How to make a temporal graph connected?

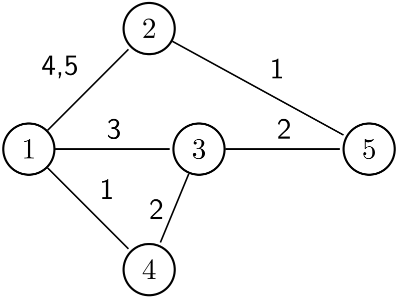
Thomas Bellitto Jules Bouton Popper Bruno Escoffier

LIP6
Sorbonne Université

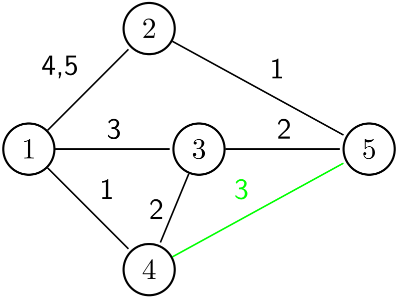
SAND 2025

Table of Contents

Example



Example



Temporal Connectivity Augmentation

Temporal Connectivity Augmentation (TCA)

Input: A temporal graph $\mathcal{G}$, a lifespan T and an integer K, (and a set F of temporal edges in the restricted case).
Question: Is there set $F' \subseteq V \times V \times \{1, \ldots, T\}$ ($F' \subseteq F$ in the restricted case) of size (=temporal cost) at most K such that $\mathcal{G} \uparrow F'$ is temporally connected?

- Unrestricted $=$ We can choose any temporal edge to augment the graph.

Temporal Connectivity Augmentation

Temporal Connectivity Augmentation (TCA)

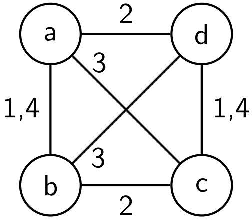
Input: A temporal graph $\mathcal{G}$, a lifespan T and an integer K, (and a set F of temporal edges in the restricted case).
Question: Is there set $F' \subseteq V \times V \times \{1, \ldots, T\}$ ($F' \subseteq F$ in the restricted case) of size (=temporal cost) at most K such that $\mathcal{G} \uparrow F'$ is temporally connected?

- Unrestricted = We can choose any temporal edge to augment the graph.
- In static $\rightarrow$ Polynomial time (trivial for graphs, Eswaran and Tarjan [ET76] for digraphs).

Temporal spanners

- Already connected temporal graph
- Remove as many edges as possible while maintaining connectivity

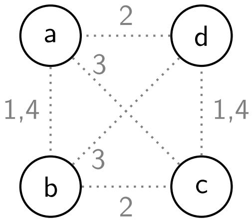
Figure: A temporal graph $\mathcal{G}$ and a spanner of $\mathcal{G}$

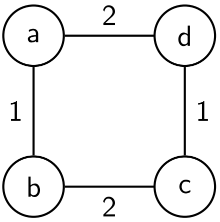


Connectivity augmentation

- Disconnected temporal graph
- Add as few *temporal* edges as possible to reach connectivity

Figure: An empty graph $\mathcal{G}$ with potential temporal edges to be added and the augmentation of $\mathcal{G}$





Spanners and connectivity augmentation

Proposition

There is a polynomial-time reduction from the temporal spanner problem to the temporal connectivity augmentation problem.

Spanners and connectivity augmentation

Proposition

There is a polynomial-time reduction from the temporal spanner problem to the temporal connectivity augmentation problem.

But the temporal connectivity problem allows starting from a non-empty graph → **TCA and spanners are not equivalent**

Figure: A graph $\mathcal{G}$ with potential temporal edges to be added and the optimal augmentation of $\mathcal{G}$

Table of Contents

2-TCA-Strict

2-TCA-Strict: Strict TCA with a temporal graph of lifespan 2

Theorem

2-TCA-Strict is NP-complete

Proof sketch:

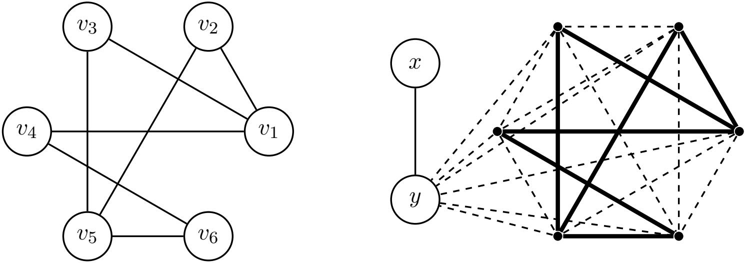
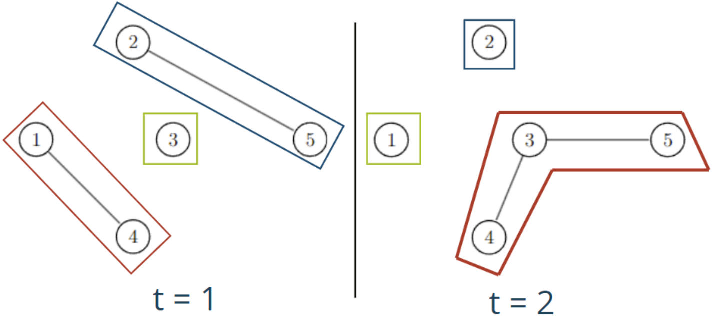


Figure: Example of a transformation of a DOMINATING SET instance into Strict TCA (unrestricted case). Dashed edges are present at time 1, solid edges are present at time 2 and the edges of the original graph are in bold.

