Eötvös Loránd University
Faculty of Science

Márk Hunor Juhász
MSc in Applied Mathematics

Optimization problems in temporal graphs

Project Work II.

Supervisor: Péter Madarasi

ELTE
EÖTVÖS LORÁND
UNIVERSITY

Budapest, 2026

1 Preliminaries and definitions

Computing a maximum matching in an undirected (static) graph is one of the most fundamental graph-algorithmic problems. It has many different variations and modifications, for example, it can also be extended to temporal graphs, whose topology is subject to discrete changes over time. In these graphs, edges can disappear and then reappear, allowing many real-world optimization problems to be modeled as a maximum matching problem in temporal graphs. However, even the maximum matching problem itself can be defined in various ways within these structures, which are in the focus of our work. We studied two temporal matching models: Δ- and γ-matching, and a closely related static model: the d-matching. In a Δ-matching [1], instead of selecting the edges of the underlying graph, we select time edges. Two selected time edges may share a vertex only if their time ticks differ by at least Δ. In a γ-matching [2], we select a block of γ consecutive appearances of the same underlying edge. Two such blocks are compatible if the corresponding underlying edges are disjoint, or if their time intervals are disjoint. In a d-matching [3, 4], we are given a bipartite graph, with an ordering on one of the vertex sets. We select the edges of the graph in such a way that two selected edges can only intersect in the unordered vertex set, and only if their other endpoints are at least d positions apart in the ordering.

Now, let us define the previous models precisely.

Definition 1.1. *A **temporal graph** $\mathcal{G} = (G, \lambda)$ consists of a finite static graph $G = (V, E)$ and a **time-labeling function** λ that assigns a non-empty finite set $\lambda(e) \subseteq \mathbb{Z}_{>0}$ to each edge $e \in E$. An edge $e \in E$ is active at time $\tau \in \mathbb{Z}_{>0}$ if $\tau \in \lambda(e)$. If e is active at time τ, then the pair (e, τ) is called a **time edge**, and we say that e **appears** at time τ. If $E \neq \emptyset$, the **lifetime** $\mathcal{T}(\mathcal{G}) < \infty$ is the largest time-label assigned to an edge, that is, $\mathcal{T}(\mathcal{G}) = \max\{\tau \in \lambda(e) : e \in E\}$. If $E = \emptyset$, we set $\mathcal{T}(\mathcal{G}) = 0$.*

Definition 1.2. *Two time edges (e, τ) and (e', τ') are **Δ-independent** if the edges e, e' do not share an endpoint or their time ticks differ by at least Δ, that is, $|\tau - \tau'| \geq \Delta$. A **$\Delta$-matching** of a temporal graph $\mathcal{G}$ is a set of pairwise Δ-independent time edges of $\mathcal{G}$.*

Remark 1.1. *Equivalently, a set M of time edges is a Δ-matching if, in every interval of Δ consecutive ticks, the time edges of M appearing in that interval form a matching in the underlying graph.*

Problem 1.1. *In the **maximum Δ-matching problem**, the input is a temporal graph $\mathcal{G}$ and a positive integer Δ. The goal is to find a Δ-matching in $\mathcal{G}$ with the largest possible cardinality.*

Definition 1.3. *Let $\mathcal{G} = (G, \lambda)$ be a temporal graph with $G = (V, E)$. For an edge $e \in E$ and a time tick $\tau \in \mathbb{Z}_{>0}$ such that $[\tau, \tau + \gamma - 1] \subseteq \lambda(e)$, define $(e, \tau)_\gamma = \{(e, \tau') : \tau' \in$*

$[\tau, \tau+\gamma-1]\}$. *We call* $(e,\tau)_\gamma$ *the* γ***-edge of*** e ***starting at time*** τ. *Two* γ*-edges* $(e,\tau)_\gamma$ *and* $(e',\tau')_\gamma$ *are* γ***-independent*** *if the edges* e, e' *do not share an endpoint or the intervals* $[\tau, \tau+\gamma-1]$ *and* $[\tau', \tau'+\gamma-1]$ *are disjoint. A* γ***-matching*** *of* $\mathcal{G}$ *is a set of pairwise* γ*-independent* γ*-edges.*

Problem 1.2. *In the* ***maximum*** $\boldsymbol{\gamma}$***-matching problem****, the input is a temporal graph* $\mathcal{G}$ *and a positive integer* γ*. The goal is to find a* γ*-matching in* $\mathcal{G}$ *with the largest possible cardinality.*

Problem 1.3. *In the* ***maximum d-matching problem****, the input is a bipartite graph* $G=(S,T,E)$ *with* $S=\{s_1,\dots,s_n\}$ *and a positive integer* d*. The goal is to find a subset* $M \subseteq E$ *of maximum cardinality such that every vertex in* S *is incident with at most one edge of* M*, and whenever* $s_it, s_jt \in M$ *for some* $t \in T$ *with* $i \neq j$*, we have* $|i-j| \geq d$*. A set of edges satisfying these two conditions is called a* ***d-matching****.*

