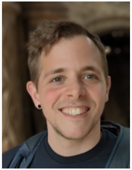
On inefficient temporal graphs

Esteban Christiann estebanc@protonmail.ch
Eric Sanlaville[1] eric.sanlaville@univ-lehavre.fr
Jason Schoeters[1 2] js2807@cam.ac.uk

Algorithmic Aspects of Temporal Graphs VI, Paderborn, July 10, 2023

[1] Supported by the RIN Tremplin Région Normandie project DynNet.
[2] Supported by the Leverhulme Trust International Professorship in Neuroeconomics.

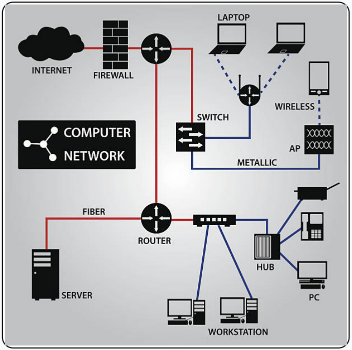
Context

Temporal graphs

Context

Temporal graphs

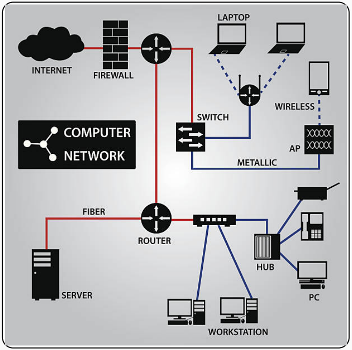
- faulty networks



Context

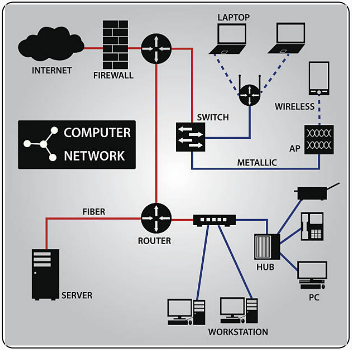
Temporal graphs

- faulty networks
- swarms of drones



Temporal graphs

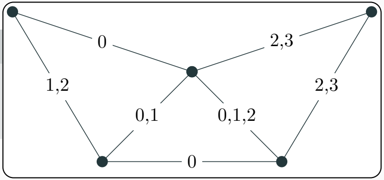
- faulty networks
- swarms of drones
- interacting population



Preliminaries

Definition temporal graph

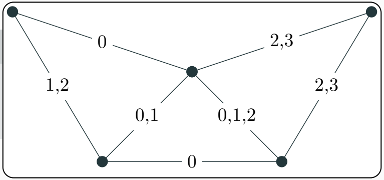
Edges appear and disappear
over the lifetime of the graph

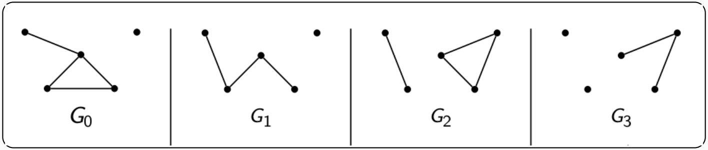


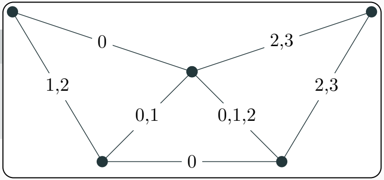
Preliminaries

Definition temporal graph

Edges appear and disappear over the lifetime of the graph

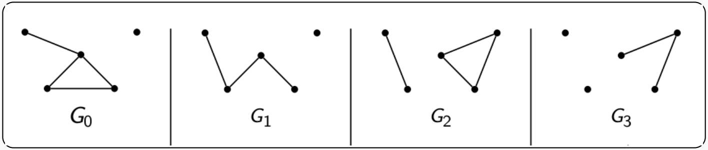






Definition temporal graph

Edges appear and disappear over the lifetime of the graph

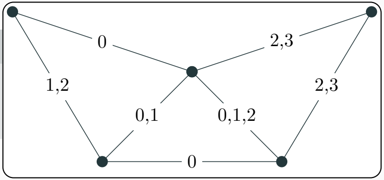


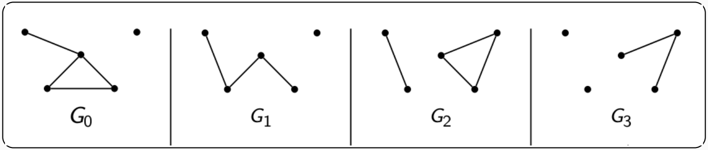
Definition footprint (or underlying graph)

Union of all snapshots G_i

Definition temporal graph

Edges appear and disappear over the lifetime of the graph





Definition footprint (or underlying graph)

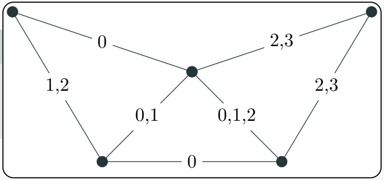
Union of all snapshots G_i

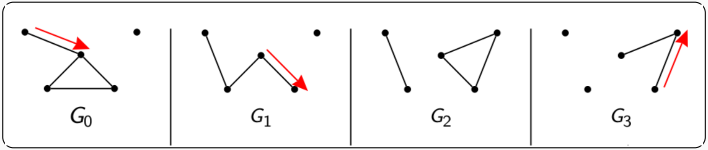
Definition journey (or temporal path)

Path in temporal graph with increasing labels on the edges

Definition temporal graph

Edges appear and disappear over the lifetime of the graph





Definition footprint (or underlying graph)

Union of all snapshots G_i

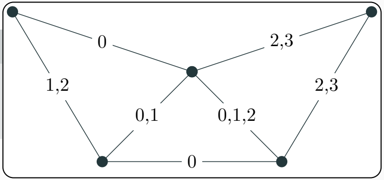
Definition journey (or temporal path)

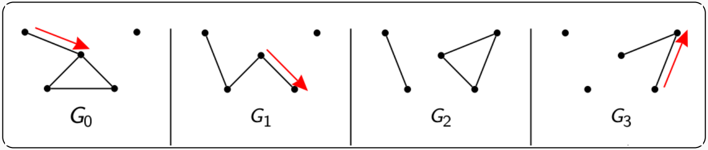
Path in temporal graph with increasing labels on the edges

Preliminaries

Definition temporal graph

Edges appear and disappear
over the lifetime of the graph





Definition footprint (or underlying graph)

Union of all snapshots G_i

Definition journey (or temporal path)

Path in temporal graph with increasing labels on the edges

Definition temporal connectivity ($\mathcal{TC}$)

A temporal graph is temporally connected
iff for all pairs of vertices (u, v) there exists a journey $u \rightsquigarrow v$

Preliminaries

Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$

Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Preliminaries

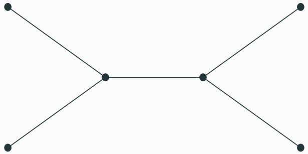
Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$

Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n - 2$?



