Finding Temporal Paths under Waiting Time Constraints

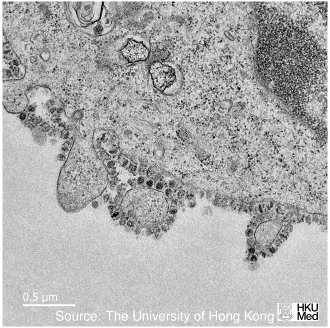
Philipp Zschoche

TU Berlin

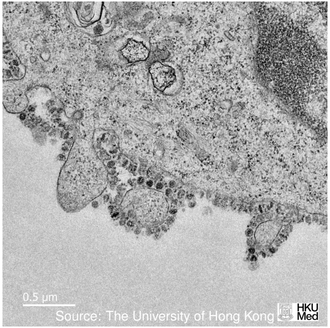
July 7 2020, Algorithmic Aspects of Temporal Graphs III

Based on joint work with Arnaud Casteigts, Anne-Sophie Himmel, and Hendrik Molter.

Motivational Example: Disease Control

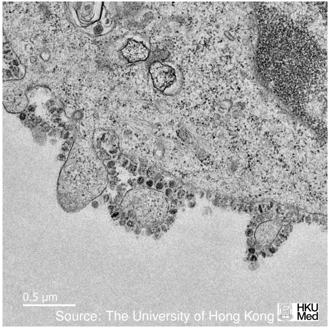


Motivational Example: Disease Control



Infectious diseases are often transmitted via physical contact.

Motivational Example: Disease Control



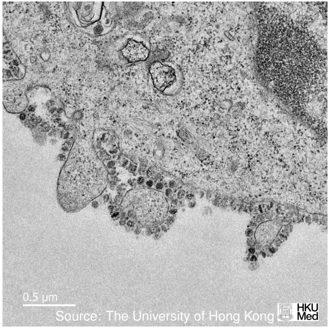
Infectious diseases are often transmitted via physical contact.

Contact Tracing:

(1) Identify and isolate infected persons

(2) Isolate all potentially infected persons (by known cases).

Motivational Example: Disease Control



Infectious diseases are often transmitted via physical contact.

Contact Tracing:

(1) Identify and isolate infected persons

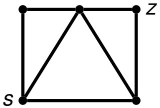
(2) Isolate all potentially infected persons (by known cases).

At time of identification: a person may started **long infection chains**.

Motivational Example: Disease Control

Can s have (indirectly) infected z?

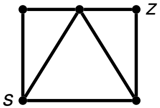
Static graph:



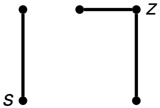
Motivational Example: Disease Control

Can s have (indirectly) infected z?

Static graph:



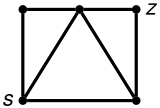
Day 1:



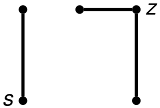
Motivational Example: Disease Control

Can s have (indirectly) infected z?

Static graph:



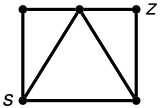
Day 1:



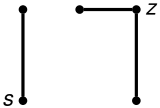
Motivational Example: Disease Control

Can *s* have (indirectly) infected *z*?

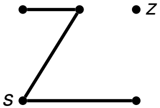
Static graph:



Day 1:



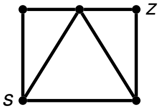
Day 2:



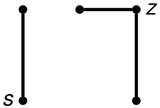
Motivational Example: Disease Control

Can *s* have (indirectly) infected *z*?

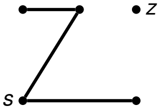
Static graph:



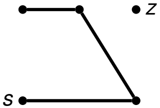
Day 1:



Day 2:



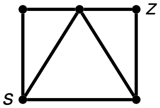
Day 3:



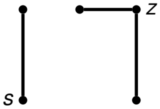
Motivational Example: Disease Control

Can s have (indirectly) infected z?

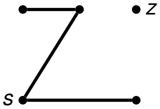
Static graph:



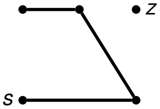
Day 1:



Day 2:



Day 3:



$\Rightarrow$ Time information is crucial for infection transmission routes.

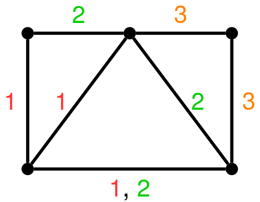
Temporal Graphs – Formal Definition

A **temporal graph** $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$ is defined as vertex set V with a list of edge sets $E_1, \dots, E_\tau$ over V, where τ is the lifetime of $\mathcal{G}$.

Temporal Graphs – Formal Definition

A **temporal graph** $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$ is defined as vertex set V with a list of edge sets $E_1, \ldots, E_\tau$ over V, where τ is the lifetime of $\mathcal{G}$.

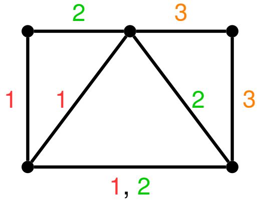
$\mathcal{G}$:



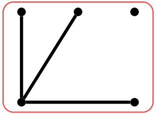
Temporal Graphs – Formal Definition

A **temporal graph** $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$ is defined as vertex set V with a list of edge sets $E_1, \ldots, E_\tau$ over V, where τ is the lifetime of $\mathcal{G}$.

$\mathcal{G}$:



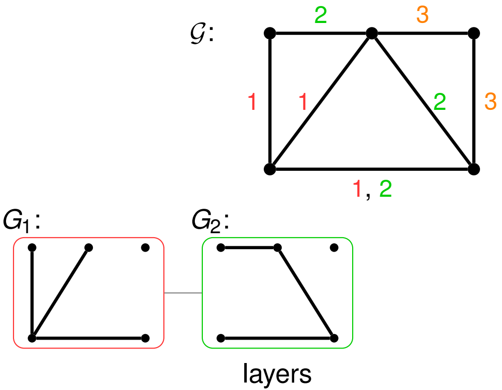
G_1:



layers

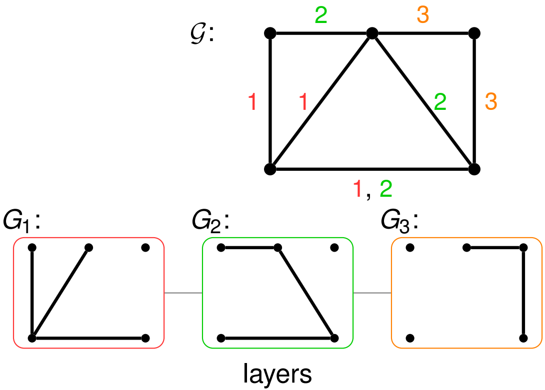
Temporal Graphs – Formal Definition

A **temporal graph** $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$ is defined as vertex set V with a list of edge sets $E_1, \ldots, E_\tau$ over V, where τ is the lifetime of $\mathcal{G}$.