Remark 1.2. *In the bipartite graph* $G=(S,T,E)$*, a set* $M \subseteq E$ *is a* d*-matching if and only if, after restricting* M *to any interval of at most* d *consecutive vertices of* S*, the remaining edges form a matching. Here, restricting* M *to a set* $X \subseteq S$ *means keeping the edges of* M *incident with vertices of* X*.*

In my BSc thesis, we focused on the previously defined temporal matching problems, and we also explored how the seemingly unrelated d-matching problem connects to temporal graphs and to the two problems. We also analyzed the maximum Δ-matching problem in the case where the underlying static graph of the temporal graph is a tree and proved three results, which are as follows. We proved that this problem becomes NP-hard as soon as every edge can appear at most twice. We also proved that it can be solved in polynomial time if every edge appears at most once. Moreover, we gave an Efficient Polynomial-Time Approximation Scheme for the problem, provided that Δ is constant.

In the previous semester, we continued examining the maximum Δ-matching problem on trees. Last year, we already established that if every edge appears at most once, then the problem can be solved in polynomial time. This raised the question of under what other conditions the problem is solvable. We managed to find several answers to this question; however, there was a fatal oversight in our theorem statement. Namely, the running time of our algorithms was $O(\text{poly}(\mathcal{T}))$, whereas a truly polynomial running time would be $O(\text{poly}(\log \mathcal{T}))$.

2 New results on solvability

The problem was the following. We treated the set of time edges in the input as if it were of size $O(|V|\mathcal{T})$, whereas the real size is $\sum_e |\lambda(e)|$. In this semester, we fixed this oversight, while keeping the main ideas of these proofs untouched. However, our approximation scheme is no longer an EPTAS, but a PTAS, which still proves that the problem is not APX-hard, unless P=NP. Thanks to the relationship between the Δ-matching and γ-matching, which was explored in [1], these results also apply to the maximum γ-matching problem. The results are as follows.

Theorem 2.1. *Consider the maximum Δ-matching problem on a temporal graph $\mathcal{G} = (G, \lambda)$ whose underlying graph $G = (V, E)$ is a tree, and let $K \in \mathbb{Z}_{>0}$ be given. For every vertex $v \in V$, let $A_v = \bigcup_{e \ni v} \lambda(e)$ be the set of time ticks at which at least one edge incident to v is active. Let $B = \max_{v \in V} |A_v|$. Suppose that there exists an optimal solution M^* such that, for every vertex $v \in V$, the time edges of M^* incident to v use at most K time ticks. Then an optimal solution can be found in $O\big((2eB)^K K^2 n\big)$ time, where $n = |V|$ and e denotes Euler's number.*

Corollary 2.1. *Consider the maximum γ-matching problem, where the underlying graph $G = (V, E)$ of the input temporal graph $\mathcal{G} = (G, \lambda)$ is a tree, and let $K \in \mathbb{Z}_{>0}$ be given. Let $B_\gamma = \max_{v \in V} |\{(e, \tau)_\gamma : e \in E,\ e \ni v,\ [\tau, \tau + \gamma - 1] \subseteq \lambda(e)\}|$. If there exists an optimal solution M^* for which the sets $M^*_v = \{(e, \tau)_\gamma \in M^* : e \text{ is incident to } v\}$ have size at most K for every $v \in V$, then an optimal solution can be found in $O\big((2eB_\gamma)^K K^2 n\big)$ time, where $n = |V|$ and e denotes Euler's number.*

Theorem 2.2. *For every fixed constant c, the maximum Δ-matching problem can be solved in polynomial time on instances whose underlying graph $G = (V, E)$ is a tree and which satisfy $\sum_{u \in N_G(v)} |\lambda(uv)| \leq c \log L$ for every $v \in V$, where $L = \sum_{e \in E} |\lambda(e)|$ and $N_G(v)$ denotes the set of neighbors of v in G.*

Corollary 2.2. *For every fixed constant c, the maximum γ-matching problem can be solved in polynomial time on instances whose underlying graph $G = (V, E)$ is a tree and which satisfy $|\{(e, \tau)_\gamma : e \in E,\ e \ni v,\ [\tau, \tau + \gamma - 1] \subseteq \lambda(e)\}| \leq c \log L_\gamma$ for every $v \in V$, where L_γ denotes the total number of γ-edges in the instance.*

Using the old version of Theorem 2.1, we managed to generalize the EPTAS given in my BSc thesis. Let us note that here—unlike in the theorem stated in the thesis—we no longer assume that Δ is constant. However, as we stated earlier, the running time of our algorithm was not an EPTAS. We also fixed this error, and proved the following theorem.

