On Computing Large Temporal (Unilateral) Connected Components

Andrea Marino

Universitá degli Studi Firenze, Italy

Joint work with
Isnard Lopes Costa[1], Raul Lopes[3], and Ana Silva[1,2]
[1]Universidade Federal do Ceará, Brazil.
[2]Universitá degli Studi Firenze, Italy.
[3]LIRMM, Université de Montpellier, France.

Slides from Raul Lopes and Ana Silva.

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

XP and FPT

Problem with input size n, associated parameter k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $O(n^k)$.

XP and FPT

Problem with input size n, associated parameter k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $O(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $O(2^k \cdot n^2)$.

XP and FPT

Problem with input size n, associated parameter k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $O(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $O(2^k \cdot n^2)$.
- Both imply polynomial running time for fixed k.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

XP and FPT

Problem with input size n, associated parameter k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $O(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $O(2^k \cdot n^2)$.
- Both imply polynomial running time for fixed k.
- W[1]-hard problem $\Rightarrow$ strong evidence that it is **not** FPT.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

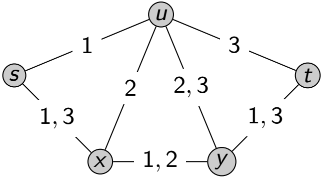
XP and FPT

Problem with input size n, associated parameter k:

- XP problem $\Rightarrow f(k) \cdot n^{g(k)}$ time algorithm.
  - Example: $O(n^k)$.
- FPT problem $\Rightarrow f(k) \cdot n^c$ time algorithm.
  - Example: $O(2^k \cdot n^2)$.
- Both imply polynomial running time for fixed k.
- W[1]-hard problem $\Rightarrow$ strong evidence that it is **not** FPT.
- k-Clique is W[1]-complete.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

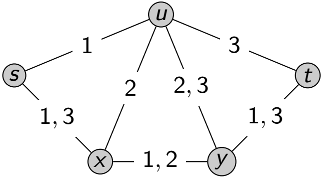
Temporal Graph



Definitions: Temporal Graph

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

Temporal Graph



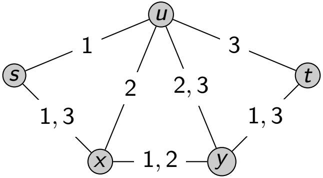
$$\tau = 3$$

Definitions: Lifetime

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- The value $\max_{e \in E(G)} \lambda(e)$ is called the *lifetime*; will be denoted by τ.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Temporal Graph

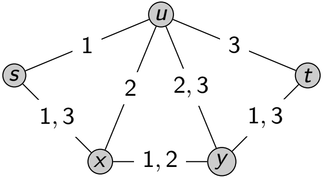


$$\tau = 3$$

Definitions: Temporal vertex/edge

- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- The value $\max_{e \in E(G)} \lambda(e)$ is called the *lifetime*; will be denoted by .
- A pair (x, i) where $x \in V(G)$ and $i \in [\tau]$ is called a temporal vertex;

Temporal Graph

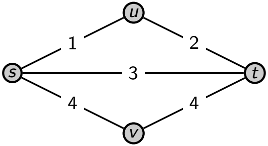


$\tau = 3$

Definitions: Temporal vertex/edge

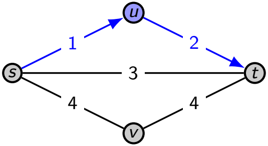
- A *temporal graph* is a pair (G, λ) where G is a simple graph, and $\lambda : E(G) \to 2^{\mathbb{N}}$;
- The value $\max_{e \in E(G)} \lambda(e)$ is called the *lifetime*; will be denoted by .
- A pair (x, i) where $x \in V(G)$ and $i \in [\tau]$ is called a temporal vertex; similarly (e, i) s.t. $e \in E(G)$ and $i \in \lambda(e)$ is called a temporal edge.

Understanding temporal paths



Strict Model: Valid walks are the ones whose labels are *strictly increasing*.