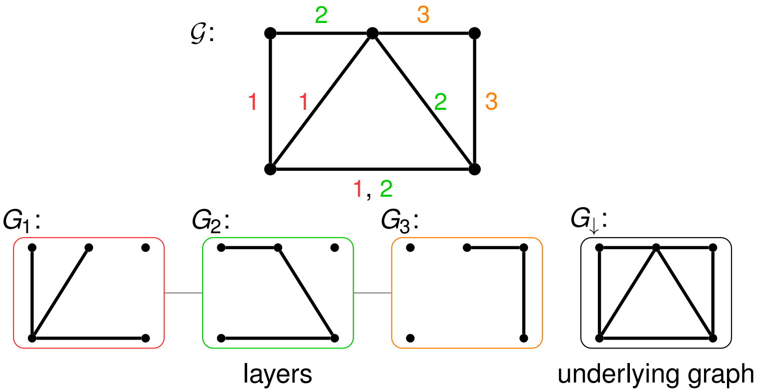
Temporal Graphs – Formal Definition

A **temporal graph** $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$ is defined as vertex set V with a list of edge sets $E_1, \ldots, E_\tau$ over V, where τ is the lifetime of $\mathcal{G}$.

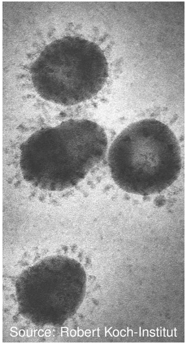


Temporal Graphs – Formal Definition

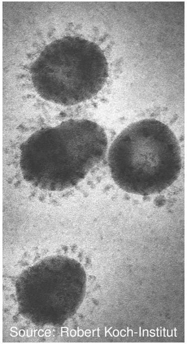
A **temporal graph** $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$ is defined as vertex set V with a list of edge sets $E_1, \ldots, E_\tau$ over V, where τ is the lifetime of $\mathcal{G}$.



Motivation: Disease Spreading

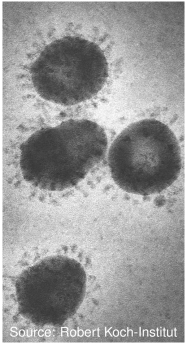


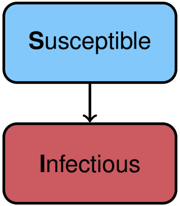
Motivation: Disease Spreading


Source: Robert Koch-Institut

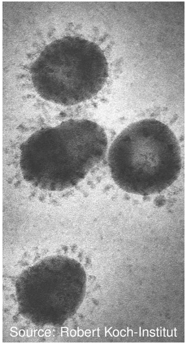
Susceptible

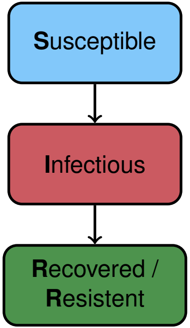
Motivation: Disease Spreading



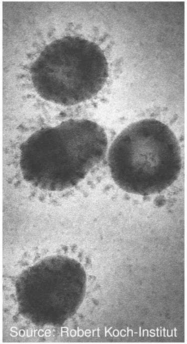


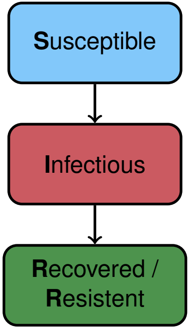
Motivation: Disease Spreading





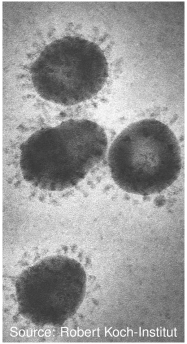
Motivation: Disease Spreading

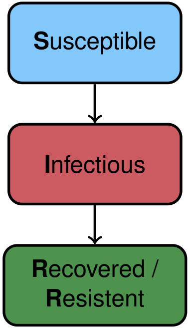




- Infectious period: 5 days.

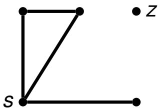
Motivation: Disease Spreading



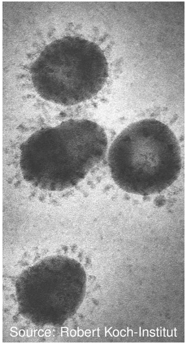


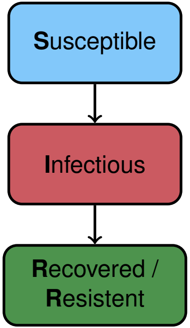
- Infectious period: 5 days.

Day 1:



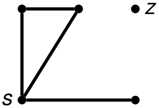
Motivation: Disease Spreading



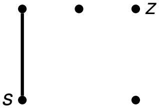


- Infectious period: 5 days.

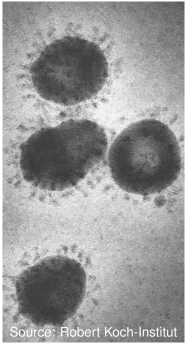
Day 1:

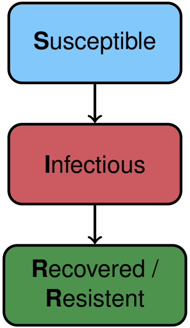


Day 4:



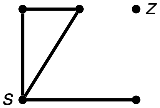
Motivation: Disease Spreading



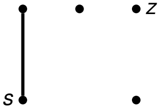


- Infectious period: 5 days.

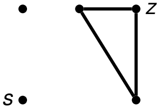
Day 1:



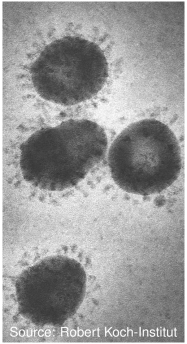
Day 4:

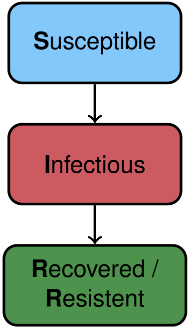


Day 7:



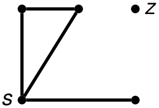
Motivation: Disease Spreading


Source: Robert Koch-Institut

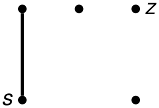


- Infectious period: 5 days.
- Disease spreads along **paths** with **bounded waiting times**.

