On Graph Problems Stemming from Temporal Graphs

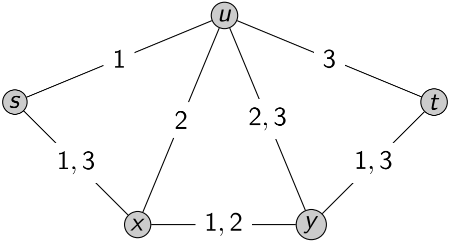
Ana Silva

ParGO (www.pargo.ufc.br), Universidade Federal do Ceará - Brazil.

ICALP - Algorithmic Aspects of Temporal Graphs
July 7 2025.

Basic definitions

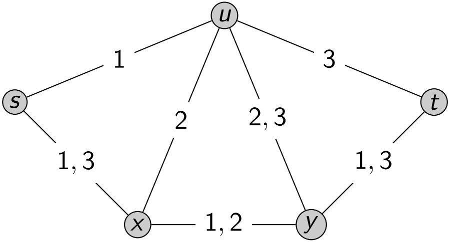
The notation we will be adopting



Definitions: Temporal Graph

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;

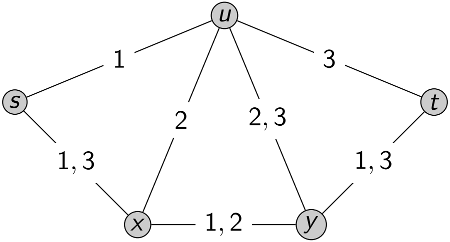
The notation we will be adopting



Definitions: Snapshot

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- The graph $G_i = (V(G), \{e \mid i \in \lambda(e)\})$ is called a *snapshot*. Sometimes the temporal graph will be described by its *sequence of snapshots*.;
- The value $\max_{e \in E(G)} \lambda(e)$ is called the *lifetime* and denoted by τ.

The notation we will be adopting

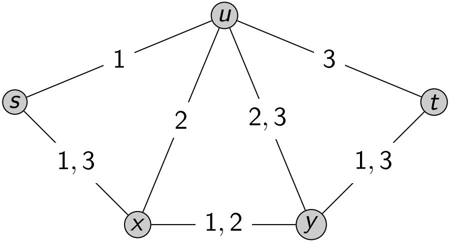


$\tau = 3$

Definitions: Lifetime

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- The graph $G_i = (V(G), \{e \mid i \in \lambda(e)\}$ is called a *snapshot*. Sometimes the temporal graph will be described by its *sequence of snapshots*.;
- The value $\max_{e \in E(G)} \lambda(e)$ is called the *lifetime* and denoted by τ.

The notation we will be adopting

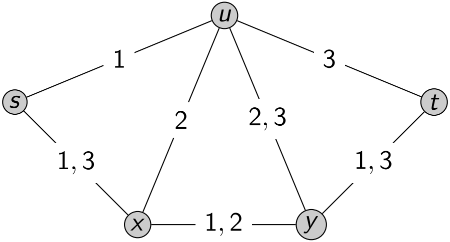


$\tau = 3$

Definitions: Temporal vertex

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- A *temporal vertex* is a pair (u, t) where $u \in V(G)$ and $t \in [\tau]$;

The notation we will be adopting

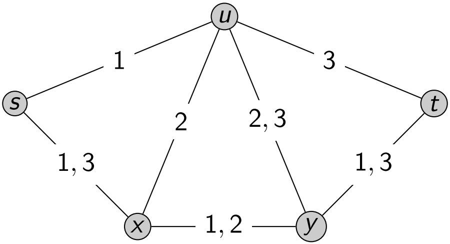


$\tau = 3$

Definitions: Temporal edge

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- A *temporal vertex* is a pair (u, t) where $u \in V(G)$ and $t \in [\tau]$;
- A *temporal edge* is a pair (e, t) where $e \in E(G)$ and $t \in \lambda(e)$;

The notation we will be adopting



$\tau = 3$

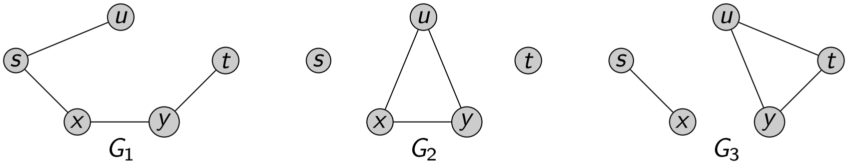
Definitions:

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- A *temporal vertex* is a pair (u, t) where $u \in V(G)$ and $t \in [\tau]$;
- A *temporal edge* is a pair (e, t) where $e \in E(G)$ and $t \in \lambda(e)$;
- A temporal graph (G, λ) with lifetime τ is called *dynamic-based* if $\lambda(e) = [\tau]$, $\forall e \in E(G)$.

