Technical University of Berlin
Electrical Engineering and Computer Science
Institute of Software Engineering and Theoretical Computer Science
Algorithmics and Computational Complexity (AKT)

Parameterized Algorithmics of Multistage Matching

Lino Steinhau

Thesis submitted in fulfillment of the requirements for the degree
"Bachelor of Science" (B. Sc.) in the field of Computer Science

November 2020

Supervisor and first reviewer: Prof. Dr. Rolf Niedermeier
Second reviewer: Prof. Dr. George B. Mertzios, Durham University, UK
Co-Supervisors: Leon Kellerhals and Philipp Zschoche

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Die selbstständige und eigenhändige Ausfertigung versichert an Eides statt

Berlin, den _____ _____

Datum Unterschrift

Zusammenfassung

Wir untersuchen die parametrisierte Komplexität von Multistage Perfect Matching (MPM): finde in einem temporalen Graphen ein perfektes Matching für jeden Zeitschritt des Graphen, sodass jeweils zwei aufeinander folgende Matchings eine minimale Anzahl an Kanten gemeinsam haben. Wir zeigen, dass MPM NP-hart ist, selbst wenn entweder die Lebensdauer des Graphen oder die symmetrische Differenz konstant sind. Das bedeutet, dass MPM für diese Parameter nicht einmal in XP ist, es sei denn, es gilt P=NP. Wir zeigen W[1]-Härte und beschreiben einen XP-Algorithmus für die Kombination dieser beiden Parameter. Diese Härteergebnisse wurden mit ausgeklügelten Reduktionen erzielt und sie sind aus technischer Sicht unsere wichtigsten Ergebnisse. Wenn zusätzlich die Baumweite des zugrundeliegenden Graphen zu diesen Parametern hinzugefügt wird, liegt MPM für diese Parameter in FPT („fixed-parameter tractable“). Weiterhin zeigen wir, dass MPM auch in FPT liegt, wenn der Parameter die Anzahl der Knoten ist, d.h. es ist einfach zu lösen, wenn die Anzahl der Knoten sehr klein ist, auch wenn der temporale Graph viele Zeitschritte umfasst.

Abstract

We study the parameterized complexity of Multistage Perfect Matching (MPM): given a temporal graph, find a perfect matching for each timestep of the graph such that each two consecutive matchings share a minimum amount of edges. We show that MPM is NP-hard, even if either the lifetime of the graph or the symmetric difference is a constant. This means that MPM is not even in XP for these parameters, unless P=NP. We prove W[1]-hardness and provide an XP-Algorithm for the combination of these two parameters. These hardness results were achieved with sophisticated reductions and they are our most important results from a technical standpoint. If additionally, the treewidth of the underlying graph is added to these parameters, then MPM becomes fixed-parameter tractable. Furthermore, we show that MPM is also in FPT if the parameter is the number of vertices, which means it is computationally simple if the number of vertices is very small, even if the temporal graph consists of many layers.

Contents

Chapter 1

Introduction

1.1 Motivation

Matching problems are amongst the most common and important graph problems and have been thoroughly studied for many decades [Kő16]. They appear in every application field where two (or sometimes more) entities from a set must be assigned to each other. A matching is a set of assignments, e.g., a set of edges in the context of graphs. If every entity of the set was assigned, then the matching is *perfect*. For a simple practical example of the Perfect Matching problem, see Figure 1.1. In contrast to other common graph problems such as Vertex Cover or Clique, which are NP-hard, a perfect matching in a graph can be computed in polynomial time.

A recent trend in the field of algorithm theory is the exploration of classic combinatorial problems in a multistage setting. In this setting, we are dealing with temporal graphs that have a static vertex set, but an edge set that changes over time. Oftentimes, the goal of multistage problems is to find solutions for each time step that are overall "similar" (or "dissimilar") by some metric.

This thesis is on the Multistage Perfect Matching (MPM) problem, which is one way to lift the definition of Perfect Matching onto temporal graphs: Given a temporal graph $\mathcal{G}$, the task is to find a perfect matching for each time step such that each two consecutive matchings differ only by a certain amount of edges.

For a practical example, we get back to the worker-task problem from Figure 1.1. Imagine that the availability of each worker for certain tasks changes within a week. This

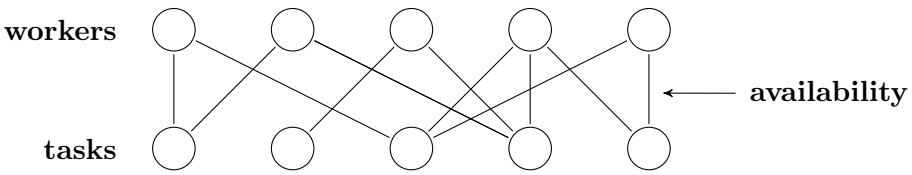


Figure 1.1: A classic perfect matching problem: There is a set of tasks that need to be completed. Each worker is available only for certain tasks. The question is: Is there an assignment of workers to tasks such that each worker does exactly one task and each task is done by exactly one worker?

most likely means that the assignment of workers to tasks will also have to be changed for each day of the week. Now, let us also assume that, whenever workers switch tasks, they need to familiarize themselves with the new task, which costs time and effort every time. Therefore, one wants the assignments of two consecutive days to be as similar as possible, so that the overhead is minimized. This problem can now be modeled as Multistage Perfect Matching.

We focus on the (mostly unknown) parameterized complexity of MPM. This means that we study the problem's complexity depending on a certain parameter such as the number of vertices or the number of time steps. This makes sense, especially for temporal graphs, since problems on temporal graphs are generally computationally hard, even for problems that are polynomial-time solvable for conventional graphs (e.g., the Perfect Matching problem). By analyzing the complexity with respect to various parameters, we can find out which parameter(s) cause this big computational expense. As a result, we will know that the problem's complexity is significantly lower if these parameters are within a certain limit.

1.2 Related Work

Many problems on temporal graph have been analyzed in the literature. Multistage matching problems in particular have been featured in the works of Chimani et al. [CTW20], who studied the polynomial-time approximability of the problem. Bampis et al. [Bam+18] explored two similar problems which they call Minimum/Maximum Multistage Perfect Matching (Min-MPM/Max-MPM). In the case of Min-MPM/Max-MPM, the temporal graph is edge-weighted and the goal is to not only find similar solutions for each time step but also to keep the total sum of edge weights as small/big as possible. They also mostly focus on approximating the two problems. Maximum Temporal Matching, which revolves around maximum matchings in a temporal graph, is studied by Mertzios et al. [Mer+20]. They proved hardness, approximation and fixed-parameter tractability results for the problem.

Fluschnik et al. [Flu+19] studied the parameterized complexity of Multistage Vertex Cover (MVC). We make use of the same method they used to provide an XP-Algorithm for MVC to prove a fixed-parameter tractability result for MPM (see Section 3.4). Fomin et al. [Fom+20] examined the problem of Diverse Pair of (Maximum/Perfect) Matchings, in which one is looking for two distinct matchings in a (static) graph that have a symmetric difference greater than an integer k. They prove that, in contrast to the classical Matching problem, this problem is not polynomial-time solvable anymore, even in graphs of maximum degree three. Mandal and Gupta [MG20] studied 0-1 Timed Matching: given a temporal graph. the task is to find an edge subset such that no two contained edges, that share an endpoint, exist in the same layer of the temporal graph. They proved NP-completeness even for bipartite temporal graphs and rooted temporal trees where each edge is only present in three layers.

Table 1.1: Overview of our complexity results for Multistage Perfect Matching. The term $\text{tw}(G_\cup)$ denotes the treewidth of the underlying graph $G_\cup$.

Constraints	Complexity	Proof
$\tau = 2$	NP-hard	Theorem 4.1
$\ell = 4$	NP-hard	Theorem 4.2
$\ell = 0, \ell \geq n$	P	Observations 3.1,3.2
Parameter		
τ	para-NP-hard	Theorem 4.1
ℓ	para-NP-hard	Theorem 4.2
$\ell + \tau$	XP,W[1]-hard	Theorems 4.9, 3.12
$\ell + \tau + \text{tw}(G_\cup)$	FPT	Thm. 3.6, Cor. 3.7
n	FPT	Theorem 3.8
$\sum_{i \in [\tau-1]} \lvert E_i \cap E_{i+1} \rvert$	FPT	Theorem 3.4

1.3 Our Contributions

We study the parameterized complexity of Multistage Perfect Matching and achieve both positive and negative results for different (combinations of) natural parameters. Our results are summarized in Table 1.1. Parameterized by the number n of vertices in the graph, the problem is fixed-parameter tractable. The remaining results focus mostly on the allowed symmetric difference ℓ between two consecutive matchings, and the number τ of time steps. Parameterized by either one of these parameters, Multistage Perfect Matching is para-NP-hard, that is, MPM is NP-hard even if the parameter is constant. However, we prove that if both parameters are combined, then the problem is contained in XP and W[1]-hard. This hardness result is our most technical one and utilizes a sophisticated reduction from Clique. If additionally the treewidth of the underlying graph is added as a parameter, then Multistage Perfect Matching becomes fixed-parameter tractable. Furthermore, we prove fixed-parameter tractability when the parameter is the sum of the size of consecutive edge set intersections.

1.4 Structure of the Work

In Chapter 2, we introduce the necessary information and definitions for this thesis. Thereafter, we present our tractability results in Chapter 3 and the intractability results in Chapter 4. In Chapter 5, we summarize all results and give an overview of open questions and starting points for future work.

Chapter 2

Preliminaries

In this chapter, we introduce the necessary information concerning basic graph theory, temporal graphs, and parameterized complexity. We start with two important notations:

$A \triangle B$ denotes the *symmetric difference* of sets A and B, formally $A \triangle B := (A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A)$.

$[k]$ denotes the *set* $\{1, 2, \ldots, k-1, k\}$, with $k \in \mathbb{N}$.

2.1 Graph Theory

We introduce basic notation for (non-temporal) graphs. Note that the purpose of this section is mainly to provide a table of reference for looking up notations that will be used later in this thesis.

Let $G = (V, E)$ denote an undirected graph, where V is the set of vertices and E is the set of edges. We denote by

$V(G)$ the *vertex set* of G;

$\binom{V(G)}{2}$ the set of all possible unordered vertex pairs, formally $\{\{u, v\} \mid u, v \in V, u \neq v\}$;

$E(G)$ the *edge set* of G with $E(G) \subseteq \binom{V(G)}{2}$; for an edge $e = \{u, v\} \in E(G)$ the two vertices u and v are called *endpoints* of e;

uv short notation for $\{u, v\}$ with $\{u, v\} \in E(G)$;

n_G the number $|V(G)|$ of *vertices*;

m_G the number $|E(G)|$ of *edges*;

$V(E')$ the *induced vertex set* of $E' \subseteq E(G)$, formally $V(E') \subseteq V(G) := \bigcup_{e \in E'} e$;

$G[V']$ the *induced subgraph* of G on $V' \subseteq V(G)$, formally, $G[V'] := (V', E(G) \cap \binom{V'}{2})$;

$G[E']$ the *induced subgraph* of G on $E' \subseteq E(G)$, formally, $G[E'] := (V(E'), E')$;

$G - E'$ the graph obtained from G by deleting the edges $E' \subseteq E(G)$, formally, $G - E' := (V(G), E(G) \setminus E')$;

$G - V'$ the graph obtained from G by deleting the vertices $V' \subseteq V(G)$, formally, $G - V' := G[V(G) \setminus V']$.

For $k \in \mathbb{N}$, we call a path consisting of k vertices and $k-1$ edges a (k)*-path*. If by context it is clear which G we refer to, then we will omit G from $V(G), E(G), n_G, m_G$ and simply write V, E, n, m.

2.2 Temporal Graphs

A temporal graph $\mathcal{G} = (V, E_1, \ldots, E_\tau)$ consists of a fixed vertex set V and a collection of $\tau \in \mathbb{N}$ edge sets $E_i, i \in [\tau]$. We denote by

$G_i(\mathcal{G})$ the *i-th stage* of $\mathcal{G}$, formally $G_i := (V, E_i)$, also called *layer*;

$n_{\mathcal{G}}$ the *number of vertices* of $\mathcal{G}$, formally $n_{\mathcal{G}} := |V|$;

$E_\cap$ the *set of edges* present in **every** stage of $\mathcal{G}$, formally $E_\cap := \bigcap_{i\in[\tau]} E_i$;

$E_\cup$ the *set of edges* present in **at least one** stage of $\mathcal{G}$, formally $E_\cup := \bigcup_{i\in[\tau]} E_i$;

$G_\cup$ the *underlying graph* of $\mathcal{G}$, formally $G_\cup := (V, E_\cup)$;

$G_\cap$ the *intersection graph* of $\mathcal{G}$, formally $G_\cap := (V, E_\cap)$.

When dealing with two consecutive stages of a temporal graph, we call the step from one stage to the next one a *transition*.

We now introduce the edge-labeled graph $L(\mathcal{G})$ as defined by Zschoche et al. [Zsc+20]. It serves as an enhancement of the *underlying graph* $G_\cup$ with the temporal properties of $\mathcal{G}$ encoded into the edges. The edge-labeling function is defined as $\omega : E(G_\cup) \to [2^\tau - 1]$ with $\omega(uv) = \sum_{i=1}^{\tau} \mathbb{1}_{uv\in E_i} \cdot 2^{i-1}$. Note that $\mathbb{1}_{uv\in E_i}$ is equal to 1 if and only if $uv \in E_i$, and 0 otherwise. Thus, each bit with index i of a label in binary indicates whether the associated edge exists in the i-th stage of $\mathcal{G}$. This edge-labeled graph can be constructed in $O(m_{G_\cup} \cdot \log(m_{G_\cup}))$ time [Zsc+20].

For more details on temporal graphs (also known as *dynamic* or *time-varying* graphs), we refer to Casteigts et al. [Cas+12], Flocchini et al. [FMS13], Kostakos [Kos09], and Michail [Mic15].

2.3 Matching & Multistage Matching

In this section, we define both the conventional and the *multistage* version of Matching.

Matching
Input: An undirected graph $G = (V, E)$ and a number $k \in \mathbb{N}$.
Question: Is there a subset $M \subseteq E$ of edges with $|M| \geq k$ such that $\sum_{v\in V} |e \cap \{v\}| \leq 1$ for all $e \in M$?

In other words, each vertex v must only be "matched" by at most one edge e in M. A matching M is *maximal* if no edge of $e \in E \setminus M$ can be added such that $M \cup \{e\}$ is also a matching. A matching M is a *maximum matching* if there is no other matching M^* such that $|M^*| > |M|$. A matching M is *perfect* if $|M| = \frac{n}{2}$.

Finding a perfect matching in a graph with n vertices can be done in $O(n^2 \cdot m)$ time using Edmond's blossom algorithm [Edm65] or even $O(n^{0.5} \cdot m)$ time using the more complicated algorithm of Micali and Vazirani [MV80]. Throughout this thesis, we will denote by $T_M(n)$ the running time of the fastest algorithm for finding a perfect matching.

We now define Multistage Perfect Matching, the centerpiece of this thesis:

Multistage Perfect Matching (MPM)

Input: A temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ and an integer $\ell \in \mathbb{N}_0$.

Question: Is there a sequence $\mathcal{M} = (M_1, \ldots, M_\tau)$ of edge sets such that

(i) M_i is a perfect matching of graph G_i, for each $i \in [\tau]$, and

(ii) the symmetric difference $M_i \triangle M_{i+1}$, for each $i \in [\tau - 1]$, is of size at most ℓ?

2.4 Parameterized Complexity

The field of parameterized complexity is a relatively new one in the field of algorithmics and complexity. It aims to analyze a problem's complexity not only dependent on the size of the input instance but also on a certain parameter. We now define the essential complexity classes and principles. For further info, we refer to the common monographies of parameterized algorithmics [Cyg+15; DF13; FG06; Nie06].

2.4.1 Basic Definition

Let Σ be a finite alphabet. Parameterized problems are languages $L \subseteq \{(x, k) \in \Sigma^* \times \mathbb{N}_0\}$. One of the foundations of parameterized complexity is the complexity class of fixed-parameter tractable problems, abbreviated FPT. It is defined as the set of parameterized problems L for which, given an instance $(x, k) \in \Sigma^* \times \mathbb{N}_0$, whether or not $(x, k) \in L$ can be decided in $f(k) \cdot n^{O(1)}$ time, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function depending only on k and $n = |x|$.

FPT is contained in the class of "slice-wise polynomial" problems, called XP. Formally, a problem L is in XP if there exists a function $g : \mathbb{N} \to \mathbb{N}$ and an algorithm that can decide in time $n^{g(k)}$ if $(x, k) \in L$. Problems in XP are solvable in polynomial time for each fixed "slice" of k, but since the exponent in $n^{g(k)}$ depends on k, the polynomial degree is potentially different for each k. This is the main difference to FPT, where the polynomial part of the running time always has a constant exponent.