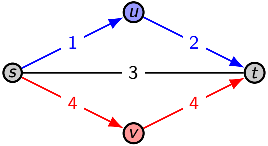
Understanding temporal paths



Strict Model: Valid walks are the ones whose labels are *strictly increasing*.
Non-Strict Model: Valid walks are the ones whose labels are *non-strictly increasing*.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Understanding temporal paths



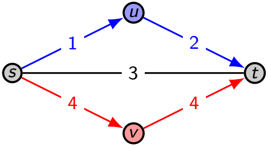
Strict Model: Valid walks are the ones whose labels are *strictly increasing*.
Non-Strict Model: Valid walks are the ones whose labels are *non-strictly increasing*.

Polynomial check even with some optimization criteria.

Xuan, B. Bui, Afonso Ferreira, and Aubin Jarry.
"Computing shortest, fastest, and foremost journeys in dynamic networks.". International Journal of Foundations of Computer Science 14.02 (2003): 267-285

Wu, Huanhuan, et al.
"Efficient algorithms for temporal path computation." IEEE Transactions on Knowledge and Data Engineering 28.11 (2016): 2927-2942.

Understanding temporal paths



Strict Model: Valid walks are the ones whose labels are *strictly increasing*.
Non-Strict Model: Valid walks are the ones whose labels are *non-strictly increasing*.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Components in Static Undirected Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in G.

Components in Static Undirected Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in G.
  - $v \in R(u) \iff u \in R(v)$ symmetric

Components in Static Undirected Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in G.
  - $v \in R(u) \iff u \in R(v)$ symmetric
  - $v \in R(u) \wedge w \in R(v) \implies w \in R(u)$ transitive.

Components in Static Undirected Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in G.
  - $v \in R(u) \iff u \in R(v)$ symmetric
  - $v \in R(u) \land w \in R(v) \implies w \in R(u)$ transitive.
- *Component:* maximal set of vertices C s.t. $v \in R(u)$ for all $u, v \in C$.

Components in Static Undirected Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in G.
  - $v \in R(u) \iff u \in R(v)$ symmetric
  - $v \in R(u) \wedge w \in R(v) \implies w \in R(u)$ transitive.
- *Component:* maximal set of vertices C s.t. $v \in R(u)$ for all $u, v \in C$.
  - Equivalence classes of $V(G)$.

Components in Static Undirected Graphs

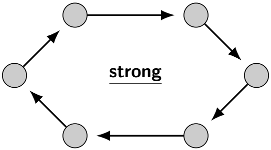
- $v \in R(u) \implies$ exists $u \to v$ *path* in G.
  - $v \in R(u) \iff u \in R(v)$ symmetric
  - $v \in R(u) \wedge w \in R(v) \implies w \in R(u)$ transitive.
- *Component:* maximal set of vertices C s.t. $v \in R(u)$ for all $u, v \in C$.
  - Equivalence classes of $V(G)$.
  - Find all components of G in poly-time.

Components in Static Directed Graphs

- $v \in R(u) \implies$ exists $u \rightarrow v$ *path* in the **digraph** G.
- *Strong component:* maximal set of vertices C s.t. $v \in R(u) \wedge u \in R(v)$ for all $u, v \in C$.

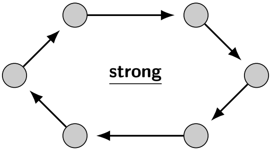
Components in Static Directed Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in the **digraph** G.
- *Strong component:* maximal set of vertices C s.t. $v \in R(u) \wedge u \in R(v)$ for all $u, v \in C$.



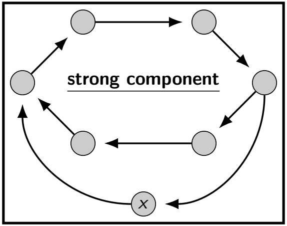
Components in Static Directed Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in the **digraph** G.
- *Strong component:* maximal set of vertices C s.t. $v \in R(u) \wedge u \in R(v)$ for all $u, v \in C$.