Non-strict TCA: Snapshot components

Definition

In the non-strict setting, *snapshot* or *connected components at time* t are the connected components of the underlying subgraph in which all the edges that are not present at time t are removed.



Non-strict TCA: Characterization

Property

Let $C_1^1, \ldots, C_{k_1}^1, \ldots, C_1^T, \ldots, C_{k_T}^T$ be the snapshot components of a temporal graph $\mathcal{G}$. $\mathcal{G}$ satisfies property $\mathcal{P}$ if for every $i_1 \leq k_1$ and $i_T \leq k_T$, there is a sequence of $i_2, \ldots, i_{T-1}$ such that

$$\forall j < T : |C_{i_j}^j \cap C_{i_{j+1}}^{j+1}| \geq 1$$

$\rightarrow$ Path of components from time 1 to the lifespan.

Non-strict TCA: Characterization

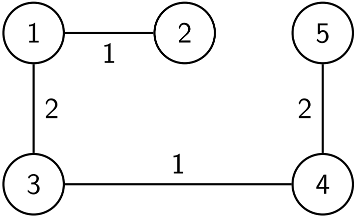
Characterization

A temporal graph $\mathcal{G}$ is temporally connected in the non-strict setting iff it verifies property $\mathcal{P}$.

Corollary

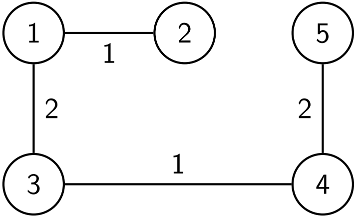
For a lifespan of 2, every snapshot component at time 1 has to intersect every snapshot component at time 2.

Component intersection matrices

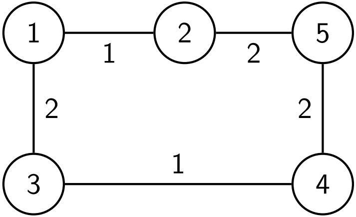


	$\{1,3\}$	$\{2\}$	$\{4,5\}$
$\{1,2\}$	1	1	0
$\{3,4\}$	1	0	1
$\{5\}$	0	0	1

Component intersection matrices



	$\{1,3\}$	$\{2\}$	$\{4,5\}$
$\{1,2\}$	1	1	0
$\{3,4\}$	1	0	1
$\{5\}$	0	0	1



	$\{1,3\}$	$\{2,4,5\}$
$\{1,2\}$	1	1
$\{3,4\}$	1	1
$\{5\}$	0	1

Reformulation with matrices

- one-filled matrix $\Leftrightarrow$ connectivity

Reformulation with matrices

- one-filled matrix ⇔ connectivity

OR-COMBINATION TO ONE-FILLED (OCTO)

Instance: A binary matrix B and a positive integer K
Question: Can B be transformed into a one-filled matrix with at most K OR-combinations?

Reformulation with matrices

- one-filled matrix ⇔ connectivity

OR-COMBINATION TO ONE-FILLED (OCTO)

Instance: A binary matrix B and a positive integer K
Question: Can B be transformed into a one-filled matrix with at most K OR-combinations?

→ Very similar to DISJOINT SET COVERS, NP-Complete! [CD05]:

- A finite set $T = \{e_1, \ldots, e_n\}$, a collection $C = \{T_1, \ldots, T_m\}$ of subsets of T and a positive integer K.
- We want to decide if C can be partitioned into at least K disjoint parts, such that every part is a cover of T

Table of Contents

Temporal Source Augmentation (TSA)

Connectivity $\rightarrow$ all pairs $\rightarrow$ Very hard
Source $\rightarrow$ only one vertex reaching all the other $\rightarrow$?

Temporal Source Augmentation (TSA)

Connectivity $\rightarrow$ all pairs $\rightarrow$ Very hard
Source $\rightarrow$ only one vertex reaching all the other $\rightarrow$?

Theorem

2-TSA and 2-STSA are NP-Complete

Asking for source is still too much...

2 pairs

Lemma

Connecting two pairs is polynomial time solvable

Proof.

Case 1:

The paths are disjoint $\rightarrow$ Easy to compute shortest paths □

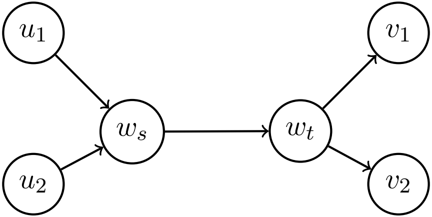
2 pairs

Lemma

Connecting two pairs is polynomial time solvable

Proof.