Temporal coloring

Coloring temporal graphs, Marino and Silva. J. Comp. System and Sciences 123 (2022) 171–185.

Temporal Coloring

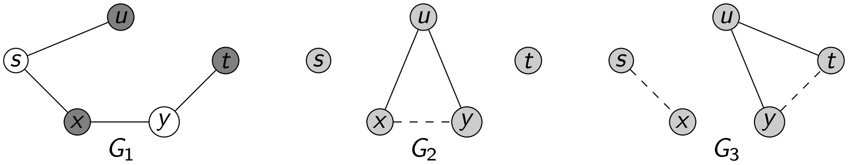


We want to minimize the number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.

Intruced in *Sliding window temporal graph coloring*, Mertzios, Molter, Zamaraev.

JCSS 120 (2021) 97–115.

Temporal Coloring

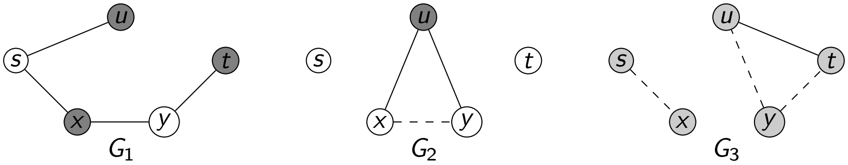


We want to minimize the number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.

Intruced in *Sliding window temporal graph coloring*, Mertzios, Molter, Zamaraev.

JCSS 120 (2021) 97–115.

Temporal Coloring

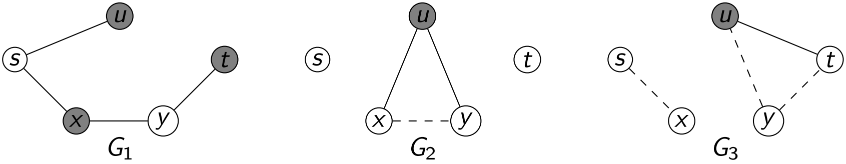


We want to minimize the number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.

Intruced in *Sliding window temporal graph coloring*, Mertzios, Molter, Zamaraev.

JCSS 120 (2021) 97–115.

Temporal Coloring

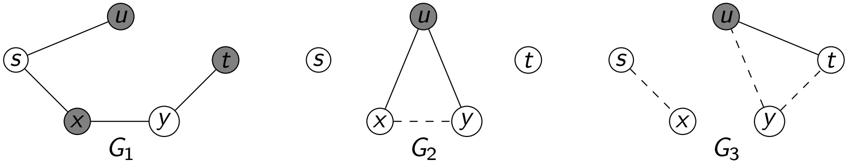


We want to minimize the number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.

Intruced in ***Sliding window temporal graph coloring***, **Mertzios, Molter, Zamaraev.**

JCSS 120 (2021) 97–115.

Temporal Coloring



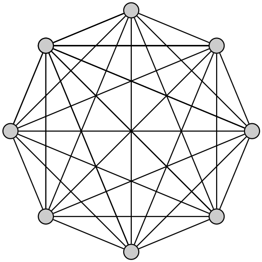
We want to minimize the number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.

Let us call it the temporal chromatic number and denote it by $t\chi(G, \lambda)$.

