Sparse Spanners in Temporal Cliques and Where to Find Them

September 3, 2024

Sebastian Angrick, **Ben Bals**, Tobias Friedrich,
Hans Gawendowicz, Niko Hastrich, Nicolas Klodt,
Pascal Lenzner, Jonas Schmidt, George Skretas,
Armin Wells

Hasso Plattner Institute,
University of Potsdam

Overview

- Short introduction to temporal graphs

Overview

- Short introduction to temporal graphs
- Spanners in temporal graphs

Overview

- Short introduction to temporal graphs
- Spanners in temporal graphs
- ”How to Reduce Temporal Cliques to Find Sparse Spanners”

Overview

- Short introduction to temporal graphs
- Spanners in temporal graphs
- ”How to Reduce Temporal Cliques to Find Sparse Spanners”
- Summary and next steps

Overview

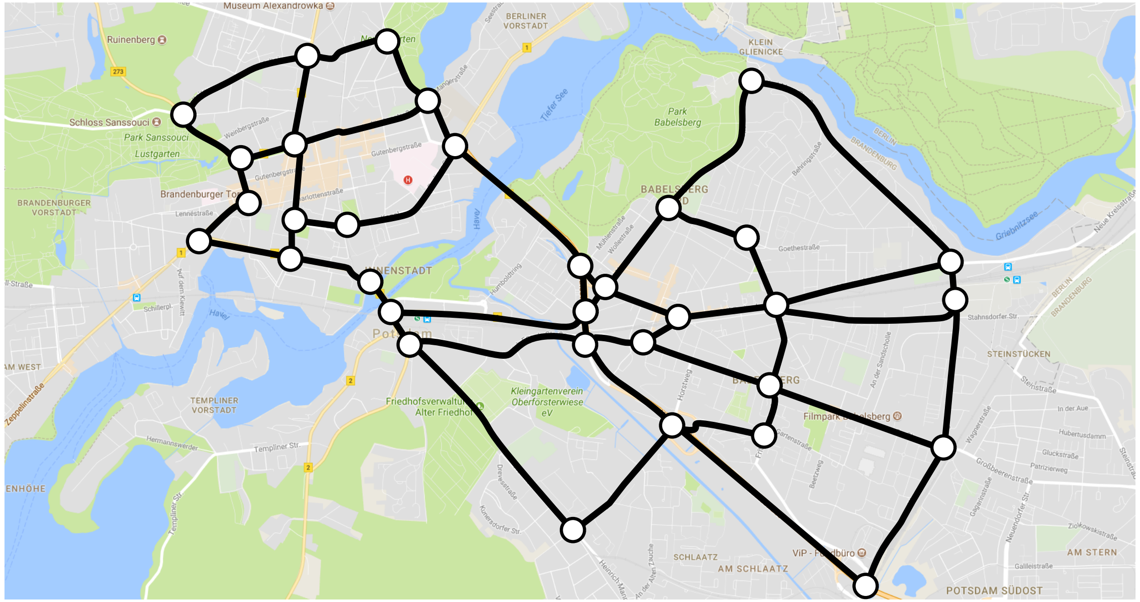
- Short introduction to temporal graphs
- Spanners in temporal graphs
- ”How to Reduce Temporal Cliques to Find Sparse Spanners”
- Summary and next steps

Questions are very welcome!

Temporal Graphs—Motivation

Static Graphs

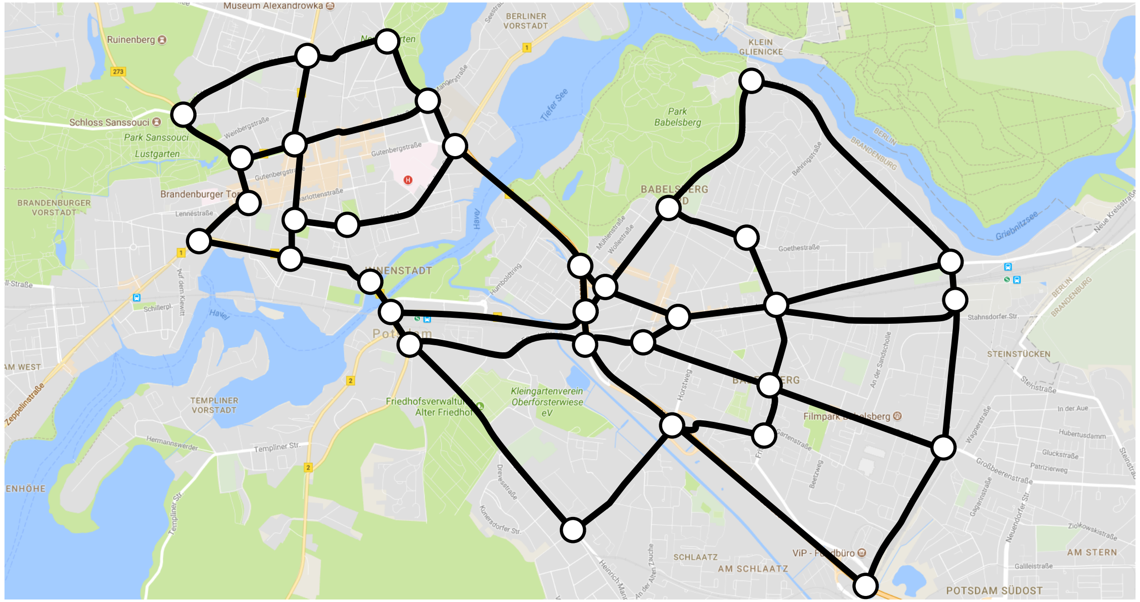
- Example: road networks



Temporal Graphs—Motivation

Static Graphs

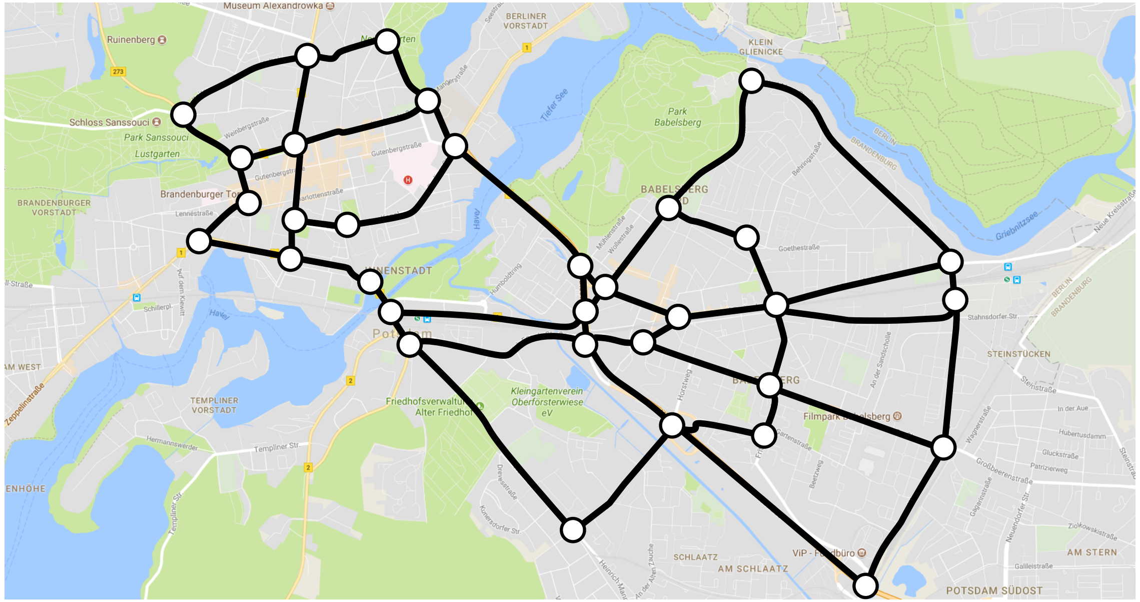
- Example: road networks
- Full algorithmic and graph-theoretic toolkit
- Common problems:
  - Shortest paths
  - Spanning trees
  - Matchings



Temporal Graphs—Motivation

Static Graphs

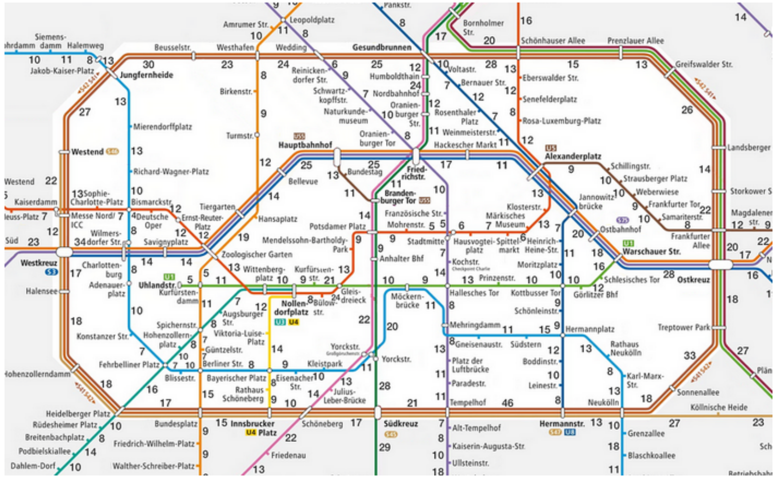
- Example: road networks
- Full algorithmic and graph-theoretic toolkit
- Common problems:
  - Shortest paths
  - Spanning trees
  - Matchings



Easy!

Temporal Graphs—Motivation

Social network graph



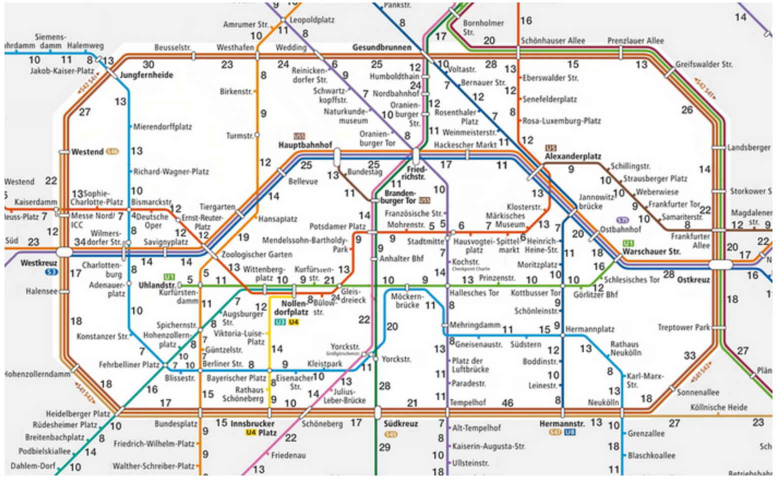
S-Bahn network of Berlin

Dynamic Graphs

- Example: friendship graphs, public transport

Temporal Graphs—Motivation

Social network graph



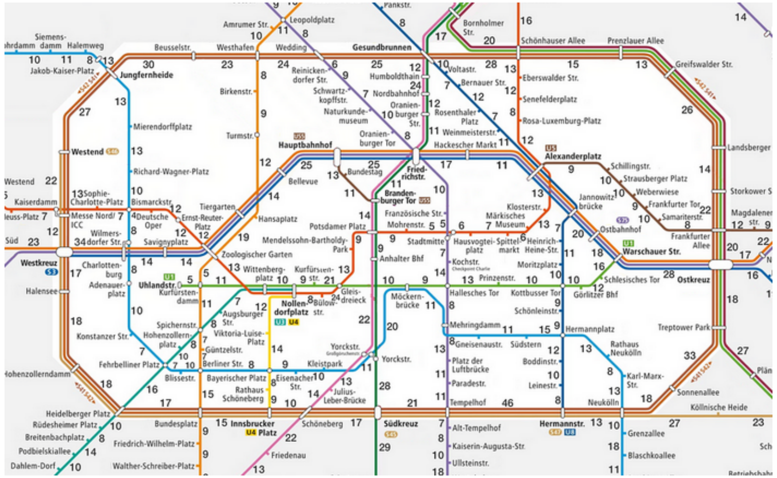
S-Bahn network of Berlin

Dynamic Graphs

- Example: friendship graphs, public transport
- Multiple settings based on updates
  - Node or edge updates
  - Online or offline

Temporal Graphs—Motivation

Social network graph



S-Bahn network of Berlin

Dynamic Graphs

- Example: friendship graphs, public transport
- Multiple settings based on updates
  - Node or edge updates
  - Online or offline
- **Temporal Graphs**: edges available at given timestamps
  - Timestamps known in advance

Static vs Temporal Graphs

Classical Problem

Temporal challenges

Static vs Temporal Graphs

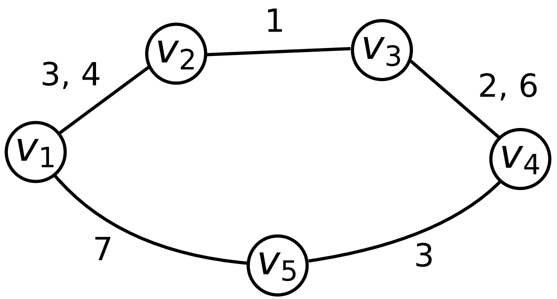
Classical Problem

Shortest Paths

Temporal challenges

Connectivity is not transitive!

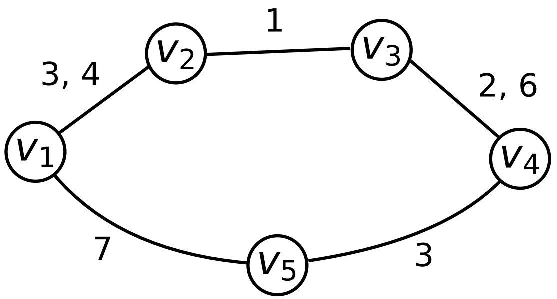
$\Rightarrow$ Can be NP-hard



There are paths $v_1 \rightsquigarrow v_2$ and $v_2 \rightsquigarrow v_3$, but not $v_1 \rightsquigarrow v_3$

Static vs Temporal Graphs

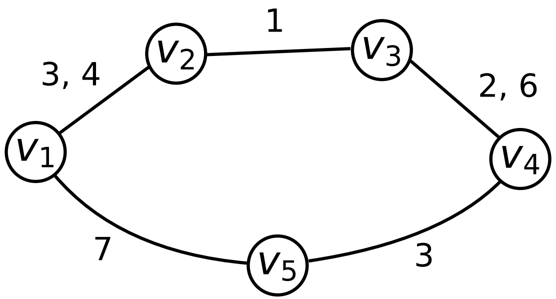
Classical Problem	Temporal challenges
Shortest Paths	Connectivity is not transitive! $\Rightarrow$ Can be NP-hard
Matchings	Menger's theorem does not hold $\Rightarrow$ Can be NP-hard



There are paths $v_1 \rightsquigarrow v_2$ and $v_2 \rightsquigarrow v_3$, but not $v_1 \rightsquigarrow v_3$

Static vs Temporal Graphs

Classical Problem	Temporal challenges
Shortest Paths	Connectivity is not transitive! $\Rightarrow$ Can be NP-hard
Matchings	Menger's theorem does not hold $\Rightarrow$ Can be NP-hard
Minimum spanner	Minimum spanners usually not trees $\Rightarrow$ Even quadratic size sometimes!



