Temporal Vertex Cover with a Sliding Time Window

George B. Mertzios
Durham University

These results have been presented in **ICALP 2018**

Joint work with:
Eleni C. Akrida, Durham University,
Paul G. Spirakis, University of Liverpool & University of Patras,
Viktor Zamaraev, University of Liverpool

Research Seminar
Computer Engineering and Informatics Department (CEID)
Univeristy of Patras, Greece
December 2019

Static and Temporal Graphs

Modern networks are highly dynamic:

- Social networks: friendships are added/removed, individuals leave, new ones enter
- Transportation networks: transportation units change with time their position in the network
- Physical systems: e.g. systems of interacting particles

The common characteristic in all these applications:

- the graph topology is subject to discrete changes over time

$\Rightarrow$ the notion of vertex adjacency must be appropriately re-defined (by introducing the time dimension in the graph definition)

Various graph concepts (e.g. reachability, connectivity):

- crucially depend on the exact temporal ordering of the edges

Temporal graphs

Formally:

Definition (Temporal Graph)

A temporal graph is a pair (G, λ) where:

- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.

Temporal graphs

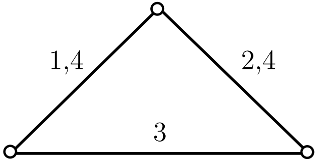
Formally:

Definition (Temporal Graph)

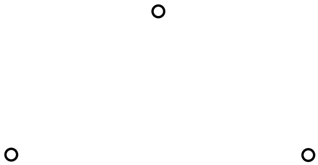
A temporal graph is a pair (G, λ) where:

- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.

temporal graph:



temporal instances:



Temporal graphs

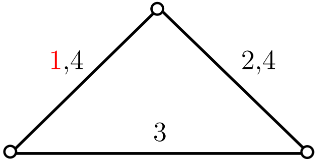
Formally:

Definition (Temporal Graph)

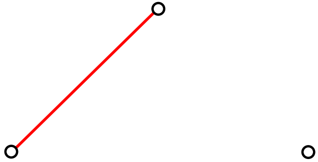
A temporal graph is a pair (G, λ) where:

- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.

temporal graph:



temporal instances:



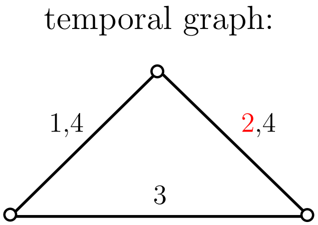
Temporal graphs

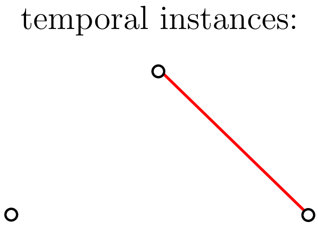
Formally:

Definition (Temporal Graph)

A temporal graph is a pair (G, λ) where:

- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.





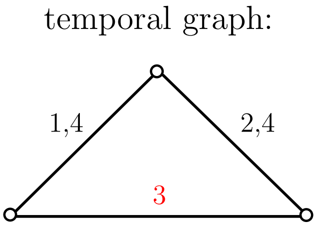
Temporal graphs

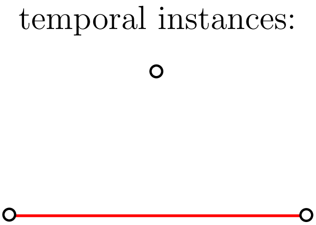
Formally:

Definition (Temporal Graph)

A temporal graph is a pair (G, λ) where:

- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.





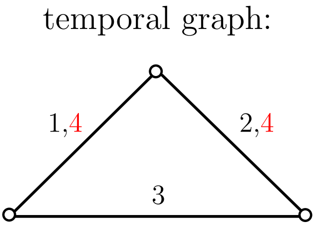
Temporal graphs

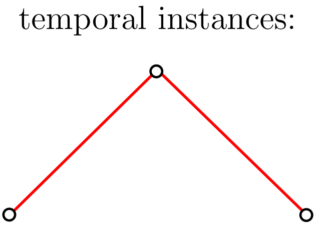
Formally:

Definition (Temporal Graph)

A temporal graph is a pair (G, λ) where:

- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.





Temporal graphs

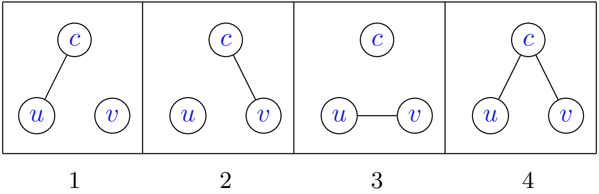
Formally:

Definition (Temporal Graph)

A temporal graph is a pair (G, λ) where:

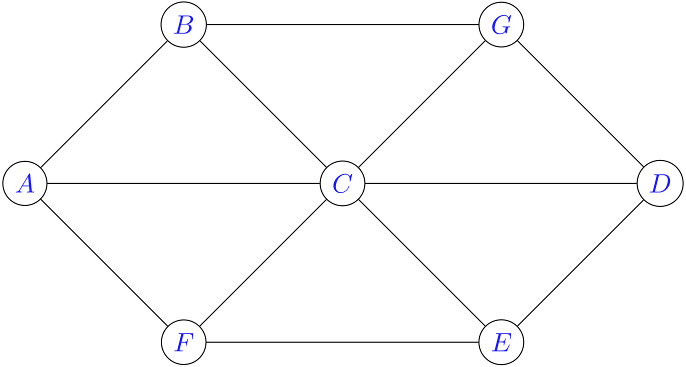
- $G = (V, E)$ is an underlying (di)graph and
- $\lambda : E \to 2^{\mathbb{N}}$ is a discrete time-labeling function.

Alternatively, we can view it as a sequence of static graphs, the snapshots:

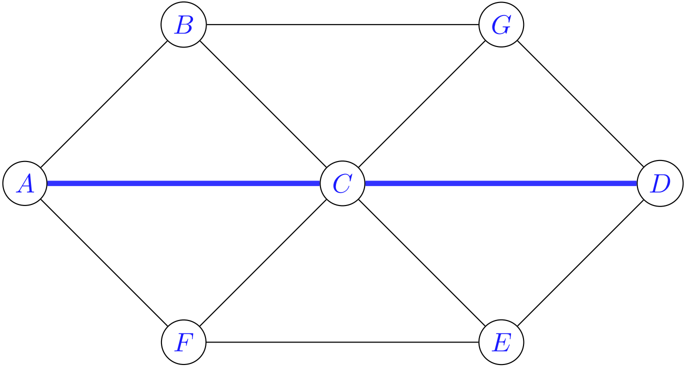


Example: static vs temporal graphs

From paths to temporal paths:

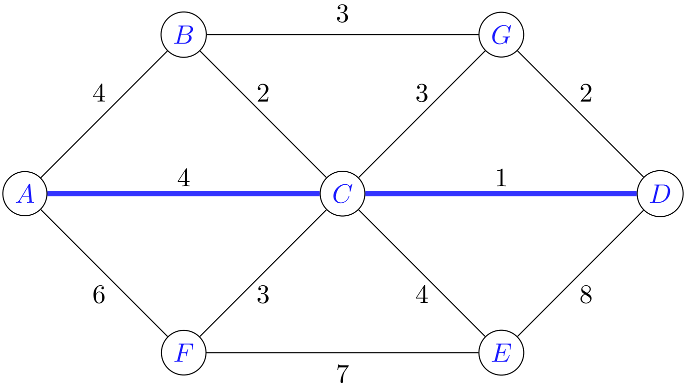


From paths to temporal paths:



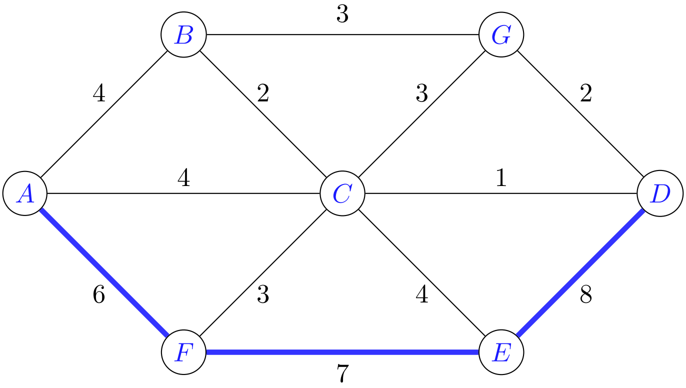
Example: static vs temporal graphs

From paths to temporal paths:



Example: static vs temporal graphs

From paths to temporal paths:



Overview

- Basic definitions
- Temporal vertex cover
- Temporal vertex cover with a sliding time window
- Open problems

Basic definitions I

To specify a temporal graph class, we can:

- either restrict the underlying graph G,
- or restrict the labeling $\lambda : E \to 2^{\mathbb{N}}$ (or both)

Basic definitions I

To specify a temporal graph class, we can:

- either restrict the underlying graph G,
- or restrict the labeling $\lambda : E \to 2^{\mathbb{N}}$ (or both)

Definition (Temporal Graph Classes)

For a class $\mathcal{X}$ of static graphs we say that a temporal graph (G, λ) is

- $\mathcal{X}$ temporal, if $G \in \mathcal{X}$;
- always $\mathcal{X}$ temporal, if $G_i \in \mathcal{X}$ for every $i \in [T] = \{1, 2, \ldots, T\}$.

Basic definitions I

To specify a temporal graph class, we can:

- either restrict the underlying graph G,
- or restrict the labeling $\lambda : E \to 2^{\mathbb{N}}$ (or both)

Definition (Temporal Graph Classes)

For a class $\mathcal{X}$ of static graphs we say that a temporal graph (G, λ) is

- $\mathcal{X}$ temporal, if $G \in \mathcal{X}$;
- always $\mathcal{X}$ temporal, if $G_i \in \mathcal{X}$ for every $i \in [T] = \{1, 2, \ldots, T\}$.

Definition (Temporal Vertex Subset)

A pair $(u, t) \in V \times [T]$ is called the appearance of vertex u at time t.
A temporal vertex subset of (G, λ) is a set $\mathcal{S} \subseteq V \times [T]$ of vertex appearances in (G, λ).

Basic definitions II

Definition (Edge is Temporally Covered)

A vertex appearance (w, t) temporally covers an edge e if:

(i) w covers e, i.e. $w \in e$, and

(ii) $t \in \lambda(e)$, i.e. the edge e is active during the time slot t.

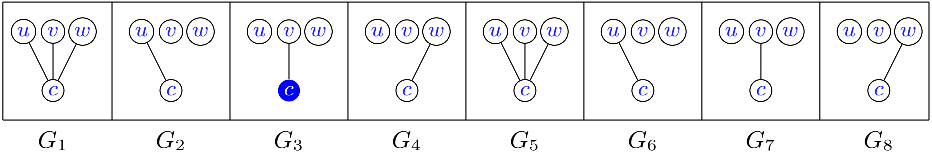
Definition (Edge is Temporally Covered)

A vertex appearance (w, t) temporally covers an edge e if:

(i) w covers e, i.e. $w \in e$, and

(ii) $t \in \lambda(e)$, i.e. the edge e is active during the time slot t.

Example:



– $(c, 3)$ temporally covers edge cv, but

– $(c, 3)$ temporally covers neither cu, nor cw.

Basic definitions: Temporal Vertex Cover

Definition (Temporal Vertex Cover)

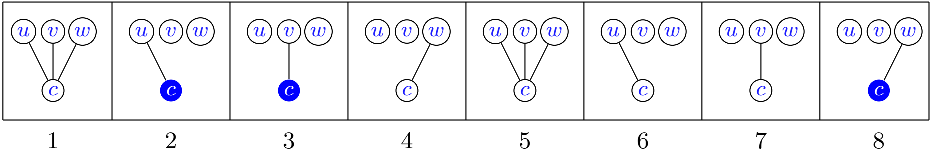
A temporal vertex cover of (G, λ) is a temporal vertex subset $\mathcal{S}$ of (G, λ) such that every edge $e \in E(G)$ is temporally covered by at least one vertex appearance in $\mathcal{S}$.

Basic definitions: Temporal Vertex Cover

Definition (Temporal Vertex Cover)

A temporal vertex cover of (G, λ) is a temporal vertex subset $\mathcal{S}$ of (G, λ) such that every edge $e \in E(G)$ is temporally covered by at least one vertex appearance in $\mathcal{S}$.

Example



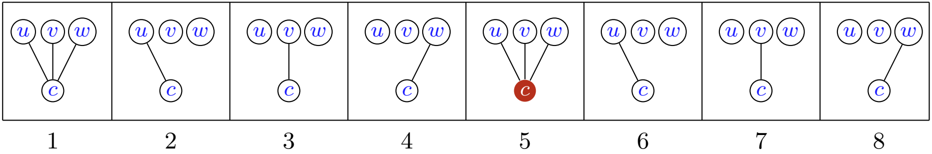
– $\{(c, 2), (c, 3), (c, 8)\}$ is a Temporal Vertex Cover

Basic definitions: Temporal Vertex Cover

Definition (Temporal Vertex Cover)

A temporal vertex cover of (G, λ) is a temporal vertex subset $\mathcal{S}$ of (G, λ) such that every edge $e \in E(G)$ is temporally covered by at least one vertex appearance in $\mathcal{S}$.

Example



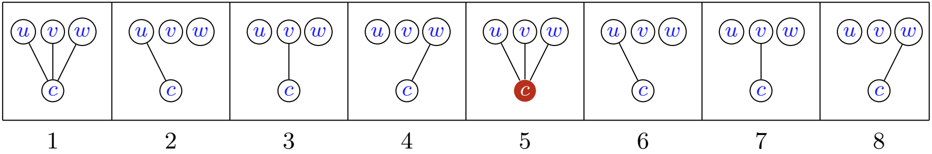
- $\{(c, 2), (c, 3), (c, 8)\}$ is a Temporal Vertex Cover
- $\{(c, 5)\}$ is a minimum Temporal Vertex Cover

Basic definitions: Temporal Vertex Cover

Definition (Temporal Vertex Cover)

A temporal vertex cover of (G, λ) is a temporal vertex subset $\mathcal{S}$ of (G, λ) such that every edge $e \in E(G)$ is temporally covered by at least one vertex appearance in $\mathcal{S}$.

