \in [m]} (|P(s_1^j, a_r)| + |P(s_1^j, b_{r'})|). \operatorname{dist}(f(\pi_1^1, a_r), y'_u) = 2 + \min_{j \in [m]} (|P(s_1^j, a_r)| + |P(s_1^j, b_{r'})|).$ $\min\left(\operatorname{dist}(f(\pi_1^1, a_r), b_{r'}) + \ell_{z'_{u'}}, |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| + |P(b_{r'}, u_{r'}^{i'})| - \ell_{u'_{u'}}\right) > 20(n + 1)$ 1). Thus every pair $\{x_u, y'_u\}$ such that $y'_u \neq b_{r'}$ is resolved by $f(\pi_1^1, a_r)$. Similarly, every pair $\{x_v, y'_u\}$ such that $y'_u \neq b_{r'}$, every pair $\{x_v, y'_v\}$ and $\{x_u, y'_v\}$ are resolved by $f(\pi_1^1, a_r)$. For a pair $\{y_u, y'_u\}$ such that $y_u \neq b_r$, $\operatorname{dist}(f^1(u_r^i, v_r^i), y_u) = 2 + |P(b_r, u_r^i)| - \ell_{y_u} < 20(n+1)$. $\operatorname{dist}(f^{1}(u_{r}^{i}, v_{r}^{i}), b_{r'}) = \min \left(2 + |P(b_{r}, v_{r}^{i})| + \min_{j \in [m]}(|P(s_{1}^{j}, b_{r})| + |P(s_{1}^{j}, b_{r'})|), 1 + |P(a_{r}, v_{r}^{i})| + |P(s_{1}^{j}, b_{r'})|\right)$ $\min_{j' \in [m]} (|P(s_1^{j'}, a_r)| + |P(s_1^{j'}, b_{r'})|)). \quad \operatorname{dist}(f^1(u_r^i, v_r^i), y_u') = \min (\operatorname{dist}(f^1(u_r^i, v_r^i), b_{r'}) + \ell_{y_u'}, 1 + \ell_{y_u'})$ $|P(a_r, v_r^i)| + |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| + |P(b_{r'}, u_{r'}^{i'})| - \ell_{y'_u}) > 20(n+1).$ Thus every pair $\{y_u, y'_u\}$ such that $y_u \neq b_r$ is resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{y_u, y'_v\}$ such that $y_u \neq b_r$, every pair $\{y_v, y'_v\}$ and $\{y_v, y'_u\}$ are resolved by $f^1(u_r^i, v_r^i)$. For a pair $\{y_u, z'_u\}, \operatorname{dist}(f^1(u^i_r, v^i_r), y_u) < 20(n+1). \operatorname{dist}(f^1(u^i_r, v^i_r), z'_u) = \min(1+|P(a_r, u^i_r)|+|P(\pi^1_1, a_r)|+|P(\pi^1_1, a_r$ $|P(\pi_1^1, c_{r'})| + \ell_{z'_u}, 1 + |P(a_r, u^i_r)| + |P(\pi_1^1, a_r)| + |P(\pi_1^1, a_{r'})| + |P(a_{r'}, u^{i'}_{r'})| + |P(c_{r'}, u^{i'}_{r'})| - \ell_{z'_u} - 2)$ if $|P(c_{r'}, u_{r'}^{i'})| - \ell_{z'_u} \ge 2$. dist $(f^1(u_r^i, v_r^i), z'_u) = \min(1 + |P(a_r, u_r^i)| + |P(\pi_1^1, a_r)| + |P(\pi_1^1, c_{r'})| + |P(\pi_1^1, c_{$ $\ell_{z'_{u}}, 1 + |P(a_{r}, u^{i}_{r})| + |P(\pi^{1}_{1}, a_{r})| + |P(\pi^{1}_{1}, a_{r'})| + |P(a_{r'}, u^{i'}_{r'})| + |P(c_{r'}, u^{i'}_{r'})| - \ell_{z'_{u}}) \text{ if } |P(c_{r'}, u^{i'}_{r'})| - \ell_{z'_{u}} < 0.5$ 2. It follows that dist $(f^1(u_r^i, v_r^i), z'_u) > 30(n+1)$. Thus every pair $\{y_u, z'_u\}$ is resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{y_u, z'_v\}, \{y_v, z'_u\}$ and $\{y_v, z'_v\}$ are resolved by $f^1(u_r^i, v_r^i)$. For a pair $\{z_u, z'_u\}$, dist $(f^1(u^i_r, v^i_r), z_u) = 1 + |P(c_r, u^i_r)| - \ell_{z_u}$ if $z_u \neq u^i_r$ and dist $(f^1(u^i_r, v^i_r), z_u) = 2$ if $z_u = u_r^i$. Thus $dist(f^1(u_r^i, v_r^i), z_u) < 30(n+1) < dist(f^1(u_r^i, v_r^i), z_u')$ and every pair $\{z_u, z_u'\}$ is resolved by $f^1(u_r^i, v_r^i)$. Similarly, every pair $\{z_u, z_v'\}, \{z_v, z_u'\}$ and $\{z_v, z_v'\}$ are resolved by

 $f^1(u_r^i, v_r^i)$. As a result, every vertex pair of $R_r \times R_{r'}$ with $r' \in \{1, 2, 3\}$ and $r' \neq r$ is resolved by S'. This completes the proof for the lemma.

Lemma 57. Every pair $\{x, y\} \in \bigcup_{i \in [n]} U_i \times \bigcup_{i \in [n]} \prod_i$ is resolved by S'.

Proof. First, we show that every pair $\{x, y\} \in U_i^h \times \bigcup_{j \in [m], r \in \{1, 2, 3\}} \Pi^h(i, j, r)$ for $i \in [n], h \in \{1, 2\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P^h(i, j', a_r)$. For a vertex $x \in P(s_i^j, p_i^h)$, let $P(x, p_i^h)$ be the subpath of $P(s_i^j, p_i^h)$ from x to p_i^h and $|P(x, p_i^h)| = \ell_x$. For a vertex $y \in P^h(i, j', a_r)$, let $P(y, \pi_i^h)$ be the subpath of $P^h(i, j', a_r)$ from y to π_i^h and $|P(y, \pi_i^h)| = \ell_y$. Let $j^* \in [m]$ be an integer such that $j^* \neq j$ and $j^* \neq j'$. Then dist $(f^h(i, j^*, a_r), x) = 3 + \ell_x$ and dist $(f^h(i, j^*, a_r), y) = 2 + \ell_y$. Thus every pair $\{x, y\}$ is resolved by $f^h(i, j^*, a_r)$ unless $\ell_y - \ell_x = 1$. For the pair $\{x, y\}$ such that $\ell_y - \ell_x = 1$, there are two cases. Case 1: j = j'. Since dist $(f^h(i, j', a_r), x) = 3 + \ell_x$ if $x \neq s_i^j$, dist $(f^h(i, j', a_r), s_i^j) = 20(n + 1) + 1$ and dist $(f^h(i, j', a_r), y) = \ell_y$ if $y \neq \pi_i^h$, dist $(f^h(i, j', a_r), \pi_i^h) = 2$, every pair $\{x, y\}$ such that $\ell_y - \ell_x = 1$ is resolved by $f^h(i, j', a_r)$. Case 2: $j \neq j'$. Since dist $(f^h(i, j', a_r), x) = 3 + \ell_x$, dist $(f^h(i, j', a_r), \pi_i^h) = 2$, every pair $\{x, y\}$ such that $\ell_y - \ell_x = 1$ is resolved by $f^h(i, j', a_r)$. It follows that every pair $\{x, y\}$ is resolved by $f^h(i, j', a_r)$, $P(s_i^j, p_i^h) \times P^h(i, j', a_r)$. Similarly, we can show that every vertex pair of $P(s_i^j, p_i^h) \times P^h(i, j', b_r)$, $P(s_i^j, p_i^h) \times P^h(i, j', c_r)$ and $P(s_i^j, p_i^h) \times P^h(i, j', p_i^{-3-h})$ are resolved by S'.

Next we show that every pair $\{x, y\} \in U_i^h \times \bigcup_{j \in [m], r \in \{1,2,3\}} \Pi^{3-h}(i, j, r)$ for $i \in [n], h \in \{1,2\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m], h \in \{1,2\}$ and $r \in \{1,2,3\}$. Suppose that $x \in P(s_i^j, p_i^h) \setminus \{s_i^j\}$ and $y \in P^{3-h}(i, j', a_r)$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the first paragraph. There are two cases. Case 1: j = j'. dist $(f^{3-h}(i, j', a_r), y) = \ell_y$ if $y \neq \pi_i^h$, dist $(f^h(i, j', a_r), \pi_i^h) = 2$. dist $(f^{3-h}(i, j', a_r), x) = \min(|P^{3-h}(i, j, a_r)| + 1 + |P(s_i^j, p_i^h)| - \ell_x, 3 + |P(\pi_i^{3-h}, c_r)| + |P(c_r, \pi_i^h)| + \ell_x) = \min(40(n+1) + 1 - \ell_x, 20(n+1) + 3 + \ell_x) \ge 20(n+1) + 1 > \text{dist}(f^{3-h}(i, j', a_r), y)$. Thus in this case, every pair $\{x, y\}$ is resolved by $f^{3-h}(i, j', a_r), x = \min(3 + |P(s_i^j, p_i^{3-h})| + |P(s_i^j, p_i^h)| - \ell_x, 3 + |P(\pi_i^{3-h}, c_r)| + |P(c_r, \pi_i^h)| + \ell_x) = \min(40(n+1) + 3 - \ell_x, 20(n+1) + 3 + \ell_x) \ge 20(n+1) + 3 > \text{dist}(f^{3-h}(i, j', a_r), y)$. Thus in this case, every pair $\{x, y\}$ is resolved by $f^{3-h}(i, j', a_r), x = \min(3 + |P(s_i^j, p_i^{3-h})| + |P(s_i^j, p_i^h)| - \ell_x, 3 + |P(\pi_i^{3-h}, c_r)| + |P(c_r, \pi_i^h)| + \ell_x) = \min(40(n+1) + 3 - \ell_x, 20(n+1) + 3 + \ell_x) \ge 20(n+1) + 3 > \text{dist}(f^{3-h}(i, j', a_r), y)$. Thus in this case, every pair $\{x, y\}$ is resolved by $f^{3-h}(i, j', a_r), x = \min\{x, y\}$ is resolved by $f^{3-h}(i, j', a_r), x = \min\{x, y\}$ is resolved by $f^{3-h}(i, j', a_r)$. It follows that every pair $\{x, y\}$ is resolved by $f^{3-h}(i, j', a_r)$. Similarly, we can show that every vertex pair of $P(s_i^j, p_i^h) \times P^{3-h}(i, j', b_r), P(s_i^j, p_i^h) \times P^{3-h}(i, j', c_r)$ and $P(s_i^j, p_i^h) \times P^{3-h}(i, j', p_i^h)$ are resolved by S'.

Finally we show that every pair $\{x, y\} \in U_i^h \times \bigcup_{j \in [m], r \in \{1, 2, 3\}} \Pi^{h'}(i', j, r)$ for $i, i' \in [n], h, h' \in \{1, 2\}$ such that $i \neq i'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h, h' \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ such that $i \neq i'$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P^{h'}(i', j', a_r)$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x in the first paragraph. Let $j^* \in [m]$ be an integer such that $j^* \neq j'$. Then $\operatorname{dist}(f^{h'}(i', j^*, a_r), y) = 2 + \ell_y$. $\operatorname{dist}(f^{h'}(i', j^*, a_r), s_i^j) = \min_{r' \in \{1, 2, 3\}} (2 + |P(\pi_{i'}^{h'}, c_{r'})| + |P(s_i^j, c_{r'})|)$. $\operatorname{dist}(f^{h'}(i', j^*, a_r), x) = \min(\operatorname{dist}(f^{h'}(i', j^*, a_r), s_i^j) + |P(s_i^j, \pi_i^h)| - \ell_x, 3 + |P(\pi_{i'}^{h'}, c_r)| + |P(\pi_i^h, c_r)| + \ell_x) > 2 + 20(n+1) \ge \operatorname{dist}(f^{h'}(i', j^*, a_r), y)$. Thus every pair $\{x, y\}$ is resolved by $f^{h'}(i', j^*, a_r)$. Similarly, we can show that every vertex pair of $P(s_i^j, p_i^h) \times P^{h'}(i', j', b_r), P(s_i^j, p_i^h) \times P^{h'}(i', j', c_r)$ and $P(s_i^j, p_i^h) \times P^{h'}(i', j', p_{i'}^{3-h'})$ are resolved by $f^{h'}(i', j^*, a_r)$. This completes the proof for the lemma.

Lemma 58. Every pair $\{x, y\} \in \bigcup_{i \in [n]} U_i \times \bigcup_{i \in [n]} L_i$ is resolved by S'.

Proof. First, we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ for $i \in [n], j \in [m]$ and $h \in \{1, 2\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j \in [m]$ and $h \in \{1, 2\}$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$. For a vertex

Next we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times P(q_i^h, \operatorname{mid}(P^{3-h}(i, j', p_i^h)))$ for $i \in [n], j, j' \in [m]$ and $h \in \{1, 2\}$ such that $j \neq j'$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m]$ and $h \in \{1, 2\}$ such that $j \neq j'$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j', p_i^h))))$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. dist $(f^{mid}(i, j', 3-h), y) = 1 + |P(q_i^h, \operatorname{mid}(P^{3-h}(i, j', p_i^h)))| - \ell_y = 30(n+1) - \ell_y$. dist $(f^{mid}(i, j', 3-h), x) = \min(1+|P^{3-h}(i, j', p_i^h)|/2 + |P(s_i^{j'}, p_i^h)| - 1 + \ell_x, 2 + |P^{3-h}(i, j', p_i^h)|/2 + |P(s_i^{j'}, p_i^{3-h})| + |P(s_i^{j'}, p_i^h)| - \ell_x) = \min(30(n+1) + \ell_x, 2 + 50(n+1) - \ell_x)$. Thus every pair $\{x, y\}$ is resolved by $f^{mid}(i, j', 3-h)$ unless $\ell_x = \ell_y = 0$, i.e. except the pair $\{p_i^h, q_i^h\}$. According to Lemma 46, $\{p_i^h, q_i^h\}$ is resolved by S'. Thus every pair $\{x, y\}$ is resolved by S'.

Then we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times P(q_i^{3-h}, \operatorname{mid}(P^h(i, j', p_i^{3-h})))$ for $i \in [n], j, j' \in [m]$ and $h \in \{1, 2\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m]$ and $h \in \{1, 2\}$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(q_i^{3-h}, \operatorname{mid}(P^h(i, j', p_i^{3-h})))$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Let $j^* \in [m]$ be an integer such that $j^* \neq j$. Then dist $(f^{3-h}(i, j^*, a_1), y) = 3 + \ell_y$. dist $(f^{3-h}(i, j^*, a_1), x) = \min(1 + |P^{3-h}(i, j, a_1)| + |P(s_i^j, p_i^h)| - \ell_x, 3 + |P(\pi_i^{3-h}, a_1)| + |P(\pi_i^h, a_1)| + \ell_x) = \min(40(n+1)+1-\ell_x, 20(n+1)+3+\ell_x) \ge 20(n+1)+2$ if $x \neq s_i^j$ and dist $(f^{3-h}(i, j^*, a_1), s_i^j) = 3+20(n+1)$. Thus any pair $\{x, y\}$ such that $\ell_y < 20(n+1)-1$ is resolved by $f^{3-h}(i, j^*, a_1)$. For the pair $\{x, y\}$ such that $20(n+1)-1 \le \ell_y \le 30(n+1)-1$, dist $(f^{mid}(i, j', h), y) = 1 + |P(q_i^{3-h}, \min(P^h(i, j', p_i^{3-h})))| - \ell_y = 30(n+1) - \ell_y \le 10(n+1) + 1$. If j' = j, then dist $(f^{mid}(i, j', h), x) = \min(2 + |P^h(i, j, p_i^{3-h})|/2 + \ell_x, 2 + |P^h(i, j, p_i^{3-h})|/2 + |P(s_i^j, p_i^h)| - \ell_x) = \min(2 + 10(n+1) + \ell_x \ge 10(n+1) + 2$. As a result, every pair $\{x, y\}$ is resolved by $f^{3-h}(i, j^*, a_1)$ or $f^{mid}(i, j', h)$.

Finally we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))$ for $i, i' \in [n], j, j' \in [m]$ and $h, h' \in \{1, 2\}$ such that $i \neq i'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m]$ and $h, h' \in \{1, 2\}$ such that $i \neq i'$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Then dist $(f^h(i, j, a_1), x) = 3 + \ell_x$ if $x \neq s_i^j$ and dist $(f^h(i, j, a_1), s_i^j) = 2 + 20(n + 1)$. dist $(f^h(i, j, a_1), y) = \min(3 + |P(\pi_i^h, a_1)| + |P(\pi_{i'}^{h'}, a_1)| + \ell_y, 2 + |P(\pi_i^h, a_1)| + |P(\pi_{i'}^{3-h'}, a_1)| + |P^{3-h'}(i', j', p_{i'}^{h'})|/2 + |P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))| - \ell_y) = \min(3 + 20(n + 1) + \ell_y, 1 + 60(n + 1) - \ell_y) \geq 3 + 20(n + 1) > \operatorname{dist}(f^h(i, j, a_1), x)$. Thus every pair $\{x, y\}$ is resolved by $f^h(i, j, a_1)$. This completes the proof for the lemma.

Lemma 59. Every pair $\{x, y\} \in \bigcup_{i \in [n]} U_i \times \bigcup_{i \in [n]} S_i$ is resolved by S'.