There are paths $v_1 \rightsquigarrow v_2$ and $v_2 \rightsquigarrow v_3$, but not $v_1 \rightsquigarrow v_3$

Spanners in Temporal Graphs

Minimum size edge set that yields
all pairs connectivity

Spanners in Temporal Graphs

Minimum size edge set that yields
all pairs connectivity

- Only consider temporally connected graphs

Spanners in Temporal Graphs

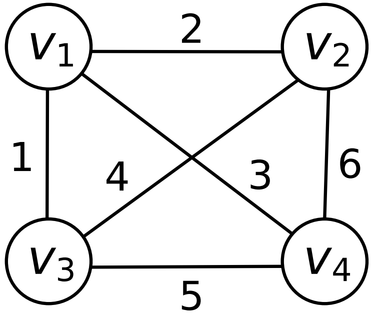
Minimum size edge set that yields all pairs connectivity

- Only consider temporally connected graphs
- For arbitrary graphs: $\Theta(n^2)$ edges needed (Axiotis et al., ICALP'16)

Spanners in Temporal Graphs

Minimum size edge set that yields
all pairs connectivity

- Only consider temporally connected graphs
- For arbitrary graphs: $\Theta(n^2)$ edges needed (Axiotis et al., ICALP'16)
- Does not hold for temporal cliques



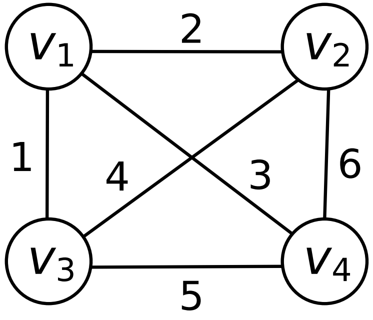
Temporal Cliques

- Underlying graph is a clique
- Each edge has exactly one unique timestamp

Spanners in Temporal Graphs

Minimum size edge set that yields all pairs connectivity

- Only consider temporally connected graphs
- For arbitrary graphs: $\Theta(n^2)$ edges needed (Axiotis et al., ICALP'16)
- Does not hold for temporal cliques
  - $\mathcal{O}(n \log(n))$ upper bound (Casteigts et al., ICALP '19)
  - No clique known with minimum spanners greater than $2n - 3$



Temporal Cliques

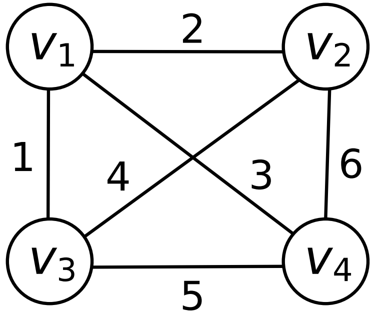
- Underlying graph is a clique
- Each edge has exactly one unique timestamp

Spanners in Temporal Graphs

Minimum size edge set that yields
all pairs connectivity

- Only consider temporally connected graphs
- For arbitrary graphs: $\Theta(n^2)$ edges needed (Axiotis et al., ICALP'16)
- Does not hold for temporal cliques
  - $\mathcal{O}(n \log(n))$ upper bound (Casteigts et al., ICALP '19)
  - No clique known with minimum spanners greater than $2n - 3$

Task: find $\Theta(n)$ sized spanners for temporal cliques!



Temporal Cliques

- Underlying graph is a clique
- Each edge has exactly one unique timestamp

Spanners in Temporal Cliques

- All pairs connectivity in temporal cliques

Spanners in Temporal Cliques

- All pairs connectivity in temporal cliques

(Casteigts et al., FOCS ’22)

- Temporal cliques permit a $O(n)$ spanner a.a.s.

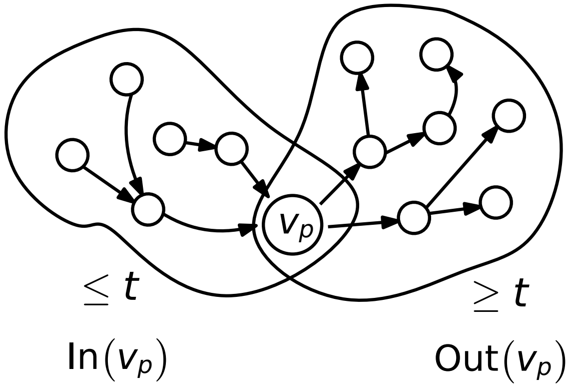
  - Timelabels assigned randomly

Spanners in Temporal Cliques

- All pairs connectivity in temporal cliques

(Casteigts et al., FOCS ’22)

- Temporal cliques permit a $O(n)$ spanner a.a.s.
  - Timelabels assigned randomly
  - Because pivot-vertices exist a.a.s.



- Construction for optimal spanners
- Structure exists a.a.s.

Spanners in Temporal Cliques

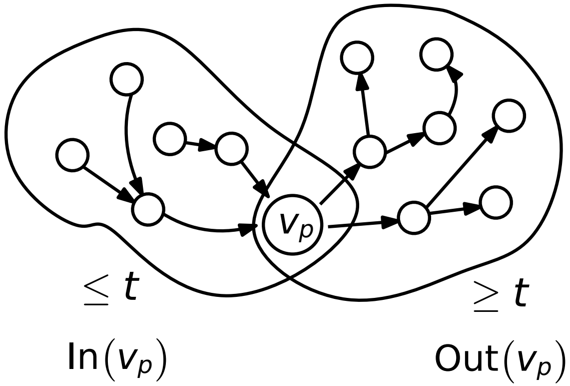
- All pairs connectivity in temporal cliques

(Casteigts et al., FOCS ’22)

- Temporal cliques permit a $O(n)$ spanner a.a.s.
  - Timelabels assigned randomly
  - Because pivot-vertices exist a.a.s.

How about all temporal cliques?

- Cliques without pivot-vertices exist



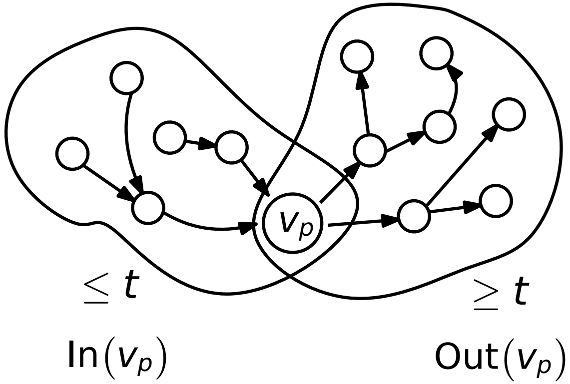
- Construction for optimal spanners
- Structure exists a.a.s.

Spanners in Temporal Cliques

- All pairs connectivity in temporal cliques

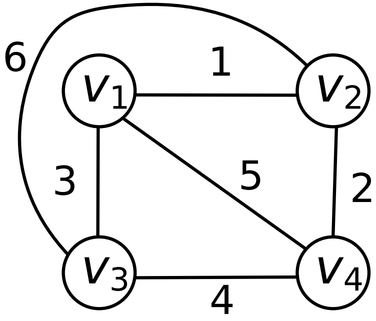
(Casteigts et al., FOCS '22)

- Temporal cliques permit a $O(n)$ spanner a.a.s.
  - Timelabels assigned randomly
  - Because pivot-vertices exist a.a.s.



How about all temporal cliques?

- Cliques without pivot-vertices exist

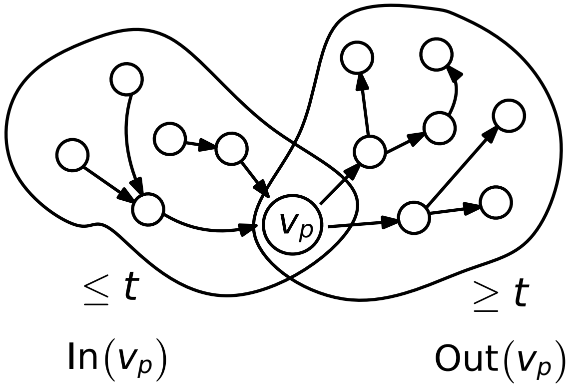


- Construction for optimal spanners
- Structure exists a.a.s.

Spanners in Temporal Cliques

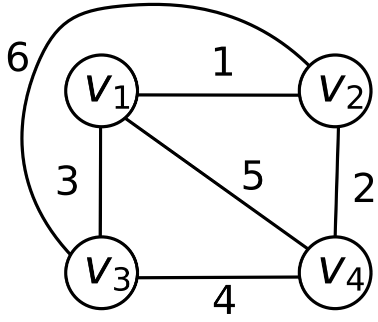
- All pairs connectivity in temporal cliques

  (Casteigts et al., FOCS '22)
- Temporal cliques permit a $O(n)$ spanner a.a.s.
  - Timelabels assigned randomly
  - Because pivot-vertices exist a.a.s.



How about all temporal cliques?

- Cliques without pivot-vertices exist

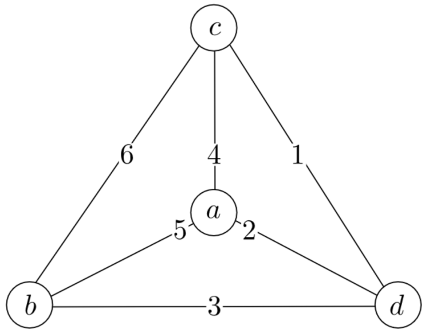


- Construction for optimal spanners
- Structure exists a.a.s.

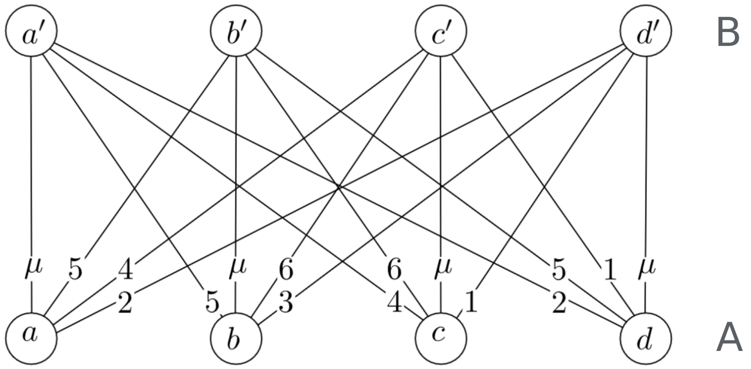
Task: find $\Theta(n)$ sized spanners for **all** temporal cliques!

Temporal Bicliques

- Bicliques are better to work with
- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity



A temporal clique



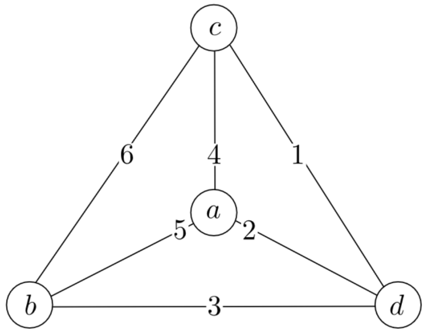
The corresponding temporal biclique

Temporal Bicliques

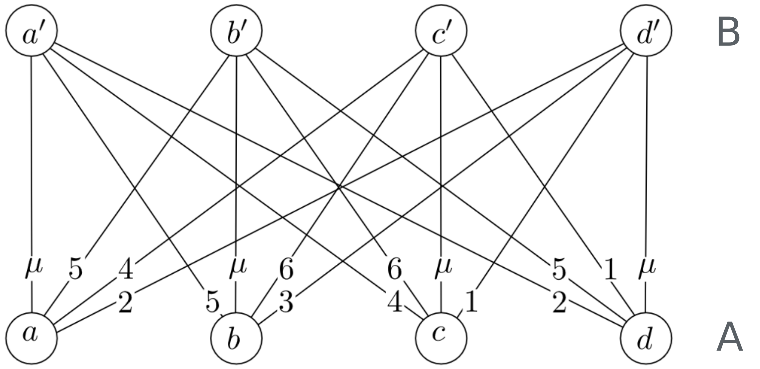
- Bicliques are better to work with
- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity

We've proven:

Theorem.
Minimal spanners for bicliques and cliques differ by constant factor



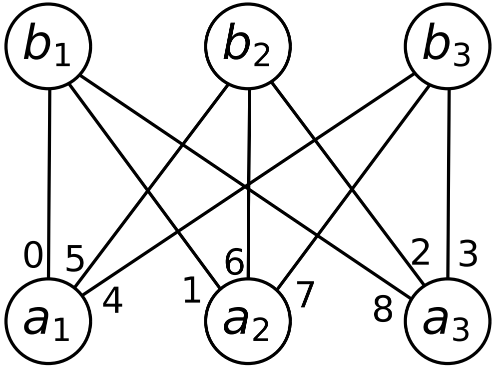
A temporal clique



The corresponding temporal biclique

(Extremal) Matchings

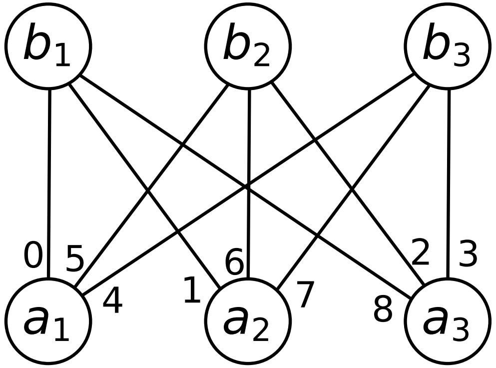
- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?