Intruced in ***Sliding window temporal graph coloring***, **Mertzios, Molter, Zamaraev.**

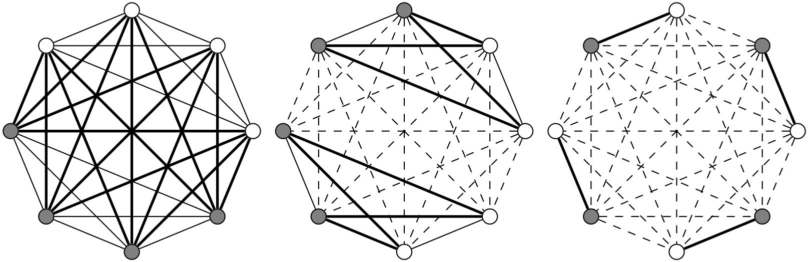
JCSS 120 (2021) 97–115.

k-partite cover of static graphs



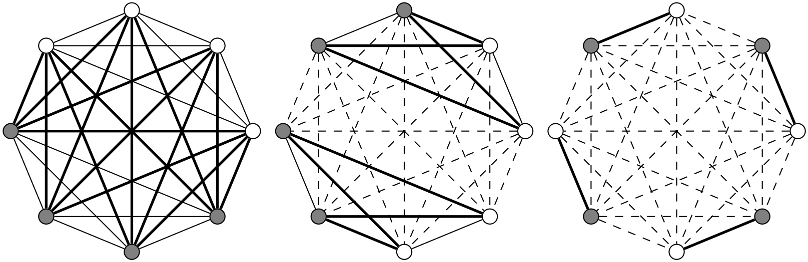
- A graph is k-partite if it can be properly colored with at most k colors.
- A k-partite cover of G is a collection of subgraphs $H_1, \ldots, H_t$ such that each H_i is k-partite and every $e \in E(G)$ belongs to at least one H_i.
- The minimum number of subgraphs t is the k-partite number of G and denoted by $\beta_k(G)$.

k-partite cover of static graphs



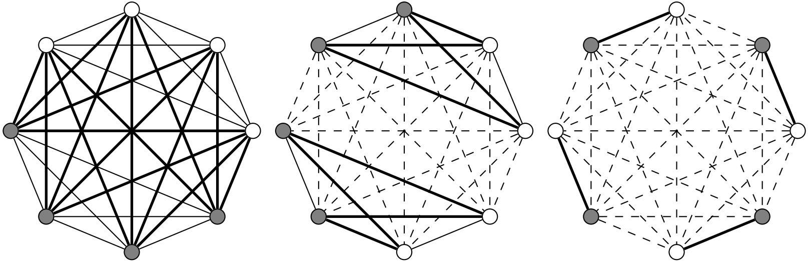
- A graph is k-partite if it can be properly colored with at most k colors.
- A k-partite cover of G is a collection of subgraphs $H_1, \ldots, H_t$ such that each H_i is k-partite and every $e \in E(G)$ belongs to at least one H_i.
- The minimum number of subgraphs t is the k-partite number of G and denoted by $\beta_k(G)$.

k-partite cover of static graphs



- A graph is k-partite if it can be properly colored with at most k colors.
- A k-partite cover of G is a collection of subgraphs $H_1, \ldots, H_t$ such that each H_i is k-partite and every $e \in E(G)$ belongs to at least one H_i.
- The minimum number of subgraphs t is the k-partite number of G and denoted by $\beta_k(G)$.

k-partite number



- The minimum number of subgraphs t is the k-partite number of G and denoted by $\beta_k(G)$.

Theorem (Khan, Alaimia, and Mahmood, 2013)

$\beta_k(G) = \lceil \log_k \chi(G) \rceil$, *where* $\chi(G)$ *denotes the chromatic number of* G *(minimum number of colors needed to properly color* G*).*

k-partite number and temporal coloring

Recall:

- temporal chromatic number of (G, λ) ($t\chi(G, \lambda)$) is the minimum number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.
- k-partite number of G ($\beta_k(G)$): minimum number of subgraphs t such that the edges of G can be covered by t subgraphs, each of which is k-partite (can be properly colored with k colors).
- $\beta_k(G) = \lceil \log_k \chi(G) \rceil$ (Khan, Alaimia, and Mahmood).

k-partite number and temporal coloring

Recall:

- temporal chromatic number of (G, λ) ($t\chi(G, \lambda)$) is the minimum number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.
- k-partite number of G ($\beta_k(G)$): minimum number of subgraphs t such that the edges of G can be covered by t subgraphs, each of which is k-partite (can be properly colored with k colors).
- $\beta_k(G) = \lceil \log_k \chi(G) \rceil$ (Khan, Alaimia, and Mahmood).

k-partite number and temporal coloring

Recall:

- temporal chromatic number of (G, λ) ($t\chi(G, \lambda)$) is the minimum number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.
- k-partite number of G ($\beta_k(G)$): minimum number of subgraphs t such that the edges of G can be covered by t subgraphs, each of which is k-partite (can be properly colored with k colors).
- $\beta_k(G) = \lceil \log_k \chi(G) \rceil$ (Khan, Alaimia, and Mahmood).

k-partite number and temporal coloring

Recall:

- temporal chromatic number of (G, λ) ($t\chi(G, \lambda)$) is the minimum number of colors needed to color the temporal vertices in a way that every edge is properly colored at least once.
- k-partite number of G ($\beta_k(G)$): minimum number of subgraphs t such that the edges of G can be covered by t subgraphs, each of which is k-partite (can be properly colored with k colors).
- $\beta_k(G) = \lceil \log_k \chi(G) \rceil$ (Khan, Alaimia, and Mahmood).

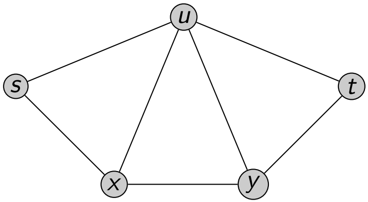
Corollary

Let G be a graph and $t = \beta_k(G)$. If (G, λ) is dynamic-based with lifetime t, then $t\chi(G, \lambda) \leq k$.
In particular, as $\chi(G) \leq n$, if (G, λ) is dynamic-based with lifetime τ, then $t\chi(G, \lambda) \leq \sqrt[\tau]{n}$.

Temporal Eulerian Walks

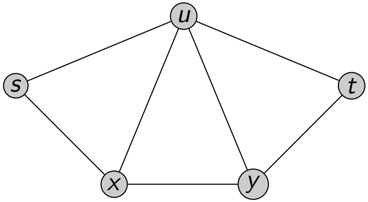
Eulerian walks in temporal graphs, Marino and Silva. Algorithmica 85 (2023) 805–830.

Eulerian Trails on Static Graphs



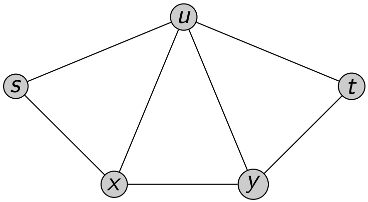
A trail is a sequence of edges that does not repeat edges. If it is a circuit if starts and finishes on the same vertex. It is Eulerian if it visits every edge of the graph.

Eulerian Trails on Static Graphs



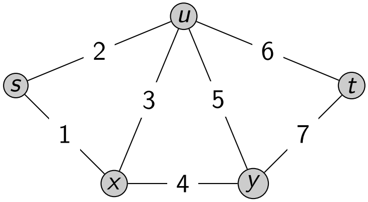
A trail is a sequence of edges that does not repeat edges. If it is a circuit if starts and finishes on the same vertex. It is Eulerian if it visits every edge of the graph.

Eulerian Trails on Static Graphs



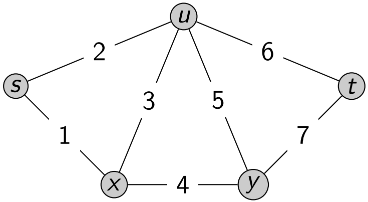
A trail is a sequence of edges that does not repeat edges. If it is a circuit if starts and finishes on the same vertex. It is Eulerian if it visits every edge of the graph.

Eulerian Trails on Static Graphs



A trail is a sequence of edges that does not repeat edges. If it is a circuit if starts and finishes on the same vertex. It is Eulerian if it visits every edge of the graph.

Eulerian Trails on Static Graphs

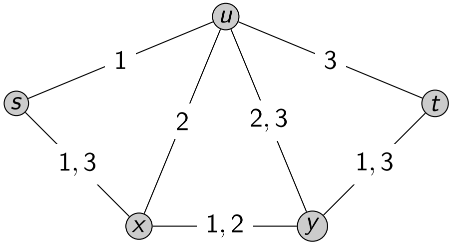


A trail is a sequence of edges that does not repeat edges. If it is a circuit if starts and finishes on the same vertex. It is Eulerian if it visits every edge of the graph.