Example



Temporal Vertex Cover (TVC)

Input: A temporal graph (G, λ).
Output: A temporal vertex cover $\mathcal{S}$ of (G, λ) with the minimum $|\mathcal{S}|$.

Basic definitions: Sliding Window Temporal Vertex Cover

Definition (Time Windows)

1. For every time slot $t \in [1, T - \Delta + 1]$:
   the time window $W_t = [t, t + \Delta - 1]$ is the sequence of the Δ consecutive time slots $t, t+1, \ldots, t + \Delta - 1$.

Basic definitions: Sliding Window Temporal Vertex Cover

Definition (Time Windows)

1. For every time slot $t \in [1, T - \Delta + 1]$:
   the time window $W_t = [t, t + \Delta - 1]$ is the sequence of the Δ consecutive time slots $t, t+1, \ldots, t+\Delta-1$.

2. $E[W_t] = \bigcup_{i \in W_t} E_i$ is the union of all edges appearing at least once in the time window W_t.

Basic definitions: Sliding Window Temporal Vertex Cover

Definition (Time Windows)

1. For every time slot $t \in [1, T - \Delta + 1]$:
   the time window $W_t = [t, t + \Delta - 1]$ is the sequence of the Δ consecutive time slots $t, t + 1, \ldots, t + \Delta - 1$.
2. $E[W_t] = \bigcup_{i \in W_t} E_i$ is the union of all edges appearing at least once in the time window W_t.
3. $\mathcal{S}[W_t] = \{(w, t) \in \mathcal{S} : t \in W_t\}$ is the restriction of the temporal vertex subset $\mathcal{S}$ to the window W_t.

Basic definitions: Sliding Window Temporal Vertex Cover

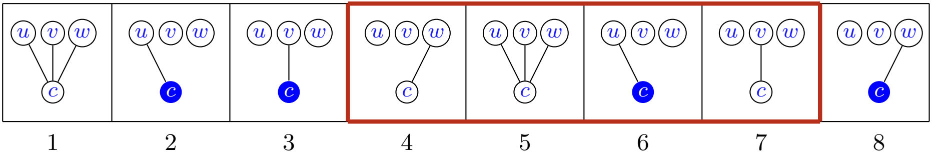
Definition (Sliding Δ-Window Temporal Vertex Cover)

A sliding Δ-window temporal vertex cover of (G, λ) is a temporal vertex subset $\mathcal{S}$ of (G, λ) such that:

- for every time window W_t and for every edge $e \in E[W_t]$,
- e is temporally covered by at least one vertex appearance $(w, t) \in \mathcal{S}[W_t]$.

Basic definitions: Sliding Window Temporal Vertex Cover

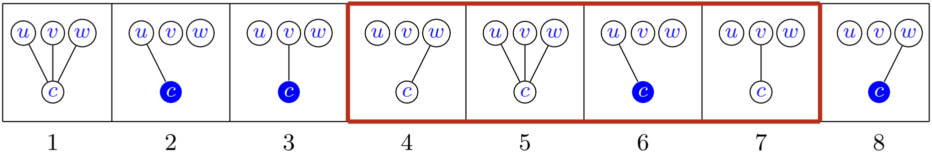
Example ($\Delta = 4$)



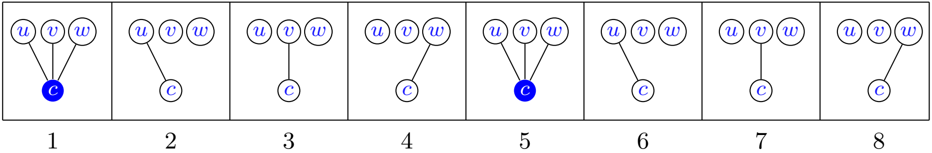
- $\{(c, 2), (c, 3), (c, 6), (c, 8)\}$ is not a sliding Δ-window temporal vertex cover, as edges $cv, cw \in E[W_4]$ are not temporally covered in window W_4.

Basic definitions: Sliding Window Temporal Vertex Cover

Example ($\Delta = 4$)



– $\{(c,2),(c,3),(c,6),(c,8)\}$ is not a sliding Δ-window temporal vertex cover, as edges $cv, cw \in E[W_4]$ are not temporally covered in window W_4.



– $\{(c,1),(c,5)\}$ is a sliding Δ-window temporal vertex cover.

Basic definitions: Sliding Window Temporal Vertex Cover

Sliding Window Temporal Vertex Cover (SW-TVC)

Input: A temporal graph (G, λ) with lifetime T, and an integer $\Delta \leq T$.
Output: A sliding Δ-window temporal vertex cover $\mathcal{S}$ of (G, λ) with the minimum $|\mathcal{S}|$.

Motivation:

- **(static) Vertex Cover:**
  network surveillance (e.g. CCTV cameras etc.)
- **Temporal Vertex Cover:**
  network surveillance in a dynamic network
- **Sliding Window Temporal Vertex Cover:**
  dynamic surveillance in every possible Δ-time window
  (e.g. for crimes that need time Δ to be performed)

Overview

- Basic definitions
- Temporal vertex cover
- Temporal vertex cover with a sliding time window
- Open problems

Temporal Vertex Cover: the star temporal case

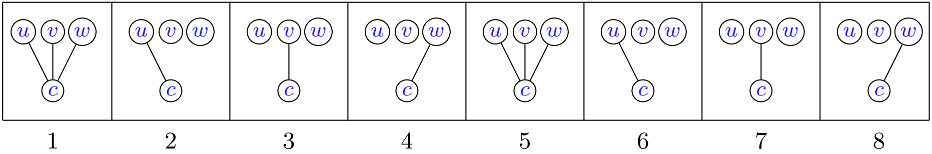
Lemma

TVC *on star temporal graphs is equivalent to* SET COVER.

- leafs of the underlying star $\leftrightarrow$ ground set of the SET COVER instance
- each snapshot graph $\leftrightarrow$ a set in the SET COVER instance

Goal: Choose sets (snapshots) to cover all elements (leafs' edges)

Example:



Temporal Vertex Cover: the star temporal case

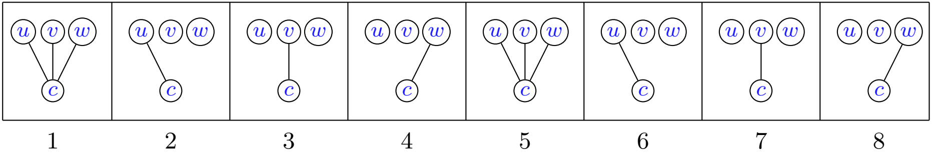
Lemma

TVC *on star temporal graphs is equivalent to* SET COVER.

- leafs of the underlying star $\leftrightarrow$ ground set of the SET COVER instance
- each snapshot graph $\leftrightarrow$ a set in the SET COVER instance

Goal: Choose sets (snapshots) to cover all elements (leafs' edges)

Example:



1. Universe: $\{u, v, w\}$
2. Sets: $S_1 = \{u, v, w\}$, $S_2 = \{u\}$, $S_3 = \{v\}$, $S_4 = \{w\}$, ...

Temporal Vertex Cover: the star temporal case

Lemma

TVC *on star temporal graphs is equivalent to* SET COVER.