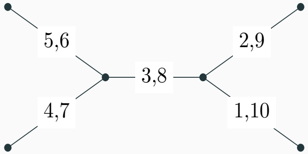
Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$

Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n - 2$?



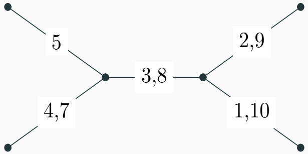
Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$

Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n - 2$? $2n - 3$?



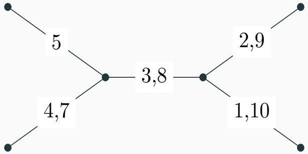
Preliminaries

Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$
Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n - 2$? $2n - 3$? $2n - 4$?



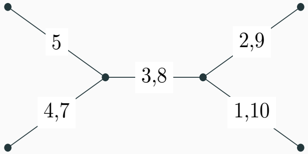
Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$

Question: does a labelling $\lambda : E \rightarrow \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n - 2$? $2n - 3$? $2n - 4$? $2n - 5$? [2]



Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$
Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n - 2$? $2n - 3$? $2n - 4$? $2n - 5$? [2]
  - **and** with at most some lifetime L? [6]

Preliminaries

Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$
Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n-2$? $2n-3$? $2n-4$? $2n-5$? [2]
  - **and** with at most some lifetime L? [6]
- Reachability of at most some R? [4]

Preliminaries

Temporal graph design problem

Input: graph $G = (V, E)$, temporal graph property $\mathcal{P}$
Question: does a labelling $\lambda : E \to \mathbb{N}$ exist such that temporal graph $\mathcal{G} = (G, L)$ have property $\mathcal{P}$?

Examples of property $\mathcal{P}$

- $\mathcal{TC}$ with amount of labels $\leq 2n-2$? $2n-3$? $2n-4$? $2n-5$? [2]
  - **and** with at most some lifetime L? [6]
- Reachability of at most some R? [4]

Remark

The amount of labels used in these problems is always minimized or given

Sparse labelling problem

Input: graph G, integer k

Question: Does a labelling λ using at most k labels exist such that temporal graph $\mathcal{G} = (G, \lambda)$ is $\mathcal{TC}$?

Problem formulation

Dense labelling problem

Input: graph G, integer k

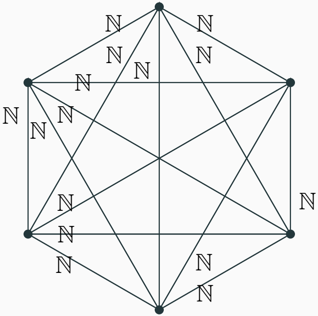
Question: Does a labelling λ using at **least** k labels exist such that temporal graph $\mathcal{G} = (G, \lambda)$ is $\mathcal{TC}$?

Problem formulation

Dense labelling problem

Input: graph G, integer k

Question: Does a labelling λ using at **least** k labels exist such that temporal graph $\mathcal{G} = (G, \lambda)$ is $\mathcal{TC}$?



Problem formulation

Dense labelling problem

Input: graph G, integer k

Question: Does a labelling λ using at **least** k labels exist such that temporal graph $\mathcal{G} = (G, \lambda)$ is $\mathcal{TC}$?

Definitions

- A label is **necessary** iff reachability reduces when removed
- A labelling is **necessary** iff it consists of only necessary labels
- A labelling is **proper** iff incident edges have distinct labels

Problem formulation

Dense labelling problem

Input: graph G, integer k

Question: Does a **proper necessary** labelling λ using at **least** k labels exist such that temporal graph $\mathcal{G} = (G, \lambda)$ is $\mathcal{TC}$?

Definitions

- A label is **necessary** iff reachability reduces when removed
- A labelling is **necessary** iff it consists of only necessary labels
- A labelling is **proper** iff incident edges have distinct labels

Problem formulation

Dense labelling problem

Input: graph G, integer k

Question: Does a **proper necessary** labelling λ using at **least** k labels exist such that temporal graph $\mathcal{G} = (G, \lambda)$ is $\mathcal{TC}$?

Definitions

- A label is **necessary** iff reachability reduces when removed
- A labelling is **necessary** iff it consists of only necessary labels
- A labelling is **proper** iff incident edges have distinct labels

Motivation

- Represents potential power of an adversary trying to waste precious network resources (even if restricted)
- Worst case scenarios for temporal spanners and greedy algorithms

Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$

Global and local measures

Measures

of a temporal graph $\mathcal{G}$ [6]:

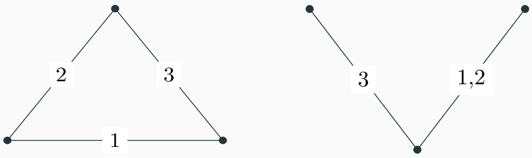
- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$

Global and local measures

Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$



Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$

of a (static) graph G:

Global and local measures

Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$

of a (static) graph G:

- **Maximum temporal cost** T^+: maximum temporal cost of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ

Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$

of a (static) graph G:

- **Maximum temporal cost** T^+: maximum temporal cost of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ
- **Maximum temporality** τ^+: maximum temporality of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ

Global and local measures

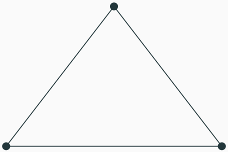
Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$

of a (static) graph G:

- **Maximum temporal cost** T^+: maximum temporal cost of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ
- **Maximum temporality** τ^+: maximum temporality of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ



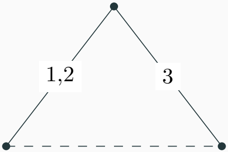
Measures

of a temporal graph $\mathcal{G}$ [6]:

- **Temporal cost** T: total amount of labels in $\mathcal{G}$
- **Temporality** τ: maximum amount of labels on some edge of $\mathcal{G}$

of a (static) graph G:

- **Maximum temporal cost** T^+: maximum temporal cost of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ
- **Maximum temporality** τ^+: maximum temporality of temporally connected $\mathcal{G} = (G, \lambda)$ among all necessary labellings λ



Bounds

Bounds on largest τ^+ and T^+

- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$

Bounds

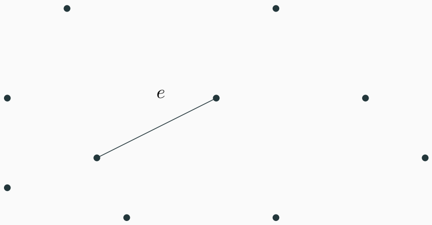
Bounds on largest τ^+ and T^+

- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$

Bounds

Bounds on largest τ^+ and T^+

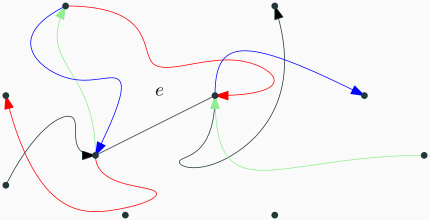
- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$