Proof. We show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times (P(\pi_{i'}^{h'}, a_r) \cup P(\pi_{i'}^{h'}, c_r))$ for $i, i' \in [n], j \in [m], h' \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j \in [m], h, h' \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(\pi_{i'}^{h'}, a_r)$. For a vertex $x \in P(s_i^j, p_i^h)$, let $P(x, p_i^h)$ be the subpath of $P(s_i^j, p_i^h)$ from x to p_i^h and $|P(x, p_i^h)| = \ell_x$. For a vertex $y \in P(\pi_{i'}^{h'}, a_r)$, let $P(y, \pi_{i'}^{h'})$ be the subpath of $P(\pi_{i'}^{h'}, a_r)$ from y to $\pi_{i'}^{h'}$ and $|P(y, \pi_{i'}^{h'})| = \ell_y$. Then dist $(f(s_i^j, a_r), y) = 2 + |P(\pi_{i'}^{h'}, a_r)| - \ell_y$. Suppose that $|P(s_i^j, a_r)| = 20(n + 1) + 10p$ for some $p \in [n]$. dist $(f(s_i^j, a_r), x) = \min(3 + |P(\pi_i^h, a_r)| + \ell_x, |P(s_i^j, a_r)| + |P(s_i^j, p_i^h)| - \ell_x) = \min(3 + 10(n + 1) + \ell_x, 40(n + 1) + 10p - \ell_x) \ge 3 + 10(n + 1) > \text{dist}(f(s_i^j, a_r), y)$. Thus every pair $\{x, y\}$ is resolved by $f(s_i^j, a_r)$. Similarly, we can show that every vertex pair of $P(s_i^j, p_i^h) \times P(\pi_{i'}^{h'}, c_r)$ for $i, i' \in [n], j \in [m], h' \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by $f(s_i^j, c_r)$. This completes the proof for the lemma. ⊔

Lemma 60. Every pair $\{x, y\} \in \bigcup_{i \in [n]} U_i \times \bigcup_{i \in [n]} H_i$ is resolved by S'.

 $\begin{aligned} &Proof. \text{ First we show that every pair } \{x,y\} \in P(s_i^j, p_i^h) \times (P(s_i^j, a_r) \cup P(s_i^j, c_r)) \text{ for } i \in [n], j \in [m], h \in \{1,2\} \text{ and } r \in \{1,2,3\} \text{ is resolved by } S'. We fix arbitrary integers } i \in [n], j \in [m], h \in \{1,2\} \text{ and } r \in \{1,2,3\}. \text{ Suppose that } x \in P(s_i^j, p_i^h) \text{ and } y \in P(s_i^j, a_r). \text{ For a vertex } x \in P(s_i^j, p_i^h), \text{ let } P(s_i^j, x) \text{ be the subpath of } P(s_i^j, p_i^h) \text{ from } s_i^j \text{ to } x \text{ and } |P(s_i^j, x)| = \ell_x. \text{ For a vertex } y \in P(s_i^j, a_r), \text{ let } P(s_i^j, y) \text{ be the subpath of } P(s_i^j, a_r) \text{ from } s_i^j \text{ to } x \text{ and } |P(s_i^j, y)| = \ell_y. \text{ Let } |P(s_i^j, a_r)| = 20(n+1) + 10\lambda \text{ for some } \lambda \in [n]. \text{ Then } \text{dist}(f(s_i^j, a_r), x) = \min (3 + |P(\pi_i^h, a_r)| + |P(s_i^j, a_r)| - \ell_y = 20(n+1) + 10\lambda - \ell_y. \text{ dist}(f(\pi_i^h, a_r), x) = \min (1 + |P(\pi_i^h, a_r)| + |P(s_i^j, p_i^h)| - \ell_x, 2 + |P(s_i^j, a_r)| + \ell_x) = \min (1 + 30(n+1) - \ell_x, 2 + 20(n+1) + 10\lambda + \ell_x). \text{ dist}(f(\pi_i^h, a_r), y) = 2 + |P(s_i^j, a_r)| - \ell_y = 2 + 20(n+1) + 10\lambda - \ell_y. \text{ For the pair } \{x,y\} \text{ which is not resolved by } f(s_i^j, a_r), \text{ is atisfies that } \text{dist}(f(s_i^j, a_r), x) = \text{dist}(f(s_i^j, a_r), y) = 3 + |P(\pi_i^h, a_r)| + |P(s_i^j, p_i^h)| - \ell_x. \text{ Thus } \text{dist}(f(\pi_i^h, a_r), x) < \text{ dist}(f(s_i^j, a_r), x) = \text{ dist}(f(\pi_i^h, a_r), y) < \text{ dist}(f(\pi_i^h, a_r), y). \text{ It follows that every pair } \{x,y\} \text{ is resolved by } f(s_i^j, a_r) \text{ or } f(\pi_i^h, a_r). \text{ Similarly, we can show that every vertex pair of } P(s_i^j, p_i^h) \times P(s_i^j, c_r) \text{ is resolved by } f(s_i^j, c_r) \text{ or } f(\pi_i^h, c_r). \end{aligned}$

Next we show that every vertex pair of $P(s_i^j, p_i^h) \times P(s_i^j, b_r)$ for $i \in [n], j \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, p_i^h) \setminus \{s_i^j\}$ and $y \in P(s_i^j, b_r) \setminus \{s_i^j\}$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Then dist $(f^{mid}(i, j, 3 - h), x) =$ $|P^{3-h}(i, j, p_i^h)|/2 + \ell_x = 10(n+1) + \ell_x$ and dist $(f^{mid}(i, j, 3 - h), y) = 2 + |P^{3-h}(i, j, p_i^h)|/2 + \ell_y = 2 + 10(n+1) + \ell_y$. For the vertex pair $\{x, y\}$ which is not resolved by $f^{mid}(i, j, 3 - h)$, i.e. $\ell_x = 2 + \ell_y$, dist $(f^{mid}(i, j, h), x) = 2 + |P^h(i, j, p_i^{3-h})|/2 + \ell_x = 2 + 10(n+1) + \ell_x$. Thus every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, 3 - h)$ or $f^{mid}(i, j, h)$.

Then we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times (P(s_{i'}^{j'}, a_r) \cup P(s_{i'}^{j'}, c_r))$ for $i, i' \in [n], j, j' \in [m], h \in \{1, 2, 3\}$ such that $i \neq i'$ or $j \neq j'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ such that $i \neq i'$ or $j \neq j'$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(s_{i'}^{j'}, a_r)$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Let $|P(s_i^j, a_r)| = 20(n+1) + 10\lambda$ and $|P(s_{i'}^{j'}, a_r)| = 20(n+1) + 10\lambda'$ for some $\lambda, \lambda' \in [n]$. Then dist $(f(s_{i'}^{j'}, a_r), x) = dist(f(\pi_{i'}^{3-h}, a_r), x) = min (3 + |P(\pi_i^h, a_r)| + |P(s_i^j, p_i^h)| - \ell_x, 2 + |P(s_i^j, a_r)| + \ell_x) = min (3 + 30(n+1) - \ell_x, 2 + 20(n+1) + 10\lambda + \ell_x)$. dist $(f(s_{i'}^{j'}, a_r), y) = dist(f(\pi_{i'}^{3-h}, a_r), y) - 2 = |P(s_{i'}^{j'}, a_r)| - \ell_y$ if $y \neq a_r$ and dist $(f(s_{i'}^{j'}, a_r), a_r) = dist(f(\pi_{i'}^{3-h}, a_r), a_r) = 2$. It follows that every pair $\{x, y\}$ is resolved by $f(s_{i'}^{j'}, a_r)$ or $f(\pi_{i'}^{3-h}, a_r)$.

Similarly, we can show that every vertex pair of $P(s_i^j, p_i^h) \times P(s_{i'}^{j'}, c_r)$ is resolved by $f(s_{i'}^{j'}, c_r)$ or $f(\pi_{i'}^{3-h}, c_r)$.

Finally we show that every vertex pair of $P(s_i^j, p_i^h) \times P(s_{i'}^{j'}, b_r)$ for $i, i' \in [n], j, j' \in [m], h \in$ $\{1,2\}$ and $r \in \{1,2,3\}$ such that $i \neq i'$ or $j \neq j'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ such that $i \neq i'$ or $j \neq j'$. Suppose that $x \in P(s_i^j, p_i^h)$ and $y \in P(s_{i'}^{j'}, b_r)$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Let $|P(s_i^j, b_r)| = 20(n+1) + 5\lambda + 1$ and $|P(s_{i'}^{j'}, b_r)| = 20(n+1) + 5\lambda' + 1$ and for some $\lambda, \lambda' \in [n]$. There are two cases. Case 1: i = i' and $j \neq j'$. dist $(f^{mid}(i, j, 3 - h), x) =$ $|P^{3-h}(i,j,p_i^h)|/2 + \ell_x = 10(n+1) + \ell_x \text{ if } x \neq s_i^j \text{ and } \operatorname{dist}(f^{mid}(i,j,3-h),s_i^j) = 2 + 10(n+1).$ $\operatorname{dist}(f^{mid}(i,j,3-h),y) = \min(2+|P^{3-h}(i,j,p_i^h)|/2+|P(s_{i'}^{j'},p_i^{3-h})| + \ell_y, 2+|P^{3-h}(i,j,p_i^h)|/2 + |P^{3-h}(i,j,p_i^h)|/2 + |$ $\begin{aligned} |P(s_i^j, b_r)| + |P(s_{i'}^{j'}, b_r)| - \ell_y) &= \min (2 + 30(n+1) + \ell_y, 4 + 50(n+1) + 5\lambda + 5\lambda' - \ell_y) \geq 2 + 30(n+1) > \\ \operatorname{dist}(f^{mid}(i, j, 3 - h), x). \text{ Thus in this case, every pair } \{x, y\} \text{ is resolved by } f^{mid}(i, j, 3 - h). \text{ Case 2:} \end{aligned}$ $i \neq i'. \operatorname{dist}(f^{mid}(i, j, 3-h), s_{i'}^{j'}) = 1 + |P^{3-h}(i, j, p_i^h)|/2 + \min_{r \in \{1, 2, 3\}}(|P(\pi_i^{3-h}, c_r)| + |P(s_{i'}^{j'}, c_r)|).$ $\operatorname{dist}(f^{mid}(i, j, 3-h), y) = \min\left(\operatorname{dist}(f^{mid}(i, j, 3-h), s_{i'}^{j'}) + \ell_y, 2 + |P^{3-h}(i, j, p_i^h)|/2 + |P(s_i^j, b_r)| + \frac{1}{2} + \frac$ $|P(s_{i'}^{j'}, b_r)| - \ell_y) = \min (1 + 40(n+1) - 10\lambda' + \ell_y, 4 + 50(n+1) + 5\lambda + 5\lambda' - \ell_y) > 30(n+1) + 5 > 100(n+1) + 5 > 100$ dist $(f^{mid}(i, j, 3 - h), x)$. Thus in this case, every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, 3 - h)$. It follows that every pair $\{x, y\}$ is resolved by $f^{mid}(i, j, 3 - h)$. This completes the proof for the lemma.

Lemma 61. Every pair $\{x, y\} \in \bigcup_{i \in [n]} U_i \times \bigcup_{r \in \{1,2,3\}} R_r$ is resolved by S'.

 $\begin{array}{l} Proof. \ \text{First we show that every pair } \{x,y\} \in P(s_i^j,p_i^h) \times (P(u_r^{i'},a_r) \cup P(v_r^{i'},a_r)) \ \text{for } i,i' \in [n], j \in [m], h \in \{1,2\} \ \text{and } r \in \{1,2,3\} \ \text{is resolved by } S'. \ \text{We fix arbitrary integers } i,i' \in [n], j \in [m], h \in \{1,2\} \ \text{and } r \in \{1,2,3\}. \ \text{Suppose that } x \in P(s_i^j,p_i^h), y \in P(u_r^{i'},a_r). \ \text{For a vertex } x \in P(s_i^j,p_i^h), \ \text{het } P(s_i^j,x) \ \text{be the subpath of } P(s_i^j,p_i^h) \ \text{from } s_i^j \ \text{to } x \ \text{and } |P(s_i^j,x)| = \ell_x. \ \text{For a vertex } y \in P(a_r,u_r^{i'}), \ \text{let } P(u_r^{i'},y) \ \text{be the subpath of } P(a_r,u_r^{i'}) \ \text{from } u_r^{i'} \ \text{to } y \ \text{and } |P(u_r^{i'},y)| = \ell_y. \ \text{Let } |P(s_i^j,a_r)| = 20(n+1) + 10\lambda \ \text{for some } \lambda \in [n] \ \text{and } |P(u_r^{i'},a_r)| = 20(n+1) - 10i'. \ \text{There are two cases. Case 1:} \ \lambda \leq i'. \ \text{dist}(f^1(u_r^{i'},v_r^{i'}),x) = \min(1+|P(a_r,u_r^{i'})|+|P(s_i^j,a_r)|+\ell_x,1+|P(a_r,u_r^{i'})|+|P(\pi_i^h,a_r)|+|P(s_i^j,a_r)|+\ell_y \ \text{if } y \neq u_r^{i'} \ \text{and } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),x) \geq 30(n+1) - 10i' + 1 \\ > \text{dist}(f^1(u_r^{i'},v_r^{i'}),y). \ \text{Case } 2: \ \lambda > i'. \ \text{dist}(f^1(u_r^{i'},v_r^{i'}),x) = \min(1+|P(s_i^j,\pi_i^h)|-\ell_x) = \min(1+|P(s_r,u_r^{i'})|+|P(s_i^j,c_r)|+\ell_x,1+|P(a_r,u_r^{i'})|+|P(s_i^j,c_r)|+\ell_x,1+|P(a_r,u_r^{i'})|+|P(s_i^j,c_r)|+\ell_x,1+|P(a_r,u_r^{i'})|+|P(s_i^j,c_r)|+|P(s_i^j,m_r^h)|-\ell_x) = \min(1+|P(a_r,u_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i'}. \ \text{Thus } \text{dist}(f^1(u_r^{i'},v_r^{i'}),y) = 2 \ \text{if } y = u_r^{i$

Next we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times (P(u_r^{i'}, b_r) \cup P(v_r^{i'}, b_r))$ for $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, p_i^h), y \in P(u_r^{i'}, b_r)$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Let $|P(s_i^j, a_r)| = 20(n + 1) + 10\lambda$ for some $\lambda \in [n]$ and $|P(u_r^{i'}, a_r)| = 20(n + 1) - 10i'$. There are two cases. Case 1: $\lambda \leq i'$. dist $(f^1(u_r^{i'}, v_r^{i'}), x) = \min(1 + |P(a_r, u_r^{i'})| + |P(s_i^j, a_r)| + \ell_x, 1 + |P(a_r, u_r^{i'})| + |P(x_i^h, a_r)| + |P(s_i^j, a_r$

 $\begin{aligned} \operatorname{dist}(f^{2}(u_{r}^{i'}, v_{r}^{i'}), x) &= \operatorname{dist}(f^{1}(u_{r}^{i'}, v_{r}^{i'}), x) - 1 = \min (|P(c_{r}, u_{r}^{i'})| + |P(s_{i}^{j}, c_{r})| + \ell_{x}, |P(a_{r}, u_{r}^{i'})| + |P(\pi_{i}^{h}, a_{r})| + |P(s_{i}^{j}, \pi_{i}^{h})| - \ell_{x}). \quad \operatorname{dist}(f^{1}(u_{r}^{i'}, v_{r}^{i'}), y) = \operatorname{dist}(f^{2}(u_{r}^{i'}, v_{r}^{i'}), y) = 2 + \ell_{y}. \quad \operatorname{Similar to} \\ \operatorname{Case 1, in this case, for a pair \{x, y\} \text{ which is not resolved by } f^{1}(u_{r}^{i'}, v_{r}^{i'}), \operatorname{dist}(f^{2}(u_{r}^{i'}, v_{r}^{i'}), x) < \\ \operatorname{dist}(f^{2}(u_{r}^{i'}, v_{r}^{i'}), y). \quad \operatorname{Thus every pair} \{x, y\} \text{ is resolved by } f^{1}(u_{r}^{i'}, v_{r}^{i'}) \text{ or } f^{2}(u_{r}^{i'}, v_{r}^{i'}). \quad \operatorname{Similarly, we} \\ \operatorname{can show that every vertex pair of <math>P(s_{i}^{j}, p_{i}^{h}) \times P(v_{r}^{i'}, b_{r}) \text{ is resolved by } f^{1}(u_{r}^{i'}, v_{r}^{i'}) \text{ or } f^{2}(u_{r}^{i'}, v_{r}^{i'}). \end{aligned}$

Finally we show that every pair $\{x, y\} \in P(s_i^j, p_i^h) \times (P(u_r^{i'}, c_r) \cup P(v_i^{i'}, c_r))$ for $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, p_i^h), y \in P(u_r^{i'}, c_r)$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Let $|P(s_i^j, c_r)| = 20(n + 1) - 10\lambda$ for some $\lambda \in [n]$ and $|P(u_r^{i'}, c_r)| = 20(n + 1) + 10i'$. Then $dist(f(s_i^j, c_r), x) = min (3 + |P(\pi_i^h, c_r)| + |P(s_i^j, p_i^h)| - \ell_x, |P(s_i^j, c_r)| + \ell_x) = min (3 + 30(n + 1) - \ell_x, 20(n + 1) - 10\lambda + \ell_x)$. $dist(f(\pi_i^h, c_r), x) = min (1 + |P(\pi_i^h, c_r)| + |P(s_i^j, p_i^h)| - \ell_x, 2 + |P(s_i^j, c_r)| + \ell_x) = min (1 + 30(n + 1) - \ell_x, 2 + 20(n + 1) - 10\lambda + \ell_x)$. $dist(f(\pi_i^{3-h}, c_r), x) = min (3 + 30(n + 1) - \ell_x, 2 + 20(n + 1) - 10\lambda + \ell_x)$. $dist(f(\pi_i^{3-h}, c_r), x) = min (3 + 30(n + 1) - \ell_x, 2 + 20(n + 1) - 10\lambda + \ell_x)$. $dist(f(\pi_i^{3-h}, c_r), y) = 2 + |P(c_r, u_r^{i'})| - \ell_y = 2 + 20(n + 1) + 10i' - \ell_y$. For a pair $\{x, y\}$ which is not resolved by $f(s_i^j, c_r)$, either $f(\pi_i^h, c_r)$ or $f(\pi_i^{3-h}, c_r)$ resolves it. Thus every pair $\{x, y\}$ is resolved by $f(s_i^j, c_r), f(\pi_i^h, c_r)$ or $f(\pi_i^{3-h}, c_r)$. Similarly, we can show that every vertex pair of $P(s_i^j, p_i^h) \times P(v_r^{i'}, c_r)$ is resolved by $f(s_i^j, c_r), f(\pi_i^h, c_r)$ or $f(\pi_i^{3-h}, c_r)$. This completes the proof for the lemma.

Lemma 62. Every pair $\{x, y\} \in \bigcup_{i \in [n]} \prod_i \times \bigcup_{i \in [n]} H_i$ is resolved by S'.