Lemma

TVC *on star temporal graphs is equivalent to* HITTING SET.

Temporal Vertex Cover: the star temporal case

Lemma

TVC on star temporal graphs is equivalent to SET COVER.

Lemma

TVC on star temporal graphs is equivalent to HITTING SET.

Consequences:

1. TVC is NP-complete even on star temporal graphs.
2. For any $\varepsilon < 1$, TVC on star temporal graphs cannot be optimally solved in $O(2^{\varepsilon T})$ time, unless SETH fails (due to Hitting Set).

Temporal Vertex Cover: the star temporal case

Lemma

TVC *on star temporal graphs is equivalent to* SET COVER*.*

Lemma

TVC *on star temporal graphs is equivalent to* HITTING SET*.*

Consequences:

1. TVC is NP-complete even on star temporal graphs.
2. For any $\varepsilon < 1$, TVC on star temporal graphs cannot be optimally solved in $O(2^{\varepsilon T})$ time, unless SETH fails (due to Hitting Set).
3. TVC does not admit a polynomial-time $(1-\varepsilon)\ln n$-approximation algorithm, unless NP has $n^{O(\log\log n)}$-time deterministic algorithms.

Temporal Vertex Cover: the star temporal case

Lemma

TVC *on star temporal graphs is equivalent to* SET COVER.

Lemma

TVC *on star temporal graphs is equivalent to* HITTING SET.

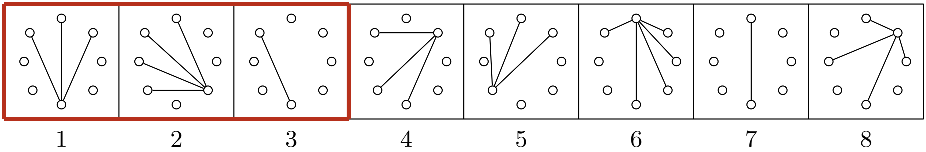
Consequences:

1. TVC is NP-complete even on star temporal graphs.
2. For any $\varepsilon < 1$, TVC on star temporal graphs cannot be optimally solved in $O(2^{\varepsilon T})$ time, unless SETH fails (due to Hitting Set).
3. TVC does not admit a polynomial-time $(1-\varepsilon)\ln n$-approximation algorithm, unless NP has $n^{O(\log\log n)}$-time deterministic algorithms.
4. TVC on star temporal graphs can be $\ln n$-approximated in polynomial time.
5. For general graphs: $2\ln n$-approximation algorithm by a similar reduction from TVC to SET COVER

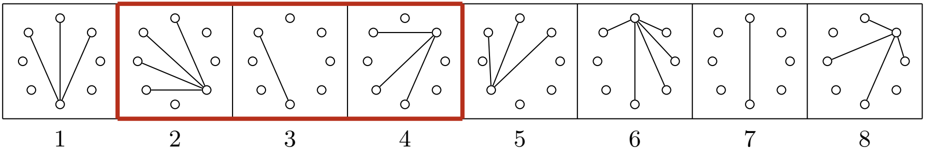
Overview

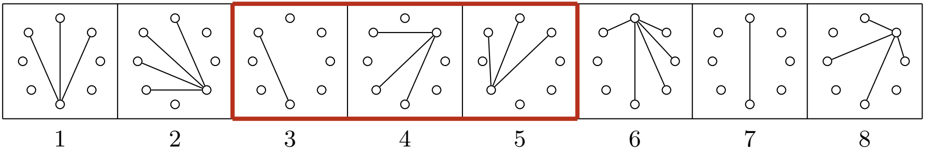
- Basic definitions
- Alternative models
- Temporal vertex cover
- Temporal vertex cover with a sliding time window
- Open problems

SW-TVC: **always star** temporal graphs

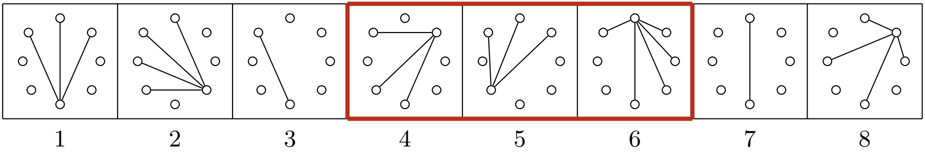


SW-TVC: **always star** temporal graphs

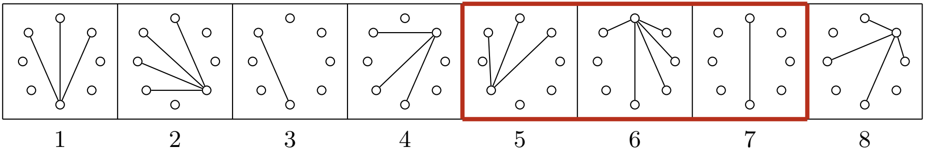



1
2
3
4
5
6
7
8

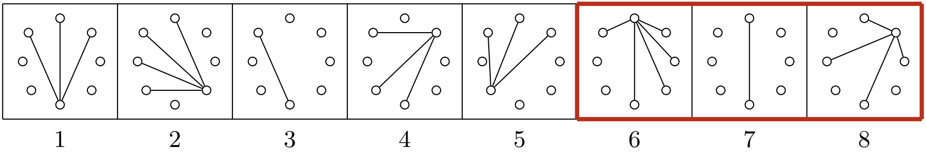
SW-TVC: **always star** temporal graphs



SW-TVC: **always star** temporal graphs

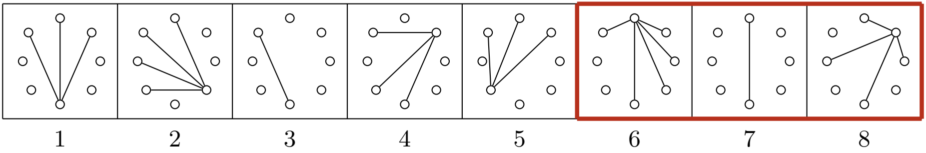


SW-TVC: **always star** temporal graphs



- On always star temporal graphs, a minimum size SW-TVC contains at most one vertex (the star center) in each snapshot

$\Rightarrow$ we assign a Boolean variable $x_i \in \{0, 1\}$ for the snapshot at time i

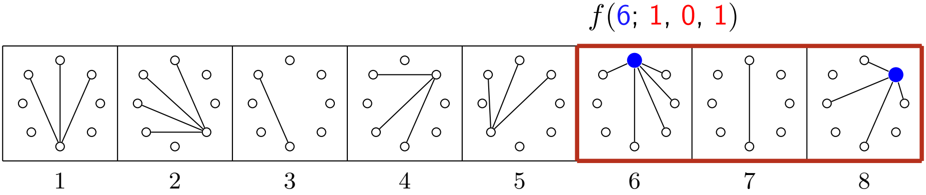


- On always star temporal graphs, a minimum size SW-TVC contains at most one vertex (the star center) in each snapshot

$\Rightarrow$ we assign a Boolean variable $x_i \in \{0,1\}$ for the snapshot at time i

- For variables $x_1, x_2, \ldots, x_\Delta$ we define $f(t; x_1, x_2, \ldots, x_\Delta)$ to be the smallest cardinality of a sliding Δ-window temporal vertex cover $\mathcal{S}$ of $(G,\lambda)|_{[1,t+\Delta-1]}$, such that the solution at times $t, t+1, \ldots, t+\Delta-1$ is defined by the variables $x_1, x_2, \ldots, x_\Delta$.