Bounds

Bounds on largest τ^+ and T^+

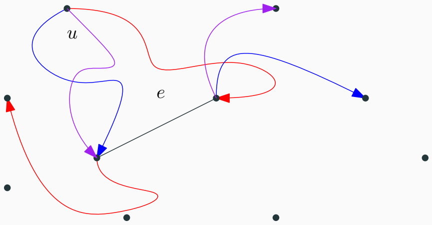
- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$



Bounds

Bounds on largest τ^+ and T^+

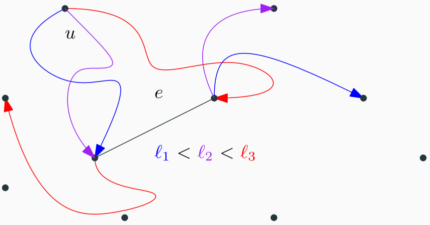
- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$



Bounds

Bounds on largest τ^+ and T^+

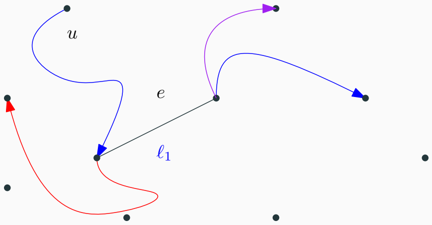
- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$



Bounds

Bounds on largest τ^+ and T^+

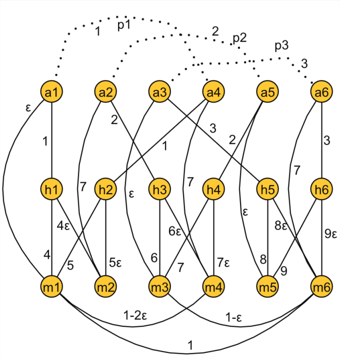
- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$



Bounds

Bounds on largest τ^+ and T^+

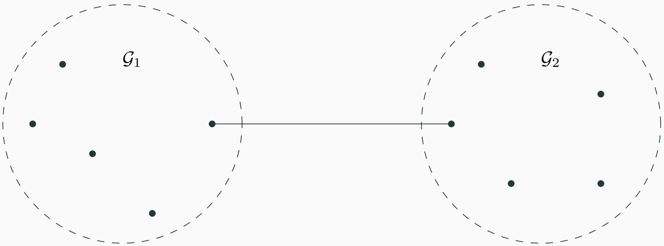
- 2 ? $\leq \tau^+$ and $\tau^+ \leq n-1$
- $\frac{1}{18}n^2 + O(n) \leq T^+$ [1] and $T^+ < (n-1)\binom{n}{2}$



The simple case of trees

Definition

A bridge edge is an edge which disconnects the graph when removed



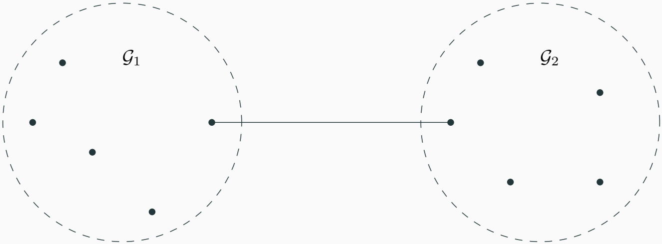
The simple case of trees

Definition

A bridge edge is an edge which disconnects the graph when removed

Lemma

In any necessary labelling, there are at most 2 labels on a bridge edge

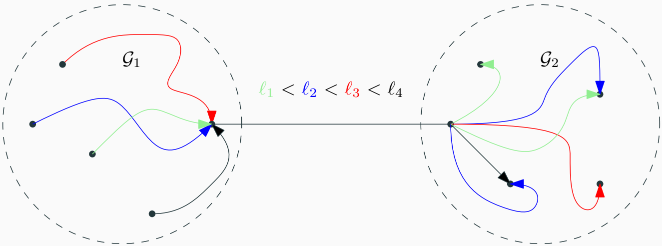


Definition

A bridge edge is an edge which disconnects the graph when removed

Lemma

In any necessary labelling, there are at most 2 labels on a bridge edge



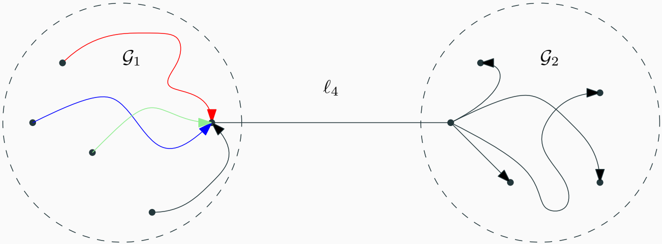
The simple case of trees

Definition

A bridge edge is an edge which disconnects the graph when removed

Lemma

In any necessary labelling, there are at most 2 labels on a bridge edge



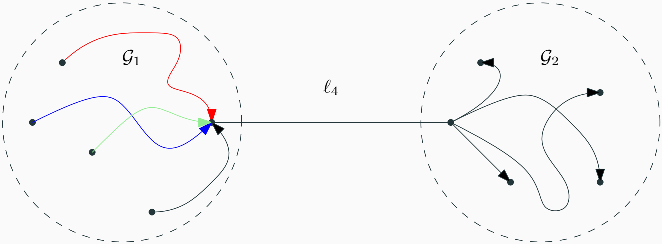
The simple case of trees

Definition

A bridge edge is an edge which disconnects the graph when removed

Lemma

In any necessary labelling, there are at most 2 labels on a bridge edge



Theorem

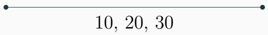
For any tree graph on n vertices, $\tau^+ = 2$ and $T^+ = 2n - 3$

Is $\tau^+ > 2$ even possible?

Is $\tau^+ > 2$ even possible?

Ad-hoc construction for $\tau^+ \geq 3$

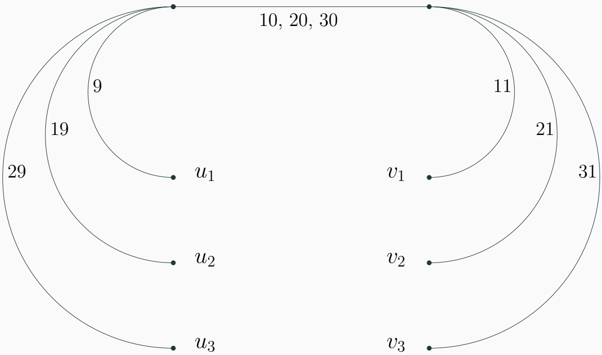
- suppose some edge e exists with 3 necessary labels;



Is $\tau^+ > 2$ even possible?