(Extremal) Matchings

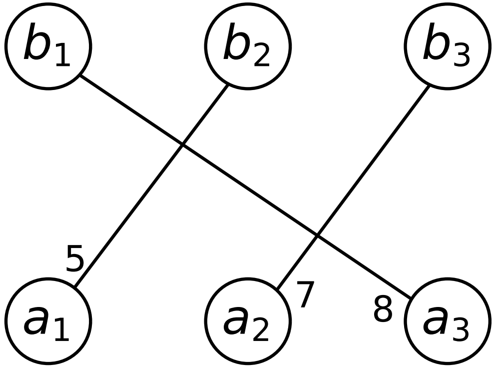
- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching



(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

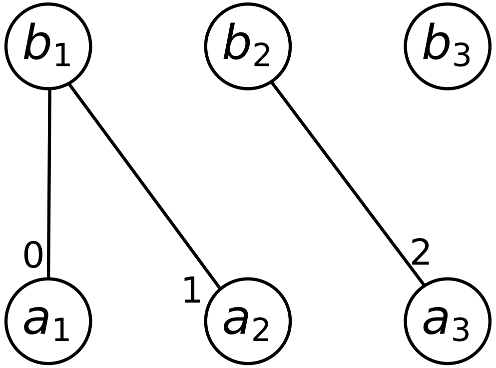


First and last edges form a complete matching

(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching

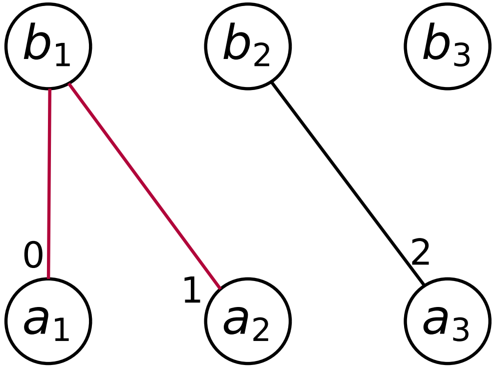


(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching

- Vertex is dismountable if it can delegate its reachability
  - Add path to proxy, delete vertex

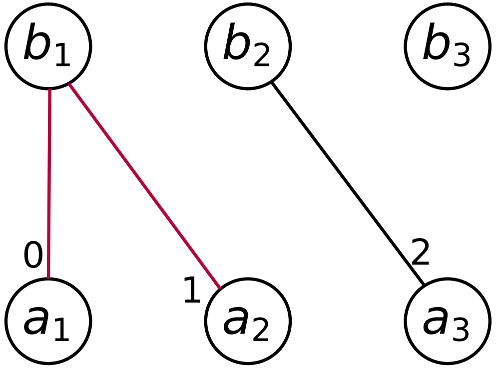


(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching

- Vertex is dismountable if it can delegate its reachability
  - Add path to proxy, delete vertex
- $\pi^- : A \sqcup B \to B \sqcup A$ is earliest (local) neighbor
- $\pi^+ : A \sqcup B \to B \sqcup A$ is latest (local) neighbor

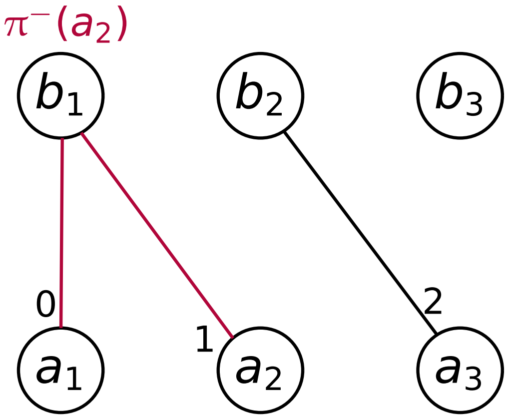


(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching

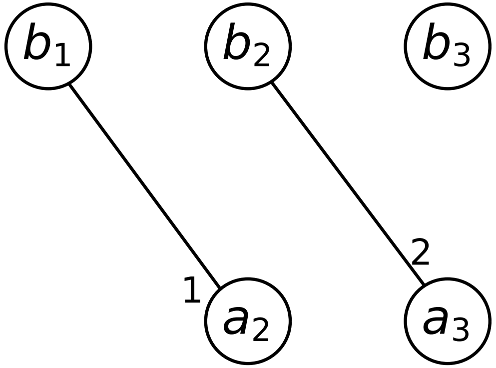
- Vertex is dismountable if it can delegate its reachability
  - Add path to proxy, delete vertex
- $\pi^- : A \sqcup B \to B \sqcup A$ is earliest (local) neighbor
- $\pi^+ : A \sqcup B \to B \sqcup A$ is latest (local) neighbor



Example of a dismountable vertex

(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?



First and last edges form a complete matching

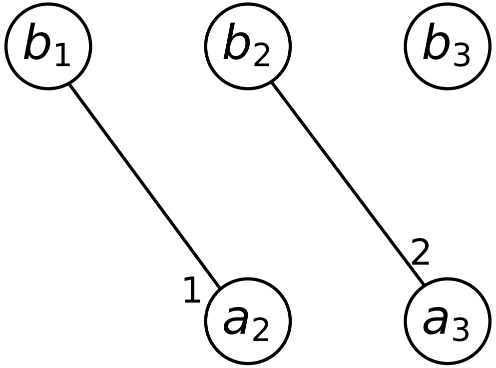
- Vertex is dismountable if it can delegate its reachability
  - Add path to proxy, delete vertex
- $\pi^- : A \sqcup B \to B \sqcup A$ is earliest (local) neighbor
- $\pi^+ : A \sqcup B \to B \sqcup A$ is latest (local) neighbor

(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching

- Vertex is dismountable if it can delegate its reachability
  - Add path to proxy, delete vertex
- $\pi^- : A \sqcup B \to B \sqcup A$ is earliest (local) neighbor
- $\pi^+ : A \sqcup B \to B \sqcup A$ is latest (local) neighbor
- Analogously holds for vertices in B with π^+



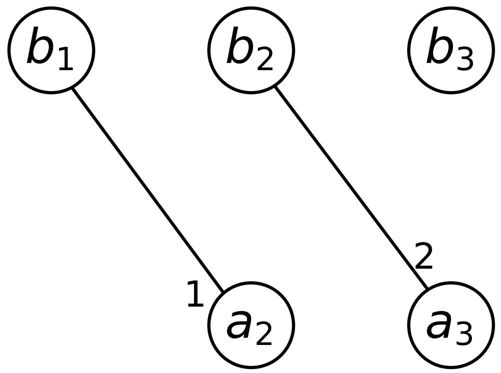
(Extremal) Matchings

- Temporal graph $G = (A \sqcup B, \lambda)$, A to B connectivity
- Any structural properties so far?

First and last edges form a complete matching

- Vertex is dismountable if it can delegate its reachability
  - Add path to proxy, delete vertex
- $\pi^- : A \sqcup B \to B \sqcup A$ is earliest (local) neighbor
- $\pi^+ : A \sqcup B \to B \sqcup A$ is latest (local) neighbor
- Analogously holds for vertices in B with π^+

Theorem. Dismount until we have a matching!



(Extremal) Matchings

- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of A and B
- Assume $|A| = |B|$

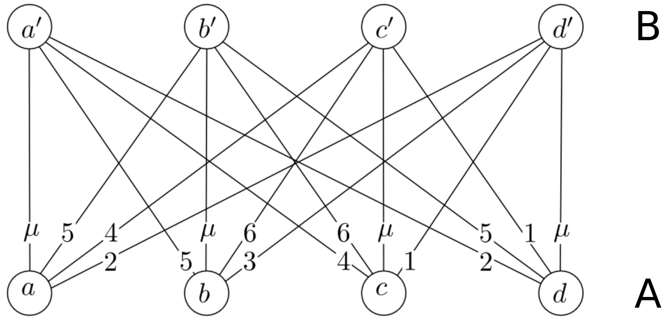
(Extremal) Matchings

- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of A and B
- Assume $|A| = |B|$

def $\texttt{biSpanner}(D = (A, B, \lambda))$:
  if $|A| = 1$ **then** **return** $E(D)$
  partition A into non-empty A_1, A_2 with $|A_1| = \lfloor |A|/2 \rfloor$
  $D_1 \leftarrow D[A_1 \sqcup B]$, $D_2 \leftarrow D[A_2 \sqcup B]$
  $(D_1^*, S_1^*) \leftarrow \texttt{dismount}(D_1)$, $(D_2^*, S_2^*) \leftarrow \texttt{dismount}(D_2)$
  return $\texttt{biSpanner}(D_1^*) \cup \texttt{biSpanner}(D_2^*) \cup S_1^* \cup S_2^*$

Algorithm for $O(n \log n)$ spanner

Utilize with divide and conquer!



Partition into A_1, A_2

(Extremal) Matchings

- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of *A* and *B*
- Assume $|A| = |B|$

Utilize with divide and conquer!

---

def `biSpanner`$(D = (A, B, \lambda))$:

  if $|A| = 1$ **then** **return** $E(D)$

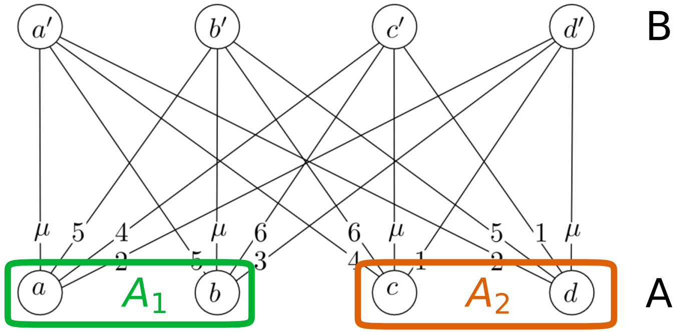
→ partition A into non-empty A_1, A_2 with $|A_1| = \lfloor |A|/2 \rfloor$

  $D_1 \leftarrow D[A_1 \sqcup B]$, $D_2 \leftarrow D[A_2 \sqcup B]$

  $(D_1^*, S_1^*) \leftarrow$ `dismount`(D_1), $(D_2^*, S_2^*) \leftarrow$ `dismount`(D_2)

  return `biSpanner`$(D_1^*) \cup$ `biSpanner`$(D_2^*) \cup S_1^* \cup S_2^*$

Algorithm for $O(n \log n)$ spanner



Partition into A_1, A_2

(Extremal) Matchings

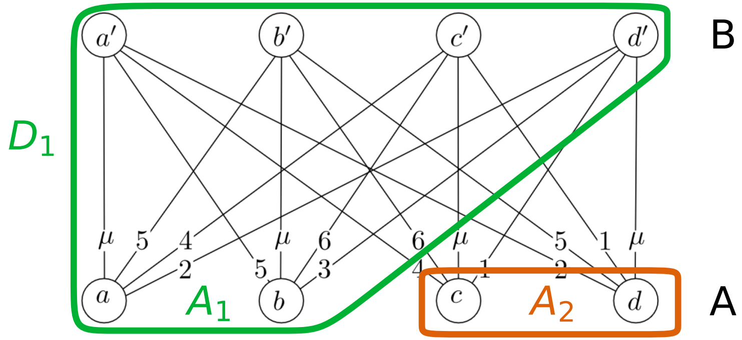
- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of *A* and *B*
- Assume $|A| = |B|$

Utilize with divide and conquer!

---

def biSpanner$(D = (A, B, \lambda))$:
    if $|A| = 1$ **then** **return** $E(D)$
    partition A into non-empty A_1, A_2 with $|A_1| = \lfloor |A|/2 \rfloor$
$\rightarrow$ $D_1 \leftarrow D[A_1 \sqcup B]$, $D_2 \leftarrow D[A_2 \sqcup B]$
    $(D_1^*, S_1^*) \leftarrow$ dismount(D_1), $(D_2^*, S_2^*) \leftarrow$ dismount(D_2)
    return biSpanner$(D_1^*) \cup$ biSpanner$(D_2^*) \cup S_1^* \cup S_2^*$

Algorithm for $O(n \log n)$ spanner



Partition into A_1, A_2

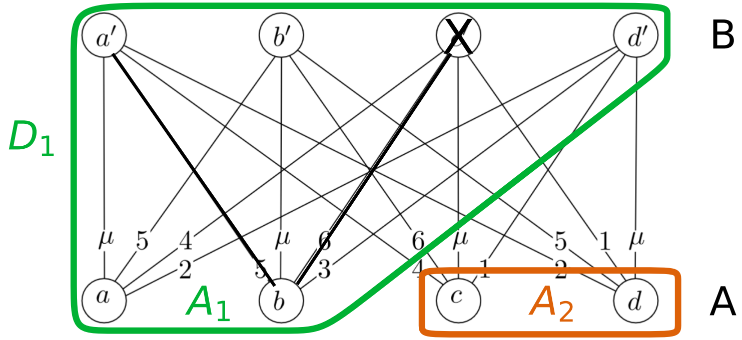
(Extremal) Matchings

- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of *A* and *B*
- Assume $|A| = |B|$

def biSpanner$(D = (A, B, \lambda))$:
    if $|A| = 1$ **then return** $E(D)$
    partition A into non-empty A_1, A_2 with $|A_1| = \lfloor |A|/2 \rfloor$
    $D_1 \leftarrow D[A_1 \sqcup B]$, $D_2 \leftarrow D[A_2 \sqcup B]$
$\rightarrow$ $(D_1^*, S_1^*) \leftarrow$ dismount(D_1), $(D_2^*, S_2^*) \leftarrow$ dismount(D_2)
    return biSpanner$(D_1^*) \cup$ biSpanner$(D_2^*) \cup S_1^* \cup S_2^*$

Algorithm for $O(n \log n)$ spanner

Utilize with divide and conquer!