Theorem 2.3. *Let $\mathcal{G} = (G, \lambda)$ be a temporal graph whose underlying graph $G = (V, E)$ is a tree, and let $n = |V|$. Write $L = \sum_{e \in E} |\lambda(e)|$ and $B = \max_{v \in V} |A_v|$, where $A_v = \bigcup_{e \ni v} \lambda(e)$. For every $0 < \varepsilon < 1$, the maximum Δ-matching problem on $\mathcal{G}$ admits a $(1-\varepsilon)$-approximation algorithm with running time $O\big(L^2 (2eB)^{\lceil 1/\varepsilon \rceil} \lceil 1/\varepsilon \rceil^2 n\big)$. In particular, since $B \leq L$, the running time is $L^{O(1/\varepsilon)}\,\mathrm{poly}(n)$, and the problem admits a PTAS.*

Since the maximum γ- and the maximum d-matching problems can be reduced to the maximum Δ-matching problem, and this reduction is approximation-preserving, this result also applies to the other two problems. However, this is only interesting if those problems are also NP-hard on trees. There had been no results on this before, but during the previous semester, we proved that the maximum γ-matching problem is NP-hard on trees. In order to do this, we proved the following lemma.

Lemma 2.1. *Consider the maximum Δ-matching problem, where the underlying graph is $G = (V, E)$, the input temporal graph is $\mathcal{G} = (G, \lambda)$, and for every edge $e \in E$, $|x - y| > \Delta$ holds for every distinct $x, y \in \lambda(e)$. In this case, the maximum Δ-matching problem can be reduced to the maximum γ-matching problem.*

Concerning the relationship between the problems, we proved last year that the maximum d-matching problem can be reduced to the maximum Δ-matching problem. However, this semester, we observed something even stronger:

Lemma 2.2. *The maximum d-matching problem is a special case of the maximum Δ-matching problem, in which every edge appears exactly once.*

With this observation, we also proved that the maximum d-matching problem is polynomial-time solvable on trees, therefore the PTAS result is only interesting in the case of the maximum γ-matching problem.

3 New results on graphs with constant treewidth

This part of our research will not be included in our manuscript, since it is only loosely related to the topic; however, the following result was also obtained during this semester. After solving the maximum Δ-matching problem on trees, we wanted to continue the research on graphs with constant treewidth, as many results obtained for trees (especially those derived using dynamic programming) also hold for such graphs. In the literature, there are many results on the solvability of those problems, where we want to optimize something on the vertices (for example vertex cover or coloring). However, if we want to optimize something on the edges (for example edge-coloring or Δ-matching), the situation is a lot harder. The most common solution here is to consider the line graph of the original graph, and therefore obtain a vertex-optimization problem. However, the line graph is not always of constant treewidth. Unfortunately, we did not manage to find another solution, where we do not take the line graph of the original graph. The theorem we proved is as follows.

Theorem 3.1. *Consider the maximum Δ-matching problem, where the underlying graph $G = (V, E)$ of the input temporal graph $\mathcal{G} = (G, \lambda)$ is of constant treewidth, that is, $\mathrm{tw}(G) \leq K$ for a given constant K. If $\mathcal{T} \leq K$ and $\delta_{\max}(G) \leq K$ also holds, where $\mathcal{T}$ is the lifetime of $\mathcal{G}$ and $\delta_{\max}(G)$ is the maximum degree in G, then the problem can be solved in polynomial time.*

In the theorem, the $\delta_{\max} \leq K$ condition guarantees the constant treewidth of the line graph. However, we can change the $\mathcal{T} \leq K$ condition to some other condition (which is similar to the ones presented in the previous section), and the problem remains solvable. In this result, we only use that many vertex-optimization problems can be solved easily on graphs with constant treewidth. However, we were unable to find a better solution than the one above for how the current edge-optimization problem could be solved, so we did not continue the research of these graphs.

4 Summary

In summary, during the semester we fixed all the oversights in our previous theorems. Using these new results, we were able to strengthen the theorem presented in my thesis, and we now have a complete picture of the computational complexity of the maximum Δ-matching problem on trees. Our refined results also apply to the maximum γ-matching problem and the d-matching problem. We continued our research by examining the case where the underlying graph of the input temporal graph is of constant treewidth. We found that most of our results concerning solvability also apply to these graphs, however, the strict conditions in the theorems make these results less interesting. Besides examining the constant treewidth case, we started writing our preprint. Now, the preprint is almost done, and we will upload it in a couple of weeks. During the semester, we also submitted an abstract to ASIACOMB 2026, and it was accepted. We will present our results in Korea, in a 20-minute talk this summer.

References

[1] George B. Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. "Computing maximum matchings in temporal graphs". In: *Journal of Computer and System Sciences* 137 (2023), pp. 1–19.

[2] Julien Baste, Binh-Minh Bui-Xuan, and Antoine Roux. "Temporal matching". In: *Theoretical Computer Science* 806 (2020), pp. 184–196.

[3] Péter Madarasi. "Matchings under distance constraints I." In: *Annals of Operations Research* 305 (2021), pp. 137–161.

[4] Péter Madarasi. "Matchings under distance constraints II." In: *Annals of Operations Research* 332 (2024), pp. 303–327.