Ad-hoc construction for $\tau^+ \geq 3$

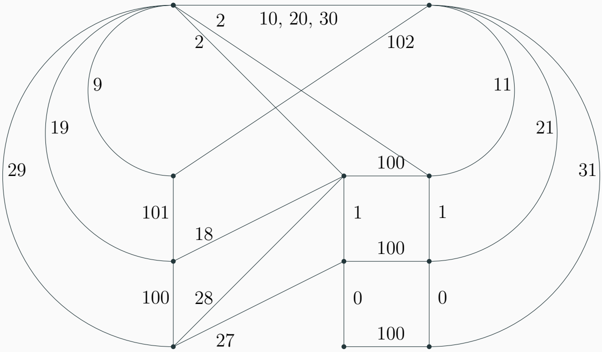
- suppose some edge e exists with 3 necessary labels;
- so those labels are necessary for some (distinct) journeys;



Is $\tau^+ > 2$ even possible?

Ad-hoc construction for $\tau^+ \geq 3$

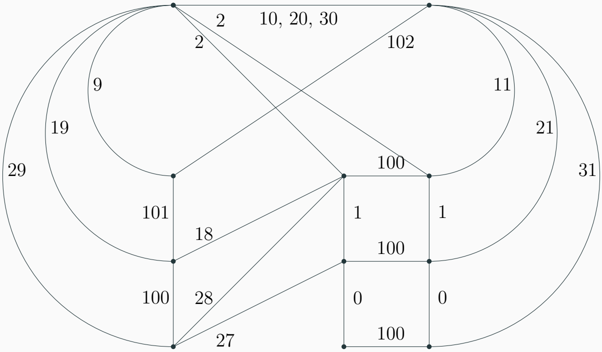
- suppose some edge e exists with 3 necessary labels;
- so those labels are necessary for some (distinct) journeys;
- add edges/labels to make sure graph is $\mathcal{TC}$



Is $\tau^+ > 2$ even possible?

Ad-hoc construction for $\tau^+ \geq 3$

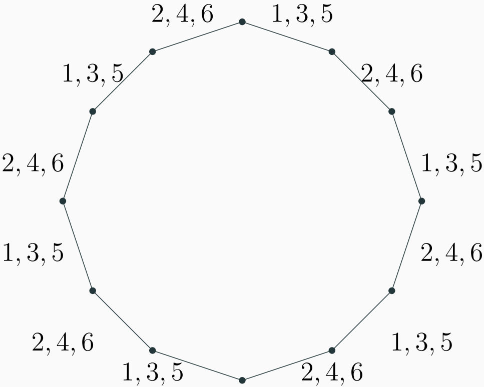
- suppose some edge e exists with 3 necessary labels;
- so those labels are necessary for some (distinct) journeys;
- add edges/labels to make sure graph is $\mathcal{TC}$



Theorem

The ad-hoc construction allows for $\tau^+ \geq \frac{1}{3}n$ and $T^+ \geq \frac{1}{18}n^2 + O(n)$

Odd/even alternating labelling



Theorem

The odd/even alternating labelling allows for $\tau^+ = \frac{1}{4}n$ and $T^+ = \frac{1}{4}n^2$

Remark

Contrary to trees, cycles allow for large τ^+ and T^+, but can we do better?

Necessary labelling generator

Idea

Extend STGen [3], a happy labelling generator using techniques based on matchings, isomorphisms, and automorphisms, to generate, given a graph, all its proper necessary labellings

Necessary labelling generator

Idea

Extend STGen [3], a happy labelling generator using techniques based on matchings, isomorphisms, and automorphisms, to generate, given a graph, all its proper necessary labellings

Some details

- Non-necessary labellings and $\mathcal{TC}$ labellings are "final", efficiently cutting branches of search space;
- Amortized constant time complexity for $\mathcal{TC}$ testing and necessity testing;
- Freely available at https://gitlab.com/echrstnn/max-temporality (Rust);

Necessary labelling generator

Idea

Extend STGen [3], a happy labelling generator using techniques based on matchings, isomorphisms, and automorphisms, to generate, given a graph, all its proper necessary labellings

Some details

- Non-necessary labellings and $\mathcal{TC}$ labellings are "final", efficiently cutting branches of search space;
- Amortized constant time complexity for $\mathcal{TC}$ testing and necessity testing;
- Freely available at https://gitlab.com/echrstnn/max-temporality (Rust);

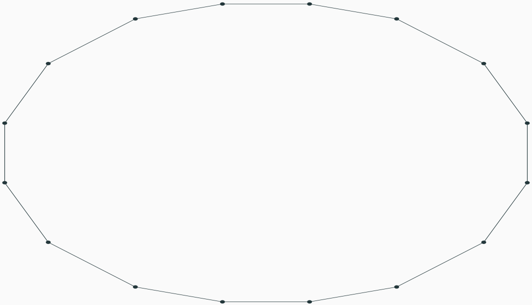
Result

Enumerated all necessary labellings of cycles of size up to $n = 14$ included, giving us the intuition for the following (empirically optimal) labelling

Odd/even distributing labelling

Definition Odd/even distributing labelling

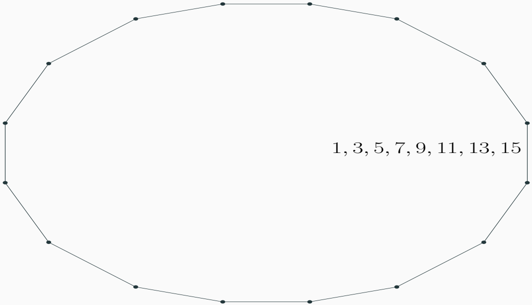
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

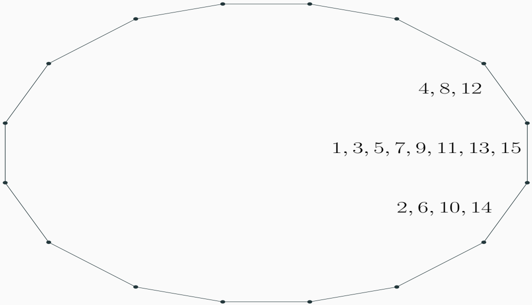
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

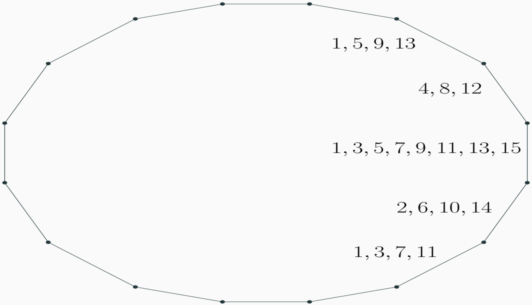
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