Proof. We show that every pair $\{x, y\} \in (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h})) \times (P^h(i, j, a_r) \cup P^h(i, j, a_r) \cup P^h(i, j, a_r))$ $(P(s_{i'}^{j'}, a_{r'}) \cup P(s_{i'}^{j'}, b_{r'}) \cup P(s_{i'}^{j'}, c_{r'}))$ for $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$ is Suppose that $x_1 \in P^h(i, j, a_r), x_2 \in P^h(i, j, b_r), x_3 \in P^h(i, j, c_r)$ and $x_4 \in P^h(i, j, p_i^{3-h})$. Suppose that $y_1 \in P(s_{i'}^{j'}, a_{r'}), y_2 \in P(s_{i'}^{j'}, b_{r'})$ and $y_3 \in P(s_{i'}^{j'}, c_{r'})$. For a vertex x_{μ} for $\mu \in \{1, 2, 3, 4\}, \text{ let } \ell_{x_{\mu}} = \text{dist}(\pi_i^h, x_{\mu}).$ For a vertex y_{ν} for $\nu \in \{1, 2, 3\}, \text{ let } \ell_{y_{\nu}} = \text{dist}(s_i^j, y_{\nu}).$ Let $|P(s_i^j, a_{r'})| = 20(n+1) + 10\lambda$ and $|P(s_{i'}^{j'}, a_{r'})| = 20(n+1) + 10\lambda'$ for some $\lambda, \lambda' \in [n]$. There are three cases. Case 1: $s_i^j = s_{i'}^{j'}$ and r' = r. Then $\operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), x_1) = \min(2 + r)$ $|P(\pi_i^h, a_r)| + \ell_{x_1}, |P(s_i^j, a_r)| + |P^h(i, j, a_r)| - 1 - \ell_{x_1}) = \min(2 + 10(n+1) + \ell_{x_1}, 40(n+1) + \ell_{x_1}) + \ell_{x_1} +$ $10\lambda - 1 - \ell_{x_1}$). dist $(f(s_{i'}^{j'}, a_{r'}), y_1) = |P(s_i^j, a_r)| - \ell_{y_1} = 20(n+1) + 10\lambda - \ell_{y_1}$ if $y_1 \neq a_r$ and $\operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), a_r) = 2. \quad \operatorname{dist}(f(\pi_i^{3-h}, a_{r'}), x_1) = \min(2 + |P(\pi_i^h, a_r)| + \ell_{x_1}, 1 + |P(s_i^j, a_r)| + \ell_{x_1}, 1 + \ell_$ $|P^{h}(i, j, a_{r})| - \ell_{x_{1}}) = \min (2 + 10(n+1) + \ell_{x_{1}}, 40(n+1) + 10\lambda + 1 - \ell_{x_{1}}). \operatorname{dist}(f(\pi_{i}^{3-h}, a_{r'}), y_{1}) = 0$ $2 + |P(s_i^j, a_r)| - \ell_{y_1} = 20(n+1) + 10\lambda + 2 - \ell_{y_1}$. Let $\gamma \in P^h(i, j, a_r)$ be the vertex such that $\operatorname{dist}(\gamma, \pi_i^h) = 20(n+1) - 1$. Obviously the pair $\{s_i^j, \gamma\}$ is resolved by $f^h(i, j, a_r)$. For the pair $\{x_1, y_1\}$ which is not resolved by $f(s_{i'}^{j'}, a_{r'})$ and $y_1 \neq s_i^j$, it satisfies that $\operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), x_1) =$ $dist(f(s_{i'}^{j'}, a_{r'}), y_1) = dist(f(\pi_i^{3-h}, a_{r'}), x_1) < dist(f(\pi_i^{3-h}, a_{r'}), y_1)$. Thus in this case, every pair $\{x_1, y_1\}$ is resolved by $f(s_{i'}^{j'}, a_{r'}), f(\pi_i^{3-h}, a_{r'})$ or $f^h(i, j, a_r)$. Case 2: $s_i^j \neq s_{i'}^{j'}$ and r' = r. Then dist $(f(s_{i'}^{j'}, a_{r'}), x_1) = \min (2 + |P(\pi_i^h, a_{r'})| + \ell_{x_1}, 1 + |P(s_i^j, a_{r'})| + |P^h(i, j, a_{r'})| - \ell_{x_1}) =$ $\min (2 + 10(n+1) + \ell_{x_1}, 40(n+1) + 10\lambda + 1 - \ell_{x_1}). \quad \operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), y_1) = |P(s_{i'}^{j'}, a_{r'})| - \ell_{y_1} = |P(s_{i'}^{j'}, a_{r'})$ $20(n+1) + 10\lambda' - \ell_{y_1}$ if $y_1 \neq a_{r'}$ and $dist(f(s_{i'}^{j'}, a_{r'}), a_{r'}) = 2$. $dist(f(\pi_i^{3-h}, a_{r'}), x_1) = \min(2 + 1)\lambda' - \ell_{y_1}$ $|P(\pi_i^h, a_{r'})| + \ell_{x_1}, |P(s_i^j, a_{r'})| + |P^h(i, j, a_{r'})| + 1 - \ell_{x_1}) = \min(2 + 10(n+1) + \ell_{x_1}, 40(n+1) + \ell_{x_1}) + 1 - \ell_{x_1} + \ell_{x_1} + 1 - \ell_{x_1} + \ell_{x_1} + 1 - \ell_{x_1} +$ $10\lambda + 1 - \ell_{x_1}$). dist $(f(\pi_i^{3-h}, a_{r'}), y_1) = 2 + |P(s_{i'}^{j'}, a_{r'})| - \ell_{y_1} = 20(n+1) + 10\lambda' + 2 - \ell_{y_1}$. For the pair $\{x_1, y_1\}$ which is not resolved by $f(s_{i'}^{j'}, a_{r'})$, it satisfies that $\operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), x_1) =$ $dist(f(s_{i'}^{j'}, a_{r'}), y_1) = dist(f(\pi_i^{3-h}, a_{r'}), x_1) < dist(f(\pi_i^{3-h}, a_{r'}), y_1)$. Thus in this case, every

pair $\{x_1, y_1\}$ is resolved by $f(s_{i'}^{j'}, a_{r'})$ or $f(\pi_i^{3-h}, a_{r'})$. Case 3: $s_i^j \neq s_{i'}^{j'}$ and $r' \neq r$. Then $\operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), x_1) = \min \left(2 + |P(\pi_i^h, a_{r'})| + \ell_{x_1}, 2 + |P(s_i^j, a_{r'})| + 1 + |P^h(i, j, a_{r'})| - \ell_{x_1}\right) = 0$ $\min (2 + 10(n+1) + \ell_{x_1}, 40(n+1) + 10\lambda + 3 - \ell_{x_1}). \operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), y_1) = |P(s_{i'}^{j'}, a_{r'})| - \ell_{y_1} = |P(s_{i'}^{j'}, a_{r'})|$ $20(n+1) + 10\lambda' - \ell_{y_1}$ if $y_1 \neq a_{r'}$ and $dist(f(s_{i'}^{j'}, a_{r'}), a_{r'}) = 2$. $dist(f(\pi_i^{3-h}, a_{r'}), x_1) = \min(2 + 10\lambda' - \ell_{y_1})$ $|P(\pi_i^h, a_{r'})| + \ell_{x_1}, 2 + |P(s_i^j, a_{r'})| + 1 + |P^h(i, j, a_{r'})| - \ell_{x_1}) = \min(2 + 10(n+1) + \ell_{x_1}, 40(n+1) + \ell_{x_1}) + 1 + |P^h(i, j, a_{r'})| + 1 + |P^h$ 1) + 10 λ + 3 - ℓ_{x_1}). dist $(f(\pi_i^{3-h}, a_{r'}), y_1) = 2 + |P(s_{i'}^{j'}, a_{r'})| - \ell_{y_1} = 20(n+1) + 10\lambda' + 2 - \ell_{y_1}$. For the pair $\{x_1, y_1\}$ which is not resolved by $f(s_{i'}^{j'}, a_{r'})$, it satisfies that $\operatorname{dist}(f(s_{i'}^{j'}, a_{r'}), x_1) =$ $dist(f(s_{i'}^{j'}, a_{r'}), y_1) = dist(f(\pi_i^{3-h}, a_{r'}), x_1) < dist(f(\pi_i^{3-h}, a_{r'}), y_1)$. Thus in this case, every pair $\{x_1, y_1\}$ is resolved by $f(s_{i'}^{j'}, a_{r'})$ or $f(\pi_i^{3-h}, a_{r'})$. It follows that every pair $\{x_1, y_1\}$ is resolved by $f^h(i, j, a_r)$, $f(s_{i'}^{j'}, a_{r'})$ or $f(\pi_i^{3-h}, a_{r'})$. In a similar way, we can show that every vertex pair $\{x_2, y_1\}$, $\{x_3, y_1\}$ and $\{x_4, y_1\}$ are resolved by S'. Also in a similar way, we can show that every vertex pair $\{x_1, y_3\}, \{x_2, y_3\}, \{x_3, y_3\}$ and $\{x_4, y_3\}$ are resolved by $f(s_{i'}^{j'}, c_{r'})$, $f(\pi_i^{3-h}, c_{r'})$ or $f^h(i, j, c_r)$. For a pair $\{x_1, y_2\}$, $dist(f^h(i, j, a_r), x_1) = \ell_{x_1}$ if $x_1 \neq \pi_i^h$ and $\operatorname{dist}(f^{h}(i, j, a_{r}), \pi^{h}_{i}) = 2. \quad \operatorname{dist}(f^{h}(i, j, a_{r}), b_{r'}) = 2 + |P(\pi^{h}_{i}, a_{r'})| + |P(a_{r'}, v^{n}_{r'})| + |P(b_{r'}, v^{n}_{r'})| + |P(b_{r$ $|P(s_{i'}^{j'}, b_{r'})| - \ell_{y_2}$. There are three cases. Case 1: i = i' and j = j'. dist $(f^h(i, j, a_r), y_2) =$ $\min\left(|P^{h}(i,j,a_{r})| + 1 + \ell_{y_{2}}, \operatorname{dist}(f^{h}(i,j,a_{r}),b_{r'}) + |P(s_{i'}^{j'},b_{r'})| - \ell_{y_{2}}\right) = \min\left(20(n+1) + 1 + 1 + \ell_{y_{2}}\right)$ $\ell_{y_2}, 55n+5\lambda+71-\ell_{y_2}) > \operatorname{dist}(f^h(i,j,a_r),x_1)$. Thus in this case, every pair $\{x_1,y_2\}$ is resolved by $f^{h}(i, j, a_{r})$. Case 2: i' = i and $j \neq j'$. Then $dist(f^{h}(i, j, a_{r}), y_{2}) = min (2 + |P^{h}(i, j, b_{r'})| + 1)$ $\ell_{y_2} - 1, \operatorname{dist}(f^h(i, j, a_r), b_{r'}) + |P(s_{i'}^{j'}, b_{r'})| - \ell_{y_2}) = \min(20(n+1) + 1 + \ell_{y_2}, 55n + 5\lambda + 71 - \ell_{y_2})$ if $y_2 \neq s_{i'}^{j'}$ and $\operatorname{dist}(f^h(i, j, a_r), s_{i'}^{j'}) = 3 + 20(n+1)$. Thus $\operatorname{dist}(f^h(i, j, a_r), y_2) \geq 20(n+1) + 2 > \operatorname{dist}(f^h(i, j, a_r), x_1)$. In this case, every pair $\{x_1, y_2\}$ is resolved by $f^h(i, j, a_r)$. Case 3: $i' \neq i$. dist $(f^h(i, j, a_r), s_{i'}^{j'}) = \min_{d \in \{1, 2, 3\}} (2 + |P(\pi_i^h, c_d)| + |P(s_{i'}^{j'}, c_d)|)$. dist $(f^h(i, j, a_r), y_2) = (i' + i') (i'$ $\min (\operatorname{dist}(f^{h}(i, j, a_{r}), s_{i'}^{j'}) + \ell_{y_{2}}, \operatorname{dist}(f^{h}(i, j, a_{r}), b_{r'}) + |P(s_{i'}^{j'}, b_{r'})| - \ell_{y_{2}}) > 20(n+1) \ge \operatorname{dist}(f^{h}(i, j, a_{r}), x_{1}).$ Thus in this case, every pair $\{x_1, y_2\}$ is resolved by $f^h(i, j, a_r)$. In a similar way, we can show that every vertex pair $\{x_2, y_2\}$, $\{x_3, y_2\}$ and $\{x_4, y_2\}$ are resolved by S'. This completes the proof for the lemma.

Lemma 63. Every pair $\{x, y\} \in \bigcup_{i \in [n]} \prod_i \times \bigcup_{i \in [n]} L_i$ is resolved by S'.

Proof. First we show every pair $\{x, y\} \in (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h})) \times P(q_i^h, \operatorname{mid}(P^{3-h}(i, j', p_i^h)))$ for $i \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x_1 \in P^h(i, j, a_r), x_2 \in P^h(i, j, b_r), x_3 \in P^h(i, j, c_r)$ and $x_4 \in P^h(i, j, p_i^{3-h})$. Suppose that $y \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j', p_i^h)))$. For a vertex x_μ for $\mu \in \{1, 2, 3, 4\}$, let $\ell_{x_\mu} = \operatorname{dist}(\pi_i^h, x_\mu)$. For a vertex y, let $\ell_y = \operatorname{dist}(q_i^h, y)$. Then $\operatorname{dist}(f^h(i, j, a_r), x_1) = \ell_{x_1}$ if $x_1 \neq \pi_i^h$ and $\operatorname{dist}(f^h(i, j, a_r), \pi_i^h) = 2$. $\operatorname{dist}(f^h(i, j, a_r), y) = \operatorname{dist}(f^h(i, j, b_r), y) = \operatorname{dist}(f^h(i, j, b_r), y) = \operatorname{dist}(f^h(i, j, b_r), y) = \operatorname{dist}(f^h(i, j, a_r), \pi_i) = 3 + \ell_y$. For the pair $\{x_1, y\}$ that is not resolved by $f^h(i, j, a_r), \operatorname{dist}(f^h(i, j, b_r), x_1)$. Thus every pair $\{x_1, y\}$ is resolved by $f^h(i, j, b_r), x_1) - 2 < \operatorname{dist}(f^h(i, j, b_r), x_1)$. Thus every pair $\{x_1, y\}$ is $resolved by f^h(i, j, b_r)$. In a similar way, we can show that every vertex pair $\{x_2, y\}, \{x_3, y\}$ and $\{x_4, y\}$ are resolved by S'.

Next we show that every pair $\{x, y\} \in P(q_i^{3-h}, \operatorname{mid}(P^h(i, j', p_i^{3-h}))) \times (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h}))$ for $i \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x_1 \in P^h(i, j, a_r), x_2 \in P^h(i, j, b_r), x_3 \in P^h(i, j, c_r) x_4 \in P^h(i, j, p_i^{3-h})$ and $y \in P(q_i^{3-h}, \operatorname{mid}(P^h(i, j', p_i^{3-h})))$. For a vertex x_μ for $\mu \in \{1, 2, 3, 4\}$, let $\ell_{x_\mu} = \operatorname{dist}(\pi_i^h, x_\mu)$. For a vertex y, let $\ell_y = \operatorname{dist}(q_i^{3-h}, y)$. Then $\operatorname{dist}(f^h(i, j, a_r), x_1) = \ell_{x_1}$ if $x_1 \neq \pi_i^h$ and $\operatorname{dist}(f^h(i, j, a_r), \pi_i^h) = 2$. $\operatorname{dist}(f^h(i, j, a_r), y) = \operatorname{dist}(f^h(i, j, b_r), y) = \operatorname{dist}(f^h(i, j, c_r), y) = \operatorname{min}(2+|P^h(i, j', p_i^{3-h})|/2+|P^h(i, j', p_i^{3-h})|/2+|P^h(i$

$$\begin{split} |P(q_i^{3-h}, \operatorname{mid}(P^h(i,j',p_i^{3-h})))| - \ell_y, 2 + |P(\pi_i^h, a_r)| + |P(\pi_i^{3-h}, a_r)| + 1 + \ell_y) &= \operatorname{min} (1 + 40(n+1) - \ell_y, 3 + 20(n+1) + \ell_y). \\ \text{For the pair} \{x_1, y\} \text{ that is not resolved by } f^h(i, j, a_r), \operatorname{dist}(f^h(i, j, b_r), y) &= \operatorname{dist}(f^h(i, j, a_r), y) = \operatorname{dist}(f^h(i, j, a_r), x_1) = \operatorname{dist}(f^h(i, j, b_r), x_1) - 2 < \operatorname{dist}(f^h(i, j, b_r), x_1). \\ \text{Thus every pair} \{x_1, y\} \text{ is resolved by } f^h(i, j, a_r) \text{ or } f^h(i, j, b_r). \\ \text{In a similar way, we can show that every vertex pair} \{x_2, y\} \text{ and } \{x_3, y\} \text{ are resolved by } S'. \\ \text{For the pair} \{x_4, y\}, \text{ there are two cases. Case 1: } j' \neq j. \\ \text{In this case, the analysis is similar to that of } \{x_1, y\} \text{ above and every pair} \{x_4, y\} \text{ is resolved by } f^h(i, j, p_i^{3-h}) \text{ or } f^h(i, j, a_r). \\ \text{Case 2: } j' = j. \\ \operatorname{dist}(f^h(i, j, p_i^{3-h}), x_4) = \ell_{x_4} \\ \text{if } x_4 \neq \pi_i^h \text{ and } \operatorname{dist}(f^h(i, j, p_i^{3-h}), \pi_i^h) = 2. \\ \operatorname{dist}(f^h(i, j, p_i^{3-h}), y) = \min(|P^h(i, j', p_i^{3-h})|/2 + |P(q_i^{3-h}, \min(|P^h(i, j', p_i^{3-h})|)| - \ell_y, 2 + |P(\pi_i^h, a_r)| + |P(\pi_i^{3-h}, a_r)| + 1 + \ell_y) = \min(40(n + 1) - 1 - \ell_y, 3 + 20(n + 1) + \ell_y). \\ \text{It satisfies that } \operatorname{dist}(f^h(i, j, p_i^{3-h}), x_4) = \operatorname{dist}(f^h(i, j, p_i^{3-h}), y) = 40(n + 1) - 1 - \ell_y = \ell_{x_4}, \text{ i.e.} \\ \operatorname{dist}(f^{mid}(i, j, h), x_4) = \operatorname{dist}(f^{mid}(i, j, h), y) \text{ and } 10(n + 1) < \ell_{x_4} \leq 20(n + 1), 20(n + 1) - 1 \leq \ell_y < 30(n + 1) - 1. \\ \text{For such pairs, } \operatorname{dist}(f^{ecc}(i, j, 3 - h, r), x_4) = 2 + 30(n + 1) - \ell_{x_4} < 2 + 20(n + 1) < \operatorname{dist}(f^{ecc}(i, j, 3 - h, r), y) = 50(n + 1) + 1 - \ell_y. \\ \text{Thus in this case, every pair} \{x_4, y\} \text{ is resolved by } f^h(i, j, p_i^{3-h}) \text{ or } f^{ecc}(i, j, 3 - h, r). \\ \end{array}$$