SW-TVC: **always star** temporal graphs

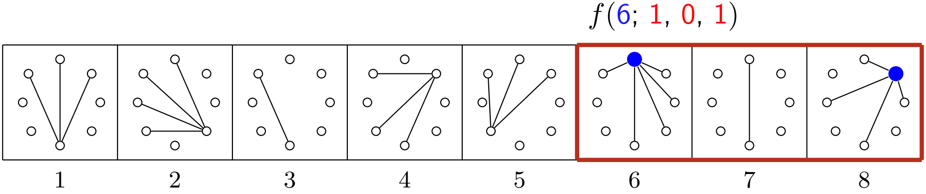


- On always star temporal graphs, a minimum size SW-TVC contains at most one vertex (the star center) in each snapshot

$\Rightarrow$ we assign a Boolean variable $x_i \in \{0, 1\}$ for the snapshot at time i

- For variables $x_1, x_2, \ldots, x_\Delta$ we define $f(t; x_1, x_2, \ldots, x_\Delta)$ to be the smallest cardinality of a sliding Δ-window temporal vertex cover $\mathcal{S}$ of $(G, \lambda)|_{[1,t+\Delta-1]}$, such that the solution at times $t, t+1, \ldots, t+\Delta-1$ is defined by the variables $x_1, x_2, \ldots, x_\Delta$.

SW-TVC: **always star** temporal graphs



- On always star temporal graphs, a minimum size SW-TVC contains at most one vertex (the star center) in each snapshot

$\Rightarrow$ we assign a Boolean variable $x_i \in \{0, 1\}$ for the snapshot at time i

- For variables $x_1, x_2, \ldots, x_\Delta$ we define $f(t; x_1, x_2, \ldots, x_\Delta)$ to be the smallest cardinality of a sliding Δ-window temporal vertex cover $\mathcal{S}$ of $(G, \lambda)|_{[1,t+\Delta-1]}$, such that the solution at times $t, t+1, \ldots, t+\Delta-1$ is defined by the variables $x_1, x_2, \ldots, x_\Delta$.

Lemma (dynamic programming)

$$f(t; x_1, x_2, \ldots, x_\Delta) = x_\Delta + \min_{y \in \{0,1\}} \{f(t-1; y, x_1, x_2, \ldots, x_{\Delta-1})\}$$

Theorem (always star temporal graphs)

SW-TVC *on always star temporal graphs can be solved in* $O(T\Delta(n+m) \cdot 2^{\Delta})$ *time.*

Theorem (always star temporal graphs)

SW-TVC *on always star temporal graphs can be solved in* $O(T\Delta(n+m) \cdot 2^{\Delta})$ *time.*

Theorem (the general case)

SW-TVC *on general temporal graphs can be solved in* $O(T\Delta(n+m) \cdot 2^{n(\Delta+1)})$ *time.*

Theorem (always star temporal graphs)

SW-TVC *on always star temporal graphs can be solved in* $O(T\Delta(n+m) \cdot 2^{\Delta})$ *time.*

Theorem (the general case)

SW-TVC *on general temporal graphs can be solved in* $O(T\Delta(n+m) \cdot 2^{n(\Delta+1)})$ *time.*

Main idea:

- for each of the Δ snapshots in the (currently) last Δ-window, we enumerate all 2^n vertex subsets,
- instead of just enumerating over the truth values of Δ Boolean variables ("always star" case)

$\mathrm{SW\text{-}TVC}$: Optimality under ETH

Theorem

For any two (arbitrarily growing) functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$, there exists a constant $\varepsilon \in (0, 1)$ such that $\mathrm{SW\text{-}TVC}$ cannot be solved in $f(T) \cdot 2^{\varepsilon n \cdot g(\Delta)}$ time assuming ETH.

SW-TVC: Optimality under ETH

Theorem

For any two (arbitrarily growing) functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$, there exists a constant $\varepsilon \in (0, 1)$ such that SW-TVC cannot be solved in $f(T) \cdot 2^{\varepsilon n \cdot g(\Delta)}$ time assuming ETH.

Proof (idea):

- reduction from VERTEX COVER
- $T = \Delta = 2$
- $G_1 = G$; G_2 is an independent set
- given f and g, choose appropriate ε

SW-TVC: Optimality under ETH

Theorem

For any two (arbitrarily growing) functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$, there exists a constant $\varepsilon \in (0, 1)$ such that $\mathrm{SW\text{-}TVC}$ cannot be solved in $f(T) \cdot 2^{\varepsilon n \cdot g(\Delta)}$ time assuming ETH.

Proof (idea):

- reduction from Vertex Cover
- $T = \Delta = 2$
- $G_1 = G$; G_2 is an independent set
- given f and g, choose appropriate ε

Corollary

Our $O(T\Delta(n + m) \cdot 2^{n(\Delta+1)})$-time algorithm is asymptotically almost optimal (assuming ETH).

SW-TVC: always bounded vertex cover number temporal graphs

Let $\mathcal{C}_k$ be the class of graphs with vertex cover number at most k.

Theorem

SW-TVC *on always* $\mathcal{C}_k$ *temporal graphs can be solved in* $O(T\Delta(n+m) \cdot n^{k(\Delta+1)})$ *time.*

SW-TVC: always bounded vertex cover number temporal graphs

Let $\mathcal{C}_k$ be the class of graphs with vertex cover number at most k.

Theorem

SW-TVC *on always* $\mathcal{C}_k$ *temporal graphs can be solved in* $O(T\Delta(n+m)\cdot n^{k(\Delta+1)})$ *time.*

Main idea:

- in the optimal solution, the choice at step i is a subset of a minimum vertex cover at this snapshot

$\Rightarrow$ for each of the Δ last snapshots, enumerate all n^k vertex subsets (candidates for vertex cover at snapshot i)

Δ-TVC

If the parameter Δ (the size of a sliding window) is fixed, we refer to $\mathrm{SW\text{-}TVC}$ as $\Delta\text{-}\mathrm{TVC}$ (i.e. Δ is a part of the problem name).

Δ-TVC

If the parameter Δ (the size of a sliding window) is fixed, we refer to $\mathrm{SW\text{-}TVC}$ as $\Delta\text{-}\mathrm{TVC}$ (i.e. Δ is a part of the problem name).