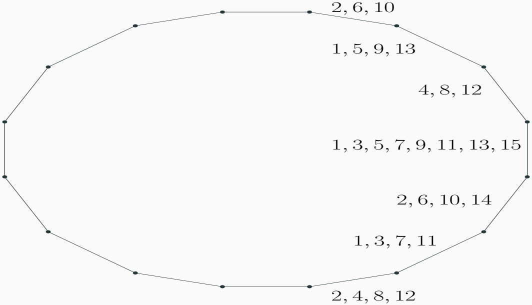
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

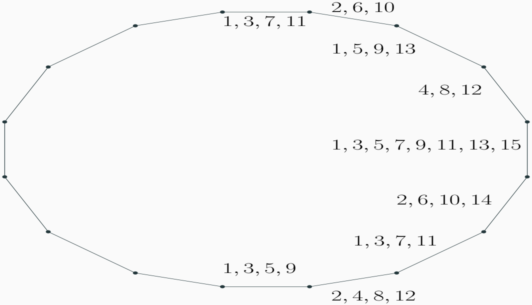
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

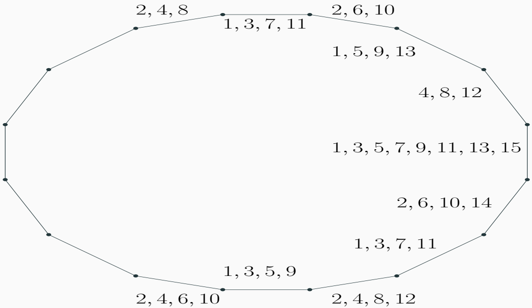
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

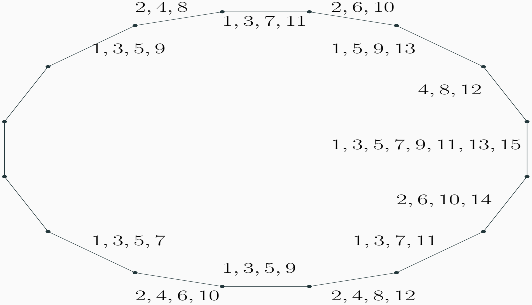
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
    - Assign P_1 to both edges e_ℓ and e_r;
    - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
    - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
    - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

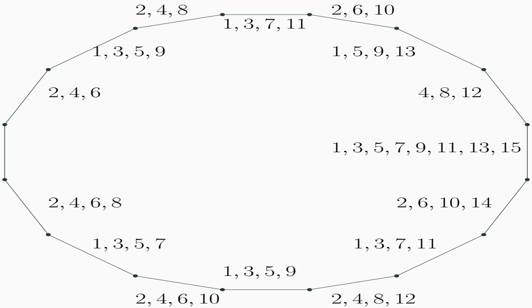
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

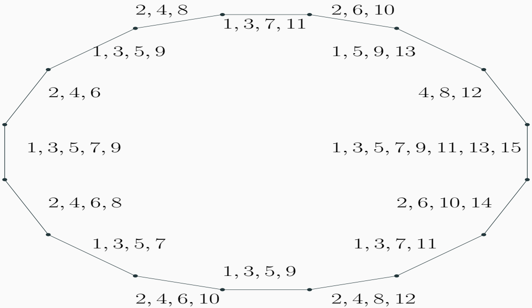
- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
    - Assign P_1 to both edges e_ℓ and e_r;
    - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
    - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
    - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Odd/even distributing labelling

Definition Odd/even distributing labelling

- Let $L_1 = (1, 3, 5, ..., n-1)$, $L_2 = (2, 4, ..., n-2)$ and $P_1, P_2 = \emptyset$;
- Choose some edge e, and let $e_\ell, e_r = e$;
- Do $\frac{n}{2}$ times:
  - Assign P_1 to both edges e_ℓ and e_r;
  - Distribute L_1 to e_ℓ and e_r, move $\min(L_1)$ to P_1 and remove $\max(L_1)$;
  - Set e_ℓ and e_r to the next clockwise and counter-clockwise edges;
  - Swap L_1 and L_2, as well as P_1 and P_2;
- Assign P_1 to e_ℓ, as well as label $\max(P_1) + 2$



Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling induces $\tau^+ \geq \frac{1}{2}n$ and $T^+ \geq \frac{1}{4}n^2 + \mathbf{1}$

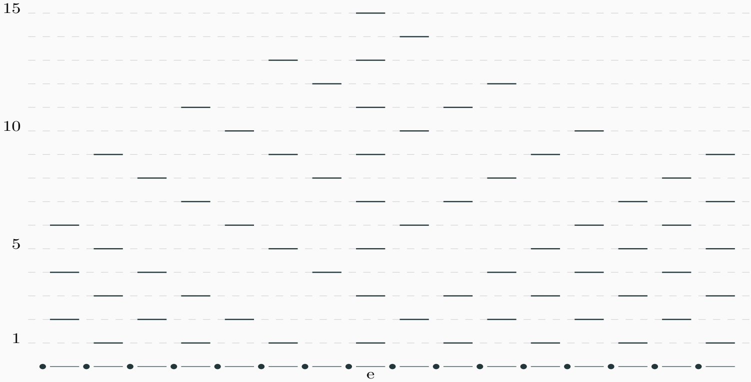
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling induces $\tau^+ \geq \frac{1}{2}n$ and $T^+ \geq \frac{1}{4}n^2 + \mathbf{1}$

Link Stream (Latapy *et al.* [5])

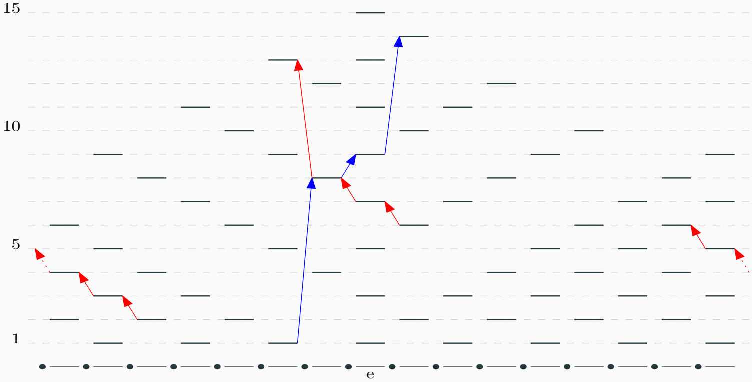
Use one axis for the vertices, and a second axis for time, allowing one to plot the time-edges (and journeys) in this two-dimensional space



Definition

A journey is:

- (counter-)clockwise iff it only uses edges towards the right (*resp.* left);