Components in Static Directed Graphs

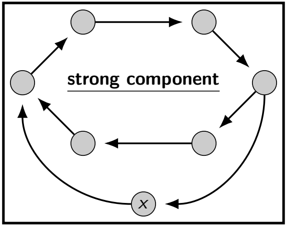
- $v \in R(u) \implies$ exists $u \to v$ *path* in the **digraph** G.
- *Strong component:* maximal set of vertices C s.t. $v \in R(u) \wedge u \in R(v)$ for all $u, v \in C$.



- Every path starting and ending in C can be added to C.

Components in Static Directed Graphs

- $v \in R(u) \implies$ exists $u \to v$ *path* in the **digraph** G.
- *Strong component:* maximal set of vertices C s.t. $v \in R(u) \wedge u \in R(v)$ for all $u, v \in C$.



- Every path starting and ending in C can be added to C.
- Key property for strong component algorithms in digraphs

Translation to temporal graphs

- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).

Translation to temporal graphs

- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).
- *Components in temporal graphs:* maximal subset $C \subseteq V(G)$ s.t. $v \in R(u)$ for all $u, v \in C$.

Translation to temporal graphs

- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).
- *Components in temporal graphs:* maximal subset $C \subseteq V(G)$ s.t. $v \in R(u)$ for all $u, v \in C$.
  - Symmetric?

Translation to temporal graphs

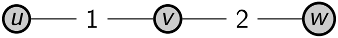
- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).
- *Components in temporal graphs:* maximal subset $C \subseteq V(G)$ s.t. $v \in R(u)$ for all $u, v \in C$.
    - ~~Symmetric?~~

$$u \;\text{——}\; 1 \;\text{——}\; v \;\text{——}\; 2 \;\text{——}\; w$$

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Translation to temporal graphs

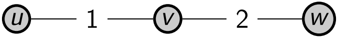
- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).
- *Components in temporal graphs:* maximal subset $C \subseteq V(G)$ s.t. $v \in R(u)$ for all $u, v \in C$.
  - ~~Symmetric?~~
  - Transitive?



Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

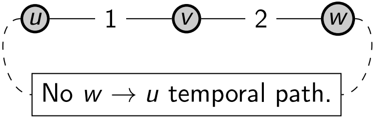
Translation to temporal graphs

- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).
- *Components in temporal graphs:* maximal subset $C \subseteq V(G)$ s.t. $v \in R(u)$ for all $u, v \in C$.
    - ~~Symmetric?~~
    - ~~Transitive?~~



Translation to temporal graphs

- $v \in R(u) \implies$ exists $u \to v$ *temporal path* in (G, λ).
- *Components in temporal graphs:* maximal subset $C \subseteq V(G)$ s.t. $v \in R(u)$ for all $u, v \in C$.
  - ~~Symmetric?~~
  - ~~Transitive?~~



Not true that every path starting and ending in C can be added to C.

Temporal component

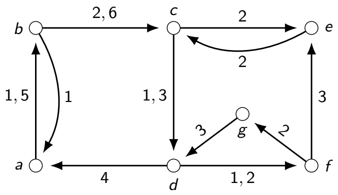
- TCC: Temporal Connected Component.

Temporal component

- TCC: Temporal Connected Component.
- $v \in R(u)$: exists temporal path from u to v in (G, λ).

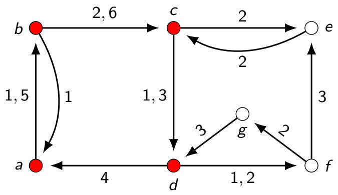
Temporal component

- TCC: Temporal Connected Component.
- $v \in R(u)$: exists temporal path from u to v in (G, λ).



Temporal component

- TCC: Temporal Connected Component.
- $v \in R(u)$: exists temporal path from u to v in (G, λ).

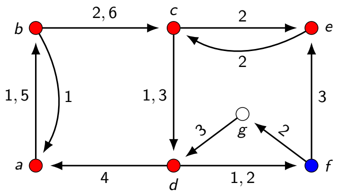


$\forall u, v \ldots$

- **Closed TCC:** $v \in R(u) \wedge u \in R(v)$ **inside** C.