Partition into A_1, A_2

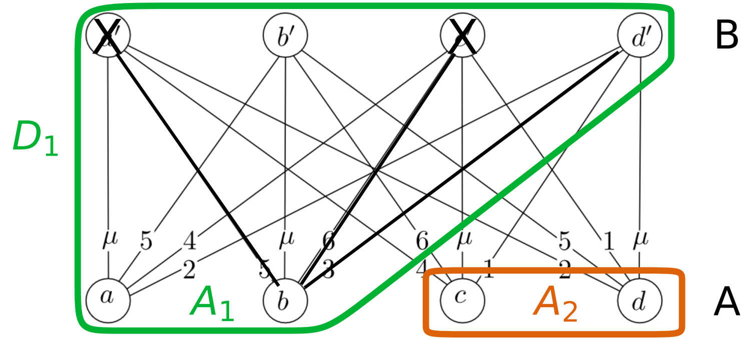
(Extremal) Matchings

- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of *A* and *B*
- Assume $|A| = |B|$

def biSpanner($D = (A, B, \lambda)$):
  if $|A| = 1$ **then return** $E(D)$
  partition A into non-empty A_1, A_2 with $|A_1| = \lfloor |A|/2 \rfloor$
  $D_1 \leftarrow D[A_1 \sqcup B]$, $D_2 \leftarrow D[A_2 \sqcup B]$
→ $(D_1^*, S_1^*) \leftarrow$ dismount(D_1), $(D_2^*, S_2^*) \leftarrow$ dismount(D_2)
  return biSpanner(D_1^*) $\cup$ biSpanner(D_2^*) $\cup S_1^* \cup S_2^*$

Algorithm for $O(n \log n)$ spanner

Utilize with divide and conquer!



Partition into A_1, A_2

(Extremal) Matchings

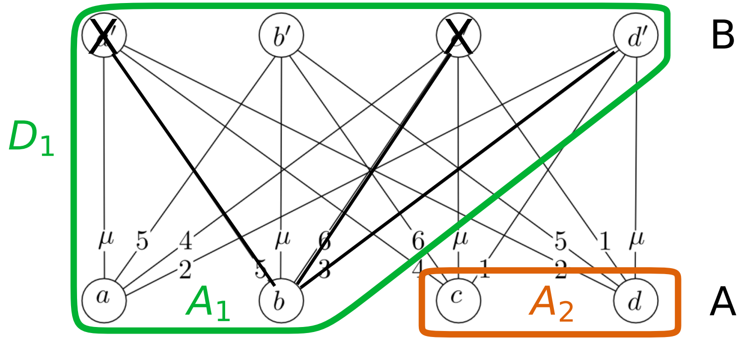
- First and last local edges form a matching!
- Take first and last edges into spanner
- Switch roles of A and B
- Assume $|A| = |B|$

def biSpanner$(D = (A, B, \lambda))$:
  if $|A| = 1$ **then** **return** $E(D)$
  partition A into non-empty A_1, A_2 with $|A_1| = \lfloor |A|/2 \rfloor$
  $D_1 \leftarrow D[A_1 \sqcup B]$, $D_2 \leftarrow D[A_2 \sqcup B]$
  $(D_1^*, S_1^*) \leftarrow$ dismount(D_1), $(D_2^*, S_2^*) \leftarrow$ dismount(D_2)
$\longrightarrow$ **return** biSpanner$(D_1^*) \cup$ biSpanner$(D_2^*) \cup S_1^* \cup S_2^*$

Algorithm for $O(n \log n)$ spanner

Utilize with divide and conquer!

- $S_1^*, S_2^* \in \Theta(n)$
- Results in $O(n \log n)$ spanner



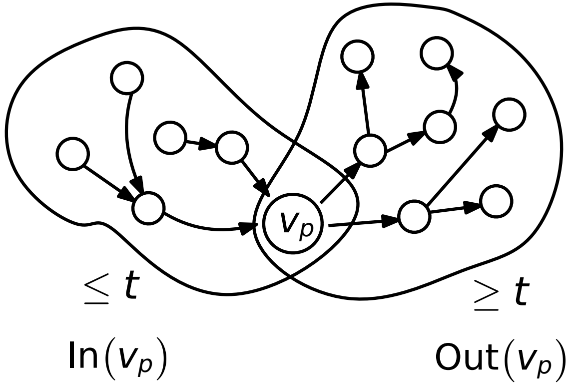
Partition into A_1, A_2

Partial Pivot Edges

- Previously: pivot vertices
  - Existence not guaranteed

Partial Pivot Edges

- Previously: pivot vertices
  - Existence not guaranteed

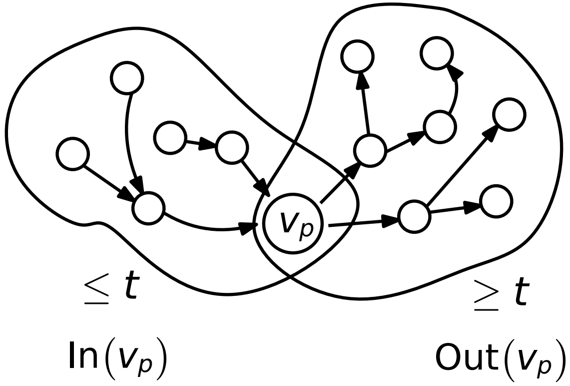


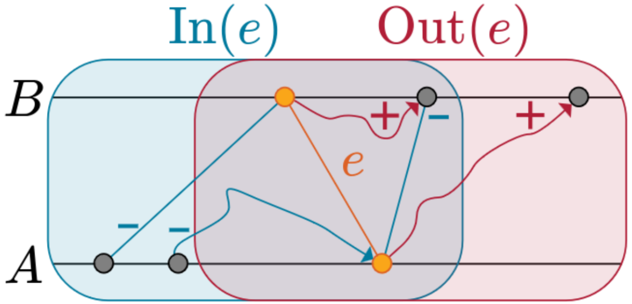
Partial Pivot Edges

- Previously: pivot vertices
  - Existence not guaranteed

Partial Pivot Edges

- Generalization of pivot vertices
- Vertices can reach or be reached by edge e
- Linear overlap suffices



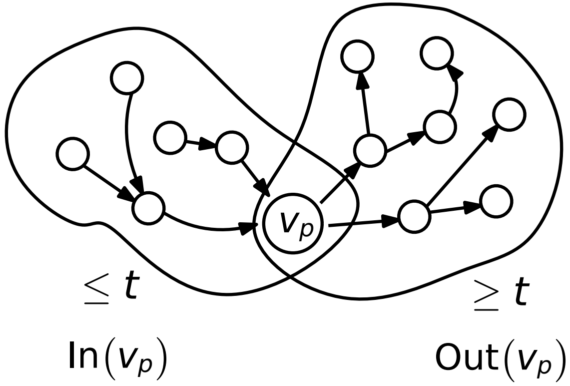


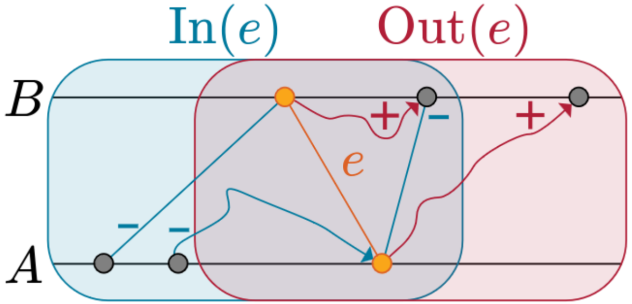
Partial Pivot Edges

- Previously: pivot vertices
  - Existence not guaranteed

Partial Pivot Edges

- Generalization of pivot vertices
- Vertices can reach or be reached by edge e
- Linear overlap suffices
- $\text{In}(e), \text{Out}(e)$: reachable vertices, via tree
  - Connect $\text{In}(e)$ to $\text{Out}(e)$ with $\mathcal{O}(|\text{In}(e)| + |\text{Out}(e)|)$ edges





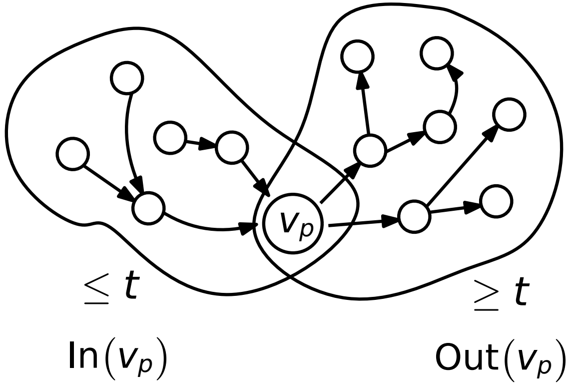
Partial Pivot Edges

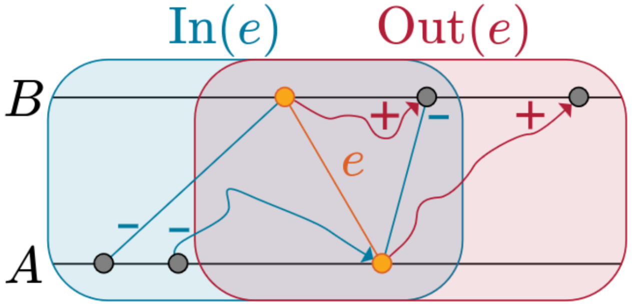
- Previously: pivot vertices
  - Existence not guaranteed

Partial Pivot Edges

- Generalization of pivot vertices
- Vertices can reach or be reached by edge e
- Linear overlap suffices
- $\text{In}(e), \text{Out}(e)$: reachable vertices, via tree
  - Connect $\text{In}(e)$ to $\text{Out}(e)$ with $\mathcal{O}(|\text{In}(e)| + |\text{Out}(e)|)$ edges

$\text{In}(e) \cap \text{Out}(e) \in \Omega(n) \Rightarrow$ reduce!





Partial Pivot Edges

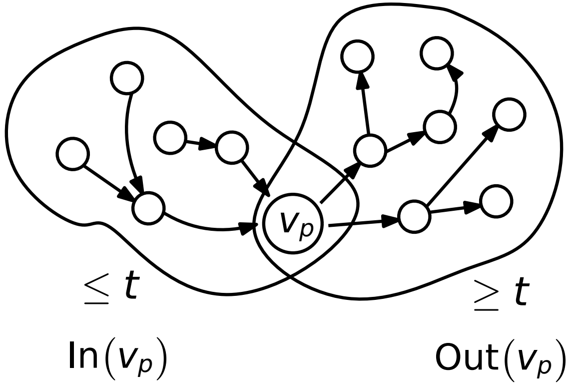
- Previously: pivot vertices
  - Existence not guaranteed

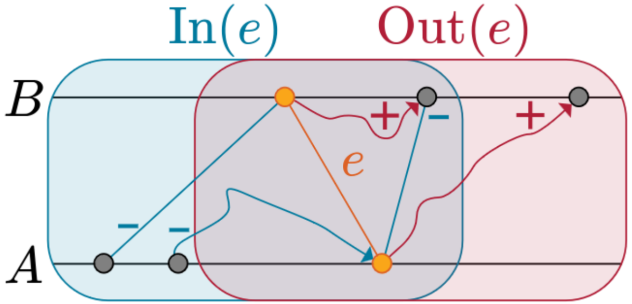
Partial Pivot Edges

- Generalization of pivot vertices
- Vertices can reach or be reached by edge e
- Linear overlap suffices
- $\text{In}(e), \text{Out}(e)$: reachable vertices, via tree
  - Connect $\text{In}(e)$ to $\text{Out}(e)$ with $\mathcal{O}(|\text{In}(e)| + |\text{Out}(e)|)$ edges

$\text{In}(e) \cap \text{Out}(e) \in \Omega(n) \Rightarrow$ reduce!

Is there a graph that has no pivot edges?





Partial Pivot Edges

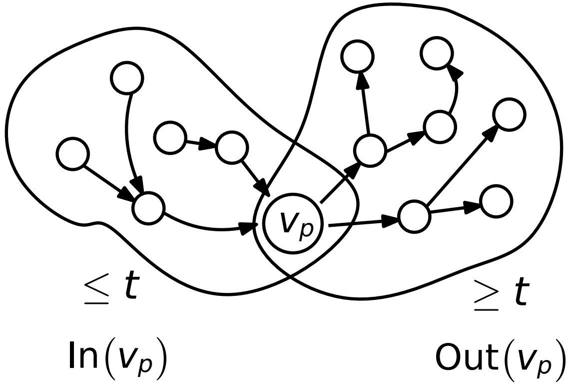
- Previously: pivot vertices
  - Existence not guaranteed

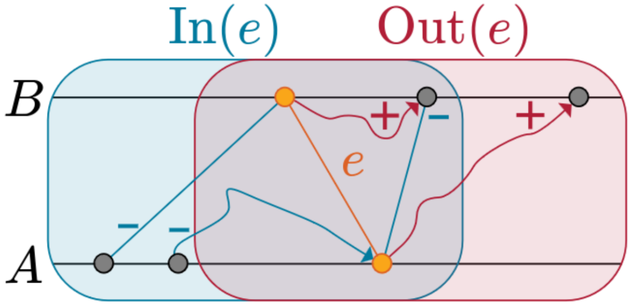
Partial Pivot Edges

- Generalization of pivot vertices
- Vertices can reach or be reached by edge e
- Linear overlap suffices
- $\text{In}(e), \text{Out}(e)$: reachable vertices, via tree
  - Connect $\text{In}(e)$ to $\text{Out}(e)$ with $\mathcal{O}(|\text{In}(e)| + |\text{Out}(e)|)$ edges

$\text{In}(e) \cap \text{Out}(e) \in \Omega(n) \Rightarrow$ reduce!