2.4.2 The W-Hierarchy & Para-NP

We now introduce the W-Hierarchy, a collection of complexity classes W[1], W[2], ... W[P], all containing FPT. Since the formal definition of the W[t] classes is of no further interest for this thesis, we will only concern ourselves with how these classes can be used to prove certain features of a parameterized problem:

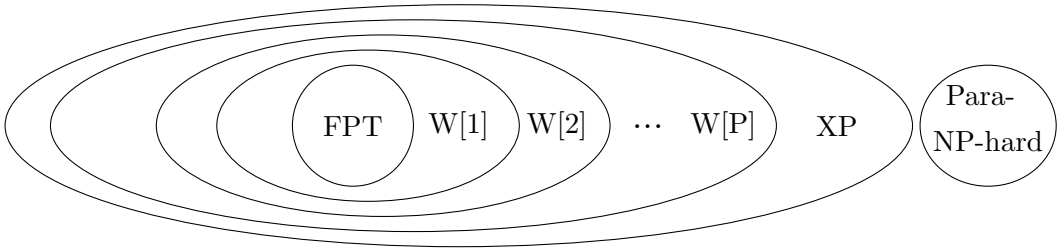


Figure 2.1: Overview of the parameterized complexity classes, assuming P≠NP, FPT⊂W[1]⊂W[2]⊂ … ⊂W[P]⊂XP and (para-NP-hard ∩ XP) = ∅

1. FPT⊆W[1]⊆W[2]⊆ … ⊆W[P]⊆XP (see Figure 2.1).
2. If a parameterized problem is W[1]-hard, then there is no $f(k) \cdot n^{O(1)}$ algorithm to solve it, unless FPT=W[1].
3. If a parameterized problem is in W[1], then there is an $n^{f(k)}$ algorithm to solve it.

Analogously to the well-known NP-hardness for classical complexity theory, para-NP-hardness for parameterized complexity is used to prove that a problem is not in XP, unless P=NP (see Figure 2.1). Note that para-NP=FPT if and only if P=NP. Also, if a parameterized problem is NP-hard for a fixed value of the parameter k, then it is para-NP-hard.

2.4.3 Parameterized Reduction

A *parameterized reduction* is a function $\phi : \Sigma^* \times \mathbb{N}_0 \to \Sigma^* \times \mathbb{N}_0$ that maps an instance (x, k) of a parameterized problem P to an instance (x', k') of another problem Q such that

- $(x, k) \in P$, if and only if $(x', k') \in Q$,
- $\phi(x, k)$ can be computed in $f(k) \cdot |x|^{O(1)}$ time, and
- $k' \leq g(k)$, for some function $g : \mathbb{N}_0 \to \mathbb{N}_0$.

Similarly to reductions in classical complexity theory, parameterized reductions can be used to prove both positive and negative results for parameterized problems. For example, if a parameterized problem A is fixed-parameter tractable for parameter k and another problem B with parameter l can be reduced to A, then B is also in FPT for parameter l. Vice versa, if a problem A is W[1]-hard for parameter k and it can be reduced to another problem B with parameter l, then B is also W[1]-hard for parameter l.

Chapter 3

Tractability Results

In this chapter we present positive fixed-parameter tractability results for Multistage Perfect Matching. We start off by laying the foundations for the results with some observations on the problem itself and certain restrictions for its parameters. Thereafter, we discuss three parameters (or parameter combinations) for which MPM is fixed-parameter tractable and one parameter combination for which it is contained in XP.

3.1 Basic Observations

We begin by stating some basic observations on the problem Multistage Perfect Matching. Firstly, we observe that certain ranges or values for parameter ℓ simplify MPM so that it becomes polynomial-time solvable. For the following observations, recall that $T_M(n)$ is the running time of the fastest algorithm for finding a perfect matching on a non-temporal graph with n vertices.

Observation 3.1. Multistage Perfect Matching *is solvable in* $\tau \cdot T_M(n)$ *time if* $\ell \geq n$.

Proof. Let $(\mathcal{G}, \ell)$ be an instance of MPM with $\ell \geq n$. A solution for this instance requires a perfect matching for each of the τ stages of $\mathcal{G}$. Since perfect matchings always have size $\frac{n}{2}$, a symmetric difference of n or greater allows two perfect matchings M_i and $M_{i+1}, i \in [\tau - 1]$, to have no elements in common for two consecutive stages G_i, G_{i+1}. Thus, the problem is now equivalent to simply finding *any* perfect matching for each stage, regardless of the relationship between the matchings. This can be done in $T_M(n)$ time for each stage. Hence, the total computation time is $\tau \cdot T_M(n)$. □

Observation 3.2. Multistage Perfect Matching *is solvable in* $O(n^2\tau) + T_M(n)$ *time if* $\ell = 0$.

Proof. Let $(\mathcal{G}, \ell)$ be an instance of MPM with $\ell = 0$. If $\ell = 0$, then the perfect matching M has to be the same for all τ stages of $\mathcal{G}$. Such a matching only exists if there is a perfect matching in the intersection graph $G_\cap$ of $\mathcal{G}$. Thus, finding a solution for $(\mathcal{G}, \ell)$ is reduced to constructing the intersection graph (in $O(n^2\tau)$ time) and finding a perfect matching in this graph in $T_M(n)$ time. □

Following these observations, henceforth we will implicitly assume that $0 < \ell < n$.

The following observation is only indirectly connected to Multistage Perfect Matching. We prove an upper bound for the number of distinct edge subsets with a certain size that can exist in a graph. This upper bound will be used for the proof in Section 3.4.

Observation 3.3. *A graph $G = (V, E)$ with n vertices can have at most $n^{\frac{n}{2}}$ distinct edge subsets of size $\frac{n}{2}$.*

Proof. We prove the observation by representing the edge subsets as ordered sequences of vertices. The number of possible *ordered* sequences is obviously higher than the number of *unordered* sets. By defining a non-injective function that maps multiple sequences to the same set, we prove by which factor the number of sequences and the number of sets differ. Consequently, we can provide an upper bound for the number of distinct edge subsets.

Let $G = (V, E)$ be a graph with an even number of vertices n and $\prod_V$ be the set of all strict orderings of V. Let $f : \prod_V \to \{E' \subseteq E\}$ be a function that maps an *ordered* sequence $V' \in \prod_V$ to an *unordered* edge subset of G such that $f((v_1, v_2, \ldots, v_{n-1}, v_n)) := \{\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}\}$. There are $n!$ possible sequences V'. Now, consider two facts:

- Let $i \in [n]$ be odd. If the two vertices at positions $i, i+1$ in the sequence V_1 are switched, forming the sequence V_2, then $f(V_1) = f(V_2)$, since

$$\begin{aligned} f(V_1) &= f((\ldots, v_i, v_{i+1}, \ldots)) \\ &= \{\ldots, \{v_i, v_{i+1}\}, \ldots\} \\ &= \{\ldots, \{v_{i+1}, v_i\}, \ldots\} \\ &= f((\ldots, v_{i+1}, v_i, \ldots)) \\ &= f(V_2). \end{aligned}$$

- Let $i, j \in [n]$ with $i \neq j$ both be odd. If the vertices at positions $i, i+1$ in the sequence V_1 and the ones at positions $j, j+1$ are switched, forming the sequence V_2, then $f(V_1) = f(V_2)$, since

$$\begin{aligned} f(V_1) &= f((\ldots, v_i, v_{i+1}, \ldots, v_j, v_{j+1}, \ldots)) \\ &= \{\ldots, \{v_i, v_{i+1}\}, \ldots, \{v_j, v_{j+1}\}, \ldots\} \\ &= \{\ldots, \{v_j, v_{j+1}\}, \ldots, \{v_{i+1}, v_i\}, \ldots\} \\ &= f((\ldots, v_j, v_{j+1}, \ldots, v_i, v_{i+1}, \ldots)) \\ &= f(V_2). \end{aligned}$$

This yields that for each sequence $V' \in \prod_V$, there is a number of sequences $V'' \neq V'$ with $f(V'') = f(V')$. To be exact, for each sequence, all vertex pairs at odd positions inside the sequence can be flipped and/or rearranged arbitrarily. There are $(\frac{n}{2})!$ possible rearrangements and $2^{\frac{n}{2}}$ possible flips of the vertex pairs. Therefore, the number of unique edge subsets with size $\frac{n}{2}$ is the number of possible sequences V' divided by the number of sequences that are mapped to the same set $f(V')$, respectively. This number is $n!/(2^{\frac{n}{2}} \cdot (\frac{n}{2})!)$, which is smaller than $n^{\frac{n}{2}}$. □

3.2 Parameter Sum of Edge Intersections

Chimani et al. [CTW20] observed that Multistage Intersection Matching and Multistage Union Matching are in FPT if parameterized by the sum of the amount of shared edges for each two consecutive stages, formally $\sum_{i\in[\tau-1]} |E_i \cap E_{i+1}|$. It follows that the same holds for Multistage Perfect Matching.

Theorem 3.4. Multistage Perfect Matching *is solvable in* $2^{O(k)} \cdot T_M(n) \cdot \tau$ *time where* $k := \sum_{i\in[\tau-1]} |E_i \cap E_{i+1}|$.

Proof. Wo provide an algorithm that solves the problem in FPT time. The idea of the algorithm is to try out all possible subsets with certain features of the shared edges between two consecutive stages. We call these subsets the *intermediate bases*. The algorithm then tries to construct perfect matchings for each stage i of the temporal graph. Each of these matchings is a superset of both the preceding intermediate basis and the subsequent intermediate basis. The features of the intermediate bases ensure that the constructed matchings have a symmteric difference that is not greater than ℓ.

Let $k := \sum_{i\in[\tau-1]} |E_i \cap E_{i+1}|$. The algorithm "guesses" all possible sequences $\mathcal{M}_\cap = (M_{1,\cap}, \ldots, M_{\tau-1,\cap})$ of *intermediate bases* such that for each $i \in [\tau - 1]$

(i) $M_{i,\cap} \subseteq E_i \cap E_{i+1}$,
(ii) $|M_{i,\cap}| \geq \frac{n-\ell}{2}$, and
(iii) $M_{j-1,\cap} \cup M_{j,\cap}$ is a perfect matching of $G_j[M_{j-1,\cap} \cup M_{j,\cap}]$, for each $j \in [\tau]$.

Note that implicitly $M_{0,\cap} = M_{\tau,\cap} = \emptyset$. We now construct the actual matchings for each stage $j \in [\tau]$ of the temporal graph: First, we construct $G'_j := G_j - V(M_{j-1,\cap} \cup M_{j,\cap})$. That is, G'_j only contains vertices not yet matched by an edge in $M_{j-1,\cap}$ or $M_{j,\cap}$. We now try to find a perfect matching M'_j of G'_j in $T_M(n)$ time. If no such matching exists, then continue with the next sequence $M_\cap$. Otherwise, we claim that $M_j := M'_j \cup M_{j-1,\cap} \cup M_{j,\cap}$ is a perfect matching for G_j. This holds, since

- $V(M'_j)$ and $V(M_{j-1,\cap} \cup M_{j,\cap})$ are disjoint and $V(M'_j) \cup V(M_{j-1,\cap} \cup M_{j,\cap}) = V$,
- the vertex sets of G'_j and $G_j[M_{j-1,\cap} \cup M_{j,\cap}]$ are disjoint,
- M'_j is a perfect matching for G'_j,
- $M_{j-1,\cap} \cup M_{j,\cap}$ is a perfect matching for $G_j[M_{j-1,\cap} \cup M_{j,\cap}]$ (by condition (iii)), and
- $G_j = G'_j \cup G_j[M_{j-1,\cap} \cup M_{j,\cap}]$.

Once each M_j has been computed successfully, the algorithm returns the sequence $\mathcal{M} = (M_j)_{j\in[\tau]}$. If every sequence has been tried unsuccessfully, then the algorithm returns nothing. There are $O(2^k)$ sequences $\mathcal{M}_\cap$ and for each sequence, the matchings can be computed in time $T_M(n) \cdot \tau$. Thus, the algorithm finishes in FPT time if parameterized by k. We claim that the algorithm returns a sequence $\mathcal{M}$ and $\mathcal{M}$ is a Multistage Perfect Matching solution for $(\mathcal{G}, \ell)$ if and only if there is a solution for $(\mathcal{G}, \ell)$.

"⇒": If the algorithm returns a sequence $\mathcal{M}$, this sequence is a solution because for each $j \in [\tau]$, M_j is a perfect matching of G_j and the symmetric difference constraint is fulfilled for the following reasons: The construction ensures that each matching M_j

contains all elements of the previous and the next intermediate basis. This implies that two consecutive matchings $M_i, M_{i+1}, i \in [\tau - 1]$, both contain all elements of the intermediate basis $M_{i,\cap}$. Since condition (ii) ensures that this basis has at least size $\frac{n-\ell}{2}$, the intersection of the two matchings is also of size at least $\frac{n-\ell}{2}$. Combined with the fact that both matchings have size $\frac{n}{2}$, this proves that their symmetric difference is at most ℓ.

"$\Leftarrow$": If there is a solution $\mathcal{M} = (M_j)_{j\in[\tau]}$ for $(\mathcal{G}, \ell)$, then $|M_i \triangle M_{i+1}| \leq \ell$, for each $i \in [\tau - 1]$. Let $M_{i,\cap} := M_i \cap M_{i+1}$. It follows that $|M_{i,\cap}| \geq \frac{n-\ell}{2}$, meaning condition (ii) of the algorithm is fulfilled. Since $M_i \subseteq E_i$ and $M_{i+1} \subseteq E_{i+1}$, it also holds that $M_{i,\cap} \subseteq E_i \cap E_{i+1}$, meaning condition (i) of the algorithm is fulfilled. Let $M_{0,\cap} = M_{\tau,\cap} = \emptyset$. We will now show that for each $j \in [\tau]$, condition (iii) of the algorithm is fulfilled for $M_{j-1,\cap}$ and $M_{j,\cap}$: Assume that there is a $v \in V(M_{j-1,\cap} \cup M_{j,\cap})$ such that $\exists e_1, e_2 \in M_{j-1,\cap} \cup M_{j,\cap} : v \in e_1 \cap e_2$, meaning that $M_{j-1,\cap} \cup M_{j,\cap}$ is not a perfect matching for $G_j[M_{j-1,\cap} \cup M_{j,\cap}]$. Since $M_{j-1,\cap} \subseteq M_j$ and $M_{j,\cap} \subseteq M_j$, we can assume that $e_1 \in M_j$ and $e_2 \in M_j$. This contradicts M_j being a perfect matching of G_j. Thus, condition (iii) is fulfilled. Since all three rules are fulfilled by each constructed intermediate basis $M_{j,\cap}$, the algorithm will always find these bases and therefore a correct sequence of matchings if a solution exists. □

3.3 Parameter $\ell + \tau$+Treewidth

In this section, we show that Multistage Perfect Matching is in FPT for the combined parameters ℓ, τ, and the *treewidth* of the underlying graph G. To this end, we use Courcelle's theorem [CE12] and results of Arnborg et al. [ALS91]. Courcelle's theorem states the following.

Theorem 3.5. *Given a graph G and an MSO formula φ describing a certain property of G, it can be decided in time $f(tw(G), |\varphi|) \cdot n^{O(1)}$ whether G has the property of interest, where $tw(G)$ is the treewidth of the graph G.*

The treewidth of a graph is an integer that describes how "tree-like" a graph is. We will not need the formal definition for this thesis, but refer to Bertelè and Brioschi [BB72] for further details on the treewidth. The second-order logic is an extension of the first-order (predicate) logic. This extension allows quantification over predicates in addition to quantification over variables. The monadic second-order (MSO) logic in particular is the fragment of second-order logic for quantification over monadic (single argument) predicates, meaning sets. MSO logic is particularly useful in the context of graphs, since the vertex and edge sets of a graph can be quantified in an MSO formula.

Theorem 3.6. *There is an **MSO** formula φ such that*

- *$\mathcal{G} \models \varphi$ if and only if $\mathcal{G}$ is a yes-instance of* Multistage Perfect Matching *and*
- *the size of φ depends only on ℓ and τ.*

Proof. We first construct the edge-labeled graph $L(\mathcal{G})$ from $\mathcal{G}$. Then we define a formula that checks whether an edge e exists in the t-th stage of $\mathcal{G}$ [Zsc+20]:

$$\text{layer}(e,t) := \bigvee_{i=1}^{\tau} \bigvee_{j \in \sigma(i,2^\tau-1)} (t = i \wedge \omega(e) = j).$$

Herein, $\sigma(i,j) := \{x \in [j] \mid i\text{-th bit of } x \text{ is } 1\}$. Note that the formula uses ω as the edge-labeling function as introduced in Section 2.2. To explain the formula, consider an edge e in a temporal graph with $\tau = 2$. Let us assume that e is only present in layer 1. The label of e would thus be $\omega(e) = 10$. If we want to check whether e is present in layer 1, we have to evaluate the formula

$$\begin{aligned}\text{layer}(e,1) =& \bigvee_{i=1}^{2} \bigvee_{j \in \sigma(i,3)} (1 = i \wedge \omega(e) = j)\\ =&(1 = 1 \wedge \omega(e) = 10) \vee (1 = 1 \wedge \omega(e) = 11) \vee\\ &(1 = 2 \wedge \omega(e) = 01) \vee (1 = 2 \wedge \omega(e) = 11).\end{aligned}$$

It is easy to see that it evaluates to *true*, as expected. If we want to check whether e is present in layer 2, we have to evaluate the formula

$$\begin{aligned}\text{layer}(e,2) =& \bigvee_{i=1}^{2} \bigvee_{j \in \sigma(i,3)} (2 = i \wedge \omega(e) = j)\\ =&(2 = 1 \wedge \omega(e) = 10) \vee (2 = 1 \wedge \omega(e) = 11) \vee\\ &(2 = 2 \wedge \omega(e) = 01) \vee (2 = 2 \wedge \omega(e) = 11).\end{aligned}$$

It is easy to see that it evaluates to *false*, as expected.