Case 2:



The paths join in the same direction

They can only join in one segment → Easy to decompose into $O(n^2)$ shortest path searchs

□

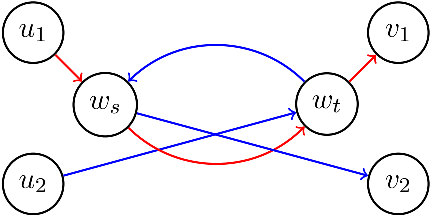
2 pairs (Strict)

Lemma

Connecting two pairs is polynomial time solvable

Proof.

Case 3:



The paths join in opposite directions

They can only join in one edge $\rightarrow$ Easy to decompose into $O(n^2)$ shortest path searchs □

Generalized connectivity

If we allow more than 2 pairs: Generalized connectivity

Theorem

Generalized connectivity (TPCA) is solvable in polynomial time with p the number of pairs fixed using Feldman et al. [FR06]

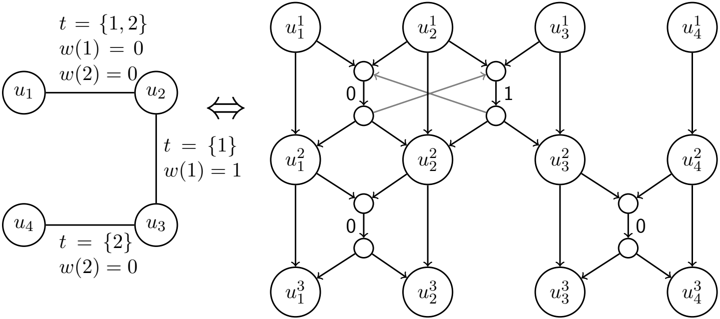


Table of Contents

Study overview

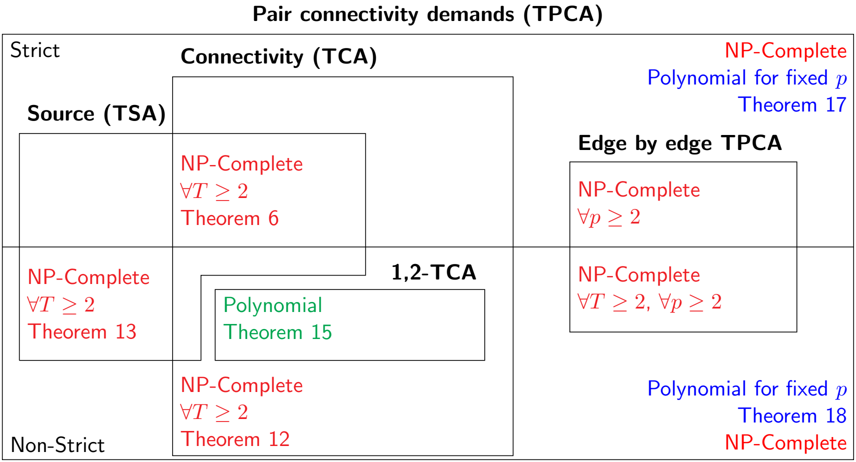


Figure: Overview of the study, with T the lifespan and p the number of pairs

Conclusion

Remarks

- Connectivity augmentation shows a jump in complexity from static to temporal;

Conclusion

Remarks

- Connectivity augmentation shows a jump in complexity from static to temporal;
- Even for the smallest lifespan of 2, the problem stays hard to solve. The same is true for the source augmentation variant;

Conclusion

Remarks

- Connectivity augmentation shows a jump in complexity from static to temporal;
- Even for the smallest lifespan of 2, the problem stays hard to solve. The same is true for the source augmentation variant;
- Looking at generalized connectivity, we find an interesting polynomial result if the number of pairs is fixed.

Conclusion

Remarks

- Connectivity augmentation shows a jump in complexity from static to temporal;
- Even for the smallest lifespan of 2, the problem stays hard to solve. The same is true for the source augmentation variant;
- Looking at generalized connectivity, we find an interesting polynomial result if the number of pairs is fixed.

Diversity of the methods we used: static to temporal, snapshot components, adapted temporal expansion

Future works

Perspectives

- Restriction on the family of the underlying graph: from paths to trees?
- Identifying temporal parameters to get FPT results (for the lifespan, most of our problems are para-NP-complete)
- Variants with temporal constraints: sliding time windows, blackout resilient, etc.
- ...

Future works

Perspectives

- Restriction on the family of the underlying graph: from paths to trees?
- Identifying temporal parameters to get FPT results (for the lifespan, most of our problems are para-NP-complete)
- Variants with temporal constraints: sliding time windows, blackout resilient, etc.
- ...

Open question: Can we extend our polynomial case?
$\rightarrow$ We prove that in the non strict case with 2 timesteps, one initially empty where all edges can be added and the other one where no edges can be added (1+1), the problem is solvable in polytime. $(T+1)$? (T_1+1+T_2)?..

Thank you!