Temporal component

- TCC: Temporal Connected Component.
- $v \in R(u)$: exists temporal path from u to v in (G, λ).

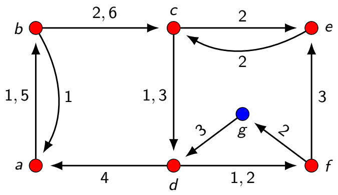


$\forall u, v \ldots$

- **Closed TCC:** $v \in R(u) \wedge u \in R(v)$ **inside** C.
- **TCC:** $v \in R(u) \wedge u \in R(v)$ (paths can use vertices not in C, like $d \to f \to e$).

Temporal component

- TCC: Temporal Connected Component.
- $v \in R(u)$: exists temporal path from u to v in (G, λ).



$\forall u, v \ldots$

- **Closed TCC:** $v \in R(u) \wedge u \in R(v)$ **inside** C.
- **TCC:** $v \in R(u) \wedge u \in R(v)$ (paths can use vertices not in C, like $d \to f \to e$).
- **Unilateral (closed) TCC (or TUCC):** $v \in R(u) \vee u \in R(v)$.

Strict vs non-Strict Components

Recall: Strict walks are the ones with labels strictly increasing. Non-strict are the ones with labels non-decreasing.

- **Strict:** Easy reduction from k-Clique, for every def of C. Given static G, give time 1 to all the edges. There is component of size $\geq k$ iff there is a clique of size $\geq k$ in G.
- **Non-Strict:** this reductions does not work. It is enough a vertex of degree at least k to make it fail. **We work in this case!**

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE
DISIA
DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Hardness results

Our results, for the non-strict case, build on top of and improves upon

Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks".
Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

Arnaud Casteigts, Timothée Corsini, and Writika Sarka.
"Simple, strict, proper, happy: A study of reachability in temporal graphs".
arXiv preprint arXiv:2208.01720, 2022.

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Hardness results

Our results, for the non-strict case, build on top of and improves upon

Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks".
Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

Arnaud Casteigts, Timothée Corsini, and Writika Sarka.
"Simple, strict, proper, happy: A study of reachability in temporal graphs".
arXiv preprint arXiv:2208.01720, 2022.

- Previous results are not parameterized reductions,

Hardness results

Our results, for the non-strict case, build on top of and improves upon

Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks".
Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

Arnaud Casteigts, Timothée Corsini, and Writika Sarka.
"Simple, strict, proper, happy: A study of reachability in temporal graphs".
arXiv preprint arXiv:2208.01720, 2022.

- Previous results are not parameterized reductions,
- and leave open cases when lifetime $= 2$ or 3.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Theorem

For any fixed $\tau \geq 2$, given a temporal graph $\mathcal{G}$ and an integer k, it is NP*-complete to decide if $\mathcal{G}$ has a (closed) TCC or a (closed) TUCC of size at least k, even if G is the line graph of a bipartite graph.*

Reduction from the Maximum Edge Biclique Problem: given a bipartite graph G and an integer k, deciding whether G has a biclique with at least k edges.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

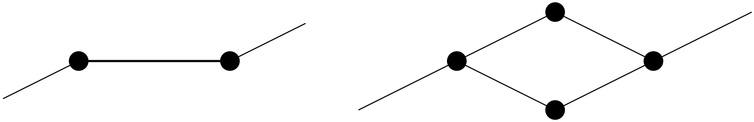
Other Reductions: the semaphore technique

- Swap edges in a graph by diamonds in a temporal graph to control behavior of paths.

Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks". Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

Other Reductions: the semaphore technique

- Swap edges in a graph by diamonds in a temporal graph to control behavior of paths.