Additionally, we need a formula that checks whether the symmetric difference of two sets is greater than ℓ:

$$\begin{aligned}\text{delta}(A,B,\ell) :=& \exists x_1, \ldots, x_{\ell+1} \Big(\bigwedge_{i,j \in [\ell+1]} i \neq j \to x_i \neq x_j \Big) \wedge\\ &\Big(\bigwedge_{i \in [\ell+1]} (x_i \in A \wedge x_i \notin B) \vee (x_i \notin A \wedge x_i \in B) \Big).\end{aligned}$$

The formula essentially checks the existence of $\ell+1$ distinct (first part of the conjunction) elements such that each element is contained in exactly one of the two sets (second part of the conjunction). If these elements exist, the symmetric difference of the two sets must be greater than ℓ.

Lastly, a formula is needed for checking whether a set M_t is a perfect matching for stage t of $\mathcal{G}$:

$$\begin{aligned}\text{pm}(M_t,t) :=& \forall v \in V \exists e \in E_\cup \Big(\big(\text{layer}(e,t) \wedge e \in M_t \wedge \text{inc}(e,v)\big) \wedge\\ &\forall f \in E_\cup \big(\text{layer}(f,t) \wedge f \in M_t \wedge \text{inc}(f,v) \to e = f\big) \Big),\end{aligned}$$

where $\text{inc}(e, v)$ is a formula that resolves to true if and only if the vertex v is incident to the edge e. In words, the formula checks whether for each vertex in the graph, it holds that

- there is an edge in M_t and G_t that matches the vertex (first part of the conjunction), and
- there is no other edge in M_t and G_t that matches the vertex (second part of the conjunction).

Finally we construct the composite formula

$$\varphi(\tau, \ell) := \exists M_1, \ldots, M_\tau \subseteq E_\cup \Big(\big(\bigwedge_{t \in [\tau]} \text{pm}(M_t, t) \big) \wedge \big(\bigwedge_{t \in [\tau-1]} \neg\text{delta}(M_t, M_{t+1}, \ell) \big) \Big).$$

The formula φ checks the existence of τ sets such that the sets are perfect matchings for the respective layers and that each two consecutive sets have a symmetric difference that is not greater than ℓ. This proves the first claim of the theorem. The second claim can be proven easily:

- The size of $\text{layer}(e, t)$ depends only on τ.
- The size of $\text{delta}(M_1, M_2, \ell)$ depends only on ℓ.
- The size of of $\text{pm}(M_t, t)$ depends on the size of $\text{layer}(e, t)$, which depends only on τ, and the size of $\text{inc}(e, v)$, which is constant.

Thus, the size of φ, which uses nothing but the above-described formulas, is only dependent on ℓ and τ. □

We can now connect Theorem 3.6 with Theorem 3.5. If we construct the edge-labeled graph $L(\mathcal{G})$ of a temporal graph $\mathcal{G}$ and the formula φ from Theorem 3.6, Courcelle's theorem tells us that we can decide whether $\mathcal{G}$ has a Multistage Perfect Matching in FPT time for the parameter $\text{tw}(L(\mathcal{G})) + |\varphi| = \text{tw}(G_\cup) + f(\ell, \tau)$, where f is some computable function. This leads to the following corollary.

Corollary 3.7. Multistage Perfect Matching *can be solved in* $O(f(\ell, \tau, tw(G_\cup)) \cdot |E(G_\cup)| \cdot log(|E(G_\cup)|)$ *time for some computable function* f *and* $tw(G)$ *being the treewidth of a graph* G.

3.4 Parameter n

In this section, we show that Multistage Perfect Matching is in FPT if parameterized by n, the number of vertices. That is, we prove the following.

Theorem 3.8. *Every instance* $(\mathcal{G}, \ell)$ *of* Multistage Perfect Matching *can be decided in* $O(\tau \cdot n^{n+1})$ *time, where* $n = |V(\mathcal{G})|$.

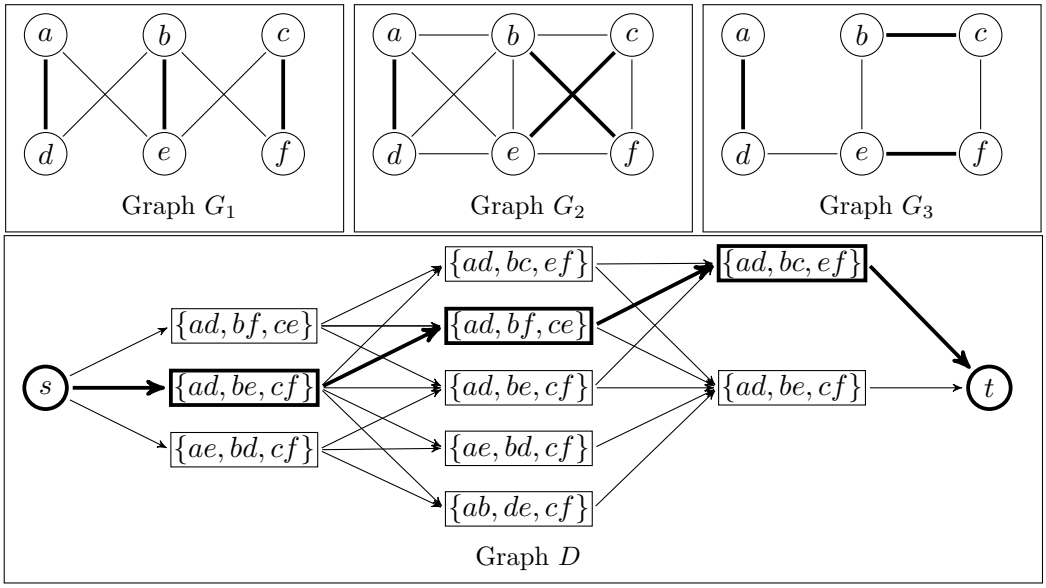


Figure 3.1: A construction of D from $(\mathcal{G} = (V, E_1, E_2, E_3), \ell = 4)$. At the top, the three layers of $\mathcal{G}$ are displayed. Below, in the configuration graph D, each "column" contains all possible perfect matchings for the corresponding stage. The bold edges in G_1, G_2, G_3 and the bold path in D represent a possible MPM solution for $\mathcal{G}$ with $\ell = 4$.

The following constructions and proofs are inspired by the ones that Fluschnik et al. [Flu+19] used to prove an XP-algorithm for Multistage Vertex Cover with parameter k. In our case, the parameter is n and thus, the algorithm has a running time of $O(n^{n+1})$ (ignoring polynomials). Since n^{n+1} is a function only depending on n, the algorithm has an FPT running time instead of "only" an XP running time. Recall that the FPT class is contained in XP (see Section 2.4).

The rough idea of the algorithm is as follows: we consider all edge subsets of size $\frac{n}{2}$ that form a perfect matching for each layer, respectively. Then, we select for each layer one of those subsets such that the symmetric difference constraint is fulfilled. Lastly, we show that these steps are equivalent to finding a source-sink path in the *configuration graph* of $\mathcal{G}$.

Definition 3.9. The *configuration graph* of an instance $(\mathcal{G}, \ell)$ of Multistage Perfect Matching is the directed graph $D = (V, A)$, with $V = V_1 \uplus \ldots \uplus V_\tau \uplus \{s, t\}$, together with a function $\gamma : V \setminus \{s, t\} \to \{E' \subseteq E(G_\cup) \mid |E'| = \frac{|V(\mathcal{G})|}{2}\}$ such that

(i) for every $i \in [\tau]$, it holds that M is a perfect matching of G_i if and only if a vertex $v \in V_i$ exists with $\gamma(v) = M$;
(ii) for every $i \in [\tau - 1], u \in V_i, v \in V_{i+1}$, there is an arc from u to v if and only if $|\gamma(u) \triangle \gamma(v)| \leq \ell$;
(iii) there is an arc (s, v) for every $v \in V_1$ and an arc (v, t) for every $v \in V_\tau$.

For an example, see Figure 3.1.

Lemma 3.10. *The configuration graph D of an instance $(\mathcal{G}, \ell)$ of* Multistage Perfect Matching*, where $n = |V(\mathcal{G})|$,*

(i) can be constructed in $O(n^{n+1} \cdot \tau)$ time, and
(ii) has at most $\tau \cdot n^{\frac{n}{2}} + 2$ vertices and $(\tau - 1) \cdot n^n + 2 \cdot n^{\frac{n}{2}}$ arcs.

Proof. The set $\mathcal{M} := \{E' \subseteq E(G_\cup) \mid |E'| = \frac{n}{2}\}$ can be computed in $O(n^{n+1})$ time, since there are at most n^2 edges in $G_\cup$ and therefore at most $\binom{n^2}{\frac{n}{2}} < (n^2)^{\frac{n}{2}} = n^n$ edge subsets of size $\frac{n}{2}$. We then check in $O(n)$ time whether $M \in \mathcal{M}$ is a perfect matching of G_i, for each $i \in [\tau]$. Let $\mathcal{M}_i$ be the set of perfect matchings for G_i. Add a vertex v to V_i for each $M \in \mathcal{M}_i$ and set $\gamma(v) = M$. At last, add s and t to D. Given that a graph with n vertices can have at most $n^{\frac{n}{2}}$ distinct perfect matchings (Observation 3.3), D has at most $\tau \cdot n^{\frac{n}{2}} + 2$ vertices. Furthermore, the vertex set was constructed in $O(n^{n+1} \cdot \tau)$ time.

Now we add the arcs to D: For each $i \in [\tau - 1], u \in V_i, v \in V_{i+1}$ check whether $|\gamma(u) \triangle \gamma(v)| \leq \ell$ in $O(n)$ time and possibly add the arc (u, v) to D. This results in $O((\tau - 1) \cdot n^n)$ time overall, since we have to consider each two consecutive stages ($\tau - 1$ in total) and each vertex of either stage (at most $n^{\frac{n}{2}}$; respectively). The only thing left to do is adding the arcs from s to every vertex in V_1 and the arcs to t from every vertex in V_τ in $O(2 \cdot n^{\frac{n}{2}})$ time. Since for each $i \in [\tau - 1]$, there are at most $n^{\frac{n}{2}}$ vertices in V_i and V_{i+1}, respectively, there can be at most $(n^{\frac{n}{2}})^2 = n^n$ arcs from a vertex of V_i to a vertex of V_{i+1}. Thus, we have now added at most n^n arcs for each two consecutive stages as well as at most $2 \cdot n^{\frac{n}{2}}$ arcs outgoing from s and ingoing to t. Thus, D has at most $(\tau - 1) \cdot n^n + 2 \cdot n^{\frac{n}{2}}$ arcs. Now, the construction of D is finished. □

Lemma 3.11. *An instance $(\mathcal{G}, \ell)$ of* Multistage Perfect Matching *is a yes-instance if and only if there is a path from s to t in the configuration graph D of $(\mathcal{G}, \ell)$.*

Proof. Let $D = (V = V_1 \uplus \ldots \uplus V_\tau \uplus \{s, t\}, A, \gamma)$.

"⇒": Let $(M_1, \ldots, M_\tau)$ be a solution to $(\mathcal{G}, \ell)$. Then, for each $i \in [\tau]$, there is a $v_i \in V_i$ such that $\gamma(v_i) = M_i$, since V_i contains a vertex for every perfect matching of G_i. For each $j \in [\tau - 1]$, the arc (v_j, v_{j+1}) is contained in A considering $|\gamma(v_j) \triangle \gamma(v_{j+1})| = |M_j \triangle M_{j+1}| \leq \ell$. Thus, $P = (\{v_1, \ldots, v_\tau\} \cup \{s, t\}, \{(s, v_1), (v_\tau, t)\} \cup \bigcup_{j=1}^{\tau-1} \{(v_j, v_{j+1})\})$ is an s-t-path in D.

"⇐": Let $P = (\{v_1, \ldots, v_\tau\} \cup \{s, t\}, \{(s, v_1), (v_\tau, t)\} \cup \bigcup_{j=1}^{\tau-1} \{(v_j, v_{j+1})\})$ be an s-t-path in D. The sequence $(M_1 = \gamma(v_1), \ldots, M_\tau = \gamma(v_\tau))$ is a solution to $(\mathcal{G}, \ell)$ since for each $i \in [\tau]$, $\gamma(v_i)$ is a perfect matching for G_i. Furthermore, for each $j \in [\tau - 1]$, the presence of the arc (v_j, v_{j+1}) implies that $|\gamma(v_j) \triangle \gamma(v_{j+1})| \leq \ell$. Thus, the proof is finished. □

We are ready to prove Theorem 3.8.

Proof of Theorem 3.8. Let $(\mathcal{G}, \ell)$ be an instance of Multistage Perfect Matching and $n = |V(\mathcal{G})|$. An algorithm that decides whether $(\mathcal{G}, \ell)$ is a yes-instance first constructs the configuration graph D in $O(n^n \cdot \tau)$ time (Lemma 3.10 (i)). Then, it searches for an s-t-path in D using breadth-first search. This requires $O(n^n \cdot \tau)$ time (Lemma 3.10 (ii)). Return *yes* if a path is found, and *no* otherwise (Lemma 3.11). □

3.5 XP-Algorithm for $\ell + \tau$

In this section, we prove that Multistage Perfect Matching parameterized by $\ell+\tau$ is in XP.

Theorem 3.12. Multistage Perfect Matching *with parameter* $\ell + \tau$ *is solvable in* $O(n^{2\cdot\ell\cdot\tau} \cdot \tau \cdot (n^2 + \ell)) \cdot T_M(n)$ *time.*

We prove Theorem 3.12 by providing an algorithm that decides an instance $(\mathcal{G} = (V, E_1, \ldots, E_\tau), \ell)$ of Multistage Perfect Matching in XP-time for parameters ℓ and τ. In a nutshell, the algorithm guesses the correct symmetric differences between each two consecutive perfect matchings. Then, the guess is extended to a perfect matching for each layer.

We begin by defining how the algorithm "guesses" the correct symmetric differences. To this end, we split the symmetric difference for each transition between two matchings into the elements that are removed (∇) and the elements that are added (Δ). Formally, the algorithm guesses the edge subsets $\nabla_1, \Delta_1, \ldots, \nabla_{\tau-1}, \Delta_{\tau-1}$ such that for each $i \in [\tau - 1]$, it holds that

- $\nabla_i \subseteq E_i$,
- $\Delta_i \subseteq E_{i+1}$,
- $|\Delta_i| = |\nabla_i| \leq \frac{\ell}{2}$,
- $V(\Delta_i) = V(\nabla_i)$, that is, the edges in Δ_i and ∇_i have the same set of endpoints,
- $\forall e \in \Delta_i : \sum_{v\in V} |e \cap \{v\}| \leq 1$, and
- $\forall e \in \nabla_i : \sum_{v\in V} |e \cap \{v\}| \leq 1$.

The last two restrictions ensure that each two edges in a set share no endpoint. Observe that there are at most n^2 edges in each edge set E_i. Thus, there are at most $\binom{n^2}{\frac{\ell}{2}} \leq (n^2)^{\frac{\ell}{2}} = n^\ell$ different ways to construct each set Δ_i and ∇_i. Overall, this leads to a running time of $O((n^\ell)^{2\tau}) = O(n^{2\cdot\ell\cdot\tau})$ to guess the correct sets.

Given these sets, the algorithm tries to construct the perfect matchings for the solution. To this end, we first construct τ *base sets*. With each layer i, we add restrictions using Δ_i and ∇_i, so that the size of the new set is always smaller or equal to the size of the previous one. Consequently, the last base set will be the smallest set. We can construct a perfect matching for layer τ based on this set, if and only if we are dealing with a yes-instance. After that, we construct all remaining perfect matchings starting with the last layer and ending with the first. The base sets $B_1, \ldots, B_\tau$ are constructed in the following way:

$$B_i := \begin{cases} E_1 & \text{if } i = 1, \\ ((B_{i-1} \cap E_i) \cup \Delta_{i-1}) \setminus \nabla_{i-1} & \text{if } i \in \{2, \ldots, \tau\}. \end{cases}$$

The construction of these sets can be done in $O(\tau \cdot (n^2 + \ell))$ time.

Now, the algorithm tries to extend $\Delta_{\tau-1}$ to a perfect matching M_τ for G_τ by finding a fitting edge subset $M'_\tau \subseteq B_\tau \setminus \Delta_{\tau-1}$ such that $M_\tau := M'_\tau \cup \Delta_{\tau-1}$. This takes $T_M(n)$

time. We can now construct the complete solution based on M_τ. For this we set $M_i := (M_{i+1} \cup \nabla_i) \setminus \Delta_i$, for each $i \in [\tau - 1]$, .

See Algorithm 1 for a compact step-by-step description of this algorithm. Note that the algorithm implicitly returns "no" ($\perp$) if it fails to perform all steps for all possible guesses.