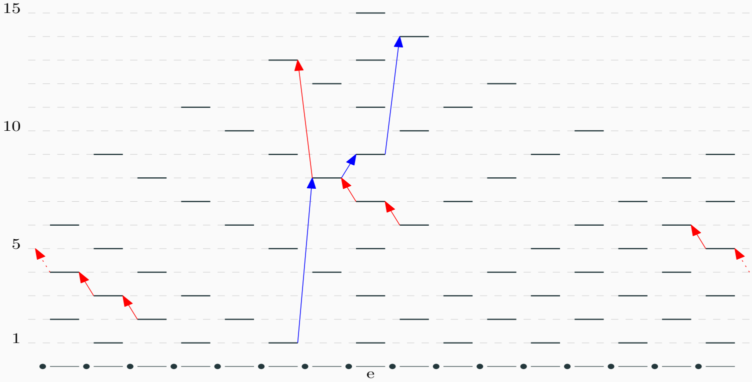


Correctness proof of odd/even distributing labelling

Definition

A journey is:

- (counter-)clockwise iff it only uses edges towards the right (*resp.* left);
- prefix-foremost iff it always uses the earliest edges possible;

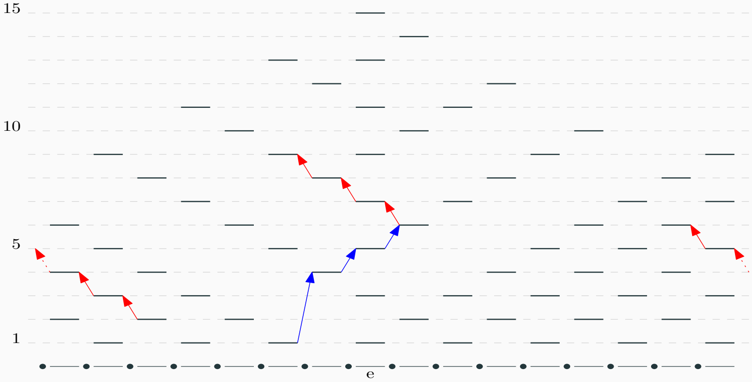


Correctness proof of odd/even distributing labelling

Definition

A journey is:

- (counter-)clockwise iff it only uses edges towards the right (*resp.* left);
- prefix-foremost iff it always uses the earliest edges possible;

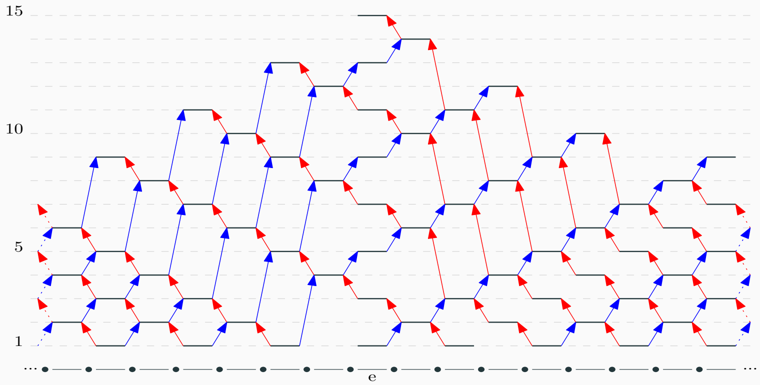


Correctness proof of odd/even distributing labelling

Definition

A journey is:

- (counter-)clockwise iff it only uses edges towards the right (*resp.* left);
- prefix-foremost iff it always uses the earliest edges possible;
- dominant iff no (same direction) journey covers its vertices or more

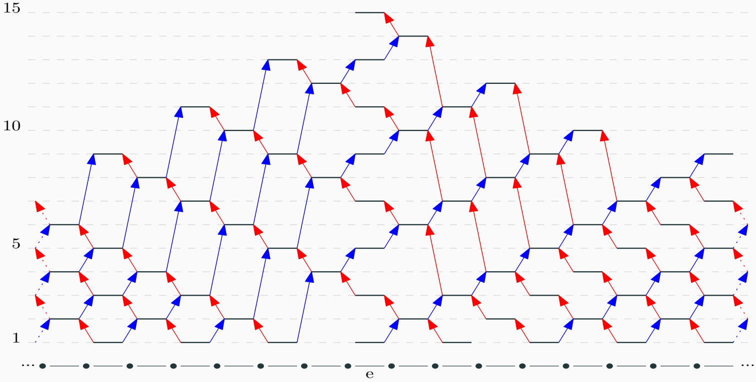


Correctness proof of odd/even distributing labelling

Lemma

A pair of clockwise and counter-clockwise journeys is necessary if:

- both start at some same vertex v;
- both are prefix-foremost;
- both are a suffix of a dominant journey;
- together they cover the whole vertex set without crossing



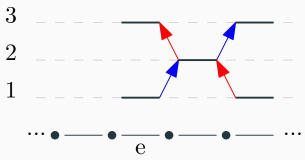
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling is necessary for all even cycles

Proof idea by induction

- base case: prove it is a necessary labelling for C_4 with Lemma
- inductive step (C_n to C_{n+2}): adds a “layer” on the link stream, lengthening journeys by 1, and adding journeys obeying Lemma $\square$



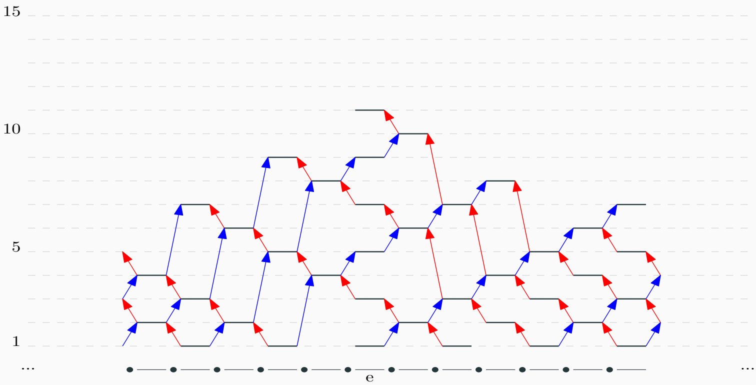
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling is necessary for all even cycles

Proof idea by induction

- base case: prove it is a necessary labelling for C_4 with Lemma
- inductive step (C_n to C_{n+2}): adds a "layer" on the link stream, lengthening journeys by 1, and adding journeys obeying Lemma □



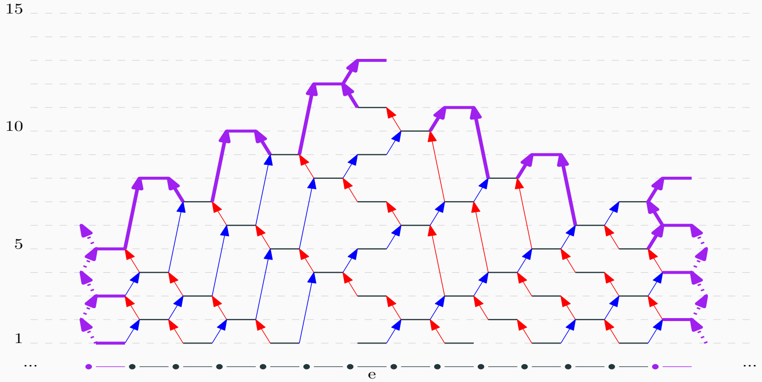
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling is necessary for all even cycles