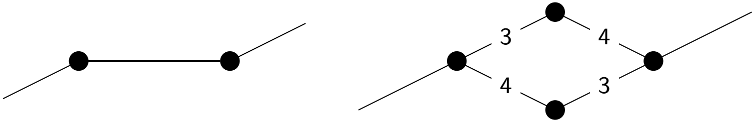


Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks". Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Other Reductions: the semaphore technique

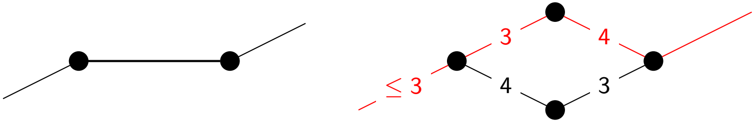
- Swap edges in a graph by diamonds in a temporal graph to control behavior of paths.



Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks". Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

Other Reductions: the semaphore technique

- Swap edges in a graph by diamonds in a temporal graph to control behavior of paths.



Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks". Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

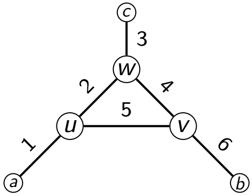
Other Reductions: the semaphore technique

- Swap edges in a graph by diamonds in a temporal graph to control behavior of paths.

Sandeep Bhadra and Afonso Ferreira.
"Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks". Ad-Hoc, Mobile, and Wireless Networks, Second International Conference, 2003.

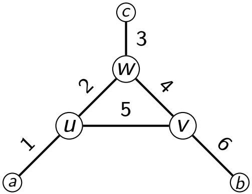
Example: from k-Clique to undir. open TCC

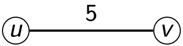
- Graph G, arbitrary order $e_1, \ldots, e_m$ for $E(G)$.

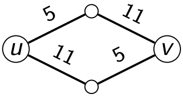


Example: from k-Clique to undir. open TCC

- Graph G, arbitrary order $e_1, \ldots, e_m$ for $E(G)$.
- For each $(u, v) \in E(G)$ with label i, add to (G, λ) diamond with times $i, m+i$ and $m+i, i$.



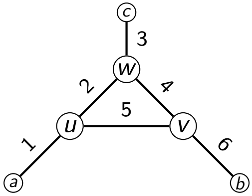


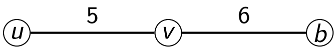


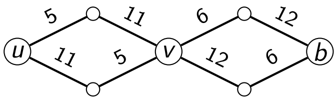
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Example: from k-Clique to undir. open TCC

- Graph G, arbitrary order $e_1, \ldots, e_m$ for $E(G)$.
- For each $(u, v) \in E(G)$ with label i, add to (G, λ) diamond with times $i, m + i$ and $m + i, i$.

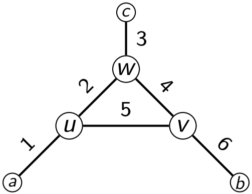


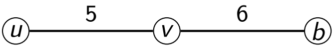


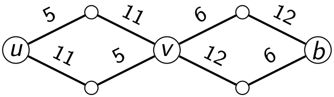


Example: from k-Clique to undir. open TCC

- Graph G, arbitrary order $e_1, \ldots, e_m$ for $E(G)$.
- For each $(u, v) \in E(G)$ with label i, add to (G, λ) diamond with times $i, m + i$ and $m + i, i$.





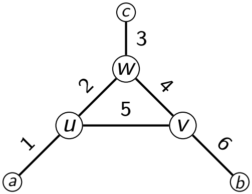


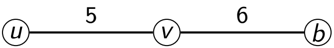
- $(x, y) \in E(G) \iff$ temporal $x \to y$ and $y \to x$ paths in (G, λ).

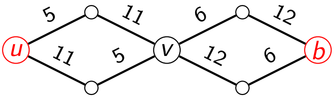
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Example: from k-Clique to undir. open TCC

- Graph G, arbitrary order $e_1, \ldots, e_m$ for $E(G)$.
- For each $(u, v) \in E(G)$ with label i, add to (G, λ) diamond with times $i, m+i$ and $m+i, i$.





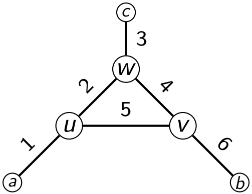


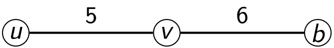
- $(x, y) \in E(G) \iff$ temporal $x \to y$ and $y \to x$ paths in (G, λ).
- no $u \to b$ temporal path.