Algorithm 1 XP-Algorithm for parameter $\ell + \tau$

Input: An instance $\mathcal{I} = (\mathcal{G} = (V, E_1, \ldots, E_\tau), \ell)$ of Multistage Perfect Matching
Output: A solution $\mathcal{M} = (M_1, \ldots, M_\tau)$ for $\mathcal{I}$ or $\perp$, if there is no solution

1: For each possible choice $\nabla_1, \Delta_1, \ldots, \nabla_{\tau-1}, \Delta_{\tau-1}$, do
2: $B_1 \leftarrow E_1$
3: **for** $i = 2$ to τ **do**
4: $B_i \leftarrow ((B_{i-1} \cap E_i) \cup \Delta_{i-1}) \setminus \nabla_{i-1}$
5: **end for**
6: Construct $M'_\tau \subseteq B_\tau \setminus \Delta_{\tau-1}$ such that $M_\tau := M'_\tau \cup \Delta_{\tau-1}$ is a perfect matching for G_τ, or return $\perp$ if this is not possible.
7: **for** $i = \tau - 1$ to 1 **do**
8: $M_i \leftarrow (M_{i+1} \cup \nabla_i) \setminus \Delta_i$
9: **end for**
10: **return** $\mathcal{M} := (M_1, \ldots, M_\tau)$, if $\mathcal{M}$ is a solution for $\mathcal{I}$

The definition ensures that, when Algorithm 1 returns a sequence $\mathcal{M}$, it is always a solution for the specified instance of MPM. It remains to be proven that the algorithm always returns a solution if a solution exists. Using this proof, we will prove Theorem 3.12 at the end of this section.

Lemma 3.13. *Let $(\mathcal{G} = (V, E_1, \ldots, E_\tau), \ell)$ be an instance of* Multistage Perfect Matching. *If a solution for $(\mathcal{G}, \ell)$ exists, then Algorithm 1 returns a sequence $\mathcal{M} := (M_1, \ldots, M_\tau)$ that is a solution.*

Proof. Let $\mathcal{M} := (M_1, \ldots, M_\tau)$ be a solution for $(\mathcal{G}, \ell)$. For each $i \in [\tau - 1]$, let $\nabla_i := M_i \setminus M_{i+1}$ and $\Delta_i := M_{i+1} \setminus M_i$. Then the following statements hold:

- $\nabla_i \subseteq E_i$, since $\nabla_i \subseteq M_i \subseteq E_i$.
- $\Delta_i \subseteq E_{i+1}$, since $\Delta_i \subseteq M_{i+1} \subseteq E_{i+1}$.
- $|\Delta_i| = |\nabla_i| \leq \frac{\ell}{2}$, since both M_i and M_{i+1} are perfect matchings and thus have the same size. This means that to obtain M_{i+1}, one has to add as many edges (Δ_i) as were removed (∇_i) from M_i. Furthermore, both Δ_i and ∇_i cannot contain more than $\frac{\ell}{2}$ edges, since $|\Delta_i \cup \nabla_i| = |M_i \triangle M_{i+1}| \leq \ell$.
- $V(\Delta_i) = V(\nabla_i)$, since every vertex that was matched by an edge of ∇_i in M_i must also be matched by an edge of Δ_i in M_{i+1}.
- $\forall e \in \Delta_i : \sum_{v \in V} |e \cap \{v\}| \leq 1$, since Δ_i is a subset of perfect matching.
- $\forall e \in \nabla_i : \sum_{v \in V} |e \cap \{v\}| \leq 1$, since ∇_i is a subset of perfect matching.

Since all the above-described conditions are fulfilled by each Δ_i and ∇_i, Algorithm 1 can guess the sets $\nabla_1, \Delta_1, \ldots, \nabla_{\tau-1}, \Delta_{\tau-1}$. To finish this proof, we have to show that, with this guess, the algorithm can construct exactly the sequence $\mathcal{M}$. Formally, we must prove that $M_\tau \subseteq B_\tau$. To this end, we prove the following claims, for $\tau \geq 2$, by induction:

$$(1) \quad B_\tau = \Big(\Big(\Big(\ldots((E_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1 \ldots\Big) \cap E_\tau\Big) \cup \Delta_{\tau-1}\Big) \setminus \nabla_{\tau-1}$$
$$(2) \quad M_\tau \subseteq \Big(\Big(\Big(\ldots((E_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1 \ldots\Big) \cap E_\tau\Big) \cup \Delta_{\tau-1}\Big) \setminus \nabla_{\tau-1}$$

1. **Induction base: Claims (1) and (2) hold for $\tau = 2$:**
   Recall the definition of the base sets: $B_1 := E_1$ and $B_2 := ((B_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1$. This yields $B_2 = ((E_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1$, which proves claim (1). Furthermore, it holds that $M_1 \subseteq E_1$, since it is a matching for G_1, and $M_2 = (M_1 \cup \Delta_1) \setminus \nabla_1$ by definition. Since M_2 is a matching for G_2, we can intersect it with E_2 without altering it: $M_2 = ((M_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1$. With everything combined, this yields $M_2 = ((E_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1$, which proves claim (2).

2. **Induction step: If claims (1) and (2) hold for $\tau \geq 2$, then they also hold for $\tau + 1$:**
   For both claims, we can simply utilize the definitions of B_τ and M_τ.
   - For (1): $B_{\tau+1} = ((B_\tau \cap E_{\tau+1}) \cup \Delta_\tau) \setminus \nabla_\tau$
     $= \Big(\Big(\Big(\ldots((E_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1 \ldots\Big) \cap E_{\tau+1}\Big) \cup \Delta_\tau\Big) \setminus \nabla_\tau.$
   - For (2): $M_{\tau+1} = ((M_\tau \cap E_{\tau+1}) \cup \Delta_\tau) \setminus \nabla_\tau$
     $\subseteq \Big(\Big(\Big(\ldots((E_1 \cap E_2) \cup \Delta_1) \setminus \nabla_1 \ldots\Big) \cap E_{\tau+1}\Big) \cup \Delta_\tau\Big) \setminus \nabla_\tau.$

This proves that Algorithm 1 can construct a set $M'_\tau \subseteq B_\tau \setminus \Delta_{\tau-1}$ such that $M_\tau = M'_\tau \cup \Delta_{\tau-1}$. Consequently, for each $i \in [\tau - 1]$, the algorithm will also construct the exact set M_i from the solution $\mathcal{M}$. □

Proof of Theorem 3.12. As the definition of Algorithm 1 and Lemma 3.13 prove, the algorithm finds a solution for an instance of Multistage Perfect Matching if and only a solution exists. Furthermore, the algorithm tests $O(n^{2 \cdot \ell \cdot \tau})$ possibilites with each test running in $O(\tau \cdot (n^2 + \ell)) \cdot T_M(n)$ time. Thus, Multistage Perfect Matching is solvable in $O(n^{2 \cdot \ell \cdot \tau} \cdot \tau \cdot (n^2 + \ell)) \cdot T_M(n)$ time. □

Chapter 4

Intractability Results

In this chapter, we prove hardness results regarding certain parameters for Multistage Perfect Matching. We first show para-NP-hardness for the lifetime τ (Section 4.1) and the symmetric difference ℓ (Section 4.2), respectively, by proving that MPM is NP-hard, even if these parameters are constant. Thereafter, in Section 4.3, we prove W[1]-hardness for the parameter $\ell + \tau$.

4.1 Para-NP-Hardness for τ

In this section, we show that Multistage Perfect Matching is para-NP-hard when parameterized by τ. To this end, we prove NP-hardness of the problem for $\tau = 2$. We first define Multistage Intersection Matching, introduced by Chimani et al. [CTW20], which will be used for the hardness proof:

Multistage Intersection Matching (MIM) [CTW20]
Input: A temporal graph $\mathcal{G} = (V, E_1, E_2, \ldots, E_\tau)$ and an integer $k \in \mathbb{N}$.
Question: Is there a sequence $\mathcal{M} = (M_1, \ldots, M_\tau)$ of edge sets such that
(i) $M_i, i \in [\tau]$, is a perfect matching of graph (V, E_i), and
(ii) for the intersection profit $p(\mathcal{M}) := \sum_{i \in [\tau-1]} |M_i \cap M_{i+1}|$, it holds that $p(\mathcal{M}) \geq k$?

For exactly t stages, we denote MIM by t-Intersection Matching (t-IM).

Theorem 4.1. Multistage Perfect Matching *is NP-hard, even if* $\tau = 2$.

Proof. We give a polynomial-time many-to-one reduction from the NP-hard problem 2-Intersection Matching (2-IM) [CTW20] to Multistage Perfect Matching.

Given an instance $\mathcal{I} := (\mathcal{G} = (V, E_1, E_2), k)$ of 2-IM, we construct an instance $\mathcal{J} := (\mathcal{G} = (V, E_1, E_2), \ell)$ of Multistage Perfect Matching. Set $\ell = |V| - 2k$. We claim that $\mathcal{I}$ is a yes-instance if and only if $\mathcal{J}$ is a yes-instance.

Let $\mathcal{M} = (M_1, M_2)$ be two perfect matchings for G_1 and G_2 with $d := |M_1 \cap M_2|$. Since M_1, M_2 are perfect matchings, $|M_1| = |M_2| = \frac{|V|}{2}$ and thus $|M_1 \setminus M_2| = |M_2 \setminus M_1| = \frac{|V|}{2} - d$. This yields $|M_1 \triangle M_2| = |M_1 \setminus M_2| + |M_2 \setminus M_1| = |V| - 2d$. Thus, $\mathcal{I}$ and $\mathcal{J}$ are both yes-instances if and only if $d \geq k$ (and therefore $\ell \geq |V| - 2d$). □

As explained in Section 2.4, if a parameterized problem is NP-hard for a fixed value of the parameter, it is para-NP-hard in general. Thus, Theorem 4.1 proves para-NP-hardness for Multistage Perfect Matching parameterized by τ.

4.2 Para-NP-Hardness for ℓ

In this section, we show that Multistage Perfect Matching is para-NP-hard when parameterized by ℓ. For that, we prove NP-hardness of the problem for $\ell = 4$.

Theorem 4.2. Multistage Perfect Matching *is NP-hard, even if* $\ell = 4$.

To prove Theorem 4.2, we give a polynomial-time many-to-one reduction from the NP-hard Vertex Cover to Multistage Perfect Matching with $\ell = 4$. To this end, we first define Vertex Cover:

Vertex Cover
Input: An undirected graph $G = (V, E)$ and a number $k \in \mathbb{N}$.
Question: Is there a subset $S \subseteq V$ of vertices such that
(i) $|S| \leq k$ and
(ii) $\forall \{u, v\} \in E : (u \in S \vee v \in S)$?

In other words, each edge of the graph needs to *covered* by at least one vertex in the vertex cover S.

Proposition 4.3. *There is a polynomial-time algorithm that maps any instance* $(G = (V, E), k)$ *of* Vertex Cover *to an equivalent instance* $(\mathcal{G}, \ell)$ *of* Multistage Perfect Matching *with* $\ell = 4$.

In the remainder of this section, we prove Proposition 4.3, which in turn proves Theorem 4.2. We next give the construction of the Multistage Perfect Matching instance, then prove the forward (Section 4.2.1) and backward (Section 4.2.2) direction of the equivalence, and finally put the pieces together in Section 4.2.3.

In a nutshell, given a graph G, we construct a temporal graph $\mathcal{G}$ whose layers can be split into two phases. In the first phase, consisting of k layers, we choose at most $2k$ specific edges for the perfect matching. Each two of these edges represent a vertex that was selected for the vertex cover in G. In the second phase, there is one layer for each edge of G. In each of these layers, two specific vertices need to be matched. These vertices each represent a single edge in G. This reflects the necessity for each edge in G to be covered by at least one vertex in the vertex cover. These specific vertices can only be matched if, for at least one of the endpoints of the corresponding edge, the above-mentioned edges were chosen in the first phase.

Construction 1. Let $\mathcal{I} := (G = (V, E), k)$ be an instance of Vertex Cover. We construct an instance $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ of Multistage Perfect Matching, with $\ell = 4, \tau = k + 2\,|E|$, in the following way:

For each edge $e \in E$, add two vertices x_e^1, x_e^2 to V'. We call these the *edge gadgets*. For each vertex $v \in V$, add four vertices $y_v^1, y_v^2, y_v^3, y_v^4$ to V'. We call these the *vertex gadgets*.

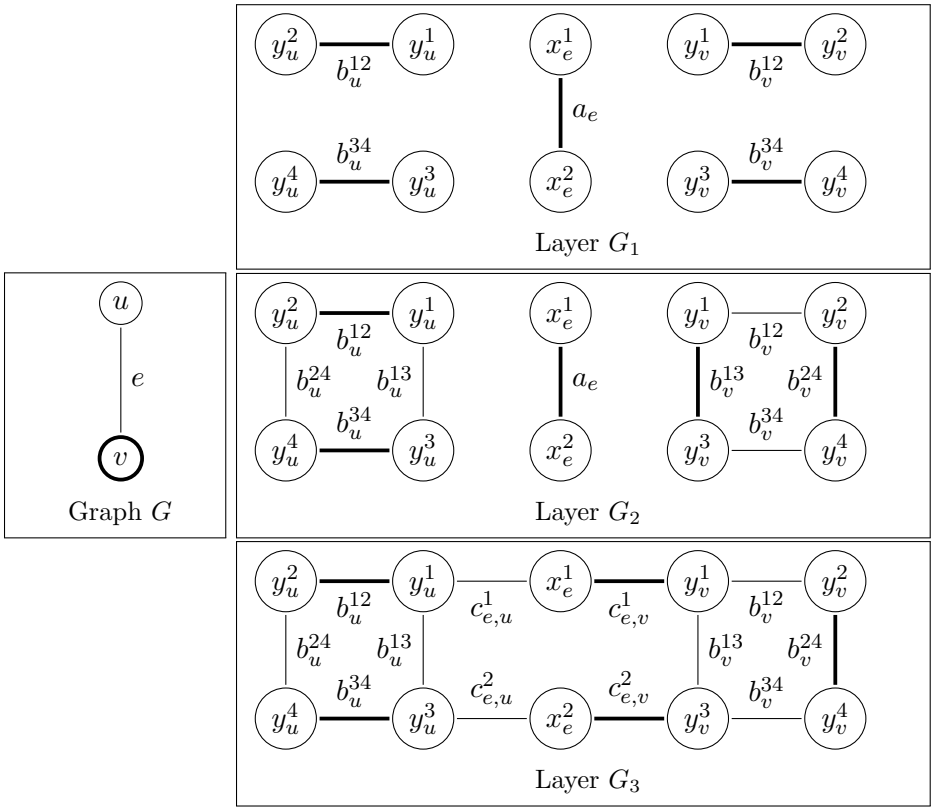


Figure 4.1: An instance $(\mathcal{G}, \ell = 4)$ created from $(G, k = 1)$ using Construction 1. The matching for G_1 is unambiguous. In graph G, the vertex v was selected for the Vertex Cover solution. Thus, in graph G_2, the vertical edges b_v^{13} and b_v^{24} were selected for the matching. Consequently, the vertices x_e^1 and x_e^2 can be matched in graph G_3 by swapping a_e, b_v^{13} for $c_{e,v}^1, c_{e,v}^2$.

We now construct the first $k+1$ layers of $\mathcal{G}$: In each layer $i \in [k+1]$, add the edge $a_e := \{x_e^1, x_e^2\}$ to E_i for each edge $e \in E$, as well as the edges $b_v^{12} := \{y_v^1, y_v^2\}, b_v^{34} := \{y_v^3, y_v^4\}$ for each $v \in V$. In each layer $j \in \{2, \ldots, k+1\}$ add the edges $b_v^{13} := \{y_v^1, y_v^3\}, b_v^{24} := \{y_v^2, y_v^4\}$ for each $v \in V$ to E_j. Let $m = |E|$. We now construct $2m - 1$ additional layers for the temporal graph: First, copy the edges of layer $k+1$ into every layer in $\{k+2, \ldots, k+2m\}$. Then, arbitrarily order the edges of E as $e_1, \ldots, e_m$. Now, for each of these edges $e_i = \{u, v\}, i \in [m]$, remove the edge a_{e_i} from layer $k+2i$. Also add the edges $c_{e_i,u}^1 := \{x_{e_i}^1, y_u^1\}, c_{e_i,u}^2 := \{x_{e_i}^2, y_u^3\}, c_{e_i,v}^1 := \{x_{e_i}^1, y_v^1\}, c_{e_i,v}^2 := \{x_{e_i}^2, y_v^3\}$ to layer $k+2i$. ■

For a visualization of this construction, see Figure 4.1.

We use the following notation:

- Layers $1, 2, \ldots, k+1$ are called *selection layers*,
- layers $k+2, k+4, \ldots, k+2m$ are called *covering layers*, and
- layers $k+1, k+3, \ldots, k+2m-1$ are called *buffering layers*.

Note that layer $k+1$ is both a selection and a buffering layer.

It is not difficult to see that the instance in Construction 1 can be computed in polynomial time. Therefore, it only remains to prove that both instances $\mathcal{I}$ and $\mathcal{J}$ are equivalent as stated in Proposition 4.3. We prove the forward direction in Section 4.2.1 and the backward direction in Section 4.2.2. Thereafter, we put everything together in Section 4.2.3.