Is there a graph that has no pivot edges? Yes

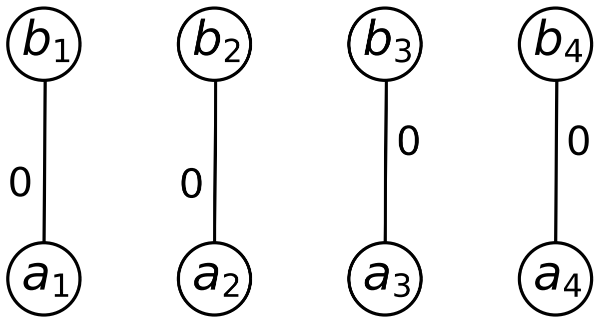




Shifted Matchings and Reverted Edges

Shifted Matching Graph

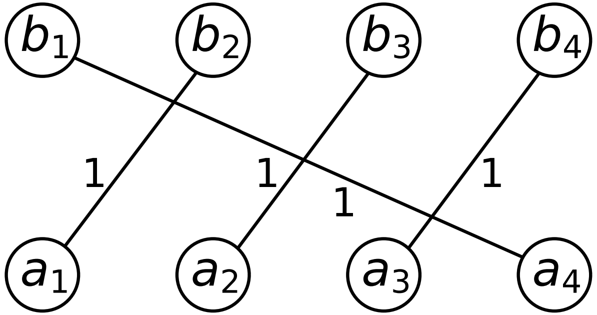
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

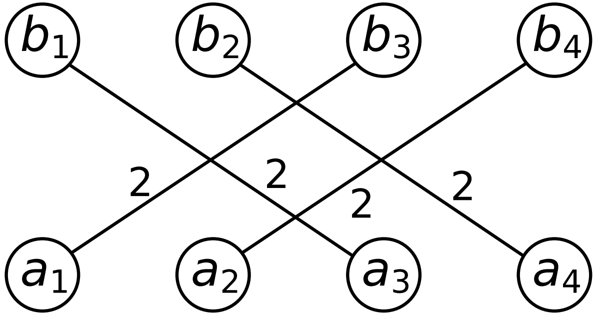
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

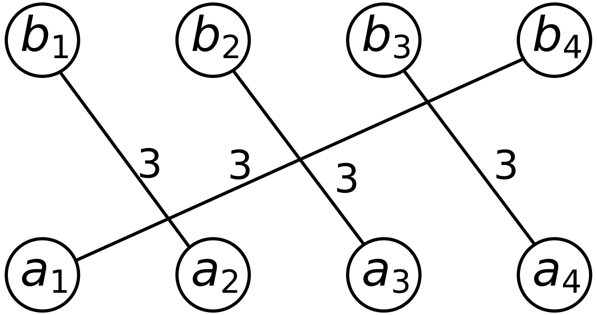
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

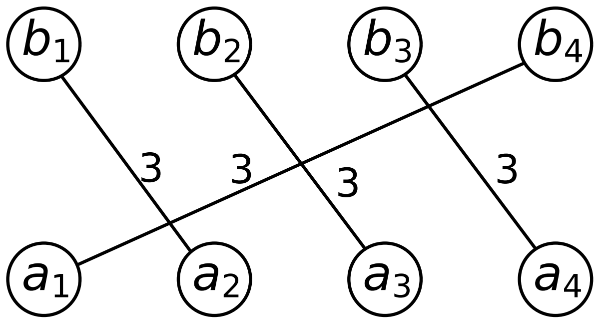
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

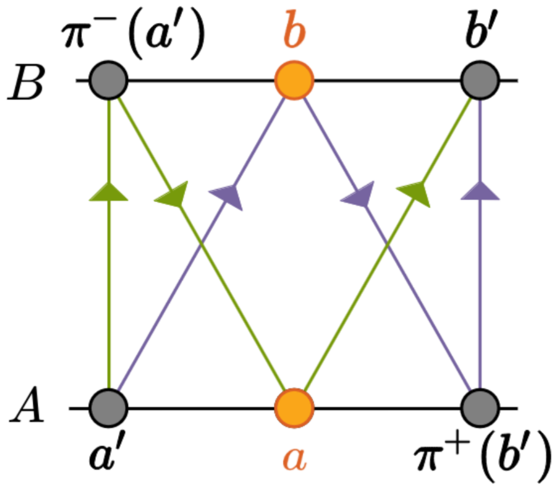
Shifted Matching Graph

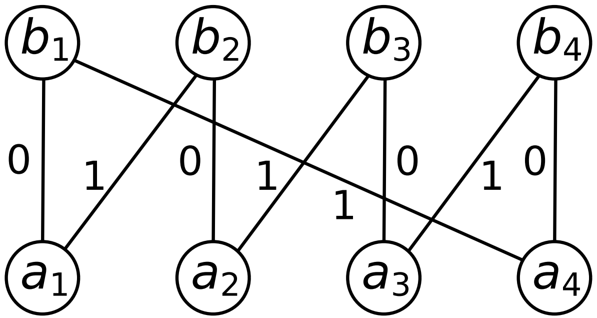
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching
- $\text{In}(\{a_i, b_j\}) \cap \text{Out}(\{a_i, b_j\}) = \{a_i, b_j\}$
  - All nodes can be reached either before an edge or after an edge



Shifted Matchings and Reverted Edges

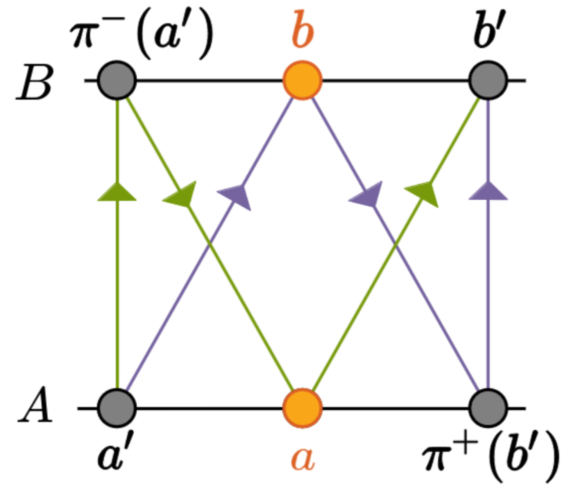
- For a fixed edge $e = \{a, b\}$, an edge $\{a', b'\}$ is reverted if the green or purple path is temporal

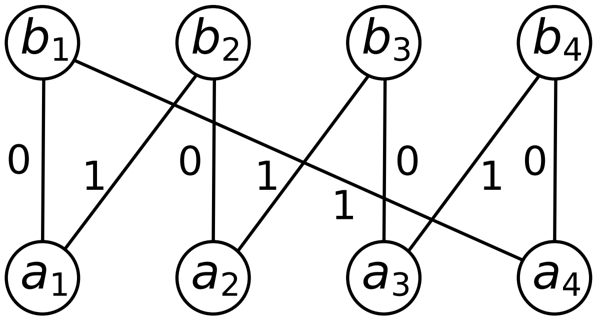




Shifted Matchings and Reverted Edges

- For a fixed edge $e = \{a, b\}$, an edge $\{a', b'\}$ is reverted if the green or purple path is temporal
- We get a spanner of size $4n - 4 + |\,\mathrm{NotRev}_e\,|$

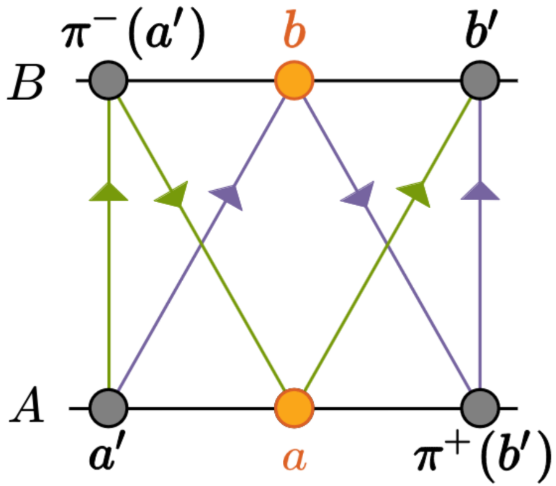


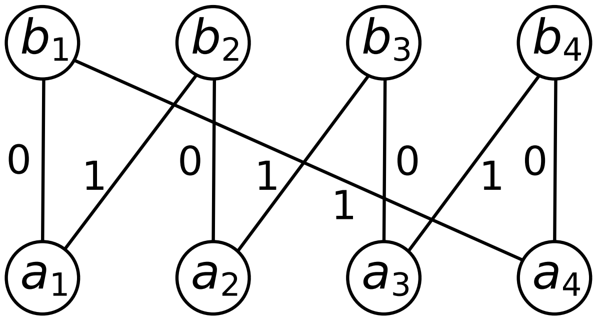


Shifted Matchings and Reverted Edges

- For a fixed edge $e = \{a, b\}$, an edge $\{a', b'\}$ is reverted if the green or purple path is temporal
- We get a spanner of size $4n - 4 + |\text{NotRev}_e|$

$\text{NotRev}_e \in \mathcal{O}(n) \Rightarrow$ Solved!



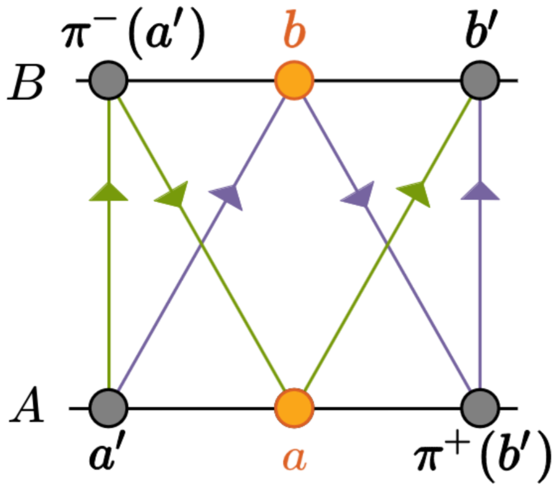


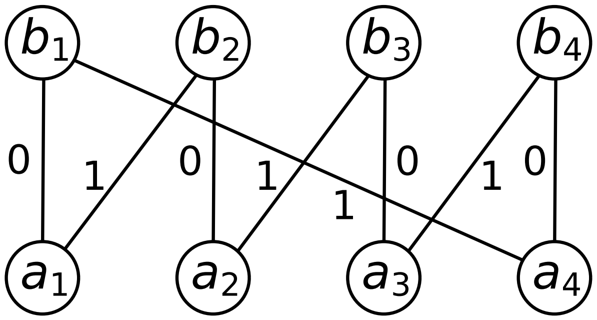
Shifted Matchings and Reverted Edges

- For a fixed edge $e = \{a, b\}$, an edge $\{a', b'\}$ is reverted if the green or purple path is temporal
- We get a spanner of size $4n - 4 + |\text{NotRev}_e|$

$\text{NotRev}_e \in \mathcal{O}(n) \Rightarrow$ Solved!

- Edges with small $|\text{NotRev}_e|$ either early or late



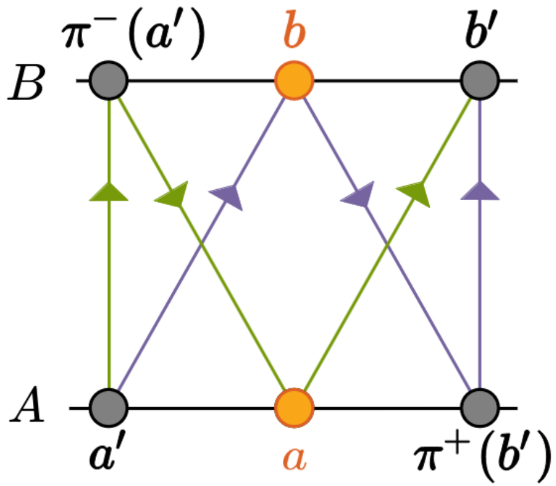


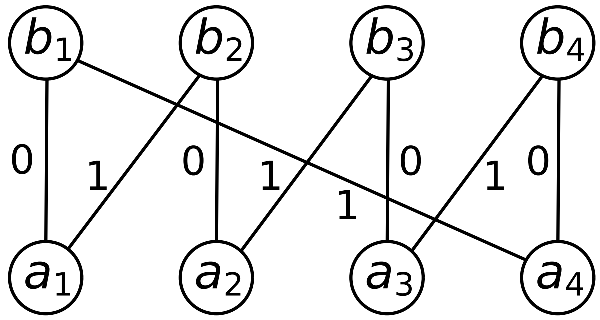
Shifted Matchings and Reverted Edges

- For a fixed edge $e = \{a, b\}$, an edge $\{a', b'\}$ is reverted if the green or purple path is temporal
- We get a spanner of size $4n - 4 + |\text{NotRev}_e|$

$\text{NotRev}_e \in \mathcal{O}(n) \Rightarrow$ Solved!

- Edges with small $|\text{NotRev}_e|$ either early or late
  - That's complimentary to pivot edges!





Shifted Matchings and Reverted Edges

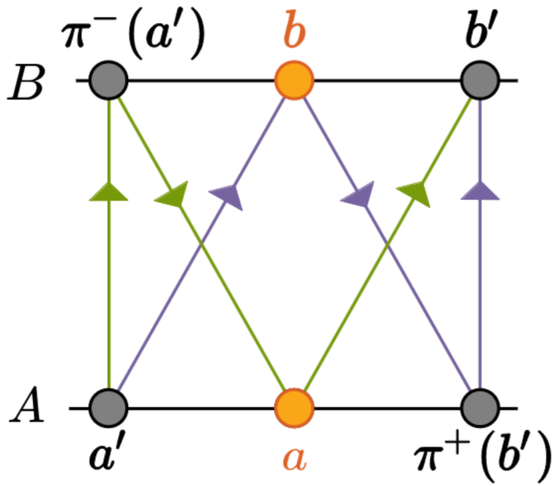
- For a fixed edge $e = \{a, b\}$, an edge $\{a', b'\}$ is reverted if the green or purple path is temporal
- We get a spanner of size $4n - 4 + |\text{NotRev}_e|$

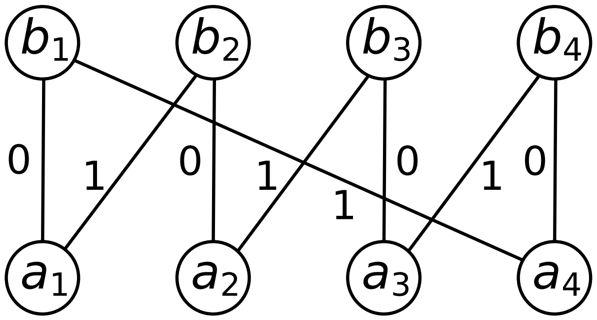
$\text{NotRev}_e \in \mathcal{O}(n) \Rightarrow$ Solved!

- Edges with small $|\text{NotRev}_e|$ either early or late
  - That's complimentary to pivot edges!