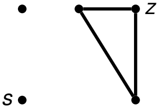
Day 1:



Day 4:



Day 7:



Restless Paths – Formal Definition

A **temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

Restless Paths – Formal Definition

A **temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s, z)-path in the underlying graph

Restless Paths – Formal Definition

A **Δ-restless temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

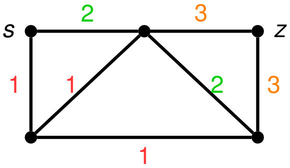
- the edges form an (s, z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

Restless Paths – Formal Definition

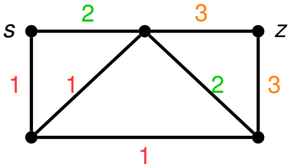
A **Δ-restless temporal** (s,z)**-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s,z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

temporal (s,z)-paths:



1-restless temporal (s,z)-path:

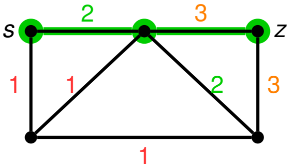


Restless Paths – Formal Definition

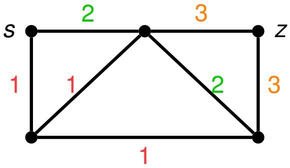
A **Δ-restless temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s, z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

temporal (s, z)-paths:



1-restless temporal (s, z)-path:

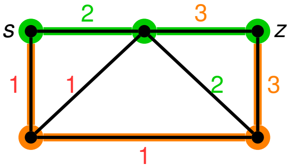


Restless Paths – Formal Definition

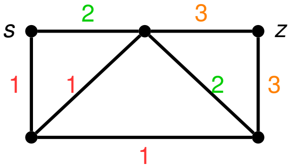
A **Δ-restless temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s, z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

temporal (s, z)-paths:



1-restless temporal (s, z)-path:

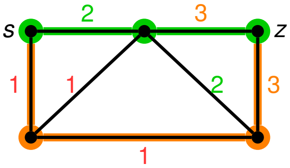


Restless Paths – Formal Definition

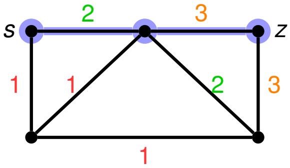
A **Δ-restless temporal (s, z)-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s, z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

temporal (s, z)-paths:



1-restless temporal (s, z)-path:

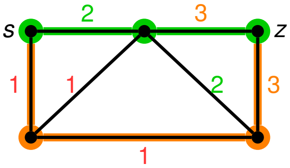


Restless Paths – Formal Definition

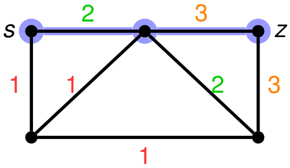
A **Δ-restless temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s, z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

temporal (s, z)-paths:



1-restless temporal (s, z)-path:



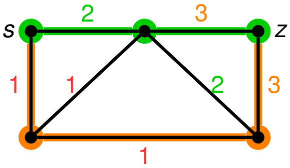
- Temporal Paths: Xuan et al. [IJFCS '03], Wu et al. [IEEE TKDE '16]

Restless Paths – Formal Definition

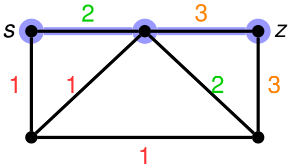
A **Δ-restless temporal** (s, z)**-path** is a list of edges labeled with **non-decreasing** time steps that

- the edges form an (s, z)-path in the underlying graph and
- consecutive time steps differ by at most Δ.

temporal (s, z)-paths:



1-restless temporal (s, z)-path:



- Temporal Paths: Xuan et al. [IJFCS '03], Wu et al. [IEEE TKDE '16]
- Restless Temporal Walks: Himmel et al. [Complex Networks '19]

Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.

Restless (s, z)-Path

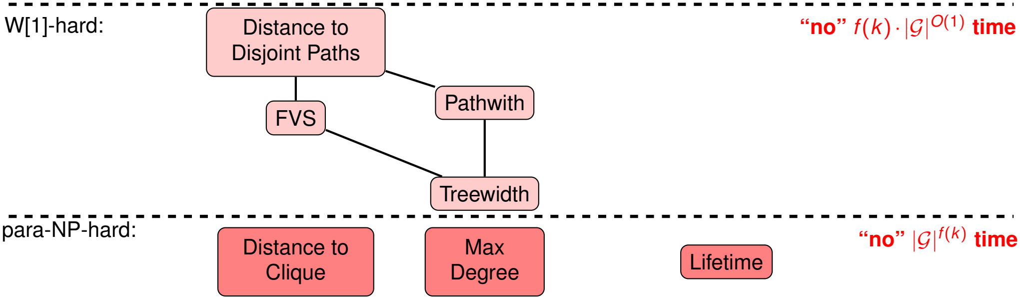
Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?

Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?

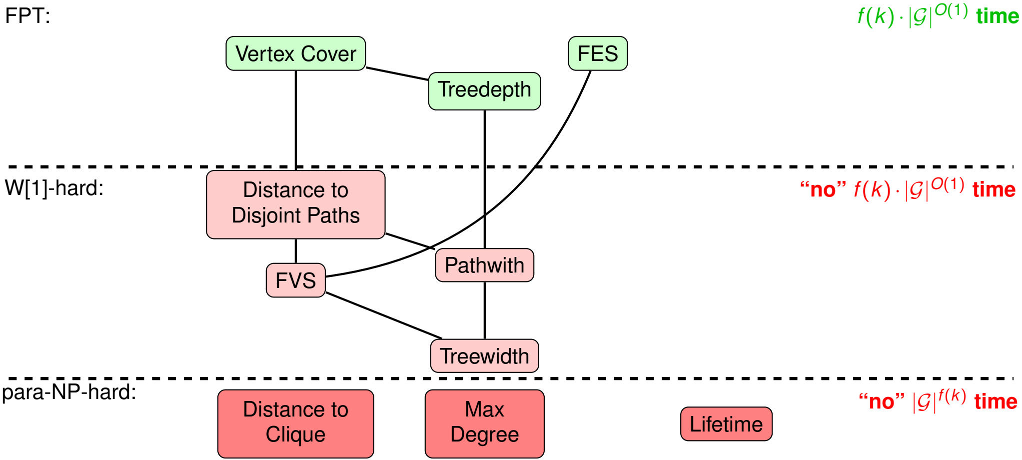
Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?