4.2.1 Forward Direction

The forward direction of Proposition 4.3 works as follows:

We are given a Vertex Cover solution S for a graph G. We construct the perfect matchings for the selection layers such that, in the last selection layer, for each vertex $v \in S$, the perfect matching contains the edges b_v^{13} and b_v^{24}. Furthermore, for each edge $e \in E$, there is exactly one covering layer in which the vertices x_e^1 and x_e^2 need to be matched by an edge other than a_e. Since S is a vertex cover, there must be a vertex $v \in S$ with $v \in e$. Thus, we can swap the edges a_e and b_v^{13} with $c_{e,v}^1$ and $c_{e,v}^2$ for the matching of this layer. The trivial rest of the solution will be explained in the proof of the following lemma.

Lemma 4.4. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Vertex Cover *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 1. If $\mathcal{I}$ is a yes-instance, then $\mathcal{J}$ is a yes-instance.*

Proof. Let $S := \{v_1, \ldots, v_k\}$ be a vertex cover with $|S| = k$. We use $m = |E|$, $\tau = k+2m$. Based on S, we show how to construct a solution $\mathcal{M} := (M_1, \ldots, M_\tau)$ for $\mathcal{J}$: In every layer $i \in [\tau]$ and for each edge $e \in E$, add the edge a_e to M_i, if it exists. Also, for each vertex $v \in V$, add the edges b_v^{12}, b_v^{34} to M_i. For each $i \in [k]$, replace $b_{v_i}^{12}, b_{v_i}^{34}$ with $b_{v_i}^{13}, b_{v_i}^{24}$ in all M_j with $j > i$. Now, every vertex in V' is matched for the selection layers and the buffering layers. This holds because

- for each $e \in E$, x_e^1 and x_e^2 are matched by a_e,
- for each $v \in V$, y_v^1, y_v^2, y_v^3 and y_v^4 are matched either by b_v^{12} and b_v^{34} or by b_v^{13} and b_v^{24}.

It remains to finish the matchings for the covering layers, because in each covering layer, there exist two vertices x_e^1 and x_e^2 that cannot be covered by the respective edge a_e, considering it is not present in the respective layer. Since S is a vertex cover, for each edge $e_i = \{u, v\} \in E$, at least one of u or v is contained in S. Without loss of generality, let v be contained in S. Let i be the index of e_i used in the construction of $\mathcal{G}$. Now, remove b_v^{13} from M_{k+2i} and add $c_{e_i,v}^1, c_{e_i,v}^2$. All vertices of the covering layers $k+2i, i \in [m]$ are now also matched, since in each layer

- $x_{e_i}^1$ and y_v^1 are matched by $c_{e_i,v}^1$,
- $x_{e_i}^2$ and y_v^3 are matched by $c_{e_i,v}^2$,
- y_v^2 and y_v^4 are matched by b_v^{24}, and
- all other vertices are matched by the same edges as in the other layers:
  - for each $e \in E \setminus \{e_i\}$, x_e^1 and x_e^2 are matched by a_e,

– for each $u \in V \setminus \{v\}$, y_u^1, y_u^2, y_u^3 and y_u^4 are either matched by b_u^{12} and b_u^{34} or by b_u^{13} and b_u^{24}.

Thus, the set $M_i \in \mathcal{M}$ is a perfect matching for G_i, for each $i \in [\tau]$. It remains to show that for each two consecutive matchings $M_i, M_{i+1}, i \in [\tau-1]$, it holds that $|M_i \triangle M_{i+1}| \leq \ell = 4$: We first look at the selection layers. In the first layer, b_v^{12} and b_v^{34} are contained in M_1, for each vertex $v \in V$. Each following layer corresponds to a vertex $v_i \in S, i \in [k]$ and starting with that layer, all following matchings contain $b_{v_i}^{13}$ and $b_{v_i}^{24}$ instead of $b_{v_i}^{12}$ and $b_{v_i}^{34}$. Hence, for each i, it holds that $|M_i \triangle M_{i+1}| = |M_i \setminus M_{i+1}| + |M_{i+1} \setminus M_i| = \left|\{b_{v_i}^{12}, b_{v_i}^{34}\}\right| + \left|\{b_{v_i}^{13}, b_{v_i}^{24}\}\right| \leq 4$. Now we look at the buffering and covering layers. Note that the matchings for all buffering layers are identical to the matching M_{k+1}. Therefore, we only need to consider the transition from a buffering layer to a covering layer and can disregard the opposite. Let $e_i = \{u, v\} \in E, i \in [m]$, be an edge, $c := k + 2i$ be the index of the corresponding covering layer and $b := c - 1$ the preceding buffering layer. Let $v \in S$ without loss of generality. The matching M_b contains a_{e_i}, b_v^{13}, and b_v^{24}. The matching M_c contains $c_{e_i,v}^1, c_{e_i,v}^2$, and b_v^{24}. Thus, $|M_b \triangle M_c| = |M_b \setminus M_c| + |M_c \setminus M_b| = \left|\{a_{e_i}, b_v^{13}\}\right| + \left|\{c_{e_i,v}^1, c_{e_i,v}^2\}\right| \leq 4$.

Since each $M_i, i \in [\tau]$, is a perfect matching and for each two consecutive matchings the symmetric difference is at most ℓ, the sequence $\mathcal{M}$ is a solution for $\mathcal{J}$ and thus, $\mathcal{J}$ is a *yes*-instance. □

4.2.2 Backward Direction

The backward direction of Proposition 4.3 works as follows:

We are given a Multistage Perfect Matching solution for a temporal graph $\mathcal{G}$. We know that in each covering layer of $\mathcal{G}$, an edge gadget needs to be matched by a vertex gadget corresponding to a vertex $v \in V$. We deduce that this is only possible if b_v^{13} and b_v^{24} are contained in the perfect matching for the last selection layer. Thus, we select for the vertex cover solution the vertices for which this condition is true.

Lemma 4.5. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Vertex Cover *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 1. If $\mathcal{J}$ is a yes-instance, then $\mathcal{I}$ is a yes-instance.*

To prove Lemma 4.5, we first define how to construct a solution for Vertex Cover from a solution of Multistage Perfect Matching. Afterwards, we prove that the construction is correct.

Construction 2. Let $\mathcal{I} := (G = (V, E), k)$ be an instance of Vertex Cover and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of Multistage Perfect Matching resulting from Construction 1. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. We construct a set $S \subseteq V$ in the following way:

For each $v \in V$, if $b_v^{13} \in M_{k+1}$ and $b_v^{24} \in M_{k+1}$, then add v to S. ■

Lemma 4.6. *The set $S \subseteq V$ resulting from Construction 2 is a* Vertex Cover *solution for instance $\mathcal{I}$.*

To prove Lemma 4.6, we have to show that

(i) $|S| \leq k$, and
(ii) S is a vertex cover of G.

We first prove (i) with the following lemma:

Lemma 4.7. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Vertex Cover *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 1. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. Then, for each $i \in [k]$ there are at most i vertices $v \in V$ with $b_v^{13} \in M_{i+1}$ and $b_v^{24} \in M_{i+1}$.*

Proof. We provide a proof by induction:

1. **Induction base: Lemma 4.7 holds for $i = 1$:**
   First, we examine the general properties of M_1 and M_2. For each edge $e \in E$, the edge a_e must be contained in both M_1 and M_2. This is due to the fact that the vertices x_e^1 and x_e^2, by construction, have no other incident edges in layers 1 and 2. In the first matching M_1, the edges b_v^{12} and b_v^{34} must be contained for every $v \in V$, since y_v^1, y_v^2, y_v^3 and y_v^4 have no other incident edges in G_1. Since $\ell = 4$, when transitioning from M_1 to M_2, only 4 edges can be changed in the matching. Assume we replace the edges b_v^{12} and b_v^{34} with the edges b_v^{13} and b_v^{24} in M_2, for any vertex $v \in V$. With this replacement, the symmetric difference of M_1 and M_2 is already at 4. Thus, no other replacements can be made and consequently, there can be at most $1 = k$ vertex v with $b_v^{13}, b_v^{24} \in M_2 = M_{i+1}$.

2. **Induction step: If Lemma 4.7 holds for $1 \leq i < k$, then it also holds for $i + 1$:**
   Let $R \subseteq V$ be the set of vertices v for which $b_v^{13}, b_v^{24} \in M_{i+1}$. By assumption, it holds that $|R| \leq i$. Similarly to the induction base, for each edge $e \in E$, the edge a_e must be contained in both M_{i+1} and M_{i+2}. Furthermore, there are at most i vertices $v \in R$ for which $b_v^{13}, b_v^{24} \in M_{i+1}$ and at least $n - i$ vertices $u \in V \setminus R$ for which $b_u^{12}, b_u^{34} \in M_{i+1}$. Since $\ell = 4$, when transitioning from M_{i+1} to M_{i+2}, only 4 edges can be changed in the matching. Assume we replace the edges b_u^{12} and b_u^{34} with the edges b_u^{13} and b_u^{24} in M_{i+2}, for any vertex $u \in V \setminus R$. With this replacement, the symmetric difference of M_{i+1} and M_{i+2} is already at 4. Thus, no other replacements can be made and consequently, there can be at most $(i) + 1$ vertices v with $b_v^{13}, b_v^{24} \in M_{i+2} = M_{(i+1)+1}$. □

We now prove (ii) with the following lemma:

Lemma 4.8. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Vertex Cover *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 1. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. Let $i \in [|E|]$ be the index used for the edge $e_i \in E$ in Construction 1. Then it holds that for each e_i, there exists a vertex $v \in V$ with*

(1) $c_{e_i,v}^1, c_{e_i,v}^2, b_v^{24} \in M_{k+2i}$ *and*
(2) $b_v^{13}, b_v^{24} \in M_{k+1}$.

Proof. Let $e_i = \{u, v\}$ with $u, v \in V$. Let $c := k + 2i$ be the index of the covering layer corresponding to the edge e_i. In layer c, there exist two vertices $x_{e_i}^1$ and $x_{e_i}^2$

without the edge a_{e_i} connecting them. Thus, the matching M_c needs to contain other edges that match these vertices. The edges $c^1_{e_i,u}, c^1_{e_i,v}$ connect the vertex $x^1_{e_i}$ to the vertices y^1_u, y^1_v. The edges $c^2_{e_i,u}, c^2_{e_i,v}$ connect the vertex $x^2_{e_i}$ to the vertices y^3_u, y^3_v. Observe that either $c^1_{e_i,u}, c^2_{e_i,u} \in M_c$ or $c^1_{e_i,v}, c^2_{e_i,v} \in M_c$ must hold. To prove this, assume that $c^1_{e_i,u}, c^2_{e_i,v} \in M_c$: The vertex y^1_u is then matched by $c^1_{e_i,u}$. Therefore, y^2_u can only be matched by b^{24}_u, which also matches y^4_u. Now, it is impossible to match y^3_u, since all its adjacent vertices are already matched by other edges. The proof in case of $c^1_{e_i,v}, c^2_{e_i,u} \in M_c$ is the same (replace u with v). We assume, without loss of generality, that $c^1_{e_i,v}, c^2_{e_i,v} \in M_c$. It follows that $b^{24}_v \in M_c$, since y^2_v and y^4_v can now only be matched by this edge. This finishes the proof for (1).

Now, consider the preceding buffering layer of layer c, having index $b = c - 1$. For every edge $e \in E$ and every $v \in V$, it holds that $a_e \in E_b$,$c^1_{e,v} \notin E_b, c^2_{e,v} \notin E_b$, and thus, $a_e \in M_b$,$c^1_{e,v} \notin M_b, c^2_{e,v} \notin M_b$. As proven, for v and e_i, the edges $c^1_{e_i,v}$ and $c^2_{e_i,v}$ are contained in M_c and the edge a_{e_i} is not contained in M_c. This results in $|M_b \triangle M_c| \geq 3$. Since both M_b and M_c are perfect matchings, they have the same size, which means that the size of their symmetric difference is always even. In this case, the size is at least 3 but at most 4 (the matchings are part of a solution for Multistage Perfect Matching with $\ell = 4$), meaning it must be exactly 4. Therefore, the symmetric difference must contain one more edge from $M_b \setminus M_c$. It is simple to observe that this edge is b^{13}_v. To prove this, note that in layer b, the vertices y^1_v and y^3_v need to be matched by edges in M_b. The only possible options to match the vertices, besides b^{13}_v, would be b^{12}_v and b^{34}_v, which are not contained in M_c. This would result in $|M_b \triangle M_c| \geq 5$, proving b^{13}_v is the only possible choice to match y^1_v and y^3_v. In conclusion, for each buffering layer b and the subsequent covering layer c, it always holds that $M_b \triangle M_c = \{a_{e_i}, b^{13}_v, c^1_{e_i,v}, c^2_{e_i,v}\}$. Furthermore, it holds that $b^{24}_v \in M_c$ and $b^{24}_v \in M_b$. It follows that $M_{b+2} \triangle M_c = \{a_{e_i}, b^{13}_v, c^1_{e_i,v}, c^2_{e_i,v}\}$ (except for $b = k + 2m - 1$), considering the buffering layers b and $b + 2$ are identical. This leads to the observation that, starting with matching M_{k+1}, all matchings M_b of a buffering layer are identical as well. This is due to the fact that the edges a_{e_i} and b^{13}_v, which are removed from the matching for the covering layer, are once again added to the matching for the subsequent buffering layer. If $b^{13}_v, b^{24}_v \in M_b$ and M_b and M_{k+1} are identical, then it follows that $b^{13}_v, b^{24}_v \in M_{k+1}$. This finishes the proof for (2). □

We are now ready to prove Lemma 4.6.

Proof of Lemma 4.6. Let $S \subseteq V$ be the set resulting from Construction 2. According to Lemma 4.7, there are at most k vertices v with $b^{13}_v, b_v 24 \in M_{k+1}$. Considering how S is constructed, it follows that $|S| \leq k$. According to Lemma 4.8, for each $e_i \in E$, there exists a vertex $v \in V$ with $b^{13}_v, b^{24}_v \in M_{k+1}$ and $c^1_{e_i,v}, c^2_{e_i,v}, b^{24}_v \in M_{k+2i}$. By Construction 1, it holds that the vertex v is incident to e_i. Also, by Construction 2, the vertex v must be contained in S. Thus, every edge of G is covered by a vertex of S, which makes S a vertex cover. Consequently, the set S is a solution for $\mathcal{I}$. □

Finally, we prove Lemma 4.5.

Proof of Lemma 4.5. If $\mathcal{J}$ is a yes-instance, then we can construct a set $S \subseteq V$ using Construction 2. Following Lemma 4.6, S is a Vertex Cover solution for instance $\mathcal{I}$. Hence, $\mathcal{I}$ is a yes-instance. □

4.2.3 Proof of Proposition 4.3

We proved the forward and backward direction of Proposition 4.3 in Section 4.2.1 and Section 4.2.2, respectively. It remains to put everything together.

Proof of Proposition 4.3. Let (G, k) be an instance of Vertex Cover and $(\mathcal{G}, \ell = 4)$ be the instance of Multistage Perfect Matching resulting from Construction 1. Observe that Construction 1 runs in polynomial time. We know that if (G, k) is a yes-instance of Vertex Cover, then $(\mathcal{G}, \ell = 4)$ is a yes-instance of Multistage Perfect Matching (Lemma 4.4) and vice versa (Lemma 4.5). □

Since Vertex Cover is NP-hard, Proposition 4.3 proves NP-hardness for MPM with $\ell = 4$ and thus also proves Theorem 4.2. Similarly to Section 4.1, this NP-hardness proves para-NP-hardness for Multistage Perfect Matching parameterized by ℓ.

4.3 W[1]-Hardness for $\ell + \tau$

In this section, we prove W[1]-hardness for Multistage Perfect Matching parameterized by $\ell + \tau$.

Theorem 4.9. Multistage Perfect Matching *with parameter* $\ell + \tau$ *is W[1]-hard.*

To prove Theorem 4.9, we give a parameterized reduction from the W[1]-hard [DF13] Clique problem with parameter k to Multistage Perfect Matching with parameter $\ell + \tau$. To this end, we first define Clique:

Clique
Input: An undirected graph $G = (V, E)$ and a number $k \in \mathbb{N}$.
Question: Is there a subset $C \subseteq V$ of vertices such that
(i) $|S| = k$ and
(ii) $\forall u, v \in C : \{u, v\} \in E$?

In other words, each vertex in the clique must have an edge to all other vertices in the clique, meaning $G[C]$ is a complete graph.

Proposition 4.10. *There is an* $|V|^{O(1)}$ *time algorithm that maps any instance* $(G = (V, E), k)$ *of* Clique *to an equivalent instance* $(\mathcal{G}, \ell)$ *of* Multistage Perfect Matching *with* $\ell = 4k - 2$ *and* $\tau = \frac{k^2+k+2}{2}$.

In the remainder of this section, we prove Proposition 4.10, which in turn proves Theorem 4.9. We next give the construction behind Proposition 4.10, then prove the forward (Section 4.3.1) and backward (Section 4.3.2) directions of the equivalence.