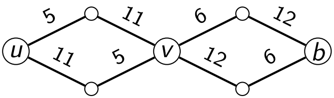
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Example: from k-Clique to undir. open TCC

- Graph G, arbitrary order $e_1, \ldots, e_m$ for $E(G)$.
- For each $(u, v) \in E(G)$ with label i, add to (G, λ) diamond with times $i, m+i$ and $m+i, i$.

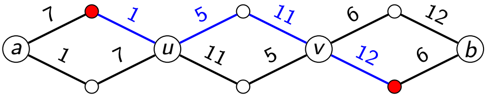




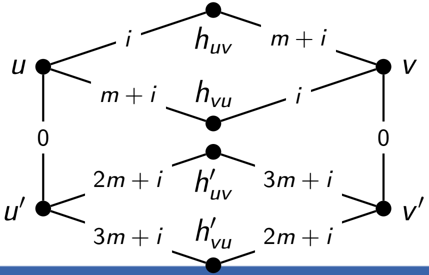


- $(x, y) \in E(G) \iff$ temporal $x \to y$ and $y \to x$ paths in (G, λ).
- no $u \to b$ temporal path.
  - Cliques in G are temporal connected sets in (G, λ).

This is not enough because we can risk to have a component made of an uncontrolled number of dummy vertices.



We solve this issue, creating a copy $\mathcal{G}'$ of the temporal graph $\mathcal{G}$ and connecting their corresponding vertices at time 0. Then we ask for a component of size at least $2k$ instead of k.



Theorem

Given integer k and temporal graph $\mathcal{G} = (G, \lambda)$,

- *deciding if $\mathcal{G}$ has a TCC of size $\geq k$ is W[1]-hard with parameter k;*

Using the reduction we have just seen

Theorem

Given integer k and temporal graph $\mathcal{G} = (G, \lambda)$,

- *deciding if $\mathcal{G}$ has a TCC of size $\geq k$ is* W*[1]-hard with parameter k;*
- *if G is directed, deciding if $\mathcal{G}$ has a TCC (TUCC) of size $\geq k$ is* W*[1]-hard with parameter k, even if $\mathcal{G}$ has lifetime 2; and*

Simple directed semaphore with times 1 and 2.

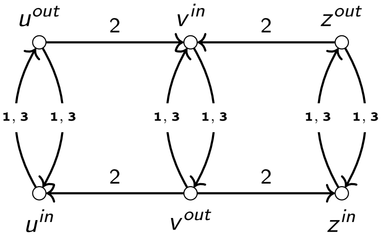
Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Theorem

Given integer k and temporal graph $\mathcal{G} = (G, \lambda)$,

- *deciding if $\mathcal{G}$ has a TCC of size $\geq k$ is* W[1]*-hard with parameter k;*
- *if G is directed, deciding if $\mathcal{G}$ has a TCC (TUCC) of size $\geq k$ is* W[1]*-hard with parameter k, even if $\mathcal{G}$ has lifetime* 2*; and*
- *if G is directed, deciding if $\mathcal{G}$ has a closed TCC (closed TUCC) of size $\geq k$ is* W[1]*-hard with parameter k, even if $\mathcal{G}$ has lifetime* 3*.*

For this we can split the nodes.



Positive results

Theorem

Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time

- $O(k^{k\cdot\tau} \cdot n)$ *that decides whether there is a* TCC *of size at least* k;

Positive results

Theorem

Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time

- $O(k^{k \cdot \tau} \cdot n)$ *that decides whether there is a* TCC *of size at least k;*
- $O(2^{k^{\tau}} \cdot n)$ *that decides whether there is a* closed TCC *of size at least k;*

Positive results

Theorem

Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time

- $O(k^{k \cdot \tau} \cdot n)$ *that decides whether there is a* TCC *of size at least k;*
- $O(2^{k^{\tau}} \cdot n)$ *that decides whether there is a* closed TCC *of size at least k;*
- $O(k^{k^2} \cdot n)$ *that decides whether there is a* TUCC *of size at least k; and*