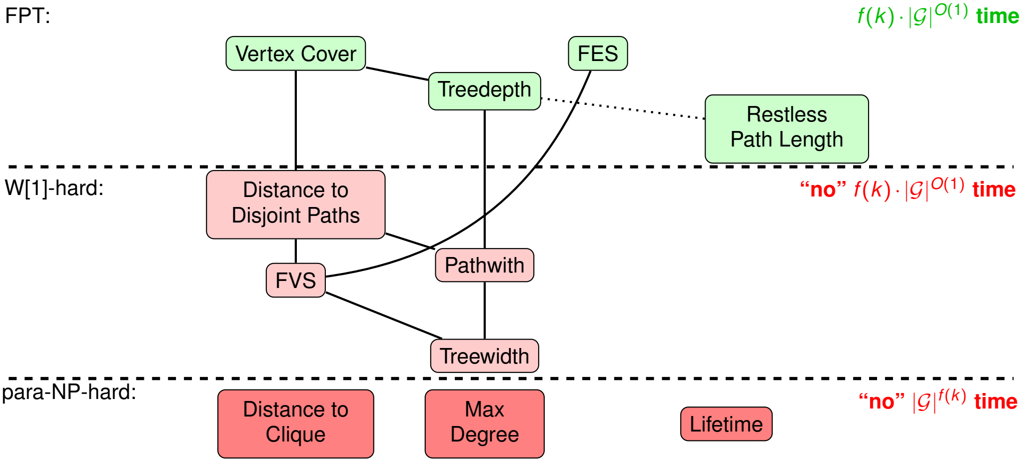
Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?



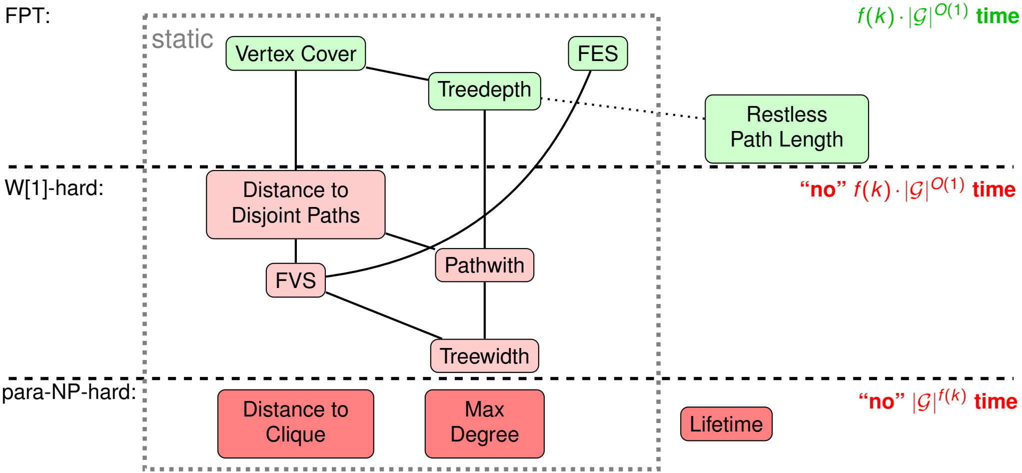
Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?



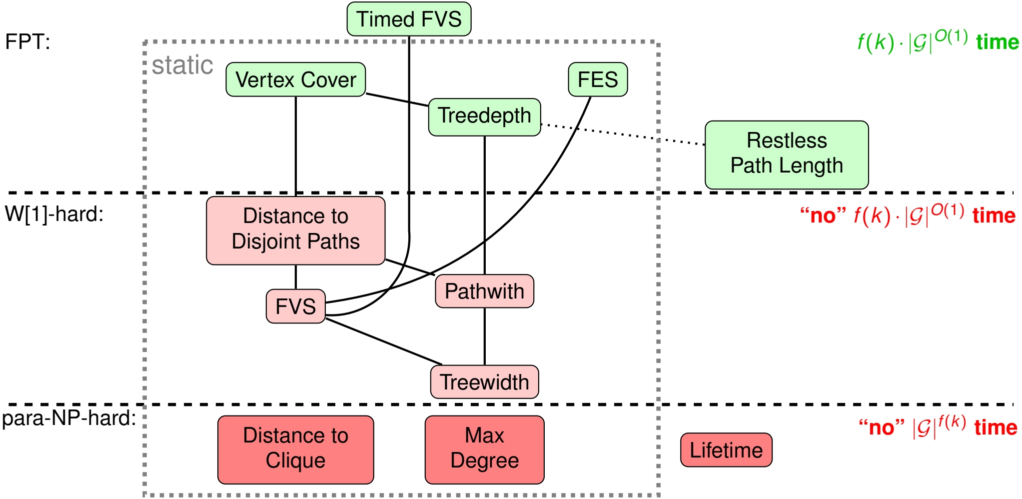
Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i\in[\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?



Restless (s, z)-Path

Input: A temporal graph $\mathcal{G} = (V, (E_i)_{i \in [\tau]})$, two vertices $s, z \in V$, and an integer Δ.
Question: Is there a Δ-restless temporal (s, z)-path in $\mathcal{G}$?



Invitation to Temporal Graph Parameters

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).

For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.

Temporal graph problems: often **intractable**:

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).

For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.

Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.
Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]
- **Temporal Matching** is **NP-hard** on path, [Mertzios et al., STACS '20]

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.
Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]
- **Temporal Matching** is **NP-hard** on path, [Mertzios et al., STACS '20]
- **Temporal Exploration** is **W[1]-hard** for pathwidth, and [Bodlaender et al., IPL '19]

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.
Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]
- **Temporal Matching** is **NP-hard** on path, [Mertzios et al., STACS '20]
- **Temporal Exploration** is **W[1]-hard** for pathwidth, and [Bodlaender et al., IPL '19]
- **Restless Path** is **W[1]-hard** for FVS.

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.
Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]
- **Temporal Matching** is **NP-hard** on path, [Mertzios et al., STACS '20]
- **Temporal Exploration** is **W[1]-hard** for pathwidth, and [Bodlaender et al., IPL '19]
- **Restless Path** is **W[1]-hard** for FVS.

With "$+\tau$": many problems become **FPT**, e.g., **Restless Path** with FVS$+\tau$.

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.
Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]
- **Temporal Matching** is **NP-hard** on path, [Mertzios et al., STACS '20]
- **Temporal Exploration** is **W[1]-hard** for pathwidth, and [Bodlaender et al., IPL '19]
- **Restless Path** is **W[1]-hard** for FVS.

With "$+\tau$": many problems become **FPT**, e.g., **Restless Path** with FVS$+\tau$.

$\rightsquigarrow$ We need new **parameters** to understand this gap.

Invitation to Temporal Graph Parameters

Parameterized Complexity: categorize on which instances a **problem** is **solvable** (or presumably **not**).
For **static graph** problems, literature has **many** parameters to do this: Vertex Cover, FVS, Treewidth, Diameter, etc.
Temporal graph problems: often **intractable**:

- **Temporal Vertex Cover** is **NP-hard** on stars, [Akrida et al., ICALP '18]
- **Temporal Matching** is **NP-hard** on path, [Mertzios et al., STACS '20]
- **Temporal Exploration** is **W[1]-hard** for pathwidth, and [Bodlaender et al., IPL '19]
- **Restless Path** is **W[1]-hard** for FVS.

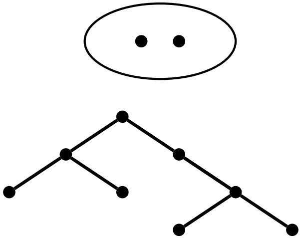
With "$+\tau$": many problems become **FPT**, e.g., **Restless Path** with FVS$+\tau$.