Before we begin the construction, consider the following: Let $G = (V, E)$ be an undirected graph. A vertex subset $C \subseteq V$ with $|C| = k$ is a clique if and only if there is an edge subset $E' \subseteq E$ with $|E'| = \frac{k^2-k}{2} = \binom{k}{2}$ and $\bigcup_{e \in E'} e = C$. In other words: If one can find exactly $\binom{k}{2}$ edges with endpoints only in C, then C must be a clique, since there can be at most $\binom{k}{2}$ edges between k vertices. We use this principle for the reduction from Clique to MPM. In a nutshell, similarly to the reduction to prove Theorem 4.2,

we construct a temporal graph $\mathcal{G}$ from a graph $G = (V, E)$. This temporal graph consists of two phases. In the first phase, we choose for the perfect matching specific edges that represent the vertices in G that form the clique. Since the size of the clique is exactly k, only k of these choices can be made. In the second phase, there are $\binom{k}{2}$ layers. In each of these layers, specific vertices, each representing a single edge $e \in E(G)$, need to be matched. Such a matching can only exist if, for both endpoints of e, the above-mentioned edges were chosen in the first phase. Thus, a perfect matching for each layer can only be found if the corresponding vertex subset $C \subseteq V(G)$ is a clique of size k.

We now describe the formal construction.

Construction 3. Let $\mathcal{I} := (G = (V, E), k)$ be an instance of Clique. To simplify this construction, we assume that $k \geq 5$. We construct an instance $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ of Multistage Perfect Matching with $\ell = 4k - 2$ and $\tau = {}^{(k^2+k+2)}/_{2}$ in the following way. We divide $\mathcal{G}$ into two phases:

- The *selection phase*, with layers $1, \ldots, k+1$.
- The *validating phase*, with layers $k+1, \ldots, k+1+\binom{k}{2} = \frac{k^2+k+2}{2}$.

Observe that layer $k+1$ is part of both the selection phase and the validating phase. Each layer consists of various gadgets, that we describe now:

- For each $i \in [k]$, add the vertex set $Z_i := \{z_i^1, z_i^2\}$. We call Z_i a *selection gadget*. Also add the edge $c_i := \{z_i^1, z_i^2\}$ to layers $[i]$. For a visualization, see Figure 4.2.

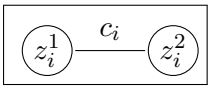


Figure 4.2: Selection gadget $Z_i, i \in [k]$ in layers $[i]$

- For each vertex $v \in V$, add the vertex set $Y_v := \{y_v^1, \ldots, y_v^{4k-4}\}$ to V'. We call Y_v a *vertex gadget*. For each $i \in [4k-5]$, also add the edge $b_v^i := \{y_v^i, y_v^{i+1}\}$ to all layers. We call b_v^i the *outer edges* if i is odd, and *inner edges* otherwise. For a visualization, see Figure 4.3.

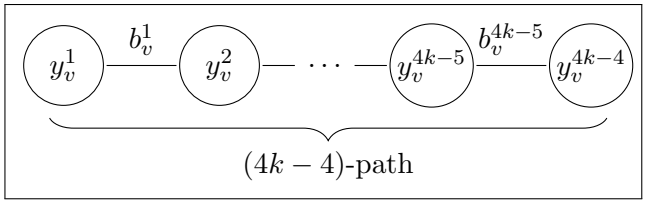


Figure 4.3: Vertex gadget $Y_v, v \in V$ in layers $[\tau]$

- For each edge $e \in E$, add the vertex set $X_e := \{x_e^1, \ldots, x_e^6\}$ to V'. We call X_e an *edge gadget*. Also add the edges $a_e^1 := \{x_e^1, x_e^2\}, a_e^2 := \{x_e^3, x_e^4\}, a_e^3 := \{x_e^5, x_e^6\}$ to all layers. For a visualization, see Figure 4.4.

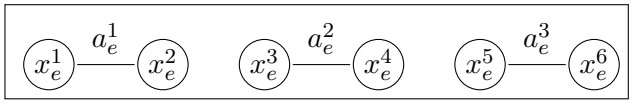


Figure 4.4: Edge gadget $X_e, e \in E$ in layers $[\tau]$

- For each $i \in [\binom{k}{2}]$, add the vertex set $W_i := \{w_i^1, w_i^2\}$ to V'. We call W_i a *validating gadget*. Also add the edge $d_i := \{w_i^1, w_i^2\}$ to layers $[k+i]$. For a visualization, see Figure 4.5.

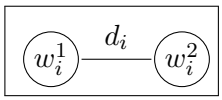


Figure 4.5: Validation gadget $W_i, i \in [\binom{k}{2}]$ in layers $[k+i]$

- Add the vertex set $R := \{r_1, \ldots, r_{4k-14}\}$ to V'. We call R the *delta gadget*. Also add the edges $s_2 := \{r_2, r_3\}, s_3 := \{r_3, r_4\}, \ldots, s_{4k-15} := \{r_{4k-15}, r_{4k-14}\}$ to all layers. For each $i \in [\tau]$:

  - If $i \leq k+1$ or $i-k$ is odd, add the edge $s_1 := \{r_1, r_2\}$ to layer i.
  - If $i > k+1$ and $i-k$ is even, add the edge $s_{4k-14} := \{r_{4k-14}, r_1\}$ to layer i.

  Thus, in the selection phase (layers 1 to $k+1$) the delta gadget is a $(4k-14)$-path from r_1 to r_{4k-14}. In the validating phase, the delta gadget alternates (for every other layer) between being a $(4k-14)$-path from r_1 to r_{4k-14} and being a $(4k-14)$-path from r_2 to r_1. The purpose of this gadget is to reduce the effective symmetric difference, that is possible for the matchings of two consecutive layers in the validating phase, to 12. This will be explained in detail in Section 4.3.2. For a visualization, see Figure 4.6

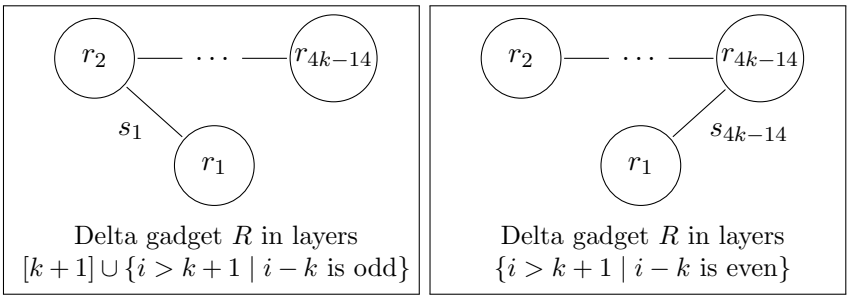
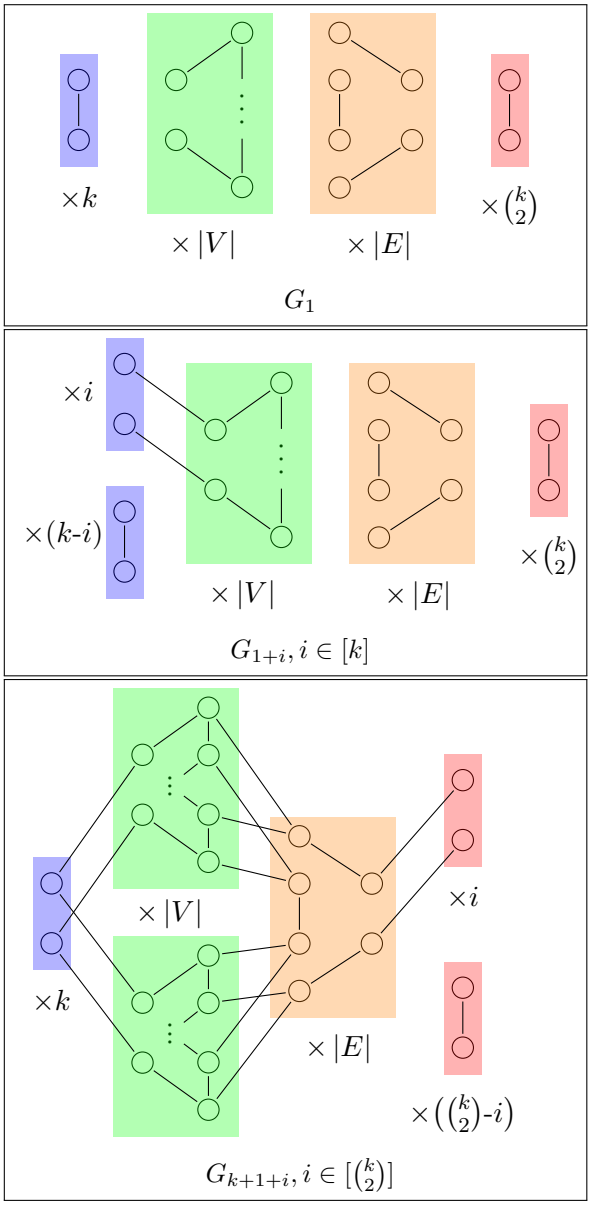


Figure 4.6: Delta gadget R



- Selection gadget $Z_i, i \in [k]$
- Vertex gadget $Y_v, v \in V$
- Edge gadget $X_e, e \in E$
- Validating gadget $W_i, i \in [\binom{k}{2}]$

With every transition in layers $2, \ldots, k+1$, one selection gadget Z_i "loses" the internal edge and is instead connected via two edges to each vertex gadget $Y_v, v \in V$.

Starting with layer $k+2$, each edge gadget $X_e, e = \{u, v\} \in E$ is connected via $2k-2$ edges to the two vertex gadgets Y_u and Y_v corresponding to the vertices u, v incident to e. With every transition in layers $k+2, \ldots, \tau$, one validating gadget W_i "loses" the internal edge and is instead connected to each edge gadget $Y_e, e \in E$.

Figure 4.7: Overview of the gadgets of $\mathcal{G} = (V, E_1, \ldots, E_\tau)$ constructed from an instance $(G = (V, E), k)$ of Vertex Cover using Construction 3. The delta gadget R is excluded.

Now we construct the edges that connect the gadgets:

- For each $e = \{u, v\} \in E$ and each $i \in [k-1]$, add the edges $g^{2i-1}_{e,v} := \{y^{4i-2}_v, x^2_e\}, g^{2i}_{e,v} := \{y^{4i-1}_v, x^3_e\}, g^{2i-1}_{e,u} := \{y^{4i-2}_u, x^4_e\}, g^{2i}_{e,u} := \{y^{4i-1}_u, x^5_e\}$ to layers $[\tau] \setminus [k+1]$.

- For each $i \in [k]$ and each $v \in V$, add the edges $f^1_{v,i} := \{z^1_i, y^1_v\}, f^2_{v,i} := \{z^2_i, y^{4k-4}_v\}$ to layers $[\tau] \setminus [i]$.

- For each $i \in [\binom{k}{2}]$ and each $e \in E$, add the edges $h^1_{e,i} := \{x^1_e, w^1_i\}, h^2_{e,i} := \{x^6_e, w^2_i\}$ to layers $[\tau] \setminus [k+i]$. ■

For a visualization of this construction, see Figure 4.7.

Since each of the $O(k^2)$ layers of the temporal graph in Construction 3 has $O(k^2 + |V|^2)$ vertices and $O(k^2 |V|^2)$ edges, it can be constructed in $|V|^{O(1)}$ time (note that $k \leq |V|$). Therefore, it only remains to prove that both instances $\mathcal{I}$ and $\mathcal{J}$ are equivalent. We prove the forward direction in Section 4.3.1 and the backward direction in Section 4.3.2.

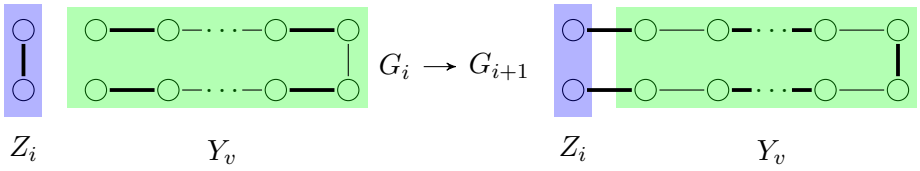
4.3.1 Forward Direction

The forward direction of Proposition 4.10 works as follows: We assume the instance of Clique is a yes-instance. In the first phase, let each $i \in [k]$ correspond to one of the k vertices in the clique solution. Starting with layer $i+1$, respectively, we then select for the matchings the edges that connect the vertex gadget of the corresponding vertex with one of the selection gadgets. To this end, we also need to select for the matchings the inner edges (instead of the outer edges like in layer 1) of each of the vertex gadgets. At layer $k+1$, we then have exactly k vertex gadgets for which the inner edges are contained in the matching. In the second phase, for each of the $\binom{k}{2}$ transitions there is one validating gadget whose two vertices need to be matched by edges other than the one incident to both of them. Thus, they need to be matched by edges connecting them to an edge gadget. For this to work, the matching must also contain the edges that connect this edge gadget to the two vertex gadgets corresponding to the incident vertices of its respective edge. To this end, the inner edges of these vertex gadgets must be contained in the matching of layer $k+1$, as described for the first phase. Since the k vertices corresponding to these vertex gadgets form a clique, there must be exactly $\binom{k}{2}$ edges that fulfill the described constraint. Thus, we can construct a Multistage Perfect Matching solution.

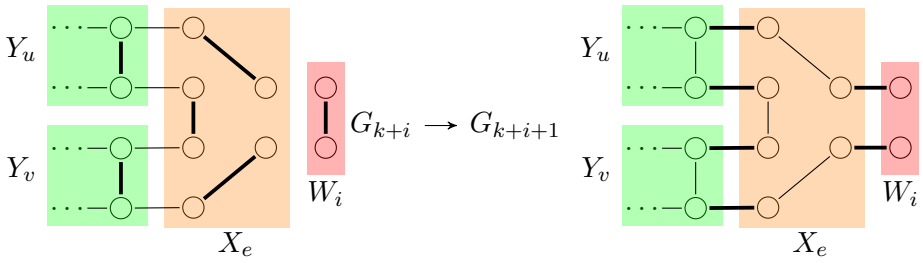
Lemma 4.11. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Clique *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 3. If $\mathcal{I}$ is a yes-instance, then $\mathcal{J}$ is a yes-instance.*

Proof. Let $C \subseteq V$ be a clique with $|C| = k$. Recall that $\tau = \frac{k^2+k+2}{2}$. Based on C, we can construct a solution $\mathcal{M} := (M_1, \ldots, M_\tau)$ for $\mathcal{J}$. Arbitrarily order the vertices of C as $v_1, \ldots, v_k$. Let $k^* := \binom{k}{2} = \frac{k^2-k}{2}$. Also order the edges of $G[C]$ as $e_1 := \{v_1, v_2\}, \ldots, e_{k-1} := \{v_1, v_k\}, e_k := \{v_2, v_3\}, \ldots, e_{2k-3} := \{v_2, v_k\}, \ldots, e_{k^*} := \{v_{k-1}, v_k\}$. Now we add the edges to the matchings:

- For each $i \in [k]$, add the edge c_i to the matchings $M_1, \ldots, M_i$.
- For each $i \in [\binom{k}{2}]$, add the edge d_i to the matchings $M_1, \ldots, M_{k+i}$.
- For each $e \in E$, add the edges a^1_e,a^2_e and a^3_e to all matchings $M_1, \ldots, M_\tau$.
- Add the edges s_1,$s_3, \ldots, s_{4k-15}$ to the matchings of layers $[k+1]$.



Selection gadget $Z_i, i \in [k]$ and vertex gadget $Y_v, v \in C$. Displayed is the transition from layer i to layer $i+1$.



Validating gadget $W_i, i \in [\binom{k}{2}]$, edge gadget $X_e, e = \{u, v\} \in E(G[C])$ and vertex gadgets Y_u, Y_v with $u, v \in C$. Displayed is the transition from layer $k+i$ to layer $k+i+1$.

Figure 4.8: Example of a transition in the selection phase and the validating phase, respectively. The bold edges are contained in the matching M_i of the respective layer $i \in [\tau]$.

- For layers $i \in [\tau] \setminus [k+1]$: If $i-k$ is odd, add the edges $s_1, s_3, \ldots, s_{4k-15}$ to the matching M_i. Else, add the edges $s_2, s_4, \ldots, s_{4k-14}$ to the matching M_i.
- For each $v_i \in C, i \in [k]$:
  - Add the edges $b^1_{v_i}, b^3_{v_i}, \ldots, b^{4k-5}_{v_i}$ to the matchings of layers $[i]$
  - Add the egdes $b^2_{v_i}, b^4_{v_i}, \ldots, b^{4k-6}_{v_i}$ and the edges $f^1_{v_i,i}, f^2_{v_i,i}$ to the matchings of layers $[\tau] \setminus [i]$.
- For each $v \in V \setminus C$, add the edges $b^1_v, b^3_v, \ldots, b^{4k-5}_v$ to all matchings $M_1, \ldots, M_\tau$.
- Let $e_i = \{v_\alpha, v_\beta\} \in E(G)$ with $i \in [k^*]$ and $v_\alpha, v_\beta \in C$. Let $1 \leq \alpha < \beta \leq k$. For layers $j \in [\tau] \setminus [k+i]$:
  - Remove $a^1_{e_i}, a^2_{e_i}, a^3_{e_i}$ from the matching M_j.
  - Add the edges $h^1_{e_i,i}, h^2_{e_i,i}$ to the matching M_j.
  - Replace $b^{4\beta-6}_{v_\alpha}$ with the edges $g^{2\beta-3}_{e_i,v_\alpha}, g^{2\beta-2}_{e_i,v_\alpha}$ in the matching M_j.
  - Replace $b^{4\alpha-2}_{v_\beta}$ with the edges $g^{2\alpha-1}_{e_i,v_\beta}, g^{2\alpha}_{e_i,v_\beta}$ in the matching M_j.