Observation

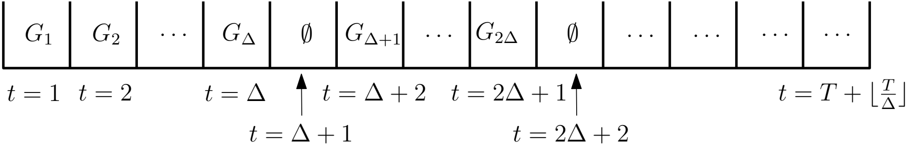
$(\Delta + 1)\text{-}\mathrm{TVC}$ *is at least as hard as* $\Delta\text{-}\mathrm{TVC}$*.*

Δ-TVC

If the parameter Δ (the size of a sliding window) is fixed, we refer to SW-TVC as Δ-TVC (i.e. Δ is a part of the problem name).

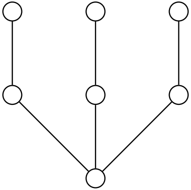
Observation

$(\Delta + 1)$-TVC *is at least as hard as* Δ-TVC.



2-TVC is hard to approximate

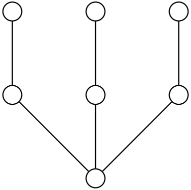
Let $\mathcal{X}$ be the class of graphs whose connected components are induced subgraphs of graph Ψ.



Ψ

2-TVC is hard to approximate

Let $\mathcal{X}$ be the class of graphs whose connected components are induced subgraphs of graph Ψ.

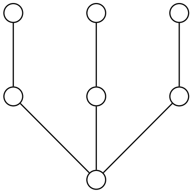


Ψ

Clearly, VERTEX COVER is linearly solvable on graphs from $\mathcal{X}$.

2-TVC is hard to approximate

Let $\mathcal{X}$ be the class of graphs whose connected components are induced subgraphs of graph Ψ.



Ψ

Clearly, Vertex Cover is linearly solvable on graphs from $\mathcal{X}$.

Theorem

There is no PTAS for 2-TVC on always $\mathcal{X}$ temporal graphs.

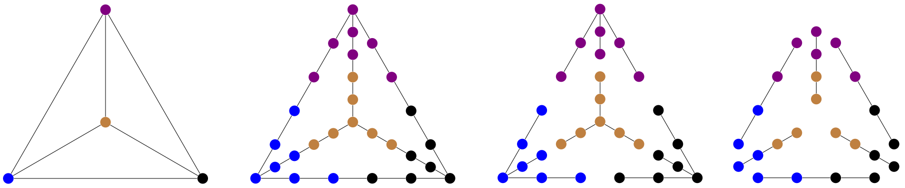
2-TVC is hard to approximate

Theorem

There is no PTAS for 2-TVC on always $\mathcal{X}$ temporal graphs.

Proof (sketch):

1. Let $\mathcal{Y}$ be the class of graphs which can be obtained from cubic graphs by subdividing every edge exactly 4 times.
2. There is no PTAS for VERTEX COVER on $\mathcal{Y}$.
3. Reduce VERTEX COVER on $\mathcal{Y}$ to 2-TVC on always $\mathcal{X}$ temporal graphs such that optimal solutions of both problems have same size.



K_4 — The 4-subdivision of K_4 — Snapshot G_1 — Snapshot G_2

SW-TVC: approximation algorithms I

Reduction from SW-TVC to Set Cover.

SW-TVC: approximation algorithms I

Reduction from SW-TVC to SET COVER.

1. The universe: the set of all pairs $(e, t) \in E \times [T - \Delta + 1]$ such that e appears (and so must be temporally covered) within window W_t.

SW-TVC: approximation algorithms I

Reduction from SW-TVC to Set Cover.

1. The universe: the set of all pairs $(e, t) \in E \times [T - \Delta + 1]$ such that e appears (and so must be temporally covered) within window W_t.
2. The sets: for every vertex appearance (v, s) we define $C_{v,s}$ to be the set of elements (e, t) in the universe, such that (v, s) temporally covers e in window W_t.

SW-TVC: approximation algorithms I

Reduction from SW-TVC to Set Cover.

1. The universe: the set of all pairs $(e, t) \in E \times [T - \Delta + 1]$ such that e appears (and so must be temporally covered) within window W_t.
2. The sets: for every vertex appearance (v, s) we define $C_{v,s}$ to be the set of elements (e, t) in the universe, such that (v, s) temporally covers e in window W_t.

Consequences:

1. $O(\ln n + \ln \Delta)$-approximation
(every set $C_{v,s}$ has at most $n\Delta$ elements
$\Rightarrow$ approximation factor $H_{n\Delta} - \frac{1}{2} \approx \ln n + \ln \Delta$)

SW-TVC: approximation algorithms I

Reduction from SW-TVC to SET COVER.

1. The universe: the set of all pairs $(e,t) \in E \times [T-\Delta+1]$ such that e appears (and so must be temporally covered) within window W_t.

2. The sets: for every vertex appearance (v,s) we define $C_{v,s}$ to be the set of elements (e,t) in the universe, such that (v,s) temporally covers e in window W_t.

Consequences:

1. $O(\ln n + \ln \Delta)$-approximation
   (every set $C_{v,s}$ has at most $n\Delta$ elements
   $\Rightarrow$ approximation factor $H_{n\Delta} - \frac{1}{2} \approx \ln n + \ln \Delta$)
2. $2k$-approximation, where k is the maximum edge Δ-frequency
   (just take both vertex appearances for every appearance of an edge)

SW-TVC: approximation algorithms I

Reduction from SW-TVC to SET COVER.

1. The universe: the set of all pairs $(e, t) \in E \times [T - \Delta + 1]$ such that e appears (and so must be temporally covered) within window W_t.

2. The sets: for every vertex appearance (v, s) we define $C_{v,s}$ to be the set of elements (e, t) in the universe, such that (v, s) temporally covers e in window W_t.

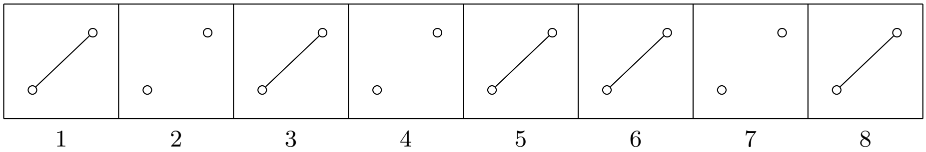
Consequences:

1. $O(\ln n + \ln \Delta)$-approximation
   (every set $C_{v,s}$ has at most $n\Delta$ elements
   $\Rightarrow$ approximation factor $H_{n\Delta} - \frac{1}{2} \approx \ln n + \ln \Delta$)
2. $2k$-approximation, where k is the maximum edge Δ-frequency
   (just take both vertex appearances for every appearance of an edge)

$\Rightarrow$ 2Δ-approximation

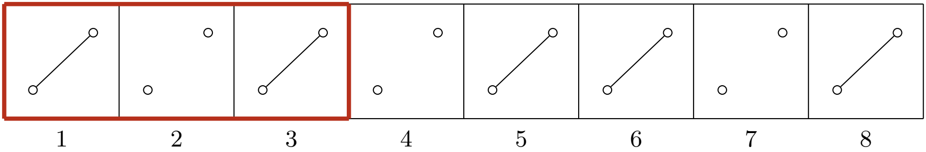
SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm



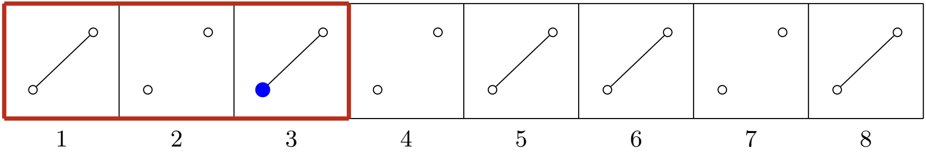
SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm



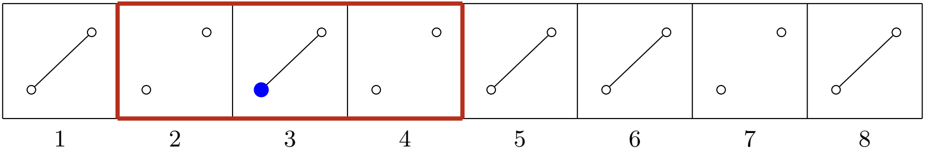
SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm



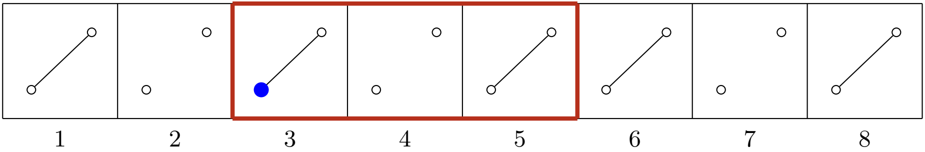
SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm



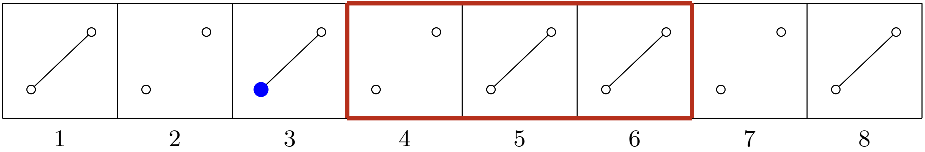
SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm



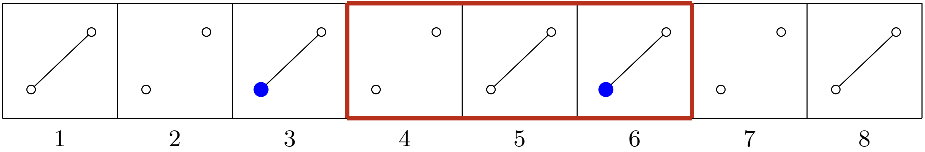
SW-TVC: approximation algorithms II

Single-edge temporal graph: **exact algorithm**



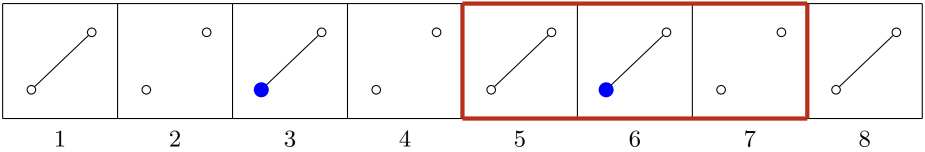
SW-TVC: approximation algorithms II

Single-edge temporal graph: **exact algorithm**



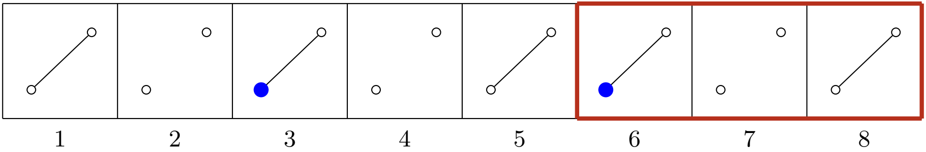
SW-TVC: approximation algorithms II

Single-edge temporal graph: **exact algorithm**



SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm



SW-TVC: approximation algorithms II

Single-edge temporal graph: exact algorithm

Algorithm SW-TVC on single-edge temporal graphs

Input: A temporal graph (G, λ) of lifetime T with $V(G) = \{u, v\}$; and $\Delta \leq T$.
Output: A minimum-cardinality sliding Δ-window temporal vertex cover $\mathcal{S}$ of (G, λ).

$\mathcal{S} \leftarrow \emptyset$
$t = 1$
while $t \leq T - \Delta + 1$ **do**
  if $\exists\ r \in [t, t + \Delta - 1]$ such that $uv \in E_r$ **then**
    choose maximum such r and add (u, r) to $\mathcal{S}$
    $t \leftarrow r + 1$
  else
    $t \leftarrow t + 1$
return $\mathcal{S}$

- greedy algorithm
- linear time

SW-TVC: approximation algorithms II

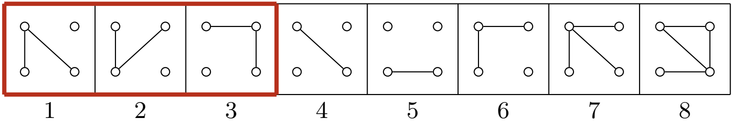
Always degree at most d temporal graphs: d-approx. algorithm

Main idea:

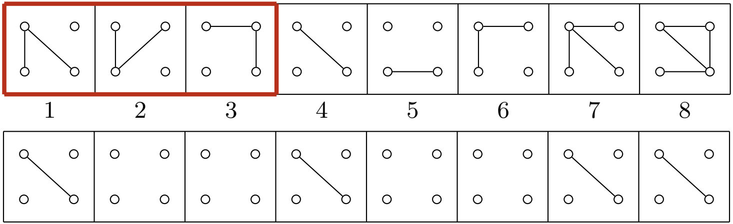
- solve independently each single-edge subgraph of G
- take the union of the solutions