$\rightsquigarrow$ We need new **parameters** to understand this gap.

Today: Lifting **Feedback Vertex Set** for path-related temporal problems.

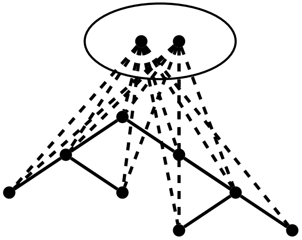
Timed Feedback Vertex Set I

Underlying FVS:
min $|S|$: $G_{\downarrow} - S$ is a forest.



Timed Feedback Vertex Set I

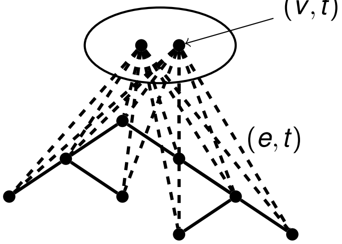
Underlying FVS:
min $|S|$: $G_{\downarrow} - S$ is a forest.



Timed Feedback Vertex Set I

Underlying FVS:
min $|S|$: $G_{\downarrow} - S$ is a forest.

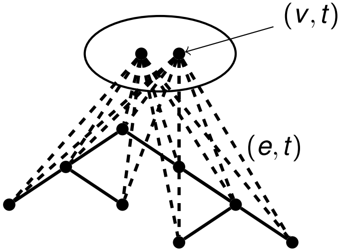
Timed FVS:
Idea: delete vertex appearances



Timed Feedback Vertex Set I

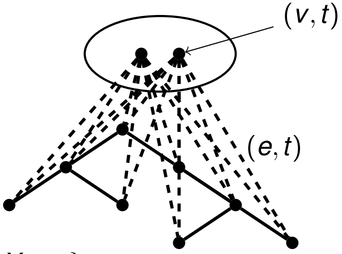
Underlying FVS:
min $|S|$: $G_{\downarrow} - S$ is a forest.

Timed FVS:
Idea: delete vertex appearances
Formal: min. size $X \subseteq V \times [\tau]$ such that $G_{\downarrow}(\mathcal{G} - X) := (V, E')$ is a forest,



Timed Feedback Vertex Set I

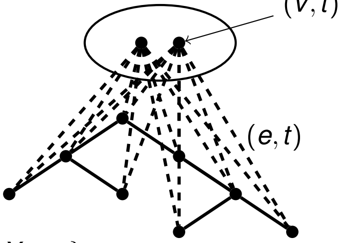
Underlying FVS:
min $|S|$: $G_{\downarrow} - S$ is a forest.



Timed FVS:
Idea: delete vertex appearances
Formal: min. size $X \subseteq V \times [\tau]$ such that $G_{\downarrow}(\mathcal{G} - X) := (V, E')$ is a forest,
where $E' := \{\{v, w\} \mid \exists t \colon \{v, w\} \in E_t \wedge \{(v, t), (w, t)\} \cap X = \emptyset\}$.

Timed Feedback Vertex Set I

Underlying FVS:
min $|S|$: $G_{\downarrow} - S$ is a forest.

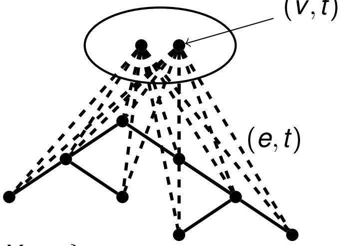


Timed FVS:
Idea: delete vertex appearances
Formal: min. size $X \subseteq V \times [\tau]$ such that $G_{\downarrow}(\mathcal{G} - X) := (V, E')$ is a forest,
where $E' := \{\{v, w\} \mid \exists t \colon \{v, w\} \in E_t \wedge \{(v, t), (w, t)\} \cap X = \emptyset\}$.

Structural properties: **Similar** to **FVS** in static graphs.

Timed Feedback Vertex Set I

Underlying FVS:
min $|S|$: $G_\downarrow - S$ is a forest.



Timed FVS:
Idea: delete vertex appearances
Formal: min. size $X \subseteq V \times [\tau]$ such that $G_\downarrow(\mathcal{G} - X) := (V, E')$ is a forest,
where $E' := \{\{v, w\} \mid \exists t \colon \{v, w\} \in E_t \wedge \{(v, t), (w, t)\} \cap X = \emptyset\}$.

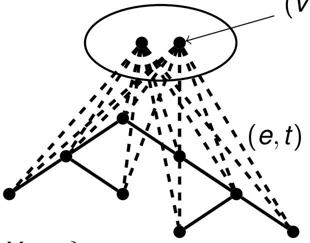
Structural properties: **Similar** to **FVS** in static graphs.

Observation

Underlying FVS $\leq$ Timed FVS $\leq$ Underlying FVS $\cdot \tau$

Timed Feedback Vertex Set I

Underlying FVS:
min $|S|\colon G_{\downarrow} - S$ is a forest.



Timed FVS:
Idea: delete vertex appearances
Formal: min. size $X \subseteq V \times [\tau]$ such that $G_{\downarrow}(\mathcal{G} - X) := (V, E')$ is a forest,
where $E' := \{\{v, w\} \mid \exists t\colon \{v, w\} \in E_t \wedge \{(v, t), (w, t)\} \cap X = \emptyset\}$.

Structural properties: **Similar** to **FVS** in static graphs.

Observation

Underlying FVS $\leq$ Timed FVS $\leq$ Underlying FVS $\cdot \tau$

algorithmically useful: **NP-hard**, but **FPT**. $O^*(4^x)$ time

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.

(2) Drop unused (v, t) from temporal graph.

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.

(2) Drop unused (v, t) from temporal graph.

(3) Guess order of the remainder in X.

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.
(2) Drop unused (v, t) from temporal graph.
(3) Guess order of the remainder in X.
(4) Check for time edges between consecutive $(v, t), (w, t) \in X$.

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.

(2) Drop unused (v, t) from temporal graph.

(3) Guess order of the remainder in X.

(4) Check for time edges between consecutive $(v, t), (w, t) \in X$.

$$(v, t_1) \in O \;\bullet \overset{?}{\text{- - - - - - - - -}} \bullet\; (w, t_2) \in O \qquad \rightsquigarrow \qquad \{v, w\} \in E_{t_1},\ t_2 - \Delta \leq t_1\,?$$

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.
(2) Drop unused (v, t) from temporal graph.
(3) Guess order of the remainder in X.
(4) Check for time edges between consecutive $(v, t), (w, t) \in X$.

$$(v, t_1) \in O \quad \bullet \overset{?}{- - - - - - - -} \bullet \quad (w, t_2) \in O \qquad \rightsquigarrow \qquad \{v, w\} \in E_{t_1},\ t_2 - \Delta \leq t_1?$$

$$(v, t_1) \in I \quad \bullet \overset{?}{- - - - - - - -} \bullet \quad (w, t_2) \in I \qquad \rightsquigarrow \qquad \{v, w\} \in E_{t_2},\ t_2 \leq t_1 + \Delta?$$

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.
(2) Drop unused (v, t) from temporal graph.
(3) Guess order of the remainder in X.
(4) Check for time edges between consecutive $(v, t), (w, t) \in X$.