This finishes the construction of $\mathcal{M}$. For a visualization, see Figure 4.8. To prove that $\mathcal{M}$ is a solution for $\mathcal{J}$, we have to show the following:

(i) For each $i \in [\tau]$, the set M_i is a perfect matching for graph G_i.
(ii) For each $i \in [\tau - 1]$, it holds that $|M_i \triangle M_{i+1}| \leq \ell = 4k - 2$.

We prove (i) for the selection phase and the validating phase, respectively:

1. Layers with index $j \in [k+1]$ (selection phase):

   - For each $i \in [\binom{k}{2}]$, the vertices w_i^1 and w_i^2 are matched by d_i.
   - For each $e \in E$, the vertices $x_e^1, x_e^2, x_e^3, x_e^4, x_e^5$ and x_e^6 are matched by a_e^1, a_e^2, a_e^3.
   - For each $i \in [2k-7]$, the vertices r_{2i-1}, r_{2i} are matched by s_{2i-1}.
   - For each $v \in V \setminus C$ and each $p \in [2k-2]$, the vertices y_v^{2p-1} and y_v^{2p} are matched by b_v^{2p-1}.
   - For each $i \in [k] \setminus [j-1]$, the vertices z_i^1 and z_i^2 are matched by c_i.
   - For each $i \in [j-1]$:
     - Let $v_i \in C$. The vertices z_i^1, z_i^2 and $y_{v_i}^1, y_{v_i}^{4k-4}$ are matched by $f_{v_i,i}^1, f_{v_i,i}^2$.
     - For each $v_i \in C$ and each $p \in [2k-3]$, the vertices $y_{v_i}^{2p}$ and $y_{v_i}^{2p+1}$ are matched by $b_{v_i}^{2p}$.

   We conclude that all vertices in V' are matched by exactly one edge in M_j, so the set M_j is a perfect matching for G_j.

2. Layers with index $j \in [\tau] \setminus [k+1]$ (validating phase):

   - For each $i \in [k]$ and $v_i \in C$, the vertices z_i^1, z_i^2 and $y_{v_i}^1, y_{v_i}^{4k-4}$ are matched by $f_{v_i,i}^1, f_{v_i,i}^2$.
   - For each $v \in V \setminus C$ and each $p \in [2k-2]$, the vertices y_v^{2p-1} and y_v^{2p} are matched by b_v^{2p-1}.
   - If $j-k$ is odd: For each $i \in [2k-7]$, the vertices r_{2i-1}, r_{2i} are matched by s_{2i-1}.
   - If $j-k$ is even: For each $i \in [2k-8]$, the vertices r_{2i}, r_{2i+1} are matched by s_{2i}. The vertices r_{4k-14} and r_1 are matched by s_{4k-14}.
   - For each $i \in [\binom{k}{2}] \setminus [j-(k+1)]$, the vertices w_i^1 and w_i^2 are matched by d_i.
   - For each $e \in E(G) \setminus E(G[C])$, the vertices $x_e^1, x_e^2, x_e^3, x_e^4, x_e^5$ and x_e^6 are matched by a_e^1, a_e^2, a_e^3.
   - For each $i \in [j-(k+1)]$: Let $e_i = \{v_\alpha, v_\beta\} \in E(G[C])$ and $1 \leq \alpha < \beta \leq k$.
     - The vertices w_i^1 and $x_{e_i}^1$ are matched by $h_{e_i,i}^1$.
     - The vertices $y_{v_\alpha}^{4\beta-6}$ and $x_{e_i}^2$ are matched by $g_{e_i,v_\alpha}^{2\beta-3}$.
     - The vertices $y_{v_\alpha}^{4\beta-5}$ and $x_{e_i}^3$ are matched by $g_{e_i,v_\alpha}^{2\beta-2}$.
     - The vertices $y_{v_\beta}^{4\alpha-2}$ and $x_{e_i}^4$ are matched by $g_{e_i,v_\beta}^{2\alpha-1}$.
     - The vertices $y_{v_\alpha}^{4\alpha-1}$ and $x_{e_i}^5$ are matched by $g_{e_i,v_\beta}^{2\alpha}$.
     - The vertices w_i^2 and $x_{e_i}^6$ are matched by $h_{e_i,i}^2$.

   Since all vertices of V' are matched by exactly one edge in M_j, the set M_j is a perfect matching for G_j.

We prove (ii) for the selection phase and the validating phase, respectively:

1. Symmetric difference between M_j and M_{j+1} with $j \in [k]$ (selection phase):

   - The vertices of all validating, edge and delta gadget are matched by exactly the same edges in matching M_j and M_{j+1}.
   - Let $v \in V \setminus C$. The vertices of the vertex gadgets Y_v are matched by exactly the same edges in matching M_j and M_{j+1}.
   - For the remaining vertex gadgets and the selection gadgets, let $i \in [k]$ and $v_i \in C$. Let $B_i^1 := \{b_{v_i}^1, b_{v_i}^3, \ldots, b_{v_i}^{4k-5}\} \cup \{c_i\}$ and $B_i^2 := \{b_{v_i}^2, b_{v_i}^4, \ldots, b_{v_i}^{4k-6}\} \cup \{f_{v_i,i}^1, f_{v_i,i}^2\}$. There are three cases to consider:
     - If $i < j$, then it holds that $B_i^1 \subseteq M_j$ and $B_i^1 \subseteq M_{j+1}$.
     - If $i > j$, then it holds that $B_i^2 \subseteq M_j$ and $B_i^2 \subseteq M_{j+1}$.
     - If $i = j$, then it holds that $B_i^1 \subseteq M_j$ and $B_i^2 \subseteq M_{j+1}$.

   Putting all of this together, it follows that $|M_j \triangle M_{j+1}| = |M_j \setminus M_{j+1}| + |M_{j+1} \setminus M_j|$ $= \left|B_j^1\right| + \left|B_j^2\right| = (2k-1) + (2k-1) = 4k-2$.

2. Symmetric difference between M_j and M_{j+1} with $j \in [\tau-1] \setminus [k]$ (validating phase):

   - Let $v \in V \setminus C$. The vertices of the vertex gadgets Y_v are matched by exactly the same edges in matching M_j and M_{j+1}.
   - Let $e \in E(G) \setminus E(G[C])$. Matchings M_j and M_{j+1} contain a_e^1, a_e^2 and a_e^3.
   - Let $i \in [k]$ and $v_i \in C$. Matchings M_j and M_{j+1} contain $f_{v_i,i}^1$ and $f_{v_i,i}^2$.
   - For the delta gadget:
     Let $S_1 := \{s_1, s_3, \ldots, s_{4k-15}\}$ and $S_2 := \{s_2, s_4, \ldots, s_{4k-14}\}$. Then, it holds that
     - if $j-k$ is odd, then $S_1 \subseteq M_j$ and $S_2 \subseteq M_{j+1}$, and
     - if $j-k$ is even, then $S_2 \subseteq M_j$ and $S_1 \subseteq M_{j+1}$.
   - For the remaining vertex and edge gadgets and all validating gadgets:
     Let $i = j-(k+1), e_i = \{v_\alpha, v_\beta\} \in E(G[C]), \alpha, \beta \in [k]$ with $\alpha < \beta$.
     Let $A_j^1 := \{d_i, a_{e_i}^1, a_{e_i}^2, a_{e_i}^3, b_{v_\alpha}^{4\beta-6}, b_{v_\beta}^{4\alpha-2}\}$
     and $A_j^2 := \{h_{e_i,i}^1, h_{e_i,i}^2, g_{e_i,v_\alpha}^{2\beta-3}, g_{e_i,v_\alpha}^{2\beta-2}, g_{e_i,v_\beta}^{2\alpha-1}, g_{e_i,v_\beta}^{2\alpha}\}$.
     Then it holds that $A_j^1 \subseteq M_j$ and $A_j^2 \subseteq M_{j+1}$.

   Putting all of this together, it follows that

   - if $j-k$ is odd, then $|M_j \triangle M_{j+1}| = |M_j \setminus M_{j+1}| + |M_{j+1} \setminus M_j|$
     $= \left|S_1 \cup A_j^1\right| + \left|S_2 \cup A_j^2\right| = ((2k-7)+6) + ((2k-7)+6) = 4k-2$, and
   - if $j-k$ is even, then $|M_j \triangle M_{j+1}| = |M_j \setminus M_{j+1}| + |M_{j+1} \setminus M_j|$
     $= \left|S_2 \cup A_j^1\right| + \left|S_1 \cup A_j^2\right| = ((2k-7)+6) + ((2k-7)+6) = 4k-2$.

We have shown that both conditions (i) and (ii) are fulfilled by all matchings in $\mathcal{M}$. Thus, $\mathcal{M}$ is a solution for $\mathcal{J}$ and, consequently, $\mathcal{J}$ is a yes-instance. □

4.3.2 Backward Direction

The backward direction of Proposition 4.10 works as follows: We assume the instance of Multistage Perfect Matching is a yes-instance. We then claim that the following is a clique in graph G: the set of vertices v for which it holds that, for any $i \in [k]$, the edges $f^1_{v,i}$ and $f^2_{v,i}$ are contained in the matching for layer $k+1$. Recall that $f^1_{v,i}$ and $f^2_{v,i}$ are the edges that connect the vertex gadget Y_v with the selection gadget Z_i. We prove that this set is a clique by first showing that there must be exactly k vertices for whose corresponding gadgets the previous constraint is fulfilled. Afterwards, we prove that there exist exactly $\binom{k}{2}$ edges that are incident only to the vertices which were selected for the supposed clique.

Lemma 4.12. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Clique *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 3. If $\mathcal{J}$ is a yes-instance, then $\mathcal{I}$ is a yes-instance.*

To prove Lemma 4.12, we first define how to construct a solution for Clique from a solution of Multistage Perfect Matching. Afterwards, we prove that the construction is correct.

Construction 4. Let $\mathcal{I} := (G = (V, E), k)$ be an instance of Clique and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of Multistage Perfect Matching resulting from Construction 3. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. We construct a set $C \subseteq V$ in the following way:

For each $v \in V$, if $f^1_{v,i} \in M_{k+1}$ and $f^2_{v,i} \in M_{k+1}$ for any $i \in [k]$, then add v to C. ■

Lemma 4.13. *The set $C \subseteq V$ resulting from Construction 4 is a* Clique *solution for instance $\mathcal{I}$.*

To prove Lemma 4.13, we have to show that

(i) $|C| = k$, and that
(ii) C is a clique in G.

We first prove (i) with the following lemma:

Lemma 4.14. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Clique *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 3. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. Then, for each $i \in [k] \cup \{0\}$, there are exactly i distinct vertices $v \in V$ with $f^1_{v,j} \in M_{i+1}$ and $f^2_{v,j} \in M_{i+1}$, for some $j \in [i]$.*

Proof. We provide a proof by induction.

1. **Induction base: Lemma 4.14 holds for $i = 0$:**
   No edge $f^1_{v,j}$ or $f^2_{v,j}$ is present in layer 1 and can therefore not be contained in M_1. Thus, there are exactly 0 distinct vertices $v \in V$ with $f^1_{v,j} \in M_1$ and $f^2_{v,j} \in M_1$, for some $j \in [0]$.

2. **Induction step: If Lemma 4.14 holds for $i \geq 0$, then it also holds for $i+1$:** Let $R \subseteq V$ be the set of vertices v for which $f^1_{v,j} \in M_{i+1}$ and $f^2_{v,j} \in M_{i+1}$, for any $j \in [i]$. By assumption, it holds that $|R| = i$. First, we examine the general properties of M_{i+1} and M_{i+2}.

   - For each edge $e \in E$, the edges a^1_e, a^2_e, a^3_e must be contained in both M_{i+1} and M_{i+2}.
   - For each $j \in [\binom{k}{2}]$, the edge d_j must be contained in both M_{i+1} and M_{i+2}.
   - For each vertex $v \in V \setminus R$, the edges $b^1_v, b^3_v, \ldots, b^{4k-5}_v$ must be contained in M_{i+1}.
   - The edge c_{i+1} must be contained in M_{i+1}.

   In layer $i+2$, the edge c_{i+1} does not exist anymore. Therefore, the vertices z^1_{i+1} and z^2_{i+1} must be matched by other edges in M_{i+2}. To this end, the only possible options are the edges $f^1_{u,i+1}$ and $f^2_{v,i+1}$, for each vertex $u, v \in V \setminus R$. Note that no vertex $w \in R$ can be chosen, since each y^1_w is already matched by $f^1_{w,j}$ for any $j \in [i]$. We will show that if $f^1_{u,i+1} \in M_{i+2}$, then $u = v$, so that $f^2_{u,i+1} \in M_{i+1}$. Let $u \in V \setminus R$ be the vertex for which y^1_u is covered by $f^1_{u,i+1}$. We will prove by induction that, for each $j \in \{2, 4, \ldots, 4k-6\}$, the edge b^j_u must be contained in M_{i+2}.

   (a) **Induction base: $b^2_u \in M_{i+2}$:**
   Since y^1_u is already matched by $f_{u,i+1}$, the edge b^1_u cannot be contained in M_{i+2}. The vertex y^2_u has degree 1 in $G_{i+2} - \{b^1_u\}$. It can only be matched by b^2_u, which must therefore be contained in M_{i+2}.

   (b) **Induction step: If $b^2_u, b^4_u, \ldots, b^j_u \in M_{i+2}$, then $b^{j+2}_u \in M_{i+2}$:**
   Since y^{j-1}_u is already matched by y^{j-2}_u, the edge b^{j-1}_u cannot be contained in M_{i+2}. The vertex y^j_u has degree 1 in $G_{i+2} - \{b^{j-1}_u\}$. It can only be matched by b^j_u, which must therefore be contained in M_{i+2}.

   Since, for each $j \in \{1, 3, \ldots, 4k-5\}$, the edge b^j_u is contained in M_{i+1}, the size of the symmetric difference of M_{i+1} and M_{i+2} is currently

$$\begin{aligned} |M_{i+1} \triangle M_{i+2}| &= |M_{i+1} \setminus M_{i+2}| + |M_{i+2} \setminus M_{i+1}| \\ &= \left|\{c_{i+1}\} \cup \{b^1_u, b^3_u, \ldots, b^{4k-5}_u\}\right| + \left|\{f^1_{u,i+1}\} \cup \{b^2_u, b^4_u, \ldots, b^{4k-6}_u\}\right| \\ &= (1 + (2k-2)) + (1 + (2k-3)) = 4k-3. \end{aligned}$$

   Thus, the symmetric difference permits only one additional edge to be contained in $M_{i+2} \setminus M_{i+1}$. Since both z^2_{i+1} and y^{4k-4}_u must be matched in M_{i+2}, this edge must be $f^2_{u,i+1}$, which is therefore contained in M_{i+2}. Consequently, there are exactly $|R|+1 = i+1$ vertices $u \in V$ with $f^1_{u,j}, f^2_{u,j} \in M_{i+2}$, for any $j \in [i+1]$. □

We now prove that the set $C \subseteq V$ resulting from Construction 4 is a clique. To this end, we first take a closer look at the delta gadget in the validating phase. Let $i \in [\tau]$ be the index of a layer.

- If $i \leq k+1$ or $i-k$ is odd, then there is a $(4k-14)$-path from r_1 to r_{4k-14}. Since all vertices in the delta gadget have no other edges, every other edge $s_{2j-1}, j \in [2k-7]$ of this path must be contained in each M_i.

- If $i > k+1$ and $i-k$ is even, then there is a $(4k-14)$-path from r_2 to r_1. Since all vertices in the delta gadget have no other edges, every other edge $s_{2j}, j \in [2k-7]$ of this path must be contained in each M_i.

Thus, starting with $i = k+1$, for each transition from matching M_i to M_{i+1}, the edges in the delta gadget must always be replaced (*swapped*). There are $2k-7$ edges that cover the vertices of a $(4k-14)$-path. Hence, the effective symmetric difference for all other gadgets results in $\ell' = \ell - 2 \cdot (2k-7) = (4k-2) - (4k-14) = 12$. Using this value for ℓ', we can perform the actual proof, with the following lemma.

Lemma 4.15. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Clique *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 3. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. Then there are exactly $\binom{k}{2}$ edges $e = \{u, v\} \in E$ with*

(1) $f^1_{u,j}, b^2_u, b^4_u, \ldots, b^{4k-6}_u, f^2_{u,j} \in M_{k+1}$ *for any $j \in [k]$, and*
(2) $f^1_{v,j}, b^2_v, b^4_v, \ldots, b^{4k-6}_v, f^2_{v,j} \in M_{k+1}$ *for any $j \in [k]$.*

We will prove Lemma 4.15 using another lemma. In the following, $\triangle_R$ denotes the *modified symmetric difference*, which excludes the edges of the delta gadget R and must therefore equal $\ell' = 12$, for each validating layer.