Theorem (Euler, 1736)

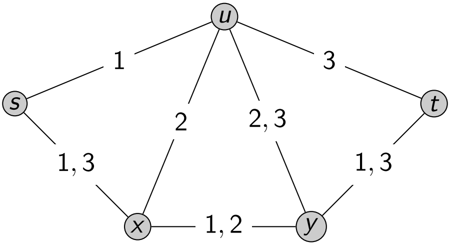
G has an Eulerian trail (circuit) iff it has exactly two (zero) odd-degree vertices.

Eulerian Local Trails on Temporal Graphs



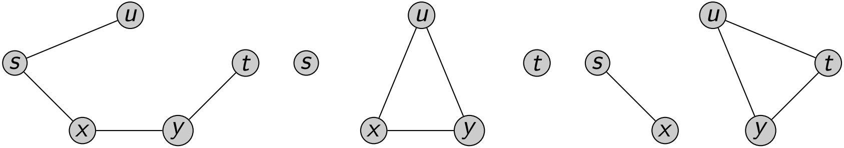
A temporal local trail is a temporal walk that forms a trail in each snapshot.

Eulerian Local Trails on Temporal Graphs



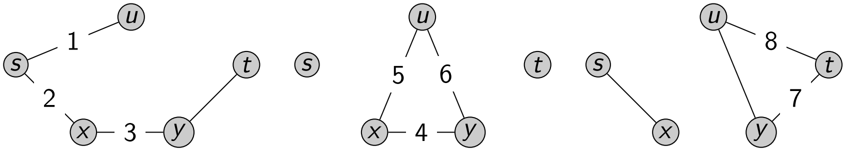
A temporal local trail is a temporal walk that forms a trail in each snapshot. It is Eulerian if each edge is visited at least once.

Eulerian Local Trails on Temporal Graphs



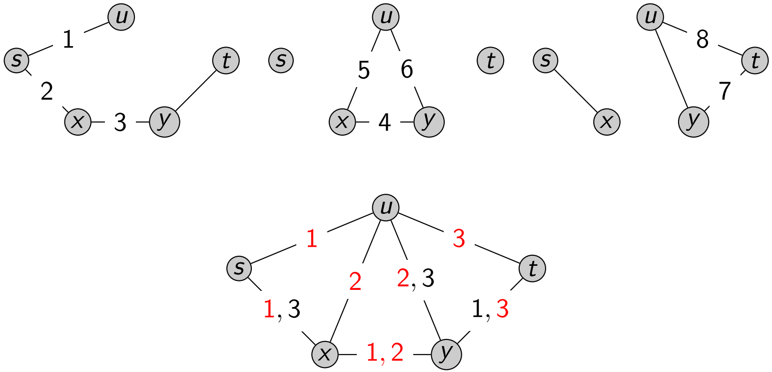
A temporal local trail is a temporal walk that forms a trail in each snapshot. It is Eulerian if each edge is visited at least once.

Eulerian Local Trails on Temporal Graphs



A temporal local trail is a temporal walk that forms a trail in each snapshot. It is Eulerian if each edge is visited at least once.

Eulerian Local Trails on Temporal Graphs



A temporal local trail is a temporal walk that forms a trail in each snapshot. It is Eulerian if each edge is visited at least once.

Eulerian Local Trails and Static Graphs

Theorem (Marino and **Silva**)

Let $\mathcal{G} = (G, \lambda)$ be a dynamic-based graph with lifetime 2 s.t. G has degree at most 4. Then, deciding whether $\mathcal{G}$ has an Eulerian local trail is NP-complete.

Corollary

Let G be a graph of degree at most 4. Deciding whether the edges of G can be covered with two trails is NP-complete.

Eulerian Local Trails and Static Graphs

Theorem (Marino and **Silva**)

Let $\mathcal{G} = (G, \lambda)$ be a dynamic-based graph with lifetime 2 s.t. G has degree at most 4. Then, deciding whether $\mathcal{G}$ has an Eulerian local trail is NP-complete.

Corollary

Let G be a graph of degree at most 4. Deciding whether the edges of G can be covered with two trails is NP-complete.

Final remarks

Temporal graphs seem to combine Graph Theory and Ordering problems in an interesting way. So I guess it is worth looking for collaborations within these three fields.