Shifted Matching graph *(SM)*

- 0 or $n - 1$ labelled edges have $\text{NotRev}_e = \emptyset$





Composed Graphs and Spanners

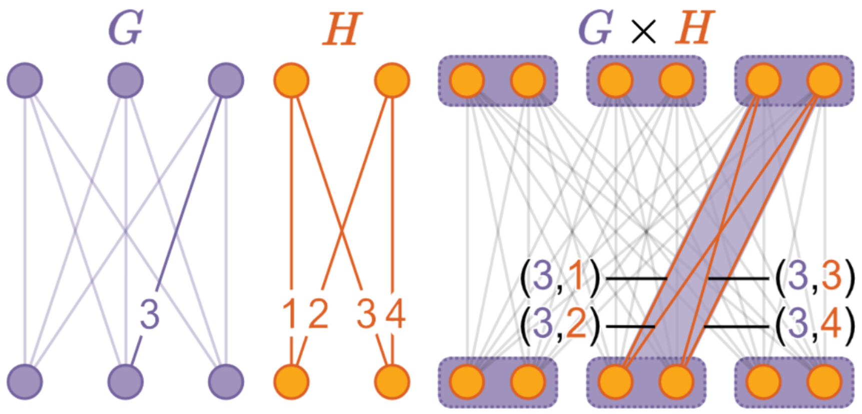
- Linear size pivot sets or many reverted edges imply linear spanners
- Unsolved: $\log(n)$ or $\sqrt{n}$ pivot edges

Composed Graphs and Spanners

- Linear size pivot sets or many reverted edges imply linear spanners
- Unsolved: $\log(n)$ or $\sqrt{n}$ pivot edges

Product graphs

- Tensor product of outer (G) and inner (H) graph

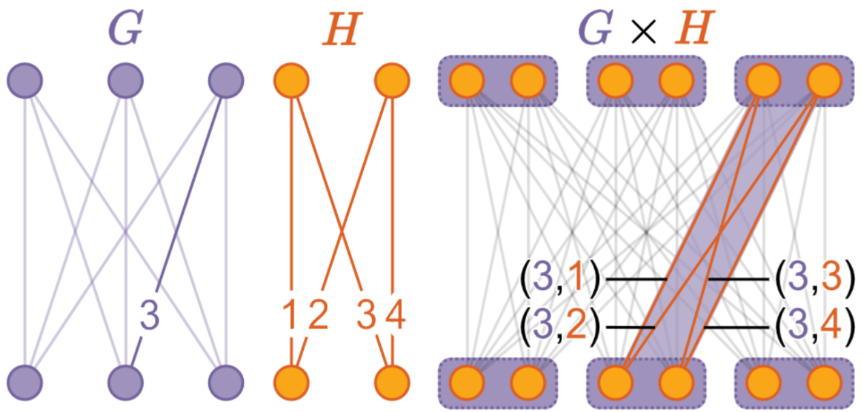


Composed Graphs and Spanners

- Linear size pivot sets or many reverted edges imply linear spanners
- Unsolved: $\log(n)$ or $\sqrt{n}$ pivot edges

Product graphs

- Tensor product of outer (G) and inner (H) graph
- Can interpolate between parameters
  - For shifted matching graph of size $\Theta(\sqrt{n})$, the graph is not solvable by partial edge pivots or reverted edges

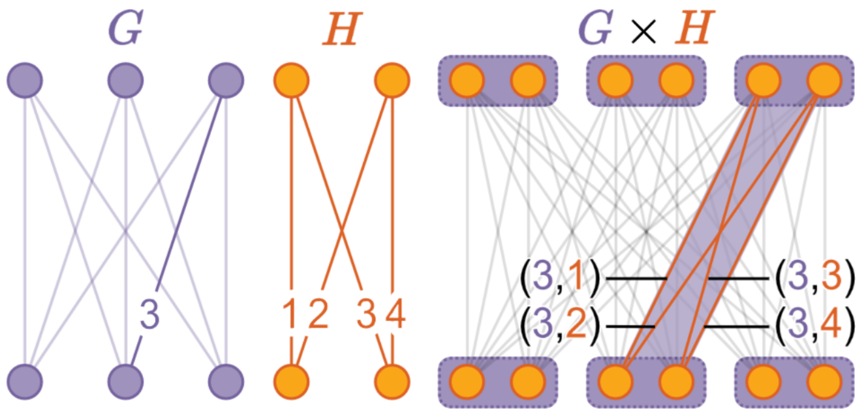


Composed Graphs and Spanners

- Linear size pivot sets or many reverted edges imply linear spanners
- Unsolved: $\log(n)$ or $\sqrt{n}$ pivot edges

Product graphs

- Tensor product of outer (G) and inner (H) graph
- Can interpolate between parameters
  - For shifted matching graph of size $\Theta(\sqrt{n})$, the graph is not solvable by partial edge pivots or reverted edges
- Compose spanner of size $|S_G| n_H + |S_H| n_G$
  - Also works if each edge is expanded by a different inner graph



Outlook

What we did

- First step in years for $\Theta(n)$ spanners in all temporal cliques

Outlook

What we did

- First step in years for $\Theta(n)$ spanners in all temporal cliques
- Full equivalence of cliques and bicliques

Outlook

What we did

- First step in years for $\Theta(n)$ spanners in all temporal cliques
- Full equivalence of cliques and bicliques
- Graphs often can be heavily reduced
  - No extremal matching? $\Rightarrow$ reduce!
  - $\text{In}(e) \cap \text{Out}(e) \in \Omega(n)$? $\Rightarrow$ reduce!
  - $\text{NotRev}_e \in \mathcal{O}(n)$? $\Rightarrow$ solved!
  - Composed structure? $\Rightarrow$ recurse!

Outlook

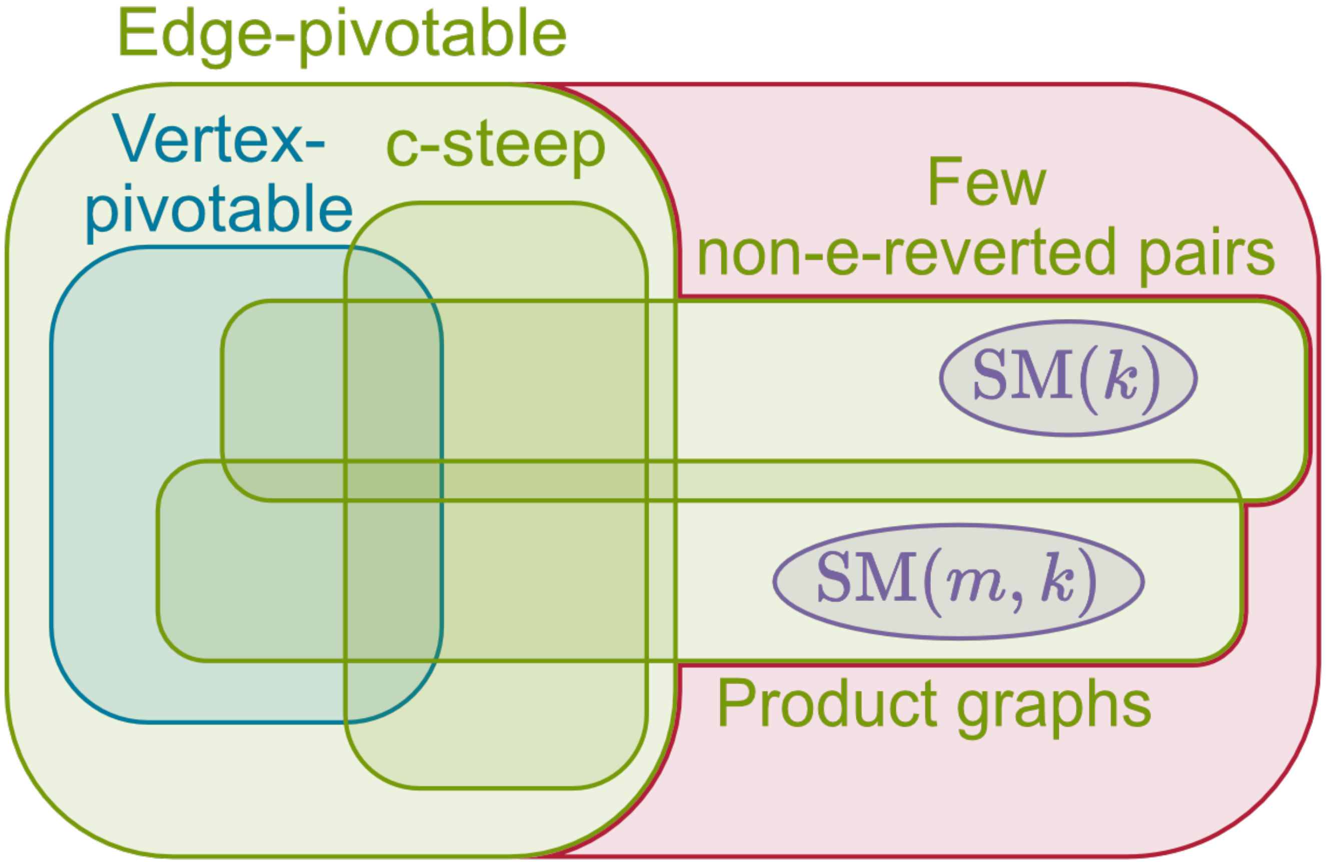
What we did

- First step in years for $\Theta(n)$ spanners in all temporal cliques
- Full equivalence of cliques and bicliques
- Graphs often can be heavily reduced
  - No extremal matching? $\Rightarrow$ reduce!
  - $\text{In}(e) \cap \text{Out}(e) \in \Omega(n)$? $\Rightarrow$ reduce!
  - $\text{NotRev}_e \in \mathcal{O}(n)$? $\Rightarrow$ solved!
  - Composed structure? $\Rightarrow$ recurse!

Future Work

- Tweaking structures may give unsolved graphs
- Allow for weaker conditions
- Compose techniques (on a local level)

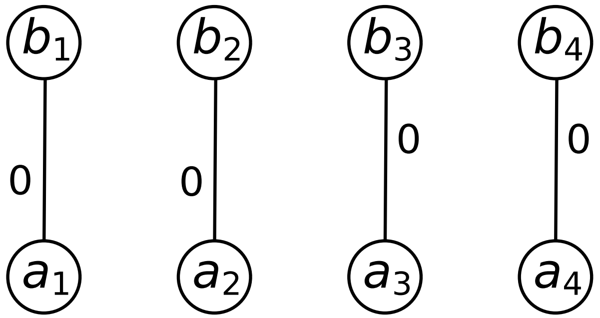
What did we solve?



Shifted Matchings and Reverted Edges

Shifted Matching Graph

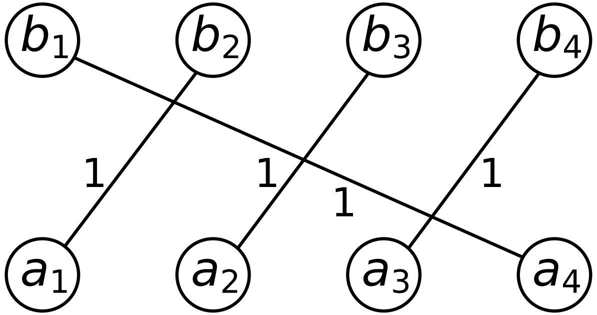
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

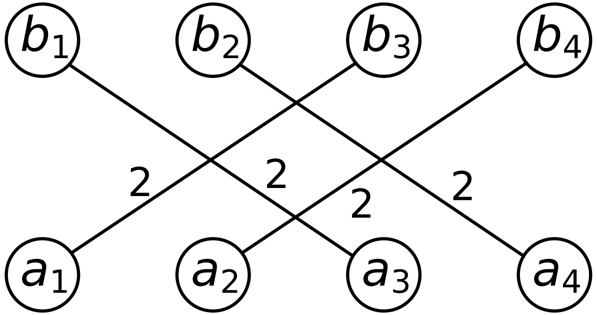
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

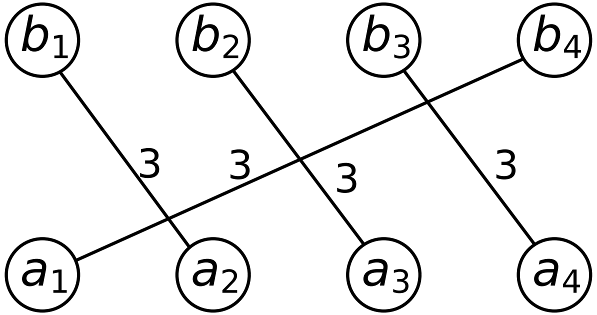
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

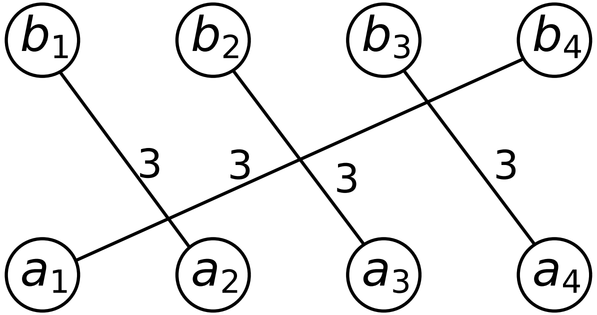
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching



Shifted Matchings and Reverted Edges

Shifted Matching Graph

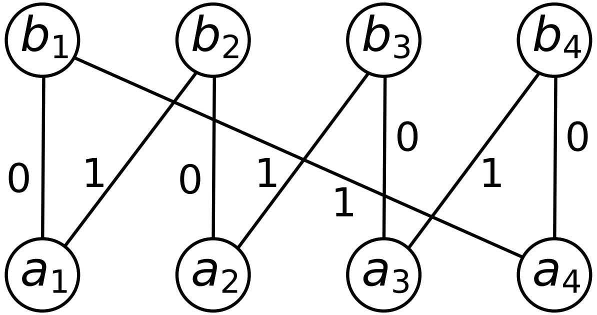
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching
- $\text{In}(\{a_i, b_j\}) \cap \text{Out}(\{a_i, b_j\}) = \{a_i, b_j\}$
  - All nodes can be reached either before an edge or after an edge