Finally we show that every pair $\{x, y\} \in (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h})) \times P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))$ for $i, i' \in [n], j, j' \in [m], h, h' \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ such that $i \neq i'$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h, h' \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ such that $i \neq i'$. Suppose that $x_1 \in P^h(i, j, a_r), x_2 \in P^h(i, j, b_r), x_3 \in P^h(i, j, c_r)$ and $x_4 \in P^h(i, j, p_i^{3-h})$. Suppose that $y \in P(q_{i'}^{3-h'}, \operatorname{mid}(P^{h'}(i', j', p_{i'}^{3-h'})))$. For a vertex x_{μ} for $\mu \in \{1, 2, 3, 4\}$, let $\ell_{x_{\mu}} = \operatorname{dist}(\pi_i^h, x_{\mu})$. For a vertex y, let $\ell_y = \operatorname{dist}(q_{i'}^{3-h'}, y)$. For the pair $\{x_1, y\}, \operatorname{dist}(f^h(i, j, a_r), x_1) = \ell_{x_1}$ if $x_1 \neq \pi_i^h$ and $\operatorname{dist}(f^h(i, j, a_r), \pi_i^h) = 2$. $\operatorname{dist}(f^h(i, j, a_r), y) = \min(2+|P(\pi_i^h, a_r)|+|P(\pi_{i'}^{h'}, a_r)|+1+\ell_y, 2+|P(\pi_i^h, a_r)|+|P(\pi_{i'}^{3-h'}, a_r)|+|P^{3-h'}(i', j', p_{i'}^{h'})|/2 + |P(q_{i'}^{h'}, \operatorname{mid}(P^{3-h'}(i', j', p_{i'}^{h'})))| - \ell_y) = \min(3+20(n+1)+\ell_y, 1+60(n+1)-\ell_y) \geq 3+20(n+1) > \operatorname{dist}(f^h(i, j, a_r), x_1)$. Thus every pair $\{x_1, y\}$ is resolved by $f^h(i, j, a_r)$. Similarly, we can show that every pair $\{x_2, y\}, \{x_3, y\}$ and $\{x_4, y\}$ are resolved by $f^h(i, j, b_r), f^h(i, j, c_r)$ and $f^h(i, j, p_i^{3-h})$ respectively. This completes the proof for the lemma. \Box

Lemma 64. Every pair $\{x, y\} \in \bigcup_{i \in [n]} \prod_i \times \bigcup_{i \in [n]} S_i$ is resolved by S'.

Proof. We show that every pair $\{x, y\} \in (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h})) \times (P^h(i, j, a_r) \cup P^h(i, j, a_r) \cup P^h(i, j, a_r))$ $(P(\pi_{i'}^{h'}, a_{r'}) \cup P(\pi_{i'}^{h'}, c_{r'}))$ for $i, i' \in [n], j \in [m], h, h' \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j \in [m], h, h' \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$. Suppose that $x_1 \in P^h(i, j, a_r)$, $x_2 \in P^h(i, j, b_r)$, $x_3 \in P^h(i, j, c_r)$ and $x_4 \in P^h(i, j, p_i^{3-h})$. Suppose that $y_1 \in P(\pi_{i'}^{h'}, a_{r'})$ and $y_2 \in P(\pi_{i'}^{h'}, c_{r'})$. For a vertex x_{μ} for $\mu \in \{1, 2, 3, 4\}$, let $\ell_{x_{\mu}} =$ $\operatorname{dist}(\pi_i^h, x_\mu)$. For a vertex y_1 , let $\ell_{y_1} = \operatorname{dist}(a_{r'}, y_1)$. For a vertex y_2 , let $\ell_{y_2} = \operatorname{dist}(c_{r'}, y_2)$. Let $|P(s_i^j, a_{r'})| = 20(n+1) + 10\lambda$ for some $\lambda \in [n]$. For a pair $\{x_1, y_1\}$, $dist(f(s_i^j, a_{r'}), y_1) = 2 + \ell_{y_1}$. $\operatorname{dist}(f(s_i^j, a_{r'}), x_1) = \min (2 + |P(\pi_i^h, a_{r'})| + \ell_{x_1}, |P(s_i^j, a_{r'})| - 1 + |P^h(i, j, a_r)| - \ell_{x_1}) = \min (2 + \ell_{x_1}) + \ell_{x_1} + \ell$ $10(n+1) + \ell_{x_1}, 40(n+1) + 10\lambda - 1 - \ell_{x_1}) \text{ if } r = r' \text{ and } \operatorname{dist}(f(s_i^j, a_{r'}), x_1) = \min \left(2 + |P(\pi_i^h, a_{r'})| + 10\lambda - 1 - \ell_{x_1}\right) + 10\lambda - 1 - \ell_{x_1} + 10\lambda - 1$ $\ell_{x_1}, |P(s_i^j, a_{r'})| + 1 + |P^h(i, j, a_r)| - \ell_{x_1}) = \min(2 + 10(n+1) + \ell_{x_1}, 40(n+1) + 10\lambda + 1 - \ell_{x_1}) \ge 10^{-10}$ $2 + 10(n+1) \ge \text{dist}(f(s_i^j, a_{r'}), y_1) \text{ if } r \ne r'. \text{ It follows that } \text{dist}(f(s_i^j, a_{r'}), x_1) \ge 2 + 10(n+1) \ge 2 + 10(n+1)$ dist $(f(s_i^j, a_{r'}), y_1)$. dist $(f(s_i^j, a_{r'}), x_1) = \text{dist}(f(s_i^j, a_{r'}), y_1)$ only when $x_1 = \pi_i^h$ and $y_1 = \pi_{i'}^{h'}$. If $i' \neq i$ or $h' \neq h$, obviously the pair $\{\pi_i^h, \pi_{i'}^{h'}\}$ is resolved by $f^h(i, j, a_r)$. Thus every pair $\{x_1, y_1\}$ is resolved by $f(s_i^j, a_{r'})$ or $f^h(i, j, a_r)$. In a similar way, we can show that every vertex pair $\{x_2, y_1\}, \{x_3, y_1\}$ and $\{x_4, y_1\}$ are resolved by $f(s_i^j, a_{r'})$ or $f^h(i, j, a_r)$. Also in a similar way, we can show that every vertex pair $\{x_1, y_2\}, \{x_2, y_2\}, \{x_3, y_2\}$ and $\{x_4, y_2\}$ are resolved by $f(s_i^j, c_{r'})$ or $f^h(i, j, a_r)$. This completes the proof for the lemma.

Lemma 65. Every pair $\{x, y\} \in \bigcup_{i \in [n]} \prod_i \times \bigcup_{r \in \{1,2,3\}} R_r$ is resolved by S'.
Proof. We show that every pair $\{x, y\} \in (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h})) \times$ $(P(u_{r'}^{i'}, a_{r'}) \cup P(v_{r'}^{i'}, a_{r'}) \cup P(u_{r'}^{i'}, c_{r'}) \cup P(v_{r'}^{i'}, c_{r'}))$ for $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r, r' \in [n], j \in [m], h \in \{1, 2\}$ $\{1,2,3\}$ is resolved by S'. We fix arbitrary integers $i,i' \in [n], j \in [m], h \in \{1,2\}$ and $r, r' \in [n], j \in [m], h \in \{1,2\}$ $\{1, 2, 3\}$. Suppose that $x_1 \in P^h(i, j, a_r), x_2 \in P^h(i, j, b_r), x_3 \in P^h(i, j, c_r) \text{ and } x_4 \in P^h(i, j, p_i^{3-h}).$ Suppose that $y_1 \in P(u_{r'}^{i'}, a_{r'}), y_2 \in P(v_{r'}^{i'}, a_{r'}), z_1 \in P(u_{r'}^{i'}, c_{r'}) \text{ and } z_2 \in P(v_{r'}^{i'}, c_{r'}).$ For a vertex x_{μ} for $\mu \in \{1, 2, 3, 4\}$, let $\ell_{x_{\mu}} = \operatorname{dist}(\pi_i^h, x_{\mu})$. For a vertex y_{ν} for $\nu \in \{1, 2\}$, let $\ell_{y_{\nu}} = \text{dist}(a_{r'}, y_{\nu}).$ For a vertex z_{η} for $\eta \in \{1, 2\}$, let $\ell_{y_{\eta}} = \text{dist}(c_{r'}, z_{\eta}).$ Then $|P(u_{r'}^{i'}, a_{r'})| =$ $|P(v_{r'}^{i'}, a_{r'})| = 20(n+1) - 10i'$ and $|P(u_{r'}^{i'}, c_{r'})| = |P(v_{r'}^{i'}, c_{r'})| = 20(n+1) + 10i'$. For a pair $\{x_1, y_1\}$, dist $(f^h(i, j, a_r), x_1) = \ell_{x_1}$ and dist $(f^h(i, j, a_r), \pi_i^h) = 2$. dist $(f^h(i, j, a_r), y_1) = \ell_{x_1}$ $\operatorname{dist}(f^{h}(i,j,b_{r}),y_{1}) = \operatorname{dist}(f^{h}(i,j,c_{r}),y_{1}) = \operatorname{dist}(f^{h}(i,j,p_{i}^{3-h}),y_{1}) = 2 + |P(\pi_{i}^{h},a_{r'})| + \ell_{y_{1}} = 2 + |P(\pi_{i}^{h}$ $2+10(n+1)+\ell_{y_1}$. For the pair $\{x_1, y_1\}$ that is not resolved by $f^h(i, j, a_r)$, $dist(f^h(i, j, a_r), x_1) =$ $dist(f^h(i, j, a_r), y_1) = dist(f^h(i, j, b_r), y_1) = dist(f^h(i, j, b_r), x_1) - 2$. Thus every pair $\{x_1, y_1\}$ is resolved by $f^h(i, j, a_r)$ or $f^h(i, j, b_r)$. In a similar way, we can show that every vertex pair $\{x_\mu, y_\nu\}$ for $\mu \in \{1, 2, 3, 4\}, \nu \in \{1, 2\}$ is resolved by S'. For a pair $\{x_1, z_1\}, \operatorname{dist}(f^h(i, j, a_r), x_1) = \ell_{x_1}$ and dist $(f^h(i, j, a_r), \pi^h_i) = 2$. dist $(f^h(i, j, a_r), z_1) = \text{dist}(f^h(i, j, b_r), z_1) = \min(2 + |P(\pi^h_i, c_{r'})| + |P(\pi^h_i$ $\ell_{z_1}, 2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| + |P(c_{r'}, u_{r'}^{i'})| - 2 - \ell_{z_1}) = \min(2 + 10(n+1) + \ell_{z_1}, 50(n+1) - \ell_{z_1}) \text{ if } i = 0$ $|P(c_{r'}, u_{r'}^{i'})| - \ell_{z_1} \ge 2. \operatorname{dist}(f^h(i, j, a_r), z_1) = \operatorname{dist}(f^h(i, j, b_r), z_1) = 2 + |P(\pi_i^h, a_{r'})| + |P(a_{r'}, u_{r'}^{i'})| +$ $|P(c_{r'}, u_{r'}^{i'})| - \ell_{z_1}$ if $|P(c_{r'}, u_{r'}^{i'})| - \ell_{z_1} < 2$. For the pair $\{x_1, z_1\}$ that is not resolved by $f^h(i, j, a_r)$, $\operatorname{dist}(f^h(i, j, a_r), x_1) = \operatorname{dist}(f^h(i, j, a_r), z_1) = \operatorname{dist}(f^h(i, j, b_r), z_1) = \operatorname{dist}(f^h(i, j, b_r), x_1) - 2$. Thus every pair $\{x_1, z_1\}$ is resolved by $f^h(i, j, a_r)$ or $f^h(i, j, b_r)$. In a similar way, we can show that every vertex pair $\{x_{\mu}, z_{\nu}\}$ for $\mu \in \{1, 2, 3, 4\}, \nu \in \{1, 2\}$ is resolved by S'.

Then we show that every pair $\{x, y\} \in (P^h(i, j, a_r) \cup P^h(i, j, b_r) \cup P^h(i, j, c_r) \cup P^h(i, j, p_i^{3-h})) \times$ $(P(u_{r'}^{i'}, b_{r'}) \cup P(v_{r'}^{i'}, b_{r'}))$ for $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$. Suppose that $x_1 \in P^h(i, j, a_r), x_2 \in P^h(i, j, b_r), x_3 \in P^h(i, j, c_r) \text{ and } x_4 \in P^h(i, j, p_i^{3-h}).$ Suppose that $w_1 \in P(u_{r'}^{i'}, b_{r'})$ and $w_2 \in P(v_{r'}^{i'}, b_{r'})$. For a vertex x_μ for $\mu \in \{1, 2, 3, 4\}$, let $\ell_{x_\mu} = \operatorname{dist}(\pi_i^h, x_\mu)$. For a vertex w_{ν} for $\nu \in \{1, 2\}$, let $\ell_{w_{\nu}} = \operatorname{dist}(b_{r'}, w_{\nu})$. Then let $|P(s_i^j, a_{r'})| = 20(n+1) + 10\lambda$ for some $\lambda \in [n], |P(u_{r'}^{i'}, b_{r'})| = 20(n+1) - 5i' - 1$ and $|P(v_{r'}^{i'}, b_{r'})| = 20(n+1) - 5i' - 2$. For a pair $\{x_1, w_1\}$, dist $(f^1(u_{r'}^{i'}, v_{r'}^{i'}), w_1) = \text{dist}(f^2(u_{r'}^{i'}, v_{r'}^{i'}), w_1) = 2 + |P(u_{r'}^{i'}, b_{r'})| - \ell_{w_1}$. For the distance between $f^{\eta}(u_{r'}^{i'}, v_{r'}^{i'})$ and x_1 for $\eta \in \{1, 2\}$, there are two cases. Case 1: $\lambda \leq 1$ *i'*. dist $(f^1(u_{r'}^{i'}, v_{r'}^{i'}), x_1) = \text{dist}(f^2(u_{r'}^{i'}, v_{r'}^{i'}), x_1) + 1 = \min(1 + |P(a_{r'}, u_{r'}^{i'})| + |P(s_i^j, a_{r'})| - 1)$ $1 + |P^{h}(i, j, a_{r})| - \ell_{x_{1}}, 1 + |P(a_{r'}, u_{r'}^{i'})| + |P(\pi_{i}^{h}, a_{r'})| + \ell_{x_{1}}) \text{ if } r = r'. \quad \text{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) = r'.$ $\operatorname{dist}(f^{2}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) + 1 = \min(1 + |P(a_{r'}, u_{r'}^{i'})| + |P(s_{i}^{j}, a_{r'})| + 1 + |P^{h}(i, j, a_{r})| - \ell_{x_{1}}, 1 + |P^{h}(i, j, a_{r'})| + 1 + |P^{h}(i, j, a$ $|P(a_{r'}, u_{r'}^{i'})| + |P(\pi_i^h, a_{r'})| + \ell_{x_1})$ if $r \neq r'$. In this case, for the pair $\{x_1, w_1\}$ which is not resolved by $f^{1}(u_{r'}^{i'}, v_{r'}^{i'})$, $\operatorname{dist}(f^{2}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) - 1 < \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), y) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) - 1 < \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), y) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) - 1 < \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), y) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) - 1 < \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), y) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) - 1 < \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), y) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) = \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1})$ $\operatorname{dist}(f^{2}(u_{r'}^{i'}, v_{r'}^{i'}), y). \quad \operatorname{Case} 2: \lambda > i'. \quad \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) = \operatorname{dist}(f^{2}(u_{r'}^{i'}, v_{r'}^{i'}), x_{1}) + 1 = \min(1 + 1)$ $|P(c_{r'}, u_{r'}^{i'})| + |P(s_i^j, c_{r'})| + 1 + |P^h(i, j, a_r)| - \ell_{x_1}, 1 + |P(a_{r'}, u_{r'}^{i'})| + |P(\pi_i^h, a_{r'})| + \ell_{x_1}).$ In this case, for the pair $\{x_1, w_1\}$ which is not resolved by $f^1(u_{r'}^{i'}, v_{r'}^{i'}), \operatorname{dist}(f^2(u_{r'}^{i'}, v_{r'}^{i'}), x_1) =$ $\operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), x_1) - 1 < \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), y) = \operatorname{dist}(f^2(u_{r'}^{i'}, v_{r'}^{i'}), y).$ In a similar way, we can show that every vertex pair $\{x_{\mu}, w_{\nu}\}$ for $\mu \in \{1, 2, 3, 4\}, \nu \in \{1, 2\}$ is resolved by S'. This completes the proof for the lemma.

Lemma 66. Every pair $\{x, y\} \in \bigcup_{i \in [n]} L_i \times \bigcup_{i \in [n]} H_i$ is resolved by S'.