Lemma 4.16. *Let $\mathcal{I} := (G = (V, E), k)$ be an instance of* Clique *and $\mathcal{J} := (\mathcal{G} = (V', E_1, \ldots, E_\tau), \ell)$ be the instance of* Multistage Perfect Matching *resulting from Construction 3. Let $\mathcal{M} = (M_1, \ldots, M_\tau)$ be a solution for $\mathcal{J}$. Then, for each $i \in [\binom{k}{2}] \cup \{0\}$, there are exactly i distinct edges $e_i = \{u, v\} \in E$ with*

$$M_{k+i} \triangle_R M_{k+i+1} = \{a^1_{e_i}, a^2_{e_i}, a^3_{e_i}, d_i, b^{2p}_u, b^{2q}_v\} \cup \{h^1_{e_i,i}, h^2_{e_i,i}, g^p_{e_i,u}, g^{p+1}_{e_i,u}, g^q_{e_i,v}, g^{q+1}_{e_i,v}\},$$

for any $p, q \in \{1, 3, ..., 2k-3\}$.

Proof. We provide a proof by induction.

1. **Induction base: Lemma 4.16 holds for $i = 0$:**
   Neither in layer k nor in layer $k+1$, the edges $h^1_{e_i,i}, h^2_{e_i,i}, g^p_{e_i,u}, g^{p+1}_{e_i,u}, g^q_{e_i,v}, g^{q+1}_{e_i,v}$ exist, for any i, e_i, u, v, p, q. Thus, there are exactly 0 edges for which Lemma 4.16 holds.

2. **Induction step: If Lemma 4.16 holds for $i-1 \geq 0$, then it also holds for i:**
   Let $E_C \subseteq E$ be the set of edges for which Lemma 4.16 holds. By assumption, it holds that $|E_C| = i-1$. First, examine the matching M_{k+i}:

   - For each $j \in [\binom{k}{2}] \setminus [i-1]$, the edge d_j must be contained in M_{k+i}.
   - For each $e \in E \setminus E_C$, the edges a^1_e, a^2_e, a^3_e must be contained in M_{k+i}.

- Following Lemma 4.14 and the assumption of the induction step, there are exactly k vertices $v \in V$ for which $f^1_{v,j}, b^2_v, b^4_v, \ldots, b^{4k-6}_v, f^2_{v,j} \in M_{k+1}$ for any $j \in [k]$. By the assumption of the induction step, for each $j \in [i-1]$, it holds: in the transition from layer $k+j$ to layer $k+j+1$, there are two (distinct) vertices $u, v \in V$, for which b^{2p}_u and b^{2q}_v are removed from the matching M_{k+j}, for any $p, q \in \{1, 3, \ldots, 2k-3\}$. Without loss of generality, let $P_u \subseteq \{1, 3, \ldots, 2k-3\}$ be the set of indices p, for which these removals were made (for vertex $u \in V$). Note that $|P_u| \leq i-1$. Thus, it holds that there are at least $(k-1) - |P_u| \geq k-i$ edges b^{2p}_u, for which $b^{2p}_u \in M_{k+i}$, for each $p \in \{1, 3, \ldots, 2k-3\} \setminus P_u$.

Now, observe that the edge $d_i = \{w^1_i, w^2_i\}$ does not exist in layer $k+i+1$, whereas it existed in every layer before $k+i$. Hence, the vertices w^1_i and w^2_i must be matched by other edges in M_{k+i+1}. To this end, the only possible options are the edges $h^1_{e,i}$ and $h^2_{e',i}$, for some edges $e, e' \in E$. We will show that if $h^1_{e,i} \in M_{k+i+1}$, then $e = e'$, so that $h^2_{e,i} \in M_{k+i+1}$. Let $e = \{u, v\} \in E$ be the edge for which $h^1_{e,i} \in M_{k+i+1}$. This results in the following causal chain:

(a) The edge a^1_e cannot be contained in M_{k+i+1}, since x^1_e is already matched by $h^1_{e,i}$.

(b) In contrast to M_{k+i}, the vertex x^2_e is not matched by a^1_e in M_{k+i+1}. It must be matched by one of the edges $g^p_{e,u}$ for each $p \in \{1, 3, \ldots, 2k-3\} \setminus P_u$. Let $p \in \{1, 3, \ldots, 2k-3\} \setminus P_u$ the index for which $g^p_{e,u} \in M_{k+i+1}$.

(c) The edge b^{2p}_u cannot be contained in M_{k+i+1}, since y^{2p+1}_u is already matched by $g^p_{e,u}$.

(d) In contrast to M_{k+i}, the vertex y^{2p+1}_u is not matched by b^{2p}_u in M_{k+i+1}. It must be matched by the edge $g^{p+1}_{e,u}$. We will later show that this must hold due to the symmetric difference constraint.

(e) The edge a^2_e cannot be contained in M_{k+i+1}, since x^3_e is already matched by $g^{p+1}_{e,u}$.

(f) In contrast to M_{k+i}, the vertex x^4_e is not matched by a^2_e in M_{k+i+1}. It must be matched by one of the edges $g^q_{e,v}$ for each $q \in \{1, 3, \ldots, 2k-3\} \setminus P_v$. Let $q \in \{1, 3, \ldots, 2k-3\} \setminus P_v$ the index for which $g^q_{e,v} \in M_{k+i}$.

(g) The edge b^{2q}_v cannot be contained in M_{k+i+1}, since y^{2q+1}_v is already matched by $g^q_{e,v}$.

(h) In contrast to M_{k+i}, the vertex y^{2q+1}_v is not matched by b^{2q}_v in M_{k+i+1}. It must be matched by the edge $g^{q+1}_{e,v}$. We will later show that this must hold due to the symmetric difference constraint.

(i) The edge a^3_e cannot be contained in M_{k+i+1}, since x^5_e is already matched by $g^{q+1}_{e,v}$.

(j) The vertex x^6_e is not matched by a^3_e in M_{k+i+1}. It must be matched by $h^2_{e,i}$.

Thus, $M_{k+i} \triangle_R M_{k+i+1} = \{a^1_e, a^2_e, a^3_e, d_i, b^{2p}_u, b^{2q}_v\} \cup \{h^1_{e,i}, h^2_{e,i}, g^p_{e,u}, g^{p+1}_{e,u}, g^q_{e,v}, g^{q+1}_{e,v}\}$, which means that there are $|E_C| + 1 = (i-1) + 1 = i$ edges for which Lemma 4.16

holds. Note that $|M_{k+i} \triangle_R M_{k+i+1}| = 12 = \ell'$. Hence, if $g_{e,u}^{p+1} \notin M_{k+i+1}$ (see step (d)), then other edges incident to y_u^{2p+1} and x_e^3 would have to be contained in M_{k+i+1}, which would result in $\triangle_R$ exceeding 12. The same holds for step (h). □

We are now ready to prove Lemma 4.15.

Proof of Lemma 4.15. Following Lemma 4.16, there are exactly $\binom{k}{2}$ edges $e = \{u, v\} \in E$ with

$$M_{k+i} \triangle_R M_{k+i+1} = \{a_e^1, a_e^2, a_e^3, d_i, b_u^{2p}, b_v^{2q}\} \cup \{h_{e,i}^1, h_{e,i}^2, g_{e,u}^p, g_{e,u}^{p+1}, g_{e,v}^q, g_{e,v}^{q+1}\},$$

for any $i \in [\binom{k}{2}]$ and any $p, q \in \{1, 3, ..., 2k-3\}$. Then it holds that $b_u^{2p}, b_v^{2q} \in M_{k+i}$. Furthermore, b_u^{2p}, b_v^{2q} must be contained in every matching from layer $k+1$ until layer $k+i$, because in each transition between these layers, both edges are not contained in the symmetric difference of two matchings. Consequently, it holds that $b_u^{2p}, b_v^{2q} \in M_{k+1}$. Since in layer $k+1$, the vertex gadget Y_v is a path without any outgoing edges (except for the two endpoints y_v^1 and y_v^{4k-4} of the path), it must hold that each edge b_v^{2p} is contained in M_{k+1}. Thus, M_{k+1} must also contain $f_{v,j}^1$ and $f_{v,j}^2$, for any $j \in [k]$. This finishes the proof. □

We are now ready to prove Lemma 4.13.

Proof of Lemma 4.13. Let $C \subseteq V$ be the set resulting from Construction 4. According to Lemma 4.14, there are exactly k vertices v with $f_{v,i}^1 \in M_{k+1}$ and $f_{v,i}^2 \in M_{k+1}$ for any $i \in [k]$. Considering how C is constructed, it follows that $|C| = k$. According to Lemma 4.15, there are exactly $\binom{k}{2}$ edges $e = \{u, v\} \in E$ for which

(1) $f_{u,i}^1 \in M_{k+1}$ and $f_{u,i}^2 \in M_{k+1}$ for any $i \in [k]$, and
(2) $f_{v,i}^1 \in M_{k+1}$ and $f_{v,i}^2 \in M_{k+1}$ for any $i \in [k]$.

Thus, for each of these $\binom{k}{2}$ egdes, both endpoints are contained in C, which makes C a clique. Consequently, the set C is a solution for $\mathcal{I}$. □

Finally, we prove Lemma 4.12.

Proof of Lemma 4.12. If $\mathcal{J}$ is a yes-instance, we can construct a set $C \subseteq V$ using Construction 4. Following Lemma 4.13, C is a Clique solution for instance $\mathcal{I}$. Hence, $\mathcal{I}$ is a yes-instance. □

We proved the forward and backward direction of Proposition 4.10. It remains to put everything together.

Proof of Proposition 4.10. Let (G, k) be an instance of Clique and $(\mathcal{G}, \ell)$ be the instance of Multistage Perfect Matching resulting from Construction 3. Observe that Construction 3 runs in FPT time. We know that if (G, k) is a yes-instance of Clique, then $(\mathcal{G}, \ell)$ is a yes-instance of Multistage Perfect Matching (Lemma 4.11) and vice versa (Lemma 4.12). This finishes the proof. □

Recall that Clique is W[1]-hard for parameter k. Furthermore, observe that the reduction in Proposition 4.10 is a *parameterized reduction*, since it runs in polynomial-time and the "new" parameter $\ell + \tau$ is bounded by a function depending only on k. Hence, this reduction proves W[1]-hardness for MPM parameterized by $\ell + \tau$ and thus also proves Theorem 4.9.

Chapter 5

Conclusion

We investigated the problem of finding perfect matchings in a temporal graph that are preferably similar by the metric of symmetric difference. We have shown that even for small values of the lifetime τ or the maximum symmetric difference ℓ, the problem remains NP-hard. This means that with respect to these two parameters, Multistage Perfect Matching is para-NP-hard and thus not fixed-parameter tractable unless P=NP. With a combination of ℓ and τ as a parameter, we proved W[1]-hardness and developed an XP-algorithm. This hardness proof is our most technical result and uses a sophisticated parameterized reduction. Furthermore, if ℓ and τ are also combined with the treewidth of the underlying graph, we proved fixed-parameter tractability by providing an MSO formula that represents MPM. We also proved fixed-parameter tractability for the sum of the size of consecutive edge set intersections. This means that the problem is computationally simple if each two consecutive edge sets in the temporal graph are very dissimilar, meaning that their intersection is very small. Lastly, we showed that MPM is in FPT if the parameter is the number of vertices n. Thus, even for many timesteps τ, if the temporal graph has very few vertices, then MPM is still easy to solve.

Future Work & Open Questions. Fruitful starting points for future investigations of Multistage Perfect Matching are the FPT algorithm for parameter n and the XP algorithm for parameter $\ell + \tau$. It is conceivable that the FPT running time for parameter n can be improved significantly, either to $c^{O(n)}$ for some constant c or by finding a similar algorithm for a parameter that is upper-bounded by n, such as the size of the smallest vertex cover of the underlying graph. It is also open whether the running time of the XP algorithm can be lowered to either $f(\ell, \tau) \cdot n^{\ell}$ or $f(\ell, \tau) \cdot n^{\tau}$, that is, a running time such that the degree of the polynomial only depends on one of ℓ and τ. Similarly, it might be possible to find an MSO formula depending either only on ℓ or only on τ, resulting in fixed-parameter tractability for parameter $\ell + \mathrm{tw}(G_{\cup})$ or $\tau + \mathrm{tw}(G_{\cup})$. Furthermore, Multistage Perfect Matching can be analyzed for certain constraints of the temporal graph, e.g. that each stage only consists of paths or cycles. Since, with MPM, we are only dealing with perfect matchings, it is also interesting to look at less restrictive but similar problems, such as Multistage Maximum Matching (MMM). Some of our results, such as the para-NP-hardness for ℓ and the W[1]-hardness for $\ell + \tau$, are directly applicable to MMM, but it is open for which results the two problems differ.

References

[ALS91] Stefan Arnborg, Jens Lagergren, and Detlef Seese. "Easy problems for tree-decomposable graphs". In: *Journal of Algorithms* 12.2 (1991), pp. 308–340. DOI: 10.1016/0196-6774(91)90006-K (cit. on p. 20).

[Bam+18] Evripidis Bampis, Bruno Escoffier, Michael Lampis, and Vangelis Th. Paschos. "Multistage Matchings". In: *Proc. of 16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*. Vol. 101. 2018, 7:1–7:13. DOI: 10.4230/LIPICS.SWAT.2018.7 (cit. on p. 10).

[BB72] Umberto Bertelè and Francesco Brioschi. *Nonserial dynamic programming*. Vol. v. 91. Mathematics in science and engineering. New York: Academic Press, 1972 (cit. on p. 20).

[Cas+12] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. "Time-Varying Graphs and Dynamic Networks". In: *International Journal of Parallel, Emergent and Distributed Systems* 27.5 (2012), pp. 387–408. DOI: 10.1080/17445760.2012.668546 (cit. on p. 14).

[CE12] B. Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Vol. 138. Encyclopedia of Mathematics and its Applications. Cambridge: Cambridge University Press, 2012. DOI: 10.1017/CBO9780511977619 (cit. on p. 20).

[CTW20] Markus Chimani, Niklas Troost, and Tilo Wiedera. *Approximating Multistage Matching Problems*. 2020. arXiv: abs/2002.06887 (cit. on pp. 10, 19, 29).

[Cyg+15] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer-Verlag, 2015. DOI: 10.1007/978-3-319-21275-3 (cit. on p. 15).

[DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. London: Springer, 2013. DOI: 10.1007/978-1-4471-5559-1 (cit. on pp. 15, 36).

[Edm65] Jack Edmonds. "Paths, Trees, and Flowers". In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467. DOI: 10.4153/CJM-1965-045-4 (cit. on p. 15).

[FG06] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. Berlin: Springer, 2006. DOI: 10.1007/3-540-29953-X (cit. on p. 15).

[Flu+19] Till Fluschnik, Rolf Niedermeier, Valentin Rohm, and Philipp Zschoche. "Multistage Vertex Cover". In: *14th International Symposium on Parameterized and Exact Computation (IPEC)*. Vol. 148. 2019, 14:1–14:14. DOI: 10.4230/LIPIcs.IPEC.2019.14 (cit. on pp. 10, 23).

[FMS13] Paola Flocchini, Bernard Mans, and Nicola Santoro. "On the exploration of time-varying networks". In: *Theoretical Computer Science* 469 (2013), pp. 53–68. DOI: 10.1016/j.tcs.2012.10.029 (cit. on p. 14).

[Fom+20] Fedor V. Fomin, Petr A. Golovach, Lars Jaffke, Geevarghese Philip, and Danil Sagunov. *Diverse Pairs of Matchings*. 2020. arXiv: 2009.04567 (cit. on p. 10).

[Kos09] Vassilis Kostakos. "Temporal graphs". In: *Physica A: Statistical Mechanics and its Applications* 388.6 (2009), pp. 1007–1023. DOI: 10.1016/j.physa.2008.11.021 (cit. on p. 14).

[Kő16] Dénes Kőnig. "Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre". In: *Mathematische Annalen* 77 (4) (1916), pp. 453–465. DOI: 10.1007/BF01456961 (cit. on p. 9).

[Mer+20] George B. Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. "Computing Maximum Matchings in Temporal Graphs". In: *37th International Symposium on Theoretical Aspects of Computer Science (STACS)*. Vol. 154. 2020, 27:1–27:14. DOI: 10.4230/LIPIcs.STACS.2020.27 (cit. on p. 10).

[MG20] Subhrangsu Mandal and Arobinda Gupta. "0-1 Timed Matching in Bipartite Temporal Graphs". In: *Algorithms and Discrete Applied Mathematics*. Vol. 12016. Springer International Publishing, 2020, pp. 331–346. DOI: 10.1007/978-3-030-39219-2_27 (cit. on p. 10).

[Mic15] Othon Michail. "An Introduction to Temporal Graphs: An Algorithmic Perspective". In: *Algorithms, Probability, Networks, and Games. Lecture Notes in Computer Science*. Vol. 9295. 2015, pp. 308–343. DOI: 10.1007/978-3-319-24024-4_18 (cit. on p. 14).

[MV80] Silvio Micali and Vijay V. Vazirani. "An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs". In: *21st Annual Symposium on Foundations of Computer Science*. IEEE, 1980, pp. 17–27. DOI: 10.1109/SFCS.1980.12 (cit. on p. 15).

[Nie06] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. DOI: 10.1093/acprof:oso/9780198566076.001.0001 (cit. on p. 15).

[Zsc+20] Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. "The Complexity of Finding Small Separators in Temporal Graphs". In: *Journal of Computer and System Sciences* 107 (2020), pp. 72–79. DOI: 10.1016/j.jcss.2019.07.006 (cit. on pp. 14, 21).