Proof idea by induction

- base case: prove it is a necessary labelling for C_4 with Lemma
- inductive step (C_n to C_{n+2}): adds a “layer” on the link stream, lengthening journeys by 1, and adding journeys obeying Lemma □



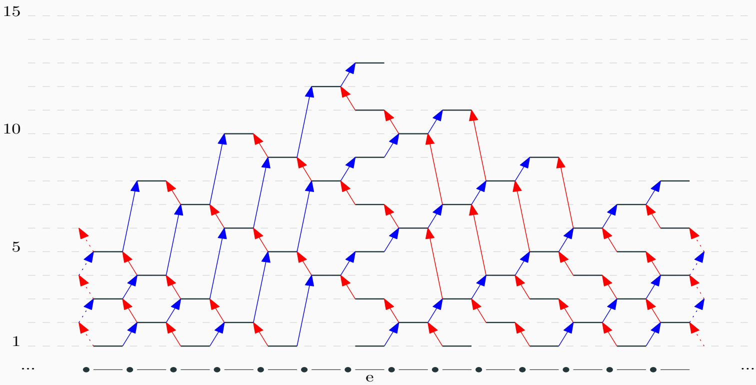
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling is necessary for all even cycles

Proof idea by induction

- base case: prove it is a necessary labelling for C_4 with Lemma
- inductive step (C_n to C_{n+2}): adds a “layer” on the link stream, lengthening journeys by 1, and adding journeys obeying Lemma □



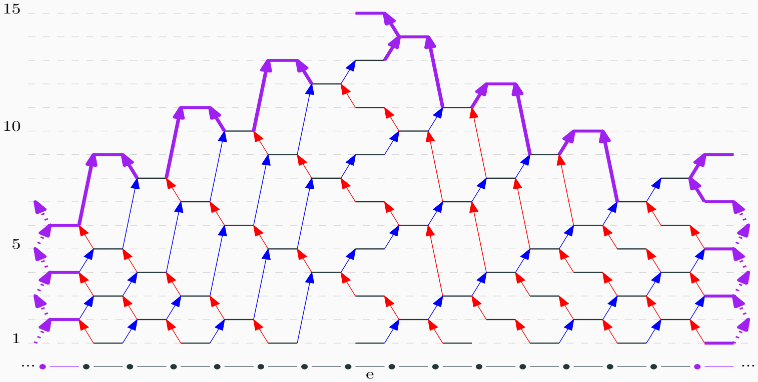
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling is necessary for all even cycles

Proof idea by induction

- base case: prove it is a necessary labelling for C_4 with Lemma
- inductive step (C_n to C_{n+2}): adds a “layer” on the link stream, lengthening journeys by 1, and adding journeys obeying Lemma □



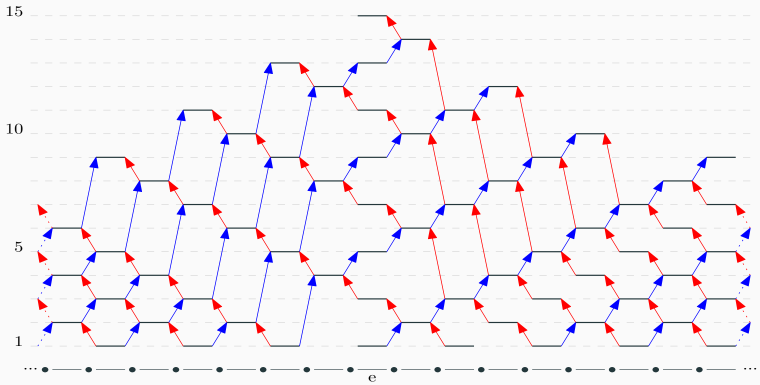
Correctness proof of odd/even distributing labelling

Theorem

The odd/even distributing labelling is necessary for all even cycles

Proof idea by induction

- base case: prove it is a necessary labelling for C_4 with Lemma
- inductive step (C_n to C_{n+2}): adds a “layer” on the link stream, lengthening journeys by 1, and adding journeys obeying Lemma □



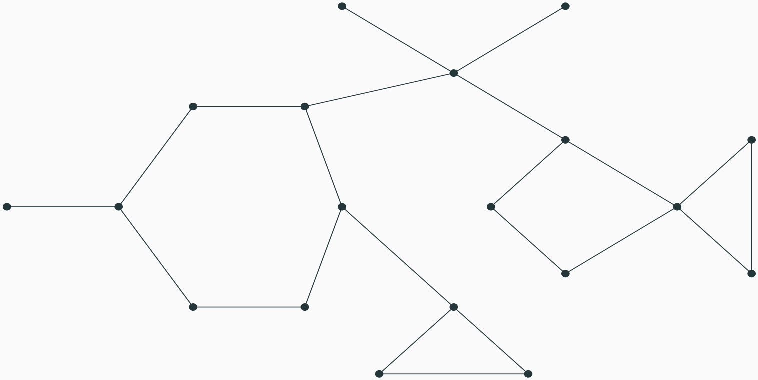
And finally: cactus graphs

Definition

A cactus graph is a tree graph with some vertices/edges replaced by cycles

Theorem

For any cactus graph on n vertices and circumference c (length of longest simple cycle), $\tau^+ \geq \frac{1}{2}c$, and $T^+ \geq \frac{1}{4}c^2 + 2(n-c) + 1$



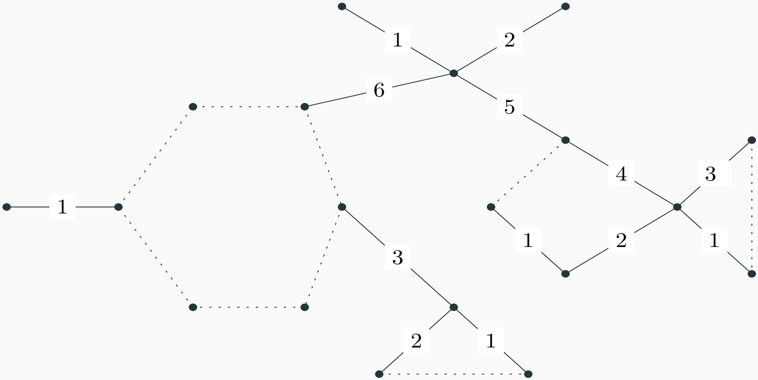
And finally: cactus graphs

Definition

A cactus graph is a tree graph with some vertices/edges replaced by cycles

Theorem

For any cactus graph on n vertices and circumference c (length of longest simple cycle), $\tau^+ \geq \frac{1}{2}c$, and $T^+ \geq \frac{1}{4}c^2 + 2(n-c) + 1$