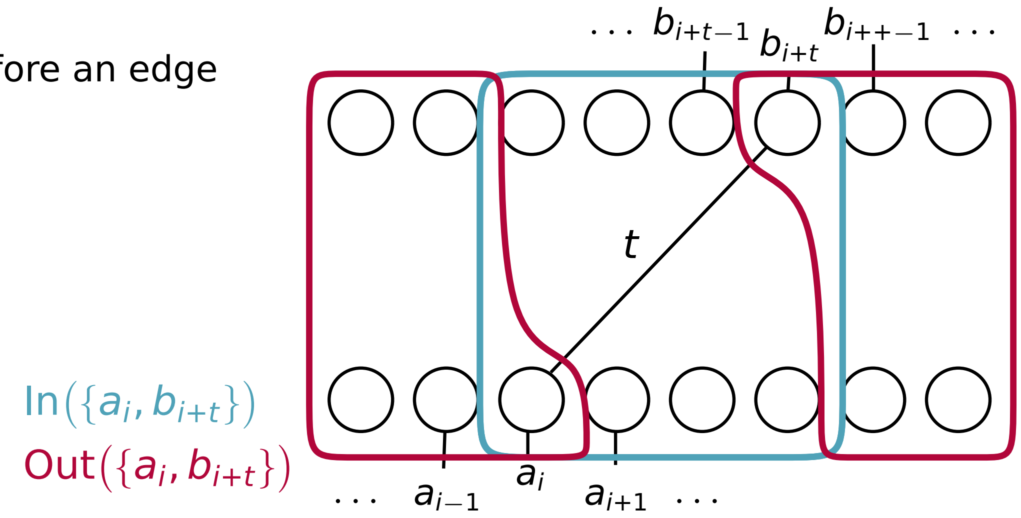


Shifted Matchings and Reverted Edges

Shifted Matching Graph

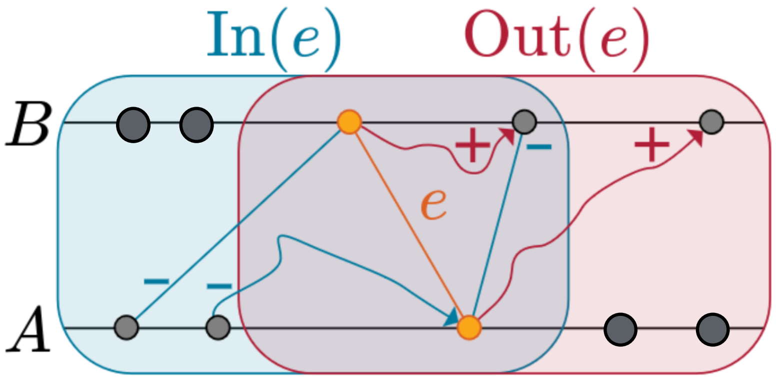
- Label $\lambda(\{a_i, b_j\}) := j - i \mod n$
- Edges of each timestamp form a matching
- $\mathrm{In}(\{a_i, b_j\}) \cap \mathrm{Out}(\{a_i, b_j\}) = \{a_i, b_j\}$
  - All nodes can be reached either before an edge or after an edge





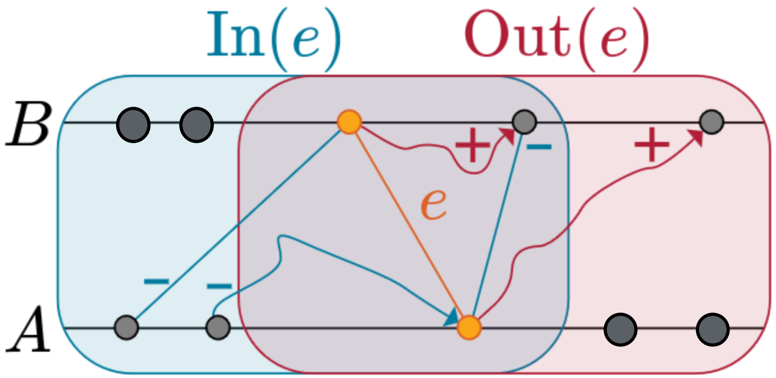
Partial Pivot Edges

- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$



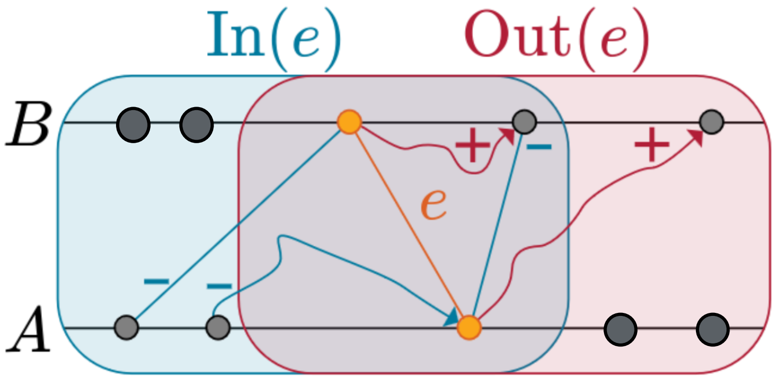
Partial Pivot Edges

- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$



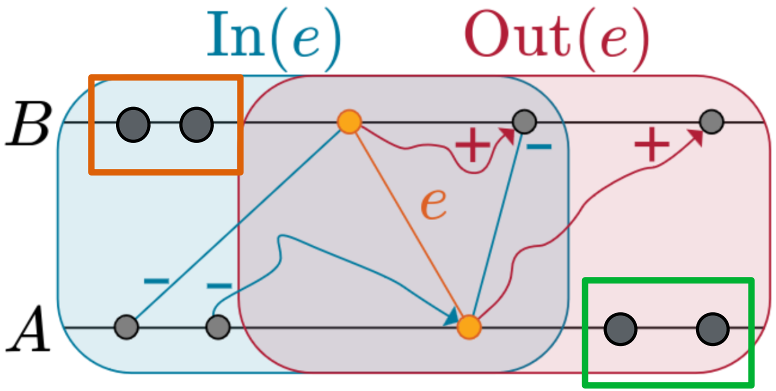
Partial Pivot Edges

- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
  - Connect $\mathrm{In}(e)$ to $\mathrm{Out}(e)$



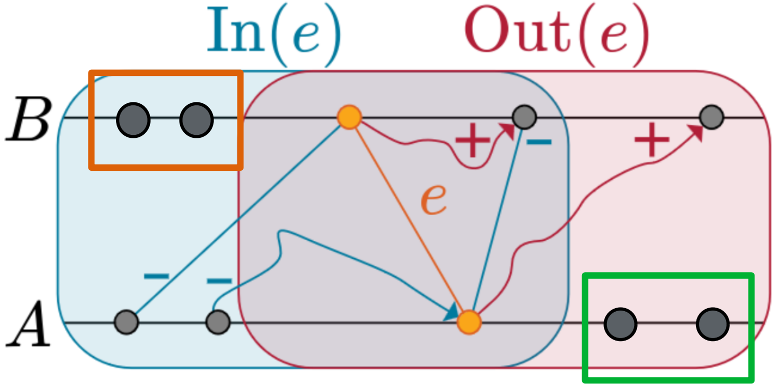
Partial Pivot Edges

- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
  - Connect $\mathrm{In}(e)$ to $\mathrm{Out}(e)$
  - Create instances $G[(A \setminus \mathrm{In}(e)) \sqcup B]$ $G[A \sqcup (B \setminus \mathrm{Out}(e))]$



Partial Pivot Edges

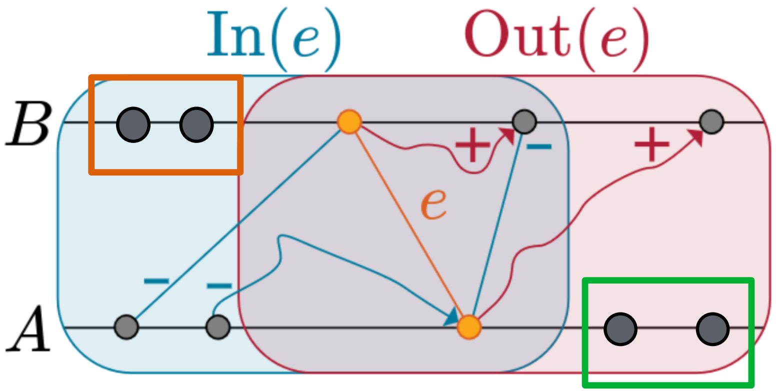
- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
  - Connect $\mathrm{In}(e)$ to $\mathrm{Out}(e)$
  - Create instances $G[(A \setminus \mathrm{In}(e)) \sqcup B]$ $G[A \sqcup (B \setminus \mathrm{Out}(e))]$
  - Size of $n - |\mathrm{In}(e)|/2$ and $n - |\mathrm{Out}(e)|/2$ vertices per side



Partial Pivot Edges

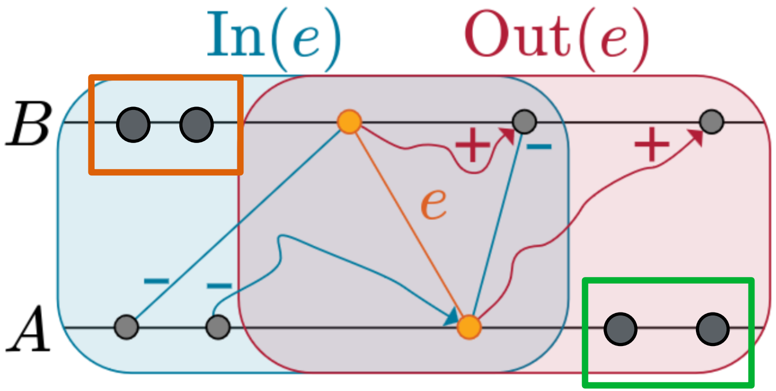
- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
  - Connect $\mathrm{In}(e)$ to $\mathrm{Out}(e)$
  - Create instances $G[(A \setminus \mathrm{In}(e)) \sqcup B]$ $G[A \sqcup (B \setminus \mathrm{Out}(e))]$
  - Size of $n - |\mathrm{In}(e)|/2$ and $n - |\mathrm{Out}(e)|/2$ vertices per side
- If $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, include constant many edges for each reduced vertex
- $\mathrm{In}(e) \cap \mathrm{Out}(e)$ called pivot set of e

$\mathrm{In}(e) \cap \mathrm{Out}(e) \in \Omega(n) \Rightarrow$ reduce!



Partial Pivot Edges

- $\mathrm{In}(e), \mathrm{Out}(e)$ can be connected with $\mathcal{O}(|\mathrm{In}(e)| + |\mathrm{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
  - Connect $\mathrm{In}(e)$ to $\mathrm{Out}(e)$
  - Create instances $G[(A \setminus \mathrm{In}(e)) \sqcup B]$ $G[A \sqcup (B \setminus \mathrm{Out}(e))]$
  - Size of $n - |\mathrm{In}(e)|/2$ and $n - |\mathrm{Out}(e)|/2$ vertices per side
- If $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, include constant many edges for each reduced vertex
- $\mathrm{In}(e) \cap \mathrm{Out}(e)$ called pivot set of e

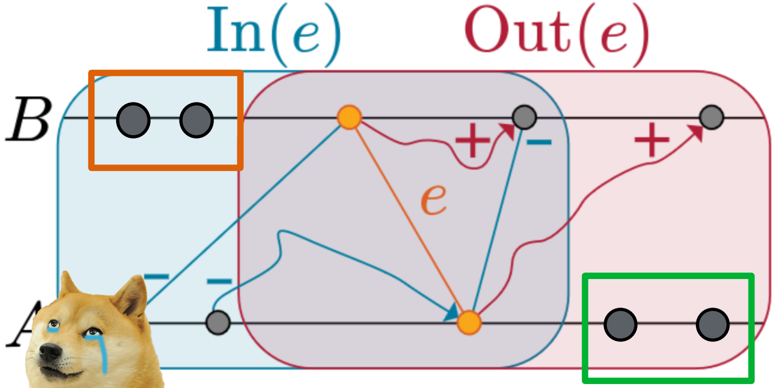


$\mathrm{In}(e) \cap \mathrm{Out}(e) \in \Omega(n) \Rightarrow$ reduce!

Is there a graph that has no pivot edges?

Partial Pivot Edges

- $\text{In}(e), \text{Out}(e)$ can be connected with $\mathcal{O}(|\text{In}(e)| + |\text{Out}(e)|)$ edges
- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\text{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\text{Out}(e)|}{2}\right) + 2|\text{In}(e)| + 2|\text{Out}(e)| - 3 \right)$
  - Connect $\text{In}(e)$ to $\text{Out}(e)$
  - Create instances $G[(A \setminus \text{In}(e)) \sqcup B]$ $G[A \sqcup (B \setminus \text{Out}(e))]$
  - Size of $n - |\text{In}(e)|/2$ and $n - |\text{Out}(e)|/2$ vertices per side
- If $|\text{In}(e) \cap \text{Out}(e)| \in \Omega(n)$, include constant many edges for each reduced vertex
- $\text{In}(e) \cap \text{Out}(e)$ called pivot set of e



$\text{In}(e) \cap \text{Out}(e) \in \Omega(n) \Rightarrow$ reduce!

Is there a graph that has no pivot edges? Yes

Partial Pivot Edges

- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
- If $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, reduce the graph

Partial Pivot Edges

- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
- If $|\,\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, reduce the graph

Forbidden structures

- Steepness

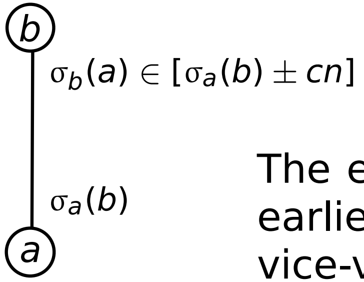
Partial Pivot Edges

- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
- If $|\,\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, reduce the graph

Forbidden structures

- Steepness

$|\{v' \in N(v) \mid i \le \sigma_{v'}(v) \le j\}| < j - i + 2cn$



The edge cannot be much earlier for a than for b (and vice-versa)!

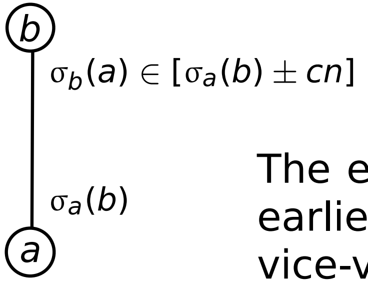
Partial Pivot Edges

- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
- If $|\,\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, reduce the graph

Forbidden structures

- Steepness
- (Label spread)
- (Activity width)

$|\{v' \in N(v) \mid i \leq \sigma_{v'}(v) \leq j\}| < j - i + 2cn$



The edge cannot be much earlier for a than for b (and vice-versa)!

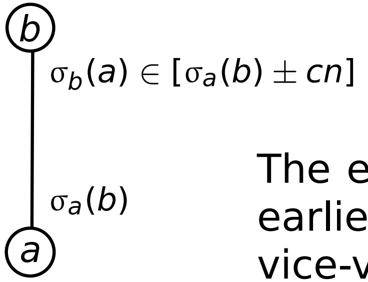
Partial Pivot Edges

- There is a spanner of size: $\min_{e \in A \otimes B} \left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3 \right)$
- If $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, reduce the graph

Forbidden structures

- Steepness
- (Label spread)
- (Activity width)

$|\{v' \in N(v) \mid i \leq \sigma_{v'}(v) \leq j\}| < j - i + 2cn$



The edge cannot be much earlier for a than for b (and vice-versa)!