Positive results

Theorem

Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time

- $O(k^{k\cdot\tau} \cdot n)$ *that decides whether there is a* TCC *of size at least k;*
- $O(2^{k^{\tau}} \cdot n)$ *that decides whether there is a* closed TCC *of size at least k;*
- $O(k^{k^2} \cdot n)$ *that decides whether there is a* TUCC *of size at least k; and*
- $O(2^{k^{k}} \cdot n)$ *that decides whether there is a* closed TUCC *of size at least k.*

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Positive results

Theorem

Given a temporal graph $\mathcal{G} = (G, \lambda)$ on n vertices and with lifetime τ, and a positive integer k, there are algorithms running in time

- $O(k^{k\cdot\tau} \cdot n)$ *that decides whether there is a* TCC *of size at least* k*;*
- $O(2^{k^{\tau}} \cdot n)$ *that decides whether there is a* closed TCC *of size at least* k*;*
- $O(k^{k^2} \cdot n)$ *that decides whether there is a* TUCC *of size at least* k*; and*
- $O(2^{k^{k}} \cdot n)$ *that decides whether there is a* closed TUCC *of size at least* k*.*

- Non-closed cases, a component is found.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Results

<table>
<tr><th></th><th>Par. τ</th><th>Par. k</th><th>Par. $k+\tau$</th></tr>
<tr><td>TCC</td><td rowspan="4">p-NP $\tau \geq 2$</td><td>W[1]-h Dir. $\tau \geq 2$ and Undir.</td><td rowspan="2">W[1]-h Dir. FPT Undir.</td></tr>
<tr><td>TUCC</td><td>W[1]-h Dir. $\tau \geq 2$ FPT Undir.</td></tr>
<tr><td>closed TCC</td><td>W[1]-h Dir. $\tau \geq 3$</td><td rowspan="2">W[1]-h Dir. FPT Undir.</td></tr>
<tr><td>closed TUCC</td><td>W[1]-h Dir. $\tau \geq 3$ FPT Undir.</td></tr>
</table>

Finanziato dall'Unione europea
NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE
DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

- Open: Parameterized complexity of deciding if an undirected temporal graph has a closed TCC (TUCC) of size $\geq k$?

- Open: Parameterized complexity of deciding if an undirected temporal graph has a closed TCC (TUCC) of size $\geq k$?
- Open: Parameterized complexity of deciding if an directed temporal graph of lifetime 2 has a closed TCC (TUCC) of size $\geq k$?

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Other results

	Check whether $X \subseteq V$ is a connected set	Check whether $X \subseteq V$ is a component
TCC	$\Theta(M^2)$	$O(n^2 \cdot M)$
TUCC		
closed TCC		NP-c
closed TUCC		

Checking connectivity cannot be subquadratic

(Notation $\tilde{O}(\cdot)$ ignores polylog factors).

Theorem

Consider a temporal graph $\mathcal{G}$ on M temporal edges. There is no algorithm running in time $\tilde{O}(M^{2-\varepsilon})$, for some ϵ, that decides whether G is temporally (unilaterally) connected, unless SETH *fails.*

In the k-SAT* problem, we have the formula, two sets X and Y each of half variables, and all the possible assignments for X and Y. We build a temporal graph not connected iff the formula is satisfiable.
It works both for strict and non-strict case.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Checking maximality is hard

Theorem

Let $\mathcal{G}$ be a (directed) temporal graph, and $Y \subseteq V(\mathcal{G})$. Deciding whether Y is a closed TCC *is* NP*-complete. The same holds for* closed TUCC.

Reduction from k-Club. We reduce from the problem of deciding whether a subset of vertices X of a given a graph G is a maximal 2-club, where a 2-club is a set of vertices C such that $G[C]$ has diameter at most 2.
It works both for strict and non-strict case.

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Thanks for the attention!

email: `andrea.marino@unifi.it`

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
UNIVERSITÀ DEGLI STUDI FIRENZE
DISIA DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"