$(v, t_1) \in O$ ●- - - -?- - - -● $(w, t_2) \in O$ $\rightsquigarrow$ $\{v, w\} \in E_{t_1}$, $t_2 - \Delta \leq t_1$?

$(v, t_1) \in I$ ●- - - -?- - - -● $(w, t_2) \in I$ $\rightsquigarrow$ $\{v, w\} \in E_{t_2}$, $t_2 \leq t_1 + \Delta$?

$(v, t_1) \in O$ ●- - - -?- - - -● $(w, t_2) \in I$ $\rightsquigarrow$ $\{v, w\} \in E_{t_2}$, $t_1 = t_2$?

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v, t) is **I**ncoming, **O**utgoing, or unused.
(2) Drop unused (v, t) from temporal graph.
(3) Guess order of the remainder in X.
(4) Check for time edges between consecutive $(v, t), (w, t) \in X$.

$(v, t_1) \in O$ ●- - - -?- - - -● $(w, t_2) \in O$ $\rightsquigarrow$ $\{v, w\} \in E_{t_1},\ t_2 - \Delta \leq t_1$?

$(v, t_1) \in I$ ●- - - -?- - - -● $(w, t_2) \in I$ $\rightsquigarrow$ $\{v, w\} \in E_{t_2},\ t_2 \leq t_1 + \Delta$?

$(v, t_1) \in O$ ●- - - -?- - - -● $(w, t_2) \in I$ $\rightsquigarrow$ $\{v, w\} \in E_{t_2},\ t_1 = t_2$?

$(v, t_1) \in I$ ●- - - -?- - - -● $(w, t_2) \in O$ $\rightsquigarrow$ $\{v, w\} \in E_{t'}$, $\max\{t_1, t_2 - \Delta\} \leq t' \leq \min\{t_2, t_1 + \Delta\}$?

How to use Timed FVS

Theorem

Restless Path is **FPT** parameterized by Timed FVS.

Alg. Outline: Let X be the Timed FVS.

(1) Guess whether (v,t) is **I**ncoming, **O**utgoing, or unused.

(2) Drop unused (v,t) from temporal graph.

(3) Guess order of the remainder in X.

(4) Check for time edges between consecutive $(v,t),(w,t) \in X$.

(5) Find "unknown parts" with **Multicolored Ind. Set on Chordal Graphs**.

$$(v,t_1) \in O \bullet \overset{?}{- - - - - - - -} \bullet (w,t_2) \in O \quad \rightsquigarrow \quad \{v,w\} \in E_{t_1},\ t_2 - \Delta \le t_1?$$

$$(v,t_1) \in I \bullet \overset{?}{- - - - - - - -} \bullet (w,t_2) \in I \quad \rightsquigarrow \quad \{v,w\} \in E_{t_2},\ t_2 \le t_1 + \Delta?$$

$$(v,t_1) \in O \bullet \overset{?}{- - - - - - - -} \bullet (w,t_2) \in I \quad \rightsquigarrow \quad \{v,w\} \in E_{t_2},\ t_1 = t_2?$$

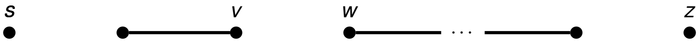
$$(v,t_1) \in I \bullet \overset{?}{- - - - - - - -} \bullet (w,t_2) \in O \quad \rightsquigarrow \quad \{v,w\} \in E_{t'},$$

Resolve Unknown Parts

First, if (v,t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.

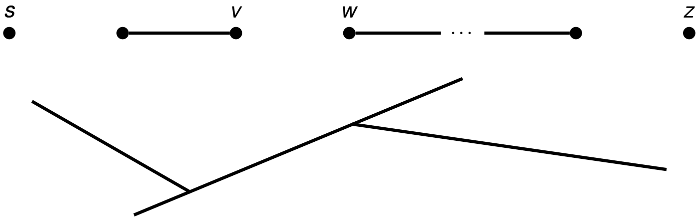
Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.



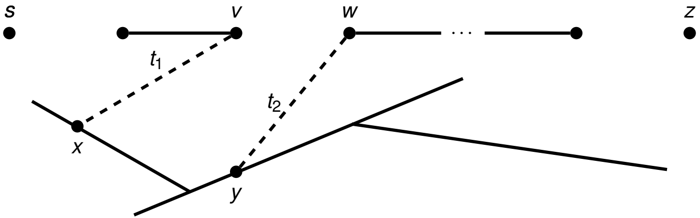
Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.



Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.

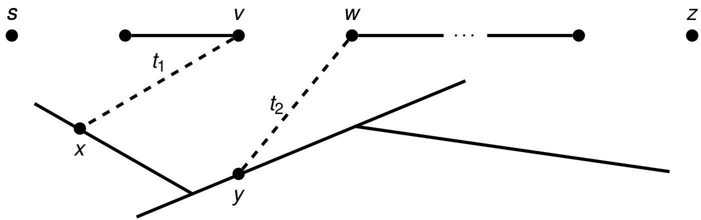


For time edge $e_1 = (\{v, x\}, t_1)$ and $e_2 = (\{w, y\}, t_2)$:

- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?

Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.

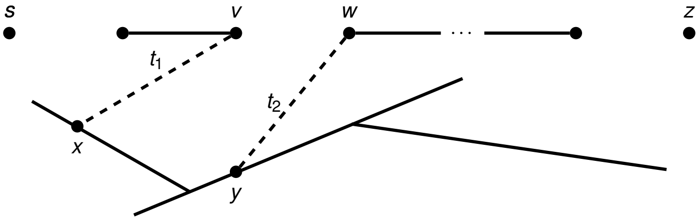


For time edge $e_1 = (\{v, x\}, t_1)$ and $e_2 = (\{w, y\}, t_2)$:

- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?
- Is there a path P from x to y in $T_\downarrow$?

Resolve Unknown Parts

First, if (v,t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G}-X)$.