Proof. First we show that every pair $\{x, y\} \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h))) \times (P(s_{i'}^{j'}, a_r) \cup P(s_{i'}^{j'}, c_r))$ for $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ and $y \in P(s_{i'}^{j'}, a_r)$. For a vertex $x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$, let $P(q_i^h, x)$ be the subpath of $P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ from q_i^h to x and $|P(q_i^h, x)| = \ell_x$. For a vertex $y \in P(s_{i'}^{j'}, a_r)$, let
$$\begin{split} P(a_r,y) & \text{ be the subpath of } P(s_{i'}^{j'},a_r) \text{ from } a_r \text{ to } y \text{ and } |P(a_r,y)| = \ell_y. \text{ Then } \operatorname{dist}(f(\pi_i^h,a_r),x) = \\ & \min \left(|P(\pi_i^h,a_r)| + 1 + \ell_x, 2 + |P(\pi_i^{3-h},a_r)| + |P^{3-h}(i,j,p_i^h)|/2 + |P(q_i^h,\operatorname{mid}(P^{3-h}(i,j,p_i^h)))| - \ell_x) = \\ & \min \left(10(n+1) + 1 + \ell_x, 50(n+1) + 1 - \ell_x\right). \quad \operatorname{dist}(f(\pi_i^{3-h},a_r),x) = \\ & \min \left(2 + |P(\pi_i^h,a_r)| + |P^{3-h}(i,j,p_i^h)|/2 + |P(q_i^h,\operatorname{mid}(P^{3-h}(i,j,p_i^h)))| - \ell_x\right) = \\ & \min \left(10(n+1) + 1 + \ell_x, 50(n+1) - 1 - \ell_x\right). \quad \operatorname{dist}(f(\pi_i^h,a_r),y) = \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y) = 2 + \ell_y. \text{ For a pair } \{x,y\} \text{ which is not resolved by } f(\pi_i^h,a_r), \text{ there are two cases. Case 1: } \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y) = 10(n+1) + 1 + \ell_x = 2 + \ell_y. \text{ In this case, } \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),x) = \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y) = 10(n+1) + 1 + \ell_x = 2 + \ell_y. \text{ In this case, } \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),x) = \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y) = 10(n+1) + 1 + \ell_x = 2 + \ell_y. \text{ In this case, } \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),x) = \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y) = 10(n+1) + 1 - \ell_x = 2 + \ell_y. \text{ In this case, } \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),x) = \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y). \text{ It follows that every pair } \{x,y\} \text{ is resolved by } \\ & f(\pi_i^h,a_r) \text{ or } \\ & \operatorname{dist}(\pi_i^{3-h},a_r). \\ & \operatorname{dist}(f(\pi_i^{3-h},a_r),y). \text{ It follows that every pair } \\ & \operatorname{dist}(P^{3-h}(i,j,p_i^h))) \times P(s_{i'}^{j'},c_r) \text{ is resolved by } \\ & f(\pi_i^h,c_r) \text{ or } \\ & f(\pi_i^{3-h},c_r). \end{aligned}$$

Then we show that every pair $\{x, y\} \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h))) \times (P(s_{i'}^{j'}, b_r) \setminus \{s_{i'}^{j'}\})$ for $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j, j' \in [m], h \in \{1, 2\}$ and $r \in \{1, 2, 3\}$. Suppose that $x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))$ and $y \in P(s_{i'}^{j'}, b_r) \setminus \{s_{i'}^{j'}\}$. We define ℓ_x and ℓ_y in a similar way to that of ℓ_x and ℓ_y in the first paragraph. Suppose that $s_{i'}^{j'}$ resolves the pair $\{u_r^{i_r}, v_r^{i_r}\}$ for some $i_r \in [n], i.e. |P(a_r, u_r^{i_r})| = 20(n+1) - 10i_r$. Then $\operatorname{dist}(f^1(u_r^{i_r}, v_r^{i_r}), x) = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), x) + 1 = \min(1 + |P(a_r, u_r^{i_r})| + |P(\pi_i^h, a_r)| + 1 + \ell_x, 1 + |P(a_r, u_r^{i_r})| + |P(\pi_i^{3-h}, a_r)| + |P^{3-h}(i, j, p_i^h)|/2 + |P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))| - \ell_x) = \min(2 + 30(n+1) - 10i_r + \ell_x, 70(n+1) - 10i_r - \ell_x)$. $\operatorname{dist}(f^1(u_r^{i_r}, v_r^{i_r}), y) = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), y) = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), x) = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), y) = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), y) = \operatorname{dist}(f^2(u_r^{i_r}, v_r^{i_r}), y) = 1$. It follows that every pair $\{x, y\}$ is resolved by $f^1(u_r^{i_r}, v_r^{i_r})$ or $f^2(u_r^{i_r}, v_r^{i_r})$. This completes the proof for the lemma.

Lemma 67. Every pair $\{x, y\} \in \bigcup_{i \in [n]} L_i \times \bigcup_{i \in [n]} S_i$ is resolved by S'.

 $\begin{array}{l} Proof. \text{ We show that every pair } \{x,y\} \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))) \times (P(\pi_{i'}^{h'},a_r) \cup P(\pi_{i'}^{h'},c_r)) \\ \text{for } i,i' \in [n], j \in [m], h, h' \in \{1,2\} \text{ and } r \in \{1,2,3\} \text{ is resolved by } S'. \text{ We fix arbitrary integers } \\ i,i' \in [n], j \in [m], h, h' \in \{1,2\} \text{ and } r \in \{1,2,3\}. \text{ Suppose that } x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))) \\ \text{and } y \in P(\pi_{i'}^{h'},a_r). \text{ For a vertex } x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))), \text{ let } P(q_i^h,x) \text{ be the subpath } \\ \text{of } P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))) \text{ from } q_i^h \text{ to } x \text{ and } |P(q_i^h,x)| = \ell_x. \text{ For a vertex } y \in P(\pi_{i'}^{h'},a_r), \text{ let } \\ P(a_r,y) \text{ be the subpath of } P(\pi_{i'}^{h'},a_r) \text{ from } a_r \text{ to } y \text{ and } |P(a_r,y)| = \ell_y. \text{ Then } \operatorname{dist}(f(s_i^j,a_r),y) = \\ 2 + \ell_y \leq 2 + 10(n+1). \quad \operatorname{dist}(f(s_i^j,a_r),x) = \min(2 + |P(\pi_i^h,a_r)| + 1 + \ell_x, 2 + |P(\pi_i^{3-h},a_r)| + |P^{3-h}(i,j,p_i^h)|/2 + |P(q_i^h,\operatorname{mid}(P^{3-h}(i,j,p_i^h)))| - \ell_x) = \min(10(n+1) + 3 + \ell_x, 50(n+1) + 1 - \ell_x) \geq \\ 3 + 10(n+1) > \operatorname{dist}(f(s_i^j,a_r),x). \text{ Thus every pair } \{x,y\} \text{ is resolved by } f(s_i^j,a_r). \text{ Similarly we can show that every pair of } P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))) \times P(\pi_{i'}^{h'},c_r) \text{ is resolved by } f(s_i^j,c_r). \text{ This completes the proof for the lemma.} \square \end{aligned}$

Lemma 68. Every pair $\{x, y\} \in \bigcup_{i \in [n]} L_i \times \bigcup_{r \in \{1,2,3\}} R_r$ is resolved by S'.

 $\begin{array}{l} Proof. \text{ We show that every pair } \{x,y\} \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))) \times (P(u_r^{i'},a_r) \cup P(v_r^{i'},a_r) \cup P(v_r^{i'},a_r) \cup P(u_r^{i'},b_r) \cup P(u_r^{i'},c_r) \cup P(v_r^{i'},c_r)) \text{ for } i,i' \in [n], j \in [m], h \in \{1,2\} \text{ and } r \in \{1,2,3\} \text{ is resolved by } S'. \text{ We fix arbitrary integers } i,i' \in [n],j \in [m], h \in \{1,2\} \text{ and } r \in \{1,2,3\}. \text{ Suppose that } x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))), y_1 \in P(u_r^{i'},a_r), y_2 \in P(v_r^{i'},a_r), z_1 \in P(u_r^{i'},b_r), z_2 \in P(v_r^{i'},b_r), w_1 \in P(u_r^{i'},c_r), w_2 \in P(v_r^{i'},c_r). \text{ For a vertex } x \in P(q_i^h, \operatorname{mid}(P^{3-h}(i,j,p_i^h))), \text{ let } P(q_i^h,x) \text{ be the subpath of } P(q_i^h,\operatorname{mid}(P^{3-h}(i,j,p_i^h))) \text{ from } q_i^h \text{ to } x \text{ and } |P(q_i^h,x)| = \ell_x. \text{ For a vertex } y_1 \in P(u_r^{i'},a_r), \text{ let } P(y_1,u_r^{i'}) \text{ be the subpath of } P(u_r^{i'},a_r) \text{ from } y_1 \text{ to } u_r^{i'} \text{ and let } |P(y_1,u_r^{i'})| = \ell_{y_1}. \text{ For a vertex } y_2 \in P(v_r^{i'},a_r), \text{ let } P(y_2,v_r^{i'}) \text{ be the subpath of } P(v_r^{i'},a_r) \text{ from } y_1 \text{ for } p(v_r^{i'},a_r) \text{ from } p(v_r^{i'},a_r) \text{ f$

 y_2 to $v_r^{i'}$ and let $|P(y_2, v_r^{i'})| = \ell_{y_2}$. For a vertex $z_1 \in P(u_r^{i'}, b_r)$, let $P(z_1, u_r^{i'})$ be the subpath of $P(u_r^{i'}, b_r)$ from z_1 to $u_r^{i'}$ and let $|P(z_1, u_r^{i'})| = \ell_{z_1}$. For a vertex $z_2 \in P(v_r^{i'}, b_r)$, let $P(z_2, v_r^{i'})$ be the subpath of $P(v_r^{i'}, b_r)$ from z_2 to $v_r^{i'}$ and let $|P(z_2, v_r^{i'})| = \ell_{z_2}$. For a vertex $w_1 \in P(u_r^{i'}, c_r)$, let $P(w_1, u_r^{i'})$ be the subpath of $P(u_r^{i'}, c_r)$ from w_1 to $u_r^{i'}$ and let $|P(w_1, u_r^{i'})| = \ell_{w_1}$. For a vertex $w_2 \in P(v_r^{i'}, c_r)$, let $P(w_2, v_r^{i'})$ be the subpath of $P(v_r^{i'}, c_r)$ from w_2 to $v_r^{i'}$ and let $|P(w_2, v_r^{i'})| = \ell_{w_2}$. For a pair $\{x, y_1\}$, dist $(f^1(u_r^{i'}, v_r^{i'}), y_1) = 1 + \ell_{y_1}$ if $y_1 \neq u_r^{i'}$ and dist $(f^1(u_r^{i'}, v_r^{i'}), u_r^{i'}) = 2$. $\operatorname{dist}(f^{1}(u_{r}^{i'}, v_{r}^{i'}), x) = \min(1 + |P(a_{r}, u_{r}^{i'})| + |P(\pi_{i}^{h}, a_{r})| + 1 + \ell_{x}, 1 + |P(a_{r}, u_{r}^{i'})| + |P(\pi_{i}^{3-h}, a_{r})| + |P(\pi_{i$ $|P^{3-h}(i,j,p_i^h)|/2 + |P(q_i^h,\operatorname{mid}(P^{3-h}(i,j,p_i^h)))| - \ell_x) = \min(2 + 30(n+1) - 10i' + \ell_x, 70(n+1) - 10i' + \ell_x) = 0$ $1) - 10i' - \ell_x$ > 1 + 20(n + 1) - 10i' $\geq \operatorname{dist}(f^1(u_r^{i'}, v_r^{i'}), y_1)$. Thus every pair $\{x, y_1\}$ is resolved by $f^1(u_r^{i'}, v_r^{i'})$. Similarly, every pair $\{x, y_2\}$ is resolved by $f^1(u_r^{i'}, v_r^{i'})$. For a pair $\{x, z_1\}, \operatorname{dist}(f^2(u_r^{i'}, v_r^{i'}), x) = \operatorname{dist}(f^1(u_r^{i'}, v_r^{i'}), x) - 1. \operatorname{dist}(f^1(u_r^{i'}, v_r^{i'}), z_1) = \operatorname{dist}(f^2(u_r^{i'}, v_r^{i'}), z_$ $2 + \ell_{z_1}$. Thus for a pair $\{x, z_1\}$ that is not resolved by $f^1(u_r^{i'}, v_r^{i'})$, $\operatorname{dist}(f^1(u_r^{i'}, v_r^{i'}), x) = \operatorname{dist}(f^1(u_r^{i'}, v_r^{i'}), z_1) = \operatorname{dist}(f^2(u_r^{i'}, v_r^{i'}), z_1) > \operatorname{dist}(f^2(u_r^{i'}, v_r^{i'}), x)$. It follows that every pair $\{x, z_1\}$ is resolved by $f^1(u_r^{i'}, v_r^{i'})$ or $f^2(u_r^{i'}, v_r^{i'})$. Similarly, every pair $\{x, z_2\}$ is resolved by $f^1(u_r^{i'}, v_r^{i'})$ or $f^{2}(u_{r}^{i'}, v_{r}^{i'})$. For a pair $\{x, w_{1}\}$, dist $(f(\pi_{i}^{h}, c_{r}), w_{1}) = \text{dist}(f(\pi_{i}^{3-h}, c_{r}), w_{1}) = \text{dist}(f(s_{i}^{j}, c_{r}), w_{1}) = \text{dist}(f(\pi_{i}^{j}, c_{r}), w_{1}) = \text{dist}(f(\pi$ $\begin{aligned} 2 + |P(c_r, u_r^{i'})| - \ell_{w_1} &= 2 + 20(n+1) + 10i' - \ell_{w_1}. \text{ dist}(f(\pi_i^h, c_r), x) &= \min(|P(\pi_i^h, a_r)| + 1 + \ell_x, 2 + |P(\pi_i^{3-h}, a_r)| + |P^{3-h}(i, j, p_i^h)|/2 + |P(q_i^h, \min(P^{3-h}(i, j, p_i^h)))| - \ell_x) &= \min(10(n+1) + 1 + \ell_x, 50(n+1) + 1 - \ell_x). \text{ dist}(f(\pi_i^{3-h}, c_r), x) &= \min(2 + |P(\pi_i^h, a_r)| + 1 + \ell_x, |P(\pi_i^{3-h}, a_r)| + 1 + \ell_x). \end{aligned}$ $\ell_x). \quad \operatorname{dist}(f(s_i^j, c_r), x) = \min \left(2 + |P(\pi_i^h, a_r)| + 1 + \ell_x, 2 + |P(\pi_i^{3-h}, a_r)| + |P^{3-h}(i, j, p_i^h)|/2 + |P(q_i^h, \operatorname{mid}(P^{3-h}(i, j, p_i^h)))| - \ell_x) = \min \left(10(n+1) + 3 + \ell_x, 50(n+1) + 1 - \ell_x\right). \text{ For a pair } \ell_x$ $\{x, w_1\}$ which is not resolved by $f(s_i^j, c_r)$, either $f(\pi_i^h, c_r)$ or $f(\pi_i^{3-h}, c_r)$ resolves it. Thus every pair $\{x, w_1\}$ is resolved by $f(s_i^j, c_r)$, $f(\pi_i^h, c_r)$ or $f(\pi_i^{3-h}, c_r)$. Similarly, we can show that every pair $\{x, w_2\}$ is resolved by $f(s_i^j, c_r)$, $f(\pi_i^h, c_r)$ or $f(\pi_i^{3-h}, c_r)$. This completes the proof for the lemma.

Lemma 69. Every pair $\{x, y\} \in \bigcup_{i \in [n]} S_i \times \bigcup_{i \in [n]} H_i$ is resolved by S'.

Proof. We show that every pair $\{x, y\} \in (P(\pi_{i'}^h, a_{r'}) \cup P(\pi_{i'}^h, c_{r'})) \times (P(s_i^j, a_r) \cup P(s_i^j, b_r) \cup P(s_i^j, b_r))$ $P(s_i^j, c_r))$ for $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], j \in [m], h \in \{1, 2\}$ and $r, r' \in \{1, 2, 3\}$. Suppose that $x_1 \in P(\pi_{i'}^h, a_{r'})$, $x_2 \in P(\pi_{i'}^h, c_{r'}), y_1 \in P(s_i^j, a_r), y_2 \in P(s_i^j, a_r) \text{ and } y_3 \in P(s_i^j, c_r).$ For a vertex $x_1 \in P(\pi_{i'}^h, a_{r'}), y_2 \in P(x_i^j, a_r)$ let $P(\pi_{i'}^h, x_1)$ be the subpath of $P(\pi_{i'}^h, a_{r'})$ from $\pi_{i'}^h$ to x_1 and let $|P(\pi_{i'}^h, x_1)| = \ell_{x_1}$. For a vertex $x_2 \in P(\pi_{i'}^h, c_{r'}), \text{ let } P(\pi_{i'}^h, x_2) \text{ be the subpath of } P(\pi_{i'}^h, c_{r'}) \text{ from } \pi_{i'}^h \text{ to } x_2 \text{ and let } |P(\pi_{i'}^h, x_2)| = \ell_{x_2}.$ For a vertex $y_1 \in P(s_i^j, a_r)$, let $P(s_i^j, y_1)$ be the subpath of $P(s_i^j, a_r)$, from s_i^j to y_1 and let $|P(s_i^j, y_1)| = \ell_{y_1}$. For a vertex $y_2 \in P(s_i^j, b_r) \setminus \{s_i^j\}$, let $P(s_i^j, y_2)$ be the subpath of $P(s_i^j, b_r)$, from s_i^j to y_2 and let $|P(s_i^j, y_2)| = \ell_{y_2}$. For a vertex $y_3 \in P(s_i^j, c_r)$, let $P(s_i^j, y_3)$ be the subpath of $P(s_i^j, c_r)$, from s_i^j to y_3 and let $|P(s_i^j, y_3)| = \ell_{y_3}$. Let $|P(s_i^j, a_r)| = 20(n+1) + 10\lambda$ for some $\lambda \in [n]$. For a vertex pair $\{x_1, y_1\}$, dist $(f^h(i', j, a_r), x_1) = 2 + \ell_{x_1}$. For the distance between $f^h(i', j, a_r)$ and y_1 , there are two cases. Case 1: i' = i. Then $dist(f^h(i', j, a_r), y_1) = min(|P^h(i, j, a_r)| +$ $\ell_{y_1} - 1, 2 + |P(\pi_i^h, a_r)| + |P(s_i^j, a_r)| - \ell_{y_1}) = \min(20(n+1) + \ell_{y_1} - 1, 30(n+1) + 10\lambda + 2 - \ell_{y_1})$ if $y_1 \neq s_i^j$ and dist $(f^h(i', j, a_r), s_i^j) = 20(n+1) + 1$. Thus dist $(f^h(i', j, a_r), y_1) \ge 2 + 10(n+1) \ge 2$ $\operatorname{dist}(f^h(i',j,a_r),x_1)$. $f^h(i',j,a_r)$ does not resolve $\{x_1,y_1\}$ only when $x_1 = a_{r'}$ and $y_1 = a_r$ with $r \neq r'$. The pair $\{a_{r'}, a_r\}$ is resolved by $f(\pi_{i'}^h, a_{r'})$. Thus in this case, every pair $\{x_1, y_1\}$ is resolved by $f^{h}(i', j, a_{r})$ or $f(\pi_{i'}^{h}, a_{r'})$. Case 2: $i' \neq i$. Then $dist(f^{h}(i', j, a_{r}), s_{i}^{j}) = \min_{d \in \{1, 2, 3\}} (2 + i)$ $|P(\pi_{i'}^h, c_d)| + |P(s_i^j, c_d)|). \operatorname{dist}(f^h(i', j, a_r), y_1) = \min \left(\operatorname{dist}(f^h(i', j, a_r), s_i^j) + \ell_{y_1}, 2 + |P(\pi_{i'}^h, a_r)| + \frac{1}{2} + \frac$ $|P(s_i^j, a_r)| - \ell_{y_1}| \ge 2 + 10(n+1) \ge \text{dist}(f^h(i', j, a_r), x_1).$ $f^h(i', j, a_r)$ does not resolve $\{x_1, y_1\}$ only when $x_1 = a_{r'}$ and $y_1 = a_r$ with $r \neq r'$. The pair $\{a_{r'}, a_r\}$ is resolved by $f(\pi_{i'}^h, a_{r'})$. It