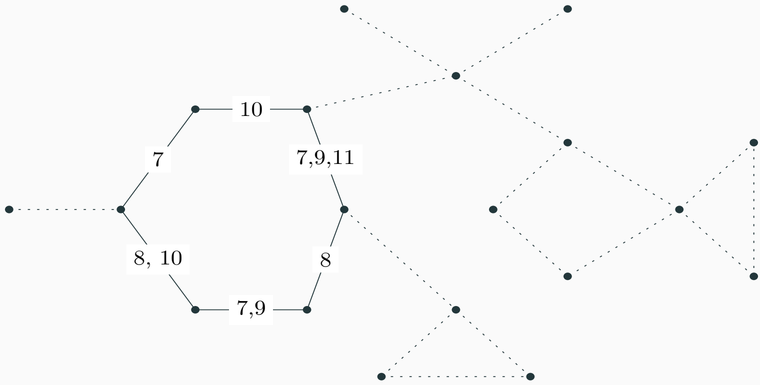
And finally: cactus graphs

Definition

A cactus graph is a tree graph with some vertices/edges replaced by cycles

Theorem

For any cactus graph on n vertices and circumference c (length of longest simple cycle), $\tau^+ \geq \frac{1}{2}c$, and $T^+ \geq \frac{1}{4}c^2 + 2(n-c) + 1$



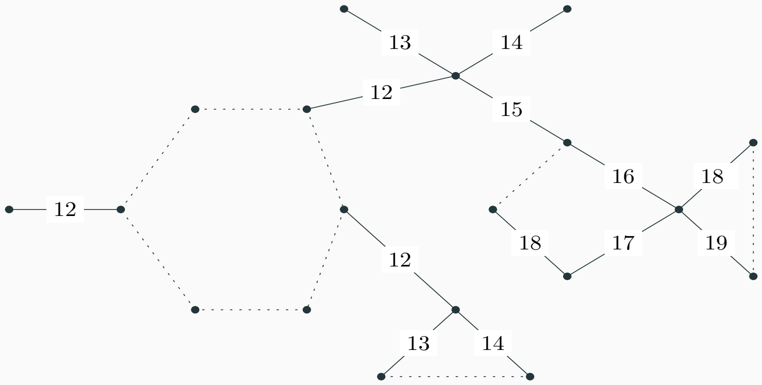
And finally: cactus graphs

Definition

A cactus graph is a tree graph with some vertices/edges replaced by cycles

Theorem

For any cactus graph on n vertices and circumference c (length of longest simple cycle), $\tau^+ \geq \frac{1}{2}c$, and $T^+ \geq \frac{1}{4}c^2 + 2(n-c) + 1$



Conclusion

Theorem

For any tree graph on n vertices, $\tau^+ = 2$ and $T^+ = 2n - 3$

Corollary

For any connected graph on n vertices, $\tau^+ \geq 2$ and $T^+ \geq 2n - 3$

Theorem

For any cycle graph on n vertices, $\tau^+ \geq \frac{1}{2}n$ and $T^+ \geq \frac{1}{4}n^2 + 1$

Corollary

For any Hamiltonian graph on n vertices, $\tau^+ \geq \frac{1}{2}n$ and $T^+ \geq \frac{1}{4}n^2 + 1$

Theorem

For any cactus graph on n vertices and circumference c, $\tau^+ \geq \frac{1}{2}c$, and $T^+ \geq \frac{1}{4}c^2 + 2(n - c) + 1$

Corollary

For any graph on n vertices and circumference c, $\tau^+ \geq \frac{1}{2}c$, and $T^+ \geq \frac{1}{4}c^2 + 2(n - c) + 1$

Conclusion

Future work and open questions

- Reduce upper bound of τ^+ in cycles, currently $n-1$ (to a tight $\frac{1}{2}n$?) by showing forbidden configurations of dominating journeys
- Can cactus graphs beat $\tau^+ = \frac{1}{2}c$?
- We adapted our labelling generator to work for general graphs, and $\tau^+ \leq \frac{1}{2}n$ (empirically) seems to hold in general as well
- Given a general graph, is it NP-hard to compute τ^+ or T^+?

Measure \ Labelling	Proper	Happy	Strict
τ^+	$\geq \frac{1}{2}n$	1 (by definition)	$> \frac{1}{2}n$? (empirically)
T^+	$\geq \frac{1}{4}n^2+1$	$\geq \frac{1}{18}n^2+O(n)$	$\geq \frac{1}{2}n^2+O(n)$

Conclusion

Future work and open questions

- Reduce upper bound of τ^+ in cycles, currently $n-1$ (to a tight $\frac{1}{2}n$?) by showing forbidden configurations of dominating journeys
- Can cactus graphs beat $\tau^+ = \frac{1}{2}c$?
- We adapted our labelling generator to work for general graphs, and $\tau^+ \leq \frac{1}{2}n$ (empirically) seems to hold in general as well
- Given a general graph, is it NP-hard to compute τ^+ or T^+?

Measure \ Labelling	Proper	Happy	Strict
τ^+	$\geq \frac{1}{2}n$	1 (by definition)	$> \frac{1}{2}n$? (empirically)
T^+	$\geq \frac{1}{4}n^2 + 1$	$\geq \frac{1}{18}n^2 + O(n)$	$\geq \frac{1}{2}n^2 + O(n)$

Thank you for your attention

Bibliography i

K. Axiotis and D. Fotakis, On the size and the approximability of minimum temporally connected subgraphs, arXiv preprint:1602.06411, (2016).

B. Baker, Gossips and telephones* brenda baker and robert shostak† discrete mathematics 2 (1972) 191–193 the following problem has circulated lately among mathematicians. other solutions have been given independently by rt bumby and by a. hajnal, ec milner and e. szemerédi., Discrete Mathematics, 2 (1972), pp. 191–193.

A. Casteigts, Efficient generation of simple temporal graphs up to isomorphism, (2020).

J. Enright, K. Meeks, and F. Skerman, Assigning times to minimise reachability in temporal graphs, Journal of Computer and System Sciences, 115 (2021), pp. 169–186.

- M. Latapy, T. Viard, and C. Magnien, Stream graphs and link streams for the modeling of interactions over time, Social Network Analysis and Mining, 8 (2018), pp. 1–29.
- G. B. Mertzios, O. Michail, I. Chatzigiannakis, and P. G. Spirakis, Temporal network optimization subject to connectivity constraints, in International Colloquium on Automata, Languages, and Programming, Springer, 2013, pp. 657–668.