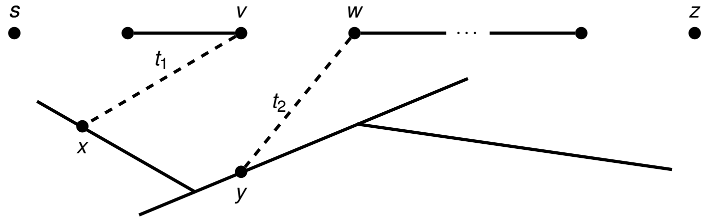


For time edge $e_1 = (\{v,x\},t_1)$ and $e_2 = (\{w,y\},t_2)$:

- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?
- Is there a path P from x to y in $T_\downarrow$?
- Is there a Δ-restless path v to w in $\mathcal{G}[P]$ plus e_1, e_2?

Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.



For time edge $e_1 = (\{v, x\}, t_1)$ and $e_2 = (\{w, y\}, t_2)$:

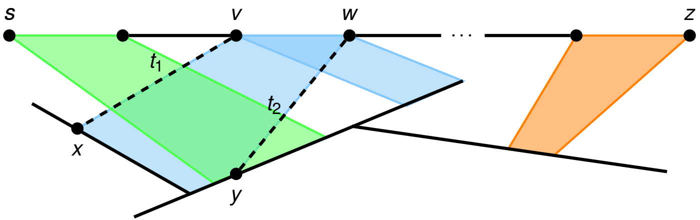
- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?
- Is there a path P from x to y in $T_\downarrow$?
- Is there a Δ-restless path v to w in $\mathcal{G}[P]$ plus e_1, e_2?

Lemma

Restless Path linear time solvable on path.

Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.



For time edge $e_1 = (\{v, x\}, t_1)$ and $e_2 = (\{w, y\}, t_2)$:

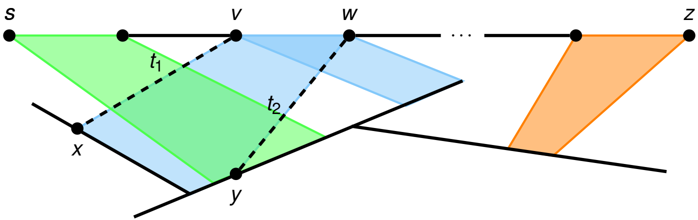
- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?
- Is there a path P from x to y in $T_\downarrow$?
- Is there a Δ-restless path v to w in $\mathcal{G}[P]$ plus e_1, e_2?

Lemma

Restless Path linear time solvable on path.

Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.



For time edge $e_1 = (\{v, x\}, t_1)$ and $e_2 = (\{w, y\}, t_2)$:

- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?
- Is there a path P from x to y in $T_\downarrow$?
- Is there a Δ-restless path v to w in $\mathcal{G}[P]$ plus e_1, e_2?

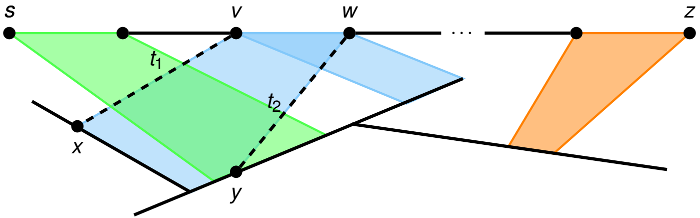
Intersection graphs of paths in trees are **chordal**.

Lemma

Restless Path linear time solvable on path.

Resolve Unknown Parts

First, if (v, t) is still in X remove v from $T_\downarrow := G_\downarrow(\mathcal{G} - X)$.



For time edge $e_1 = (\{v, x\}, t_1)$ and $e_2 = (\{w, y\}, t_2)$:

- Is departure time t_1 and arrival time t_2 **"valid"** (4 cases again)?
- Is there a path P from x to y in $T_\downarrow$?
- Is there a Δ-restless path v to w in $\mathcal{G}[P]$ plus e_1, e_2?

Lemma

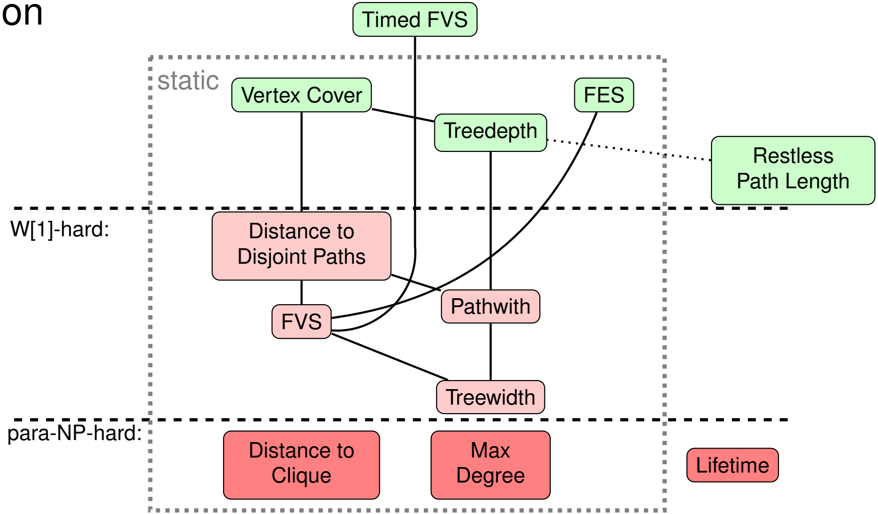
Restless Path linear time solvable on path.

Intersection graphs of paths in trees are **chordal**.

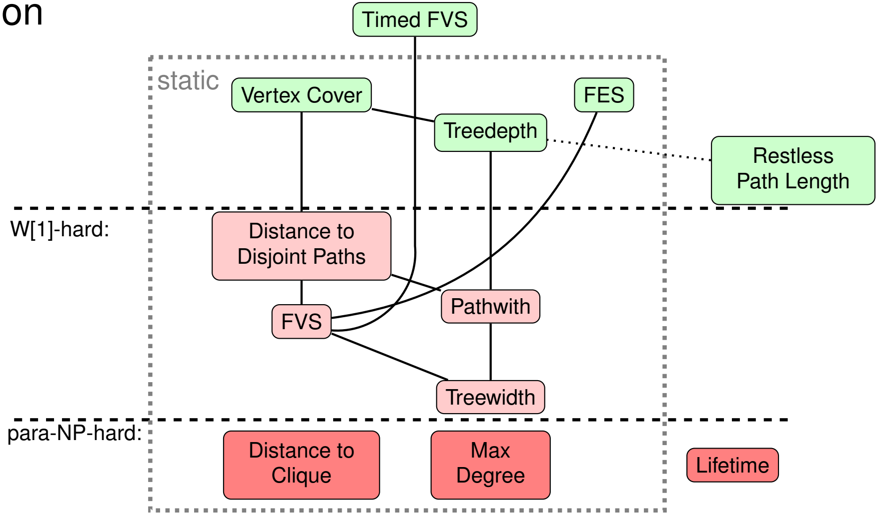
Lemma (Bentert et al., J. Scheduling 19)

Multicolored Ind. Set on Chordal Graphs is **FPT** with # of colors.