follows that every pair $\{x_1, y_1\}$ is resolved by $f^h(i', j, a_r)$ or $f(\pi_{i'}^h, a_{r'})$. In a similar way, we can show that every pair $\{x_2, y_1\}$, $\{x_1, y_3\}$ and $\{x_2, y_3\}$ are resolved by S'. For a vertex pair $\{x_1, y_2\}$, $dist(f^h(i', j, b_r), x_1) = 2 + \ell_{x_1}$. For the distance between $f^h(i', j, b_r)$ and y_2 , there are two cases. Case 1: i' = i. Then $dist(f^h(i', j, b_r), y_2) = \min(|P^h(i', j, b_r)| + \ell_{y_2} - 1, 2 + |P(\pi_i^h, a_r)| + |P(a_r, v_r^n)| + |P(b_r, v_r^n)| + |P(s_i^j, b_r)| - \ell_{y_2}) \ge 20(n + 1) > dist(f^h(i', j, b_r), x_1)$. Case 2: $i' \neq i$. Then $dist(f^h(i', j, b_r), s_i^j) = \min_{d \in \{1, 2, 3\}}(2 + |P(\pi_{i'}^h, c_d)| + |P(s_i^j, c_d)|)$. $dist(f^h(i', j, b_r), y_2) = \min(dist(f^h(i', j, b_r), s_i^j) + \ell_{y_2}, 2 + |P(\pi_{i'}^h, a_r)| + |P(a_r, v_r^n)| + |P(b_r, v_r^n)| + |P(s_i^j, b_r)| - \ell_{y_2}) > 20(n+1) > dist(f^h(i', j, b_r), x_1)$. Thus in both cases, every pair $\{x_1, y_2\}$ is resolved by $f^h(i', j, b_r)$. In a similar way, we can show that every pair $\{x_2, y_2\}$ is resolved by $f^h(i', j, b_r)$. This completes the proof for the lemma.

Lemma 70. Every pair $\{x, y\} \in \bigcup_{i \in [n]} S_i \times \bigcup_{r \in \{1,2,3\}} R_r$ is resolved by S'.

 $\begin{array}{l} Proof. \text{ We show that every pair } \{x,y\} \in (P(\pi_i^h,a_r) \cup P(\pi_i^h,c_r)) \times (P(u_{i'}^{i'},a_{r'}) \cup P(v_{i'}^{i'},a_{r'}) \cup P(u_{i''}^{i'},c_{r'}) \cup P(v_{i''}^{i'},c_{r'}) \cup P(v_{i''}^{i'},c_{r'})) \text{ for } i,i' \in [n], h \in \{1,2\} \text{ and } r,r' \in \{1,2,3\} \text{ is resolved by } S'. \text{ We fix arbitrary integers } i,i' \in [n], j \in [m], h \in \{1,2\} \text{ and } r,r' \in \{1,2,3\}. \text{ Suppose that } x_1 \in P(\pi_i^h,a_r), x_2 \in P(\pi_i^h,c_r), y_1 \in P(u_{r'}^{i'},a_{r'}), y_2 \in P(v_{i'}^{i'},a_{r'}), z_1 \in P(u_{r'}^{i'},b_{r'}), z_2 \in P(v_{i'}^{i'},b_{r'}), w_1 \in P(u_{r'}^h,c_{r'}), w_2 \in P(v_{i'}^{i'},c_{r'}). \text{ For a vertex } x_1 \in P(\pi_i^h,a_r), \text{ let } P(\pi_i^h,x_1) \text{ be the subpath of } P(\pi_i^h,a_r) \text{ from } \pi_i^h \text{ to } x_1 \text{ and let } |P(\pi_i^h,x_1)| = \ell_{x_1}. \text{ For a vertex } x_2 \in P(\pi_i^h,c_r), \text{ let } P(\pi_i^h,a_r) \text{ and } r,r' \in \{1,2,3\}. \text{ Then } \text{dist}(f^h(i,j,a_r),x_1) = 2 + \ell_{x_1} \leq 2 + 10(n+1) \text{ and } \text{dist}(f^h(i,j,a_r),x_2) = 2 + \ell_{x_2} \leq 2 + 10(n+1). \text{ dist}(f^h(i,j,a_r),a_{r'}) = \text{dist}(f^h(i,j,a_r),c_{r'}) = 2 + 10(n+1). \text{ dist}(f^h(i,j,a_r),b_{r'}) = 2 + |P(\pi_i^h,a_{r'})| + |P(a_{r'},v_{r'}^n)| + |P(b_{r'},v_{r'}^n)| > 2 + 10(n+1). \text{ We see that any shortest path from } f^h(i,j,a_r) \text{ to a vertex of } \{y_1,y_2,z_1,z_2,w_1,w_2\} \text{ goes through } a_{r'},b_{r'} \text{ or } c_{r'}. \text{ Thus the distance from } f^h(i,j,a_r) \text{ to any vertex of } \{y_1,y_2,z_1,z_2,w_1,w_2\} \text{ is at least } 2 + 10(n+1) \text{ and the equality holds only when } y_1 = y_2 = a_{r'} \text{ or } w_1 = w_2 = c_{r'}. \text{ Obviously } f(\pi_i^h,a_r) \text{ resolves the pairs } \{a_r,a_{r'}\} \text{ and } \{a_r,c_{r'}\} \text{ and } f(\pi_i^h,c_r) \text{ resolves the pairs } \{c_r,a_{r'}\} \text{ and } \{c_r,c_{r'}\} \text{ with } r \neq r'. \text{ As a result, every vertex pair of } \bigcup_{i \in [n]} S_i \times \bigcup_{r \in \{1,2,3\}} R_r \text{ is resolved by } f^h(i,j,a_r), f(\pi_i^h,a_r) \text{ or } f(\pi_i^h,c_r). \text{ This completes the proof for the lemma. \square$

Lemma 71. Every pair $\{x, y\} \in \bigcup_{i \in [n]} H_i \times \bigcup_{r \in \{1,2,3\}} R_r$ is resolved by S'.

 $\begin{array}{l} Proof. \mbox{ First we show that every pair } \{x,y\} \in P(s_i^j,a_r) \times (P(u_{i'}^{i'},a_{r'}) \cup P(v_{r'}^{i'},a_{r'}) \cup P(u_{r'}^{i'},b_{r'}) \cup P(v_{r'}^{i'},c_{r'}) \cup P(v_{r'}^{i'},c_{r'})) \mbox{ for } i,i' \in [n],j \in [m] \mbox{ and } r,r' \in \{1,2,3\} \mbox{ is resolved by } S'. \mbox{ We fix arbitrary integers } i,i' \in [n],h \in \{1,2\},j \in [m] \mbox{ and } r,r' \in \{1,2,3\}. \mbox{ Suppose that } x \in P(s_i^j,a_r),y_1 \in P(u_{r'}^{i'},a_{r'}),y_2 \in P(v_{r'}^{i'},a_{r'}),z_1 \in P(u_{r'}^{i'},b_{r'}),z_2 \in P(v_{r'}^{i'},b_{r'}),w_1 \in P(u_{r'}^{i'},c_{r'}),w_2 \in P(v_{r'}^{i'},c_{r'}). \mbox{ For a vertex } x \in P(s_i^j,a_r), \mbox{ be the subpath of } P(s_i^j,a_r) \mbox{ for } a_{r'},a_{r'}) \mbox{ for } y_1 \in P(u_{r'}^{i'},a_{r'}), \mbox{ be the subpath of } P(s_i^j,a_r) \mbox{ for } a_{r'},a_{r'}) \mbox{ for } y_1^{i'},a_{r'}) \mbox{ for } y_1^{i'},a_{r'}) \mbox{ for } y_1^{i'},a_{r'}, \mbox{ for } y_1^{i'},a_{r'}) \mbox{ for } y_1^{i'},a_{r'}, \mbox{ for } y_1^{i'},a_{r'}) \mbox{ for } y_1^{i'},a_{r'}, \mbox \mbox{$

 $|P(\pi_i^h, a_r)| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| - \ell_{y_1}$. For a vertex pair $\{x, y_1\}$ that is not resolved by $f(s_i^j, a_r)$, $dist(f(\pi_i^h, a_r), x) > dist(f(s_i^j, a_r), x) = dist(f(s_i^j, a_r), y_1) > dist(f(\pi_i^h, a_r), y_1)$. Thus in this case, every pair $\{x, y_1\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$. Similarly, every pair $\{x, y_2\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$. For a vertex pair $\{x, z_1\}$, there are two cases. Case 1: r' = r. dist $(f(s_i^j, a_r), z_1) = dist(f(\pi_i^h, a_r), z_1) = min (2 + |P(u_{r'}^{i'}, a_{r'})| + \ell_{z_1}, 2 + \ell_{z_1},$ $|P(u_{r'}^n, a_{r'})| + |P(u_{r'}^n, b_{r'})| + |P(u_{r'}^{i'}, b_{r'})| - \ell_{z_1}).$ For a vertex pair $\{x, z_1\}$ that is not resolved by $f(s_i^j, a_r)$, $dist(f(s_i^j, a_r), x) = dist(f(s_i^j, a_r), z_1) = dist(f(\pi_i^h, a_r), z_1) < dist(f(\pi_i^h, a_r), x)$. Thus in this case, every pair $\{x, z_1\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$. Case 2: $r' \neq i$ r. dist $(f(\pi_i^h, a_r), b_{r'}) = \min_{\alpha \in [m]} (2 + |P(s_i^\alpha, a_r)| + |P(s_i^\alpha, b_{r'})|)$. Then dist $(f(\pi_i^h, a_r), z_1) = (1 + |P(s_i^\alpha, a_r)| + |P(s_i^\alpha, b_{r'})|)$. $\min\left(|P(\pi_i^h, a_r)| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| + \ell_{z_1}, \operatorname{dist}(f(\pi_i^h, a_r), b_{r'}) + |P(u_{r'}^{i'}, b_{r'})| - \ell_{z_1}\right) > 0$ $30(n+1) > \text{dist}(f(\pi_i^h, a_r), x)$. Thus in this case, every pair $\{x, z_1\}$ is resolved by $f(\pi_i^h, a_r)$. Similarly, every pair $\{x, z_2\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$. For a vertex pair $\{x, w_1\}$, there are two cases. Case 1: r' = r. dist $(f(s_i^j, a_r), w_1) = \min(2 + |P(u_r^{i'}, a_{r'})| - 2 + \ell_{w_1}, 2 + \ell_{w_1}$ $|P(\pi_i^h, a_r)| + |P(\pi_i^h, c_r)| + |P(u_r^{i'}, c_r)| - \ell_{w_1}) \text{ if } \ell_{w_1} \ge 2. \text{ dist}(f(s_i^j, a_r), w_1) = 2 + |P(u_r^{i'}, a_{r'})| + \ell_{w_1}$ if $\ell_{w_1} < 2$. dist $(f(\pi_i^h, a_r), w_1) = \min(2 + |P(u_r^{i'}, a_{r'})| - 2 + \ell_{w_1}, |P(\pi_i^h, a_r)| + |P(\pi_i^h, c_r)| + |P(\pi_i^h, c_$ $|P(u_r^{i'}, c_r)| - \ell_{w_1})$ if $\ell_{w_1} \geq 2$. dist $(f(\pi_i^h, a_r), w_1) = 2 + |P(u_r^{i'}, a_{r'})| + \ell_{w_1}$ if $\ell_{w_1} < 2$. It follows that dist $(f(s_i^j, a_r), w_1) \ge \text{dist}(f(\pi_i^h, a_r), w_1)$. For a pair $\{x, w_1\}$ that is not resolved by $f(s_i^j, a_r)$, $dist(f(\pi_i^h, a_r), x) > dist(f(s_i^j, a_r), x) = dist(f(s_i^j, a_r), w_1) \ge dist(f(\pi_i^h, a_r), w_1)$. Thus in this case, every pair $\{x, w_1\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$. Case 2: $r' \neq r$. $\operatorname{dist}(f(s_i^j, a_r), w_1) = \min \left(2 + |P(\pi_i^h, a_r)| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| - 2 + \ell_{w_1}, 2 + |P(\pi_i^h, a_r)| + |P(\pi_i^h, a_{r'})| + |P(\pi_i^h, a_{r'$ $\begin{aligned} |P(\pi_i^h, c_{r'})| + |P(u_{r'}^{i'}, c_{r'})| - \ell_{w_1}) & \text{if } \ell_{w_1} \ge 2. \ \operatorname{dist}(f(s_i^j, a_r), w_1) = 2 + |P(\pi_i^h, a_r)| + |P(\pi_i^h, a_{r'})| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| + \ell_{w_1} & \text{if } \ell_{w_1} < 2. \ \operatorname{dist}(f(\pi_i^h, a_r), w_1) = \min (|P(\pi_i^h, a_r)| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| + |P(u_{r'$ $\ell_{w_1} - 2, |P(\pi_i^h, a_r)| + |P(\pi_i^h, c_{r'})| + |P(u_{r'}^{i'}, c_{r'})| - \ell_{w_1}) \text{ if } \ell_{w_1} \ge 2. \text{ dist}(f(\pi_i^h, a_r), w_1) = |P(\pi_i^h, a_r)| + |P(\pi_i^h, a_r)$ $|P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| + \ell_{w_1} \text{ if } \ell_{w_1} < 2. \text{ It follows that } \operatorname{dist}(f(s_i^j, a_r), w_1) > \operatorname{dist}(f(\pi_i^h, a_r), w_1).$ For a pair $\{x, w_1\}$ that is not resolved by $f(s_i^j, a_r)$, $\operatorname{dist}(f(\pi_i^h, a_r), x) > \operatorname{dist}(f(s_i^j, a_r), x) =$ $\operatorname{dist}(f(s_i^j, a_r), w_1) > \operatorname{dist}(f(\pi_i^h, a_r), w_1)$. Thus in this case, every pair $\{x, w_1\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$. Similarly, every pair $\{x, w_2\}$ is resolved by $f(s_i^j, a_r)$ or $f(\pi_i^h, a_r)$.

Then we show that every pair $\{x, y\} \in P(s_i^j, c_r) \times (P(u_{r'}^{i'}, a_{r'}) \cup P(v_{r'}^{i'}, a_{r'}) \cup P(u_{r'}^{i'}, b_{r'}) \cup$ $P(v_{r'}^{i'}, b_{r'}) \cup P(u_{r'}^{i'}, c_{r'}) \cup P(v_{r'}^{i'}, c_{r'}))$ for $i, i' \in [n], j \in [m]$ and $r, r' \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], h \in \{1, 2\}, j \in [m]$ and $r, r' \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, c_r), y_1 \in P(u_{r'}^{i'}, a_{r'}), y_2 \in P(v_{r'}^{i'}, a_{r'}), z_1 \in P(u_{r'}^{i'}, b_{r'}), z_2 \in P(v_{r'}^{i'}, b_{r'}), z_1 \in$ $w_1 \in P(u_{r'}^{i'}, c_{r'}), w_2 \in P(v_{r'}^{i'}, c_{r'}).$ We define $\ell_x, \ell_{y_1}, \ell_{y_2}, \ell_{z_1}, \ell_{z_2}, \ell_{w_1}$ and ℓ_{w_2} in a similar way to that of ℓ_x, ℓ_{y_1} in the first paragraph. Then $\operatorname{dist}(f(s_i^j, c_r), x) = \operatorname{dist}(f(\pi_i^h, c_r), x) - 2 = \ell_x$ if $x \neq c_r$ and dist $(f(s_i^j, c_r), c_r) = \text{dist}(f(\pi_i^h, c_r), c_r) = 2$. For a pair $\{x, y_1\}$, there are two cases. Case 1: r' = r. dist $(f(s_i^j, c_r), y_1) = \min(2 + |P(u_r^{i'}, c_{r'})| - 2 + \ell_{y_1}, 2 + |P(\pi_i^h, c_r)| + \ell_{y_1}, 2 + |P(\pi_i^h, c_r)| + \ell_{y_1}, 2 + \ell_{y_1}, 2$ $|P(\pi_i^h, a_r)| + |P(u_r^{i'}, a_r)| - \ell_{y_1}) \text{ if } \ell_{y_1} \ge 2. \quad \text{dist}(f(s_i^j, c_r), y_1) = 2 + |P(u_r^{i'}, c_{r'})| + \ell_{y_1} \text{ if } \ell_{y_1} < 2.$ $\operatorname{dist}(f(\pi_i^h, c_r), y_1) = \min (2 + |P(u_r^{i'}, c_{r'})| - 2 + \ell_{y_1}, |P(\pi_i^h, c_r)| + |P(\pi_i^h, a_r)| + |P(u_r^{i'}, a_r)| - \ell_{y_1}) \text{ if }$ $\ell_{y_1} \ge 2. \operatorname{dist}(f(\pi_i^h, c_r), y_1) = 2 + |P(u_r^{i'}, c_{r'})| + \ell_{y_1} \text{ if } \ell_{y_1} < 2.$ It follows that $\operatorname{dist}(f(s_i^j, c_r), y_1) \ge 2$ $\operatorname{dist}(f(\pi_i^h, c_r), y_1)$. For a pair $\{x, y_1\}$ that is not resolved by $f(s_i^j, c_r)$, $\operatorname{dist}(f(\pi_i^h, c_r), x) > 0$ $\operatorname{dist}(f(s_i^j, c_r), x) = \operatorname{dist}(f(s_i^j, c_r), y_1) \ge \operatorname{dist}(f(\pi_i^h, c_r), y_1)$. Thus in this case, every pair $\{x, y_1\}$ is resolved by $f(s_i^j, c_r)$ or $f(\pi_i^h, c_r)$. Case 2: $r' \neq r$. dist $(f(s_i^j, c_r), y_1) = \text{dist}(f(\pi_i^h, c_r), y_1) + 2 = 2 + 2$ $|P(\pi_i^h, c_r)| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| - \ell_{y_1}.$ It follows that $\operatorname{dist}(f(s_i^j, c_r), y_1) > \operatorname{dist}(f(\pi_i^h, c_r), y_1).$ For a pair $\{x, y_1\}$ that is not resolved by $f(s_i^j, c_r)$, $\operatorname{dist}(f(\pi_i^h, c_r), x) > \operatorname{dist}(f(s_i^j, c_r), x) =$ $\operatorname{dist}(f(s_i^j, c_r), y_1) > \operatorname{dist}(f(\pi_i^h, c_r), y_1)$. Thus in this case, every pair $\{x, y_1\}$ is resolved by $f(s_i^j, c_r)$ or $f(\pi_i^h, c_r)$. Similarly, every pair $\{x, y_2\}$ is resolved by $f(s_i^j, c_r)$ or $f(\pi_i^h, c_r)$. For