SW-TVC: approximation algorithms II

Always degree at most d temporal graphs: d-approx. algorithm

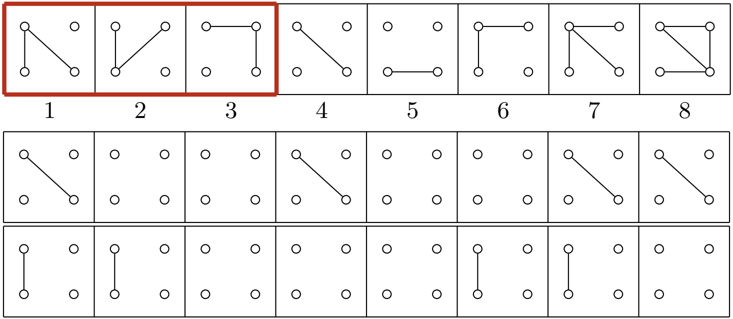


Always degree at most d temporal graphs: **d-approx. algorithm**



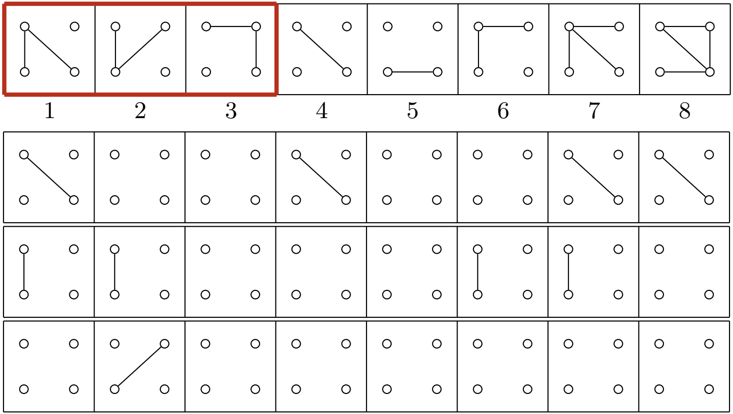
SW-TVC: approximation algorithms II

Always degree at most d temporal graphs: d-approx. algorithm



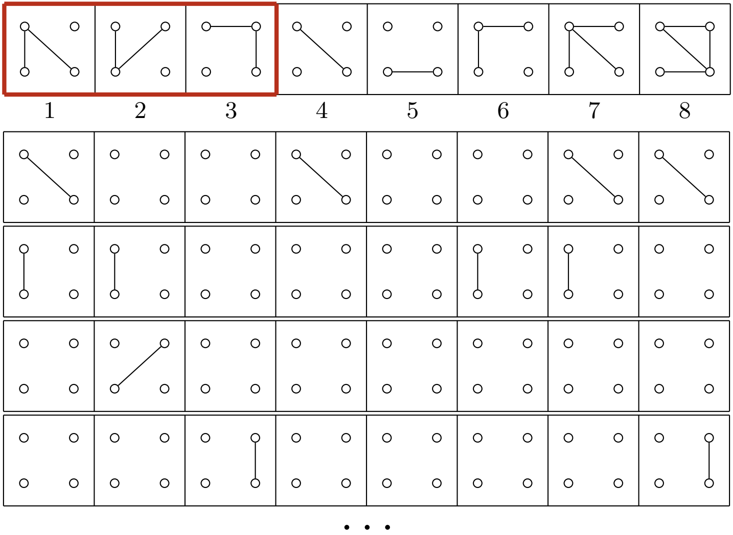
SW-TVC: approximation algorithms II

Always degree at most d temporal graphs: d-approx. algorithm



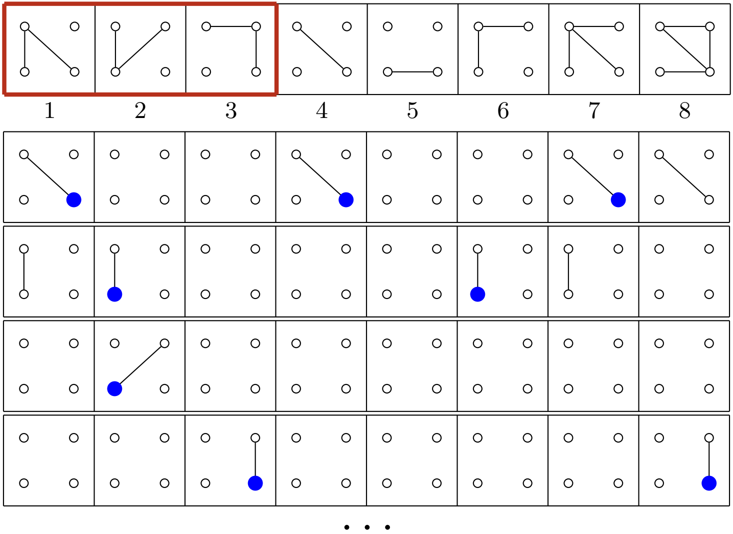
SW-TVC: approximation algorithms II

Always degree at most d temporal graphs: d-approx. algorithm



SW-TVC: approximation algorithms II

Always degree at most d temporal graphs: **d-approx. algorithm**



$\mathrm{SW\text{-}TVC}$: approximation algorithms II

Always degree at most d temporal graphs: d-approx. algorithm

Algorithm d-approximation of $\mathrm{SW\text{-}TVC}$ on always degree at most d temporal graphs

Input: An always degree at most d temporal graph (G, λ) of lifetime T, and $\Delta \leq T$.
Output: A sliding Δ-window temporal vertex cover $\mathcal{S}$ of (G, λ).

1: **for** every edge $uv \in E(G)$ **do**
2: Compute an optimal solution $\mathcal{S}_{uv}$ of the problem for $(G[\{u, v\}], \lambda)$
3: $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_{uv}$
return $\mathcal{S}$

Lemma

The above algorithm is a $O(mT)$-time d-approximation algorithm for $\mathrm{SW\text{-}TVC}$ on always degree at most d temporal graphs.

SW-TVC: approximation algorithms II

Always degree at most d temporal graphs: d-approx. algorithm

Algorithm d-approximation of SW-TVC on always degree at most d temporal graphs

Input: An always degree at most d temporal graph (G, λ) of lifetime T, and $\Delta \leq T$.
Output: A sliding Δ-window temporal vertex cover $\mathcal{S}$ of (G, λ).

```
for every edge uv ∈ E(G) do
    Compute an optimal solution S_uv of the problem for (G[{u,v}], λ)
    S ← S ∪ S_uv
return S
```

Lemma

The above algorithm is a $O(mT)$-time d-approximation algorithm for SW-TVC on always degree at most d temporal graphs.

Corollary

SW-TVC can be optimally solved in $O(mT)$ time on the class of always degree at most 1 (matching) temporal graphs.

Overview

- Basic definitions
- Alternative models
- Temporal vertex cover
- Temporal vertex cover with a sliding time window
- Open problems

Open Problems

Problem 1

Determine the complexity status of Δ-TVC *on degree at most 2 temporal graphs.*

1. Δ-TVC on always degree at most 1 can be solved in linear time.

Open Problems

Problem 1

Determine the complexity status of Δ-TVC on degree at most 2 temporal graphs.

1. Δ-TVC on always degree at most 1 can be solved in linear time.
2. Δ-TVC on always degree at most 3: no PTAS

Open Problems

Problem 1

Determine the complexity status of Δ-TVC on degree at most 2 temporal graphs.

1. Δ-TVC on always degree at most 1 can be solved in linear time.
2. Δ-TVC on always degree at most 3: no PTAS, even when:
    1. the underlying graph has degree at most 3; and
    2. connected components of snapshots have at most 7 vertices.

Open Problems

Problem 1

Determine the complexity status of Δ-TVC on degree at most 2 temporal graphs.

1. Δ-TVC on always degree at most 1 can be solved in linear time.
2. Δ-TVC on always degree at most 3: no PTAS, even when:
   1. the underlying graph has degree at most 3; and
   2. connected components of snapshots have at most 7 vertices.

Problem 2

Can Δ-TVC on general graphs be approximated within a factor better than 2Δ?

Open Problems

Problem 1

Determine the complexity status of Δ-TVC on degree at most 2 temporal graphs.

1. Δ-TVC on always degree at most 1 can be solved in linear time.
2. Δ-TVC on always degree at most 3: no PTAS, even when:
   1. the underlying graph has degree at most 3; and
   2. connected components of snapshots have at most 7 vertices.

Problem 2

Can Δ-TVC on general graphs be approximated within a factor better than 2Δ?

Problem 3

Can Δ-TVC on always degree at most d temporal graphs be approximated within a factor better than d?

Thank you!