Conclusion

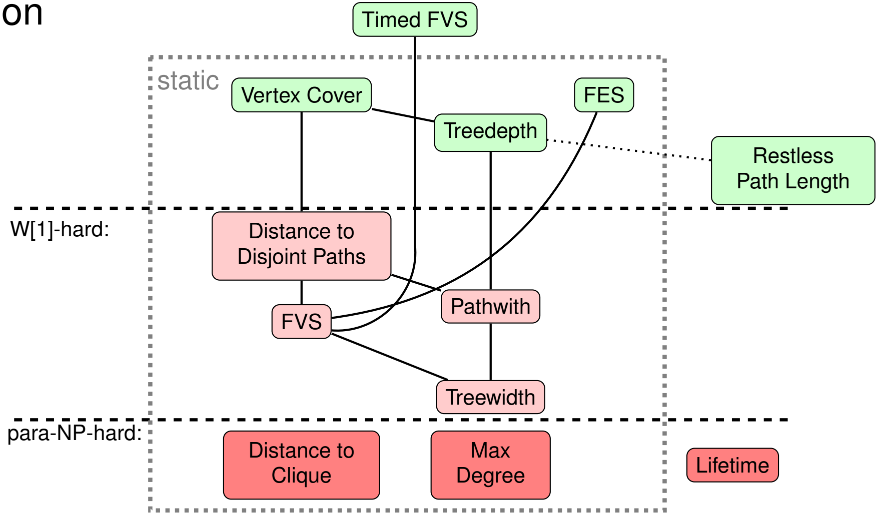

Timed FVS
static
Vertex Cover
FES
Treedepth
Restless
Path Length
W[1]-hard:
Distance to
Disjoint Paths
Pathwith
FVS
Treewidth
para-NP-hard:
Distance to
Clique
Max
Degree
Lifetime

Conclusion



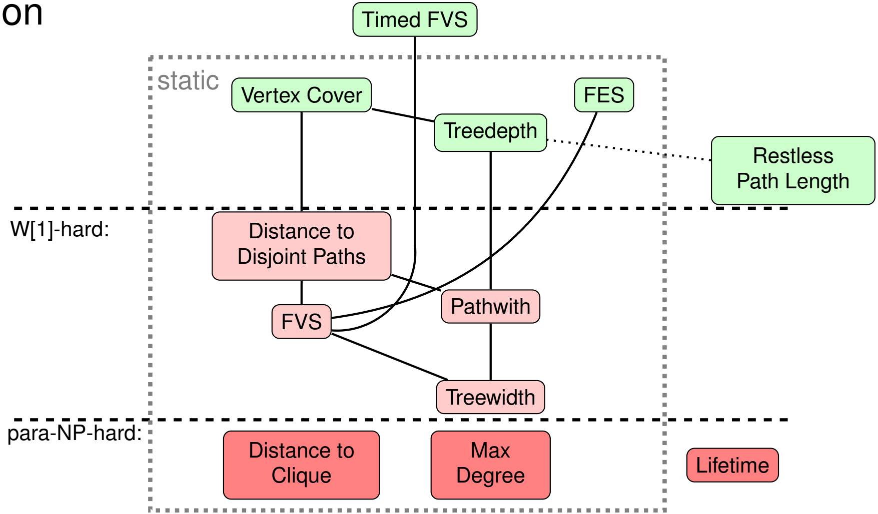
Future: temporal parameters!

Conclusion



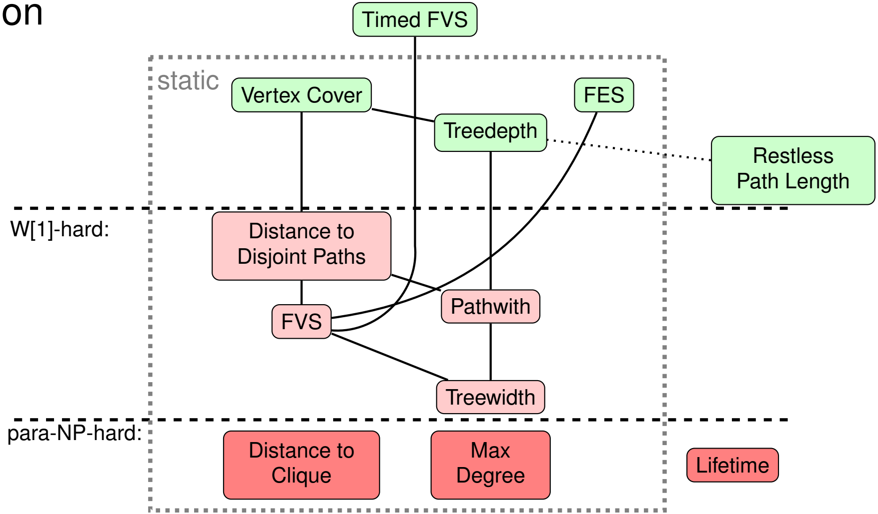
Future: temporal parameters!
How to design?

Conclusion



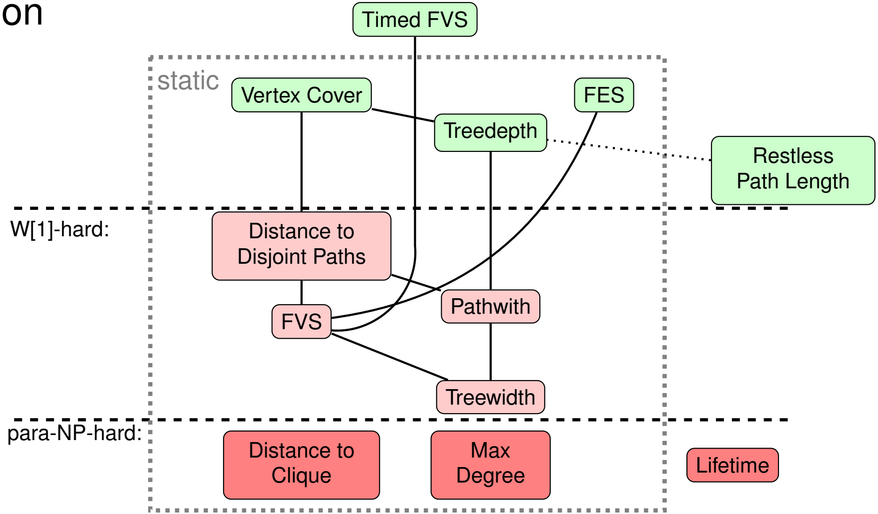
Future: temporal parameters!
How to design? How to compute?

Conclusion



Future: temporal parameters!
How to design? How to compute? Experiments?

Conclusion



Future: temporal parameters!
How to design? How to compute? Experiments?

Thank you!