a pair $\{x, z_1\}$, there are two cases. Case 1: r' = r. Let $s_i^{j^*}$ be a vertex which resolves the pair $\{u_r^n, v_r^n\}$. Then $|P(s_i^{j^*}, c_r)| + |P(s_i^{j^*}, b_r)| = 40(n + 1) - 5n + 1$. dist $(f(\pi_i^h, c_r), b_r) = 2 + |P(s_i^{j^*}, c_r)| + |P(s_i^{j^*}, b_r)| = 40(n + 1) - 5n + 3$. dist $(f(\pi_i^h, c_r), z_1) = \min(2 + |P(u_r^{i'}, c_{r'})| + \ell_{z_1}, \text{dist}(f(\pi_i^h, c_r), b_r) + |P(u_r^{i'}, b_r)| - \ell_{z_1}) > 20(n + 1) > \text{dist}(f(\pi_i^h, c_r), x)$. Thus in this case, every pair $\{x, z_1\}$ is resolved by $f(\pi_i^h, c_r)$. Case 2: $r' \neq r$. dist $(f(\pi_i^h, c_r), b_{r'}) = \min_{\alpha \in [m]}(2 + |P(s_i^{\alpha}, c_r)| + |P(s_i^{\alpha}, b_{r'})|)$. Then dist $(f(\pi_i^h, c_r), z_1) = \min(|P(\pi_i^h, c_r)| + |P(\pi_i^h, a_{r'})| + |P(u_{r'}^{i'}, a_{r'})| + \ell_{z_1}, \text{dist}(f(\pi_i^h, c_r), b_{r'}) + |P(u_{r'}^{i'}, b_{r'})| - \ell_{z_1}) > 20(n + 1) > \text{dist}(f(\pi_i^h, c_r), x)$. Thus in this case, every pair $\{x, z_1\}$ is resolved by $f(\pi_i^h, c_r)$. Similarly, every pair $\{x, z_2\}$ is resolved by $f(\pi_i^h, c_r)$. For a pair $\{x, w_1\}$, there are two cases. Case 1: r' = r. dist $(f(s_i^j, c_r), w_1) = \text{dist}(f(\pi_i^h, c_r), w_1) = 2 + |P(u_{r'}^{i'}, c_{r'})| - \ell_{w_1}$. For a vertex pair $\{x, w_1\}$ that is not resolved by $f(s_i^j, c_r)$, $dist(f(\pi_i^h, c_r), w_1) = 2 + |P(u_{r'}^{i'}, c_{r'})| - \ell_{w_1}$. For a vertex pair $\{x, w_1\}$ that is not resolved by $f(s_i^j, c_r), (dist(f(s_i^j, c_r), x)) = \text{dist}(f(s_i^j, c_r), w_1) = \text{dist}(f(\pi_i^h, c_r), w_1) < 2 + |P(\pi_i^h, c_r)| + |P(\pi_i^h, c_{r'})| - \ell_{w_1}$. For a vertex pair $\{x, w_1\}$ that is not resolved by $f(s_i^j, c_r), (dist(f(\pi_i^h, c_r), w_1) = min (|P(\pi_i^h, c_r)| + |P(\pi_i^h, c_{r'})| - \ell_{w_1}$. Thus in this case, every pair $\{x, w_1\}$ is resolved by $f(s_i^j, c_r)$ or $f(\pi_i^h, c_r) + |P(\pi_i^h, a_{r'})| + |P(\pi_i^h, a_{r$

Finally we show that every pair $\{x, y\} \in P(s_i^j, b_r) \times (P(u_{r'}^{i'}, a_{r'}) \cup P(v_{r'}^{i'}, a_{r'}) \cup P(u_{r'}^{i'}, b_{r'}) \cup P(u_{r'}^{i$ $P(v_{r'}^{i'}, b_{r'}) \cup P(u_{r'}^{i'}, c_{r'}) \cup P(v_{r'}^{i'}, c_{r'}))$ for $i, i' \in [n], j \in [m]$ and $r, r' \in \{1, 2, 3\}$ is resolved by S'. We fix arbitrary integers $i, i' \in [n], h \in \{1, 2\}, j \in [m]$ and $r, r' \in \{1, 2, 3\}$. Suppose that $x \in P(s_i^j, b_r), y_1 \in P(u_{r'}^{i'}, a_{r'}), y_2 \in P(v_{r'}^{i'}, a_{r'}), z_1 \in P(u_{r'}^{i'}, b_{r'}), z_2 \in P(v_{r'}^{i'}, b_{r'}), z_1 \in$ $w_1 \in P(u_{r'}^{i'}, c_{r'}), w_2 \in P(v_{r'}^{i'}, c_{r'}).$ We define $\ell_x, \ell_{y_1}, \ell_{y_2}, \ell_{z_1}, \ell_{z_2}, \ell_{w_1}$ and ℓ_{w_2} in a similar way to that of ℓ_x, ℓ_{y_1} in the first paragraph. For a pair $\{x, y_1\}$, there are two cases. Case 1: r = r'. $dist(f(\pi_i^h, a_r), x) = \min (2 + |P(s_i^j, a_r)| + |P(s_i^j, b_r)| - \ell_x, 2 + |P(a_r, u_r^n)| + |P(b_r, u_r^n)| + \ell_x) > 0$ 20(n+1). dist $(f(\pi_i^h, a_r), y_1) = |P(a_r, u_r^{i'})| - \ell_{y_1} < 20(n+1)$. Thus in this case, every pair $\{x, y_1\}$ is resolved by $f(\pi_i^h, a_r)$. Similarly, every pair $\{x, y_2\}$ is resolved by $f(\pi_i^h, a_r)$. Case 2: $r \neq r'$. dist $(f^1(u_{r'}^{i'}, v_{r'}^{i'}), y_1) = 1 + \ell_{y_1}$ if $y_1 \neq u_{r'}^{i'}$ and dist $(f^1(u_{r'}^{i'}, v_{r'}^{i'}), u_{r'}^{i'}) = 2$. $\operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), x) = \min\left(\operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), s_{i}^{j}) + |P(s_{i}^{j}, b_{r})| - \ell_{x}, \operatorname{dist}(f^{1}(u_{r'}^{i'}, v_{r'}^{i'}), b_{r}) + \ell_{x}\right) >$ $20(n+1) > \text{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), y_1)$. Thus in this case, every pair $\{x, y_1\}$ is resolved by $f^1(u_{r'}^{i'}, v_{r'}^{i'})$. Similarly, every pair $\{x, y_2\}$ is resolved by $f^1(u_{r'}^{i'}, v_{r'}^{i'})$. For a pair $\{x, z_1\}$, there are two cases. Suppose that $P(s_i^j, b_r) = 20(n+1) + 5\lambda + 1$ for some $\lambda \in [n]$. Case 1: r = r'. dist $(f^{mid}(i, j, h), x) =$ $|P^{h}(i, j, p_{i}^{3-h})|/2 + 2 + |P(s_{i}^{j}, b_{r})| - \ell_{x} = 30(n+1) + 5\lambda + 3 - \ell_{x}. \operatorname{dist}(f^{mid}(i, j, h), z_{1}) = \min(2 + 1) + 2\lambda + 3 - \ell_{x}.$ $|P^{h}(i,j,p_{i}^{3-h})|/2 + |P(s_{i}^{j},b_{r})| + |P(b_{r},u_{r}^{i'})| - \ell_{z_{1}}, 1 + |P^{h}(i,j,p_{i}^{3-h})|/2 + |P(\pi_{i}^{h},a_{r})| + |P(a_{r},u_{r}^{i'})| + |P(a_$ $\ell_{z_1} = \min (50(n+1) + 5\lambda + 2 - 5i' - \ell_{z_1}, 40(n+1) + 1 - 10i' + \ell_{z_1}). \operatorname{dist}(f^{ecc}(i, j, h, r), x) = 0$ $|P^{h}(i, j, a_{r})|/2 + 1 + |P(s_{i}^{j}, b_{r})| - \ell_{x} = \operatorname{dist}(f^{mid}(i, j, h), x) - 1. \operatorname{dist}(f^{ecc}(i, j, h, r), z_{1}) = \min(1 + i)$ $|P^{h}(i,j,a_{r})|/2 + |P(s_{i}^{j},b_{r})| + |P(b_{r},u_{r}^{i'})| - \ell_{z_{1}}, 2 + |P^{h}(i,j,a_{r})|/2 + |P(\pi_{i}^{h},a_{r})| + |P(a_{r},u_{r}^{i'})| + \ell_{z_{1}}) = 0$ min $(50(n+1) + 5\lambda + 1 - 5i' - \ell_{z_1}, 40(n+1) + 2 - 10i' + \ell_{z_1})$. For a vertex pair $\{x, z_1\}$ that is not resolved by $f^{mid}(i, j, h)$, $dist(f^{mid}(i, j, h), x) = dist(f^{mid}(i, j, h), z_1) = 1 + |P^h(i, j, p_i^{3-h})|/2 + 1$ $|P(\pi_i^h, a_r)| + |P(a_r, u_r^{i'})| + \ell_{z_1} > \operatorname{dist}(f^{ecc}(i, j, h, r), x).$ If $\operatorname{dist}(f^{ecc}(i, j, h, r), z_1) = 1 + |P^h(i, j, a_r)|/2$ $|P(s_i^j, b_r)| + |P(b_r, u_r^{i'})| - \ell_{z_1}$, then obviously $f^{ecc}(i, j, h, r)$ resolves this pair. Otherwise, $\operatorname{dist}(f^{ecc}(i,j,h,r),z_1) = 40(n+1) + 2 - 10i' + \ell_{z_1} > \operatorname{dist}(f^{mid}(i,j,h),z_1) > \operatorname{dist}(f^{ecc}(i,j,h,r),x).$ It follows that every pair $\{x, z_1\}$ is resolved by $f^{mid}(i, j, h)$ or $f^{ecc}(i, j, h, r)$. Similarly, every pair $\{x, z_2\}$ is resolved by $f^{mid}(i, j, h)$ or $f^{ecc}(i, j, h, r)$. Case 2: $r \neq r'$. dist $(f^1(u_{r'}^{i'}, v_{r'}^{i'}), z_1) =$ $2 + \ell_{z_1} < 20(n+1)$. dist $(f^1(u_{r'}^{i'}, v_{r'}^{i'}), x) = \min(\operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), s_i^j) + |P(s_i^j, b_r)| - \ell_x,$ $dist(f^1(u_{r'}^{i'}, v_{r'}^{i'}), b_r) + \ell_x) > 20(n+1) > dist(f^1(u_{r'}^{i'}, v_{r'}^{i'}), z_1)$. Thus in this case, every pair $\{x, z_1\}$ is resolved by $f^1(u_{r'}^{i'}, v_{r'}^{i'})$. Similarly, every pair $\{x, z_2\}$ is resolved by $f^1(u_{r'}^{i'}, v_{r'}^{i'})$. For a pair $\{x, w_1\}$, there are two cases. Case 1: r = r'. dist $(f(\pi_i^h, c_r), b_r) = \min_{\alpha \in [m]} (2 + |P(s_i^\alpha, c_r)| + |P(s_i^\alpha, c_r)|)$ $|P(s_i^{\alpha}, b_r)| = 3 + 40(n+1) - 5n > 30(n+1)$. dist $(f(\pi_i^h, c_r), x) = \min(2 + |P(s_i^j, c_r)| + |P(s_i^j, b_r)| - 1)$

$$\begin{split} \ell_x, \operatorname{dist}(f(\pi_i^h, c_r), b_r) + \ell_x). \ \operatorname{dist}(f(s_i^j, c_r), x) &= \min \left(|P(s_i^j, c_r)| + |P(s_i^j, b_r)| - \ell_x, \operatorname{dist}(f(\pi_i^h, c_r), b_r) + \ell_x \right). \ \operatorname{dist}(f(\pi_i^h, c_r), w_1) &= \operatorname{dist}(f(s_i^j, c_r), w_1) = 2 + |P(c_r, u_{r'}^{i'})| - \ell_{w_1} = 2 + 20(n+1) + 10i' - \ell_{w_1} < 30(n+1). \ \text{ For a pair } \{x, w_1\} \ \text{ that is not resolved by } f(\pi_i^h, c_r), \ \operatorname{dist}(f(\pi_i^h, c_r), x) = \\ \operatorname{dist}(f(\pi_i^h, c_r), w_1) &= 2 + |P(s_i^j, c_r)| + |P(s_i^j, b_r)| - \ell_x = \operatorname{dist}(f(s_i^j, c_r), w_1) > \operatorname{dist}(f(s_i^j, c_r), x) = \\ |P(s_i^j, c_r)| + |P(s_i^j, b_r)| - \ell_x. \ \text{ Thus in this case, every pair } \{x, w_1\} \ \text{ is resolved by } f(\pi_i^h, c_r), u_{r'}^{i'}) = 2. \\ \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), x) &= \min \left(\operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), s_i^j) + |P(s_i^j, b_r)| - \ell_x, \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), u_{r'}^{i'}) = 2. \\ \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), x) &= \min \left(\operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), s_i^j) + |P(s_i^j, b_r)| - \ell_x, \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), u_{r'}^{i'}) = 2. \\ \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), x) &= \min \left(\operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), s_i^j) + |P(s_i^j, b_r)| - \ell_x, \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), b_r) + \ell_x) > \\ \operatorname{30}(n+1) &> \operatorname{dist}(f^1(u_{r'}^{i'}, v_{r'}^{i'}), w_1). \ \text{Thus in this case, every pair } \{x, w_1\} \ \text{is resolved by } f^1(u_{r'}^{i'}, v_{r'}^{i'}). \\ \operatorname{Similarly, every pair } \{x, w_2\} \ \text{is resolved by } f^1(u_{r'}^{i'}, v_{r'}^{i'}). \ \text{This completes the proof for the lemma.} \\ \Box \end{array}$$

Lemma 72. For any vertex $v_f \in \mathcal{F}$, every vertex pair $\{x, y\} \in \{v_f\} \times V(G') \setminus \{v_f\}$ is resolved by S'.

Proof. Without loss of generality, suppose that $v_1, v_2, v_c \in F^1(u_r^i, v_r^i)$ for some $r \in \{1, 2, 3\}, i \in [n]$, where v_c is the connecting vertex of $F^1(u_r^i, v_r^i)$, v_1, v_2 are the false twins and $v_1 \in S'$. Then obviously every vertex pair of $\{v_1\} \times V(G') \setminus \{v_1\}$ is resolved by v_1 . Every vertex pair of $\{v_2\} \times V(G') \setminus \{v_2\}$ is resolved v_1 except the vertex pair $\{v_2, v_c\}$. Let w_f be an arbitrary vertex of $S' \setminus F^1(u_r^i, v_r^i)$. Then there is a shortest path from w_f to v_2 going through v_c . Thus $\operatorname{dist}(w_f, v_2) = \operatorname{dist}(w_f, v_c) + 1$ and $\{v_2, v_c\}$ is resolved by w_f . For any vertex $u \in V(G') \setminus F^1(u_r^i, v_r^i)$, $\operatorname{dist}(v_1, u) > \operatorname{dist}(v_1, v_c) = 1$. Then the correctness of the lemma follows. \Box

With Lemmas 51-72, we show that every pair of distinct vertices of G' is resolved by some vertex of S'. It follows that Lemma 50 is true and this proves the completeness of the reduction. Finally, with Lemmas 44, 48, 50 and 49 in hand, we can prove the correctness of Theorem 8.

Bibliography

- [1] Encyclopedia of Optimization, Second Edition. Springer, 2009.
- [2] Amit Agarwal, Noga Alon, and Moses Charikar. Improved approximation for directed cut problems. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium* on Theory of Computing, San Diego, California, USA, June 11-13, 2007, pages 671–680. ACM, 2007.
- [3] Akanksha Agrawal, Sushmita Gupta, Saket Saurabh, and Roohani Sharma. Improved algorithms and combinatorial bounds for independent feedback vertex set. In 11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24-26, 2016, Aarhus, Denmark, volume 63 of LIPIcs, pages 2:1–2:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
- [4] Takuya Akiba and Yoichi Iwata. Branch-and-reduce exponential/FPT algorithms in practice: A case study of vertex cover. *Theoretical Computer Science*, 609:211–225, 2016.
- [5] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.
- [6] Thomas Bellitto, Shaohua Li, Karolina Okrasa, Marcin Pilipczuk, and Manuel Sorge. The complexity of connectivity problems in forbidden-transition graphs and edge-colored graphs. In Yixin Cao, Siu-Wing Cheng, and Minming Li, editors, 31st International Symposium on Algorithms and Computation, ISAAC 2020, December 14-18, 2020, Hong Kong, China (Virtual Conference), volume 181 of LIPIcs, pages 59:1–59:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- [7] Rémy Belmonte, Fedor V. Fomin, Petr A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. SIAM J. Discret. Math., 31(2):1217–1243, 2017.
- [8] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. Journal of Computational Biology, 6(3/4):281–297, 1999.
- [9] Kristóf Bérczi and Yusuke Kobayashi. The directed disjoint shortest paths problem. In Kirk Pruhs and Christian Sohler, editors, 25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria, volume 87 of LIPIcs, pages 13:1–13:13. Schloss Dagstuhl -Leibniz-Zentrum für Informatik, 2017.
- [10] S. Böcker, S. Briesemeister, Q.B.A. Bui, and A. Truss. Going weighted: Parameterized algorithms for cluster editing. *Theoretical Computer Science*, 410(52):5467–5480, 2009.
- [11] Sebastian Böcker. A golden ratio parameterized algorithm for cluster editing. Journal of Discrete Algorithms, 16:79–89, 2012.
- [12] Sebastian Böcker and Jan Baumbach. Cluster editing. In Paola Bonizzoni, Vasco Brattka, and Benedikt Löwe, editors, Proceedings of the 9th Conference on Computability in Europe (CiE 2013), volume 7921 of Lecture Notes in Computer Science, pages 33–44. Springer, 2013.
- [13] Sebastian Böcker, Sebastian Briesemeister, and Gunnar W. Klau. Exact algorithms for cluster editing: Evaluation and experiments. Algorithmica, 60(2):316–334, 2011.
- [14] Sebastian Böcker and Peter Damaschke. Even faster parameterized cluster deletion and cluster editing. Information Processing Letters, 111(14):717-721, 2011.
- [15] Hans L. Bodlaender. On disjoint cycles. Int. J. Found. Comput. Sci., 5(1):59-68, 1994.
- [16] Hans L. Bodlaender, Michael R. Fellows, Pinar Heggernes, Federico Mancini, Charis Papadopoulos, and Frances A. Rosamond. Clustering with partial information. *Theoretical Computer Science*, 411(7-9):1202–1211, 2010.
- [17] Édouard Bonnet and Nidhi Purohit. Metric dimension parameterized by treewidth. In Bart M. P. Jansen and Jan Arne Telle, editors, 14th International Symposium on Parameterized and Exact

Computation, IPEC 2019, September 11-13, 2019, Munich, Germany, volume 148 of LIPIcs, pages 5:1–5:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

- [18] Yixin Cao and Jianer Chen. Cluster editing: Kernelization based on edge cuts. Algorithmica, 64(1):152–169, 2012.
- [19] Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set new measure and new structures. In Algorithm Theory - SWAT 2010, 12th Scandinavian Symposium and Workshops on Algorithm Theory, Bergen, Norway, June 21-23, 2010. Proceedings, volume 6139 of Lecture Notes in Computer Science, pages 93–104. Springer, 2010.
- [20] Chandra Chekuri, Sudipto Guha, and Joseph Naor. The steiner k-cut problem. SIAM J. Discrete Math., 20(1):261–271, 2006.
- [21] Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for the feedback vertex set problems. In Algorithms and Data Structures, 10th International Workshop, WADS 2007, Halifax, Canada, August 15-17, 2007, Proceedings, volume 4619 of Lecture Notes in Computer Science, pages 422–433. Springer, 2007.
- [22] Jianer Chen and Iyad A. Kanj. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. J. Comput. Syst. Sci., 67(4):833–847, 2003.
- [23] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM, 55(5):21:1–21:19, 2008.
- [24] Jianer Chen and Jie Meng. A 2k kernel for the cluster editing problem. Journal of Computer and System Sciences, 78(1):211–220, 2012.
- [25] Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. SIAM J. Comput., 45(4):1171–1229, 2016.
- [26] Rajesh Chitnis, László Egri, and Dániel Marx. List h-coloring a graph by removing few vertices. Algorithmica, 78(1):110–146, 2017.
- [27] Rajesh Hemant Chitnis, Marek Cygan, Mohammad Taghi Hajiaghayi, and Dániel Marx. Directed subset feedback vertex set is fixed-parameter tractable. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 230–241. Springer, 2012.
- [28] Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. SIAM J. Comput., 42(4):1674–1696, 2013.
- [29] Maria Chudnovsky, Alex Scott, and Paul Seymour. Disjoint paths in tournaments. Advances in Mathematics, 270:582–597, 2015.
- [30] Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971.
- [31] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [32] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [33] Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. The planar directed kvertex-disjoint paths problem is fixed-parameter tractable. In 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA, pages 197–206. IEEE Computer Society, 2013.
- [34] Marek Cygan, Marcin Pilipczuk, and Michal Pilipczuk. On group feedback vertex set parameterized by the size of the cutset. *Algorithmica*, 74(2):630–642, 2016.
- [35] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013.
- [36] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. ACM Transactions on Computation Theory, 5(1):3:1–3:11, 2013.
- [37] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feed-

back vertex set is fixed-parameter tractable. SIAM J. Discrete Math., 27(1):290–309, 2013.

- [38] Peter Damaschke. Fixed-parameter enumerability of cluster editing and related problems. *Theory* of Computing Systems, 46(2):261–283, 2010.
- [39] Josep Díaz, Olli Pottonen, Maria J. Serna, and Erik Jan van Leeuwen. Complexity of metric dimension on planar graphs. J. Comput. Syst. Sci., 83(1):132–158, 2017.
- [40] Rodney G. Downey and Michael R. Fellows. Fixed parameter tractability and completeness. In Complexity Theory: Current Research, Dagstuhl Workshop, February 2-8, 1992, pages 191–225. Cambridge University Press, 1992.
- [41] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.
- [42] Rodney G. Downey and Michael R. Fellows. Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, 2013.
- [43] Tali Eilam-Tzoreff. The disjoint shortest paths problem. Discret. Appl. Math., 85(2):113–138, 1998.
- [44] David Eppstein. Metric dimension parameterized by max leaf number. J. Graph Algorithms Appl., 19(1):313–323, 2015.
- [45] Leah Epstein, Asaf Levin, and Gerhard J. Woeginger. The (weighted) metric dimension of graphs: Hard and easy cases. Algorithmica, 72(4):1130–1171, 2015.
- [46] Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
- [47] Michael R. Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. Graph-based data clustering with overlaps. *Discrete Optimization*, 8(1):2–17, 2011.
- [48] Henning Fernau, Pinar Heggernes, Pim van 't Hof, Daniel Meister, and Reza Saei. Computing the metric dimension for chain graphs. Inf. Process. Lett., 115(9):671–676, 2015.
- [49] Jörg Flum and Martin Grohe. Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.
- [50] Fedor V. Fomin and Dieter Kratsch. Exact Exponential Algorithms. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.
- [51] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Yngve Villanger. Subexponential fixed-parameter tractability of cluster editing. CoRR, abs/1112.4419, 2011.
- [52] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Kernelization: theory of parameterized preprocessing. Cambridge University Press, 2019.
- [53] Steven Fortune, John Hopcroft, and James Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980.
- [54] Vincent Froese. Fine-Grained Complexity Analysis of Some Combinatorial Data Science Problems. PhD thesis, Technische Universität Berlin, 2018.
- [55] M. R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.
- [56] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. SIAM J. Comput., 25(2):235–251, 1996.
- [57] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. J. Algorithms, 50(1):49–61, 2004.
- [58] Shivam Garg and Geevarghese Philip. Raising the bar for vertex cover: Fixed-parameter tractability above A higher guarantee. In Krauthgamer [91], pages 1152–1166.
- [59] Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory of Computing Systems*, 38(4):373–392, 2005.
- [60] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. Discrete Optimization, 8(1):61–71, 2011.
- [61] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete* Optimization, 8(1):61–71, 2011.
- [62] Jiong Guo, Jens Gramm, Falk Hüffner, Rolf Niedermeier, and Sebastian Wernicke. Compressionbased fixed-parameter algorithms for feedback vertex set and edge bipartization. J. Comput. Syst. Sci., 72(8):1386–1396, 2006.

- [63] Jiong Guo, Iyad A. Kanj, Christian Komusiewicz, and Johannes Uhlmann. Editing graphs into disjoint unions of dense clusters. *Algorithmica*, 61(4):949–970, 2011.
- [64] Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. A more relaxed model for graph-based data clustering: s-plex cluster editing. SIAM Journal on Discrete Mathematics, 24(4):1662–1683, 2010.
- [65] Frank Harary and Robert A Melter. On the metric dimension of a graph. Ars Combin, 2(1):191– 195, 1976.
- [66] Sepp Hartung and André Nichterlein. On the parameterized and approximation hardness of metric dimension. In Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013, pages 266–276. IEEE Computer Society, 2013.
- [67] Yoichi Iwata. Linear-time kernelization for feedback vertex set. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, volume 80 of LIPIcs, pages 68:1–68:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
- [68] Yoichi Iwata. Linear-time kernelization for feedback vertex set. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017), volume 80 of Leibniz International Proceedings in Informatics (LIPIcs), pages 68:1–68:14. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- [69] Yoichi Iwata and Yusuke Kobayashi. Improved analysis of highest-degree branching for feedback vertex set. In Bart M. P. Jansen and Jan Arne Telle, editors, 14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany, volume 148 of LIPIcs, pages 22:1–22:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [70] Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, lp-branching, and FPT algorithms. SIAM J. Comput., 45(4):1377–1411, 2016.
- [71] Yoichi Iwata, Magnus Wahlström, and Yuichi Yoshida. Half-integrality, LP-branching, and FPT algorithms. SIAM J. Comput., 45(4):1377–1411, 2016.
- [72] Bart M.P. Jansen, Christian Schulz, and Hisao Tamaki. NII shonan meeting report no. 144 parameterized graph algorithms and data reduction, 2019.
- [73] Iyad A. Kanj, Michael J. Pelsmajer, and Marcus Schaefer. Parameterized algorithms for feedback vertex set. In Parameterized and Exact Computation, First International Workshop, IWPEC 2004, Bergen, Norway, September 14-17, 2004, Proceedings, volume 3162 of Lecture Notes in Computer Science, pages 235–247. Springer, 2004.
- [74] David R. Karger, Philip N. Klein, Clifford Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.
- [75] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
- [76] Richard M Karp. Reducibility among combinatorial problems. In Complexity of computer computations, pages 85–103. Springer, 1972.
- [77] Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce Reed. The disjoint paths problem in quadratic time. Journal of Combinatorial Theory, Series B, 102(2):424–435, 2012.
- [78] Samir Khuller, Balaji Raghavachari, and Azriel Rosenfeld. Landmarks in graphs. Discret. Appl. Math., 70(3):217–229, 1996.
- [79] Krzysztof Kiljan and Marcin Pilipczuk. Experimental evaluation of parameterized algorithms for feedback vertex set. In Gianlorenzo D'Angelo, editor, Proceedings of the 17th International Symposium on Experimental Algorithms (SEA 2018), volume 103 of Leibniz International Proceedings in Informatics (LIPIcs), pages 12:1–12:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [80] Lefteris M Kirousis and Christos H Papadimitriou. Interval graphs and seatching. Discrete Mathematics, 55(2):181–184, 1985.

- [81] Tomasz Kociumaka and Marcin Pilipczuk. Faster deterministic feedback vertex set. Inf. Process. Lett., 114(10):556–560, 2014.
- [82] Christian Komusiewicz and Johannes Uhlmann. Alternative parameterizations for cluster editing. In Ivana Cerná, Tibor Gyimóthy, Juraj Hromkovic, Keith G. Jeffery, Rastislav Královic, Marko Vukolic, and Stefan Wolf, editors, SOFSEM 2011: Theory and Practice of Computer Science -37th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, January 22-28, 2011. Proceedings, volume 6543 of Lecture Notes in Computer Science, pages 344–355. Springer, 2011.
- [83] Anton Kotzig. Moves without forbidden transitions in a graph. Matematický časopis, 18(1):76–80, 1968.
- [84] Lukasz Kowalik, Shaohua Li, Wojciech Nadara, Marcin Smulewicz, and Magnus Wahlström. Many visits TSP revisited. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, 28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference), volume 173 of LIPIcs, pages 66:1–66:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- [85] Stefan Kratsch. A randomized polynomial kernelization for vertex cover with a smaller parameter. SIAM Journal on Discrete Mathematics, 32(3):1806–1839, 2018.
- [86] Stefan Kratsch, Shaohua Li, Dániel Marx, Marcin Pilipczuk, and Magnus Wahlström. Multibudgeted directed cuts. In Christophe Paul and Michal Pilipczuk, editors, 13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland, volume 115 of LIPIcs, pages 18:1–18:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
- [87] Stefan Kratsch, Shaohua Li, Dániel Marx, Marcin Pilipczuk, and Magnus Wahlström. Multibudgeted directed cuts. Algorithmica, 82(8):2135–2155, 2020.
- [88] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, pages 450–459. IEEE Computer Society, 2012.
- [89] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, pages 450–459. IEEE Computer Society, 2012.
- [90] Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. ACM Transactions on Algorithms, 10(4):20, 2014.
- [91] Robert Krauthgamer, editor. Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016. SIAM, 2016.
- [92] Jason Li and Jesper Nederlof. Detecting feedback vertex sets of size k in $O^*(2.7^k)$ time. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 971–989. SIAM, 2020.
- [93] Shaohua Li, Qilong Feng, Xiangzhong Meng, and Jianxin Wang. An improved FPT algorithm for the flip distance problem. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark, volume 83 of LIPIcs, pages 65:1–65:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
- [94] Shaohua Li and Marcin Pilipczuk. An improved FPT algorithm for independent feedback vertex set. In Andreas Brandstädt, Ekkehard Köhler, and Klaus Meer, editors, Graph-Theoretic Concepts in Computer Science - 44th International Workshop, WG 2018, Cottbus, Germany, June 27-29, 2018, Proceedings, volume 11159 of Lecture Notes in Computer Science, pages 344–355. Springer, 2018.
- [95] Shaohua Li and Marcin Pilipczuk. An improved FPT algorithm for independent feedback vertex set. *Theory Comput. Syst.*, 64(8):1317–1330, 2020.
- [96] Shaohua Li and Marcin Pilipczuk. Hardness of metric dimension in graphs of constant treewidth. CoRR, abs/2102.09791, 2021.
- [97] Shaohua Li, Marcin Pilipczuk, and Manuel Sorge. Cluster editing parameterized above

modification-disjoint p_3 -packings. In Markus Bläser and Benjamin Monmege, editors, 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference), volume 187 of LIPIcs, pages 49:1–49:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

- [98] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Slightly superexponential parameterized problems. SIAM J. Comput., 47(3):675–702, 2018.
- [99] Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. ACM Transactions on Algorithms, 11(2):15:1–15:31, 2014.
- [100] Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. Linear time parameterized algorithms for subset feedback vertex set. ACM Trans. Algorithms, 14(1):7:1–7:37, 2018.
- [101] Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Meirav Zehavi. Parameterized complexity and approximability of directed odd cycle transversal. CoRR, abs/1704.04249, 2017.
- [102] Meena Mahajan and Venkatesh Raman. Parameterizing above guaranteed values: Maxsat and maxcut. Journal of Algorithms, 31(2):335–354, 1999.
- [103] Dániel Marx. Parameterized graph separation problems. Theor. Comput. Sci., 351(3):394–406, 2006.
- [104] Dániel Marx. What's next? Future directions in parameterized complexity. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution* and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday, volume 7370 of Lecture Notes in Computer Science, pages 469–496. Springer, 2012.
- [105] Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. ACM Transactions on Algorithms, 9(4):30, 2013.
- [106] Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. SIAM J. Comput., 43(2):355–388, 2014.
- [107] Matthias Middendorf and Frank Pfeiffer. On the complexity of the disjoint paths problem. Combinatorica, 13(1):97–107, 1993.
- [108] Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, and Saket Saurabh. On parameterized independent feedback vertex set. *Theor. Comput. Sci.*, 461:65–75, 2012.
- [109] Neeldhara Misra, Geevarghese Philip, Venkatesh Raman, Saket Saurabh, and Somnath Sikdar. FPT algorithms for connected feedback vertex set. J. Comb. Optim., 24(2):131–146, 2012.
- [110] John H. Morris, Leonard Apeltsin, Aaron M. Newman, Jan Baumbach, Tobias Wittkop, Gang Su, Gary D. Bader, and Thomas E. Ferrin. clusterMaker: a multi-algorithm clustering plugin for Cytoscape. BMC Bioinformatics, 12(1):436, 2011.
- [111] Yehoshua Perl and Yossi Shiloach. Finding two disjoint paths between two pairs of vertices in a graph. J. ACM, 25(1):1–9, 1978.
- [112] Marcin Pilipczuk and Magnus Wahlström. Directed multicut is W[1]-hard, even for four terminal pairs. In Krauthgamer [91], pages 1167–1178.
- [113] Fábio Protti, Maise Dantas da Silva, and Jayme Luiz Szwarcfiter. Applying modular decomposition to parameterized cluster editing problems. *Theory of Computing Systems*, 44(1):91–104, 2009.
- [114] Igor Razgon and Barry O'Sullivan. Almost 2-sat is fixed-parameter tractable. J. Comput. Syst. Sci., 75(8):435–450, 2009.
- [115] Neil Robertson and Paul D Seymour. Graph minors. xiii. the disjoint paths problem. Journal of combinatorial theory, Series B, 63(1):65–110, 1995.
- [116] Petra Scheffler. A practical linear time algorithm for disjoint paths in graphs with bounded treewidth. TU, Fachbereich 3, 1994.
- [117] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. Discrete Applied Mathematics, 144(1-2):173–182, 2004.
- [118] Peter J Slater. Leaves of trees. Congr. Numer, 14(37):549–559, 1975.
- [119] Stefan Szeider. Finding paths in graphs avoiding forbidden transitions. Discret. Appl. Math., 126(2-3):261-273, 2003.
- [120] René van Bevern, Vincent Froese, and Christian Komusiewicz. Parameterizing edge modification problems above lower bounds. *Theory of Computing Systems*, 62(3):739–770, 2018.

- [121] Vijay V Vazirani. Approximation algorithms. Springer Science & Business Media, 2013.
- [122] David P. Williamson and David B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.
- [123] Tobias Wittkop, Dorothea Emig, Sita Lange, Sven Rahmann, Mario Albrecht, John H. Morris, Sebastian Böcker, Jens Stoye, and Jan Baumbach. Partitioning biological data with transitivity clustering. *Nature Methods*, 7(6):419–420, 2010.