Is there a graph that has no pivot edges?

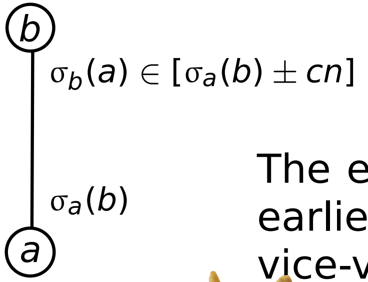
Partial Pivot Edges

- There is a spanner of size: $\min_{e \in A \otimes B}\left(\mathcal{D}\left(n - \frac{|\mathrm{In}(e)|}{2}\right) + \mathcal{D}\left(n - \frac{|\mathrm{Out}(e)|}{2}\right) + 2|\mathrm{In}(e)| + 2|\mathrm{Out}(e)| - 3\right)$
- If $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \in \Omega(n)$, reduce the graph

Forbidden structures

- Steepness
- (Label spread)
- (Activity width)

$|\{v' \in N(v) \mid i \leq \sigma_{v'}(v) \leq j\}| < j - i + 2cn$



The edge cannot be much earlier for a than for b (and vice-versa)!

Is there a graph that has no pivot edges?

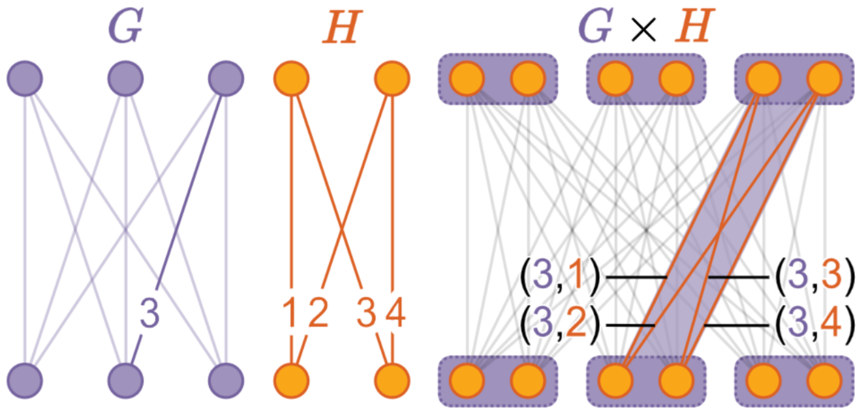
Yes

Composed Graphs—Construction

- Shifted Matchings counterpart for Partial Pivot Edges
- in between: $\log(n)$ or $\sqrt{n}$ pivots / non-reverted edges

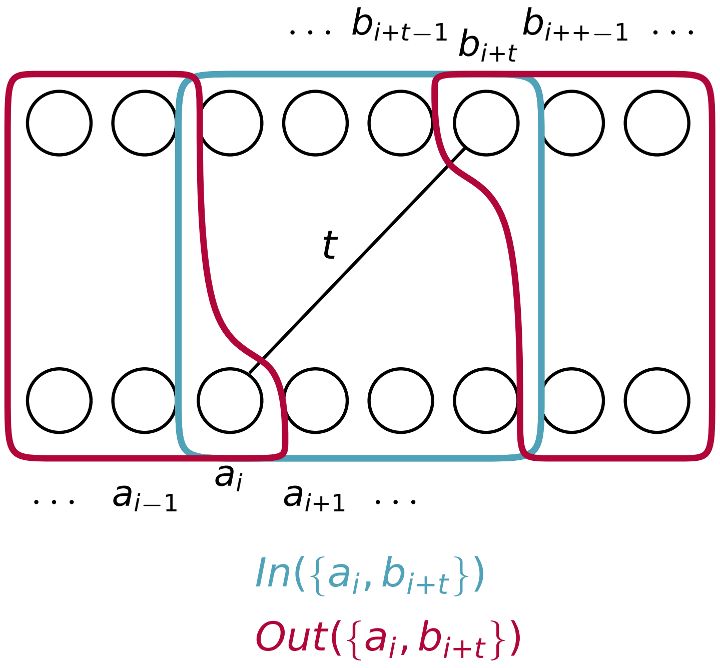
Product graphs

- tensor product of outer (G) and inner (H) graph
- for any $f : \mathbb{N} \to \mathbb{N}^+$ with $f(n) \in \mathcal{O}(n)$, can construct bi-clique s.t. for all $e \in A \otimes B$:
  - $\text{In}(e) \cap \text{Out}(e) \in \mathcal{O}(f(n))$
  - $|\text{NotRev}_e| \in \Omega(n \cdot f(n))$
- can interpolate between parameters



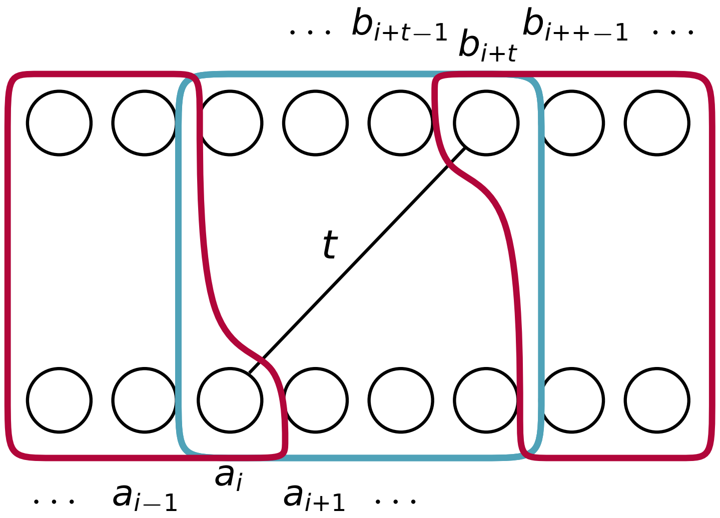
Composed Graphs—Construction

- $SM(m, k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\,\mathrm{In}(e) \cap \mathrm{Out}(e)| \leq 2k$
  - for any e: $|\,\mathrm{NotRev}_e\,| \geq (m-1)\binom{k}{2}$



Composed Graphs—Construction

- $SM(m,k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\text{In}(e) \cap \text{Out}(e)| \leq 2k$
  - for any e: $|\text{NotRev}_e| \geq (m-1)\binom{k}{2}$
- let $(g_1, h_1)(g_2, h_2) \ldots (g_\ell, h_\ell)$ be temporal
  - $g_1 g_2 \ldots g_\ell$ is temporal

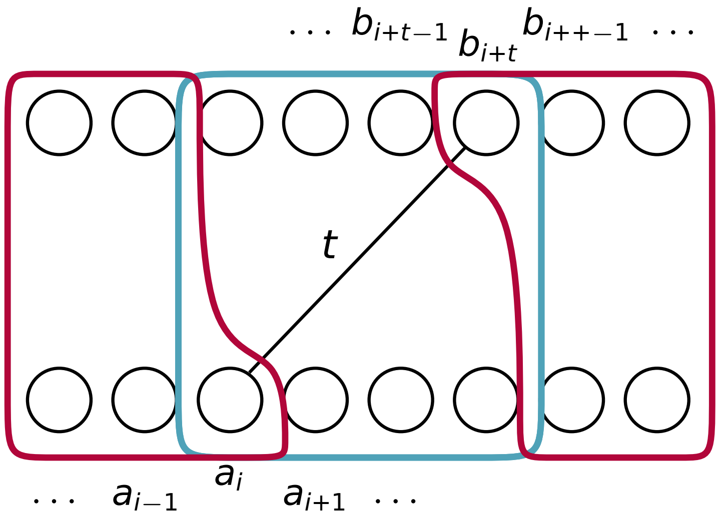


$In(\{a_i, b_{i+t}\})$

$Out(\{a_i, b_{i+t}\})$

Composed Graphs—Construction

- $SM(m, k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\text{In}(e) \cap \text{Out}(e)| \leq 2k$
  - for any e: $|\text{NotRev}_e| \geq (m-1)\binom{k}{2}$
- let $(g_1, h_1)(g_2, h_2)\ldots(g_\ell, h_\ell)$ be temporal
  - $g_1 g_2 \ldots g_\ell$ is temporal
- let $\phi(g, h) = g$ for all $(g, h) \in G \otimes H$
  - $\phi(\text{In}(e)) \subseteq In(\phi(e))$
  - $\phi(\text{In}(e) \cap \text{Out}(e)) \subseteq In(\phi(e)) \cap Out(\phi(e))$
  - $In(\{a_i, b_j\}) \cap Out(\{a_i, b_j\}) = \{a_i, b_j\}$ in SM
  - can only reach two bags

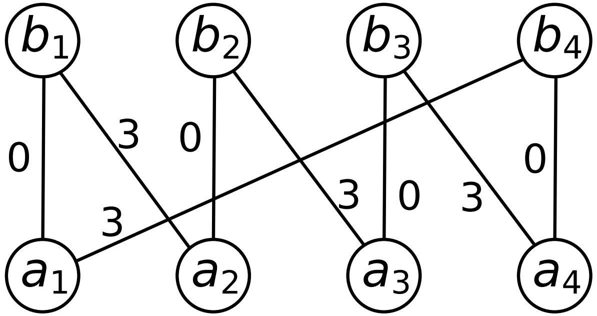


$In(\{a_i, b_{i+t}\})$

$Out(\{a_i, b_{i+t}\})$

Composed Graphs—Construction

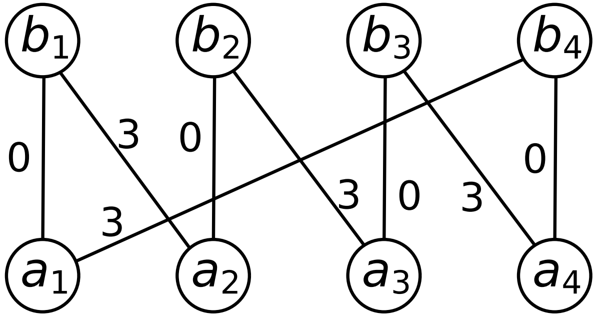
- $SM(m,k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\text{In}(e) \cap \text{Out}(e)| \leq 2k$
  - for any e: $|\text{NotRev}_e| \geq (m-1)\binom{k}{2}$



Vertex	Edge Label			
	0	1	2	3
a_0	b_0	b_1	b_2	b_3
a_1	b_1	b_2	b_3	b_0
a_2	b_2	b_3	b_0	b_1
a_3	b_3	b_0	b_1	b_2

Composed Graphs—Construction

- $SM(m,k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \leq 2k$
  - for any e: $|\mathrm{NotRev}_e| \geq (m-1)\binom{k}{2}$



- $e = \{a,b\}$, how many non-e-reverted edges
  - let $a', a'' \in A_\ell$ for $1 \leq \ell \leq m-1$ with $a' \prec_b a''$
  - $\mathrm{NotRev}_e = \{\{a', \pi^+(a'')\} \mid a' \succeq_b a'' \text{ or } \pi^-(a') \succeq_a \pi^+(a'')\}$
  - $\pi^-(a') \in B_\ell$, $\pi^+(a'') \in B_{\ell-1}$, thus $\pi^+(a'') \prec_a \pi^-(a')$
  - $|\mathrm{NotRev}_e| \geq (m-1)\binom{k}{2}$

Vertex	Edge Label			
	0	1	2	3
a_0	b_0	b_1	b_2	b_3
a_1	b_1	b_2	b_3	b_0
a_2	b_2	b_3	b_0	b_1
a_3	b_3	b_0	b_1	b_2

Composed Graphs—Construction

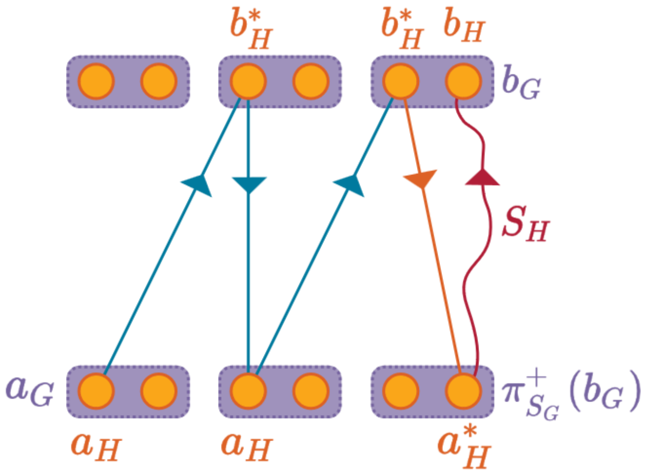
- $SM(m, k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\mathrm{In}(e) \cap \mathrm{Out}(e)| \leq 2k$
  - for any e: $|\mathrm{NotRev}_e| \geq (m-1)\binom{k}{2}$
- for any $f : \mathbb{N} \to \mathbb{N}^+$ with $f(n) \in \mathcal{O}(n)$, can construct bi-clique s.t. for all $e \in A \otimes B$:
  - $\mathrm{In}(e) \cap \mathrm{Out}(e) \in \mathcal{O}(f(n))$
  - $|\mathrm{NotRev}_e| \in \Omega(n \cdot f(n))$

Composed Graphs—Construction

- $SM(m, k)$: outer graph $SM(m)$, inner graph $SM(k)$
  - for any e: $|\text{In}(e) \cap \text{Out}(e)| \leq 2k$
  - for any e: $|\text{NotRev}_e| \geq (m-1)\binom{k}{2}$
- for any $f : \mathbb{N} \to \mathbb{N}^+$ with $f(n) \in \mathcal{O}(n)$, can construct bi-clique s.t. for all $e \in A \otimes B$:
  - $\text{In}(e) \cap \text{Out}(e) \in \mathcal{O}(f(n))$
  - $|\text{NotRev}_e| \in \Omega(n \cdot f(n))$
- construct $SM(m, k)$ with $m := 1 + \lceil \frac{n}{f(n)} \rceil$, $k := 1 + f(n)$
  - for any e: $|\text{In}(e) \cap \text{Out}(e)| \leq 2 \cdot (1 + f(n))$
  - $|\text{NotRev}_e| \geq \frac{n}{f(n)} \binom{1+f(n)}{2} \geq \frac{nf(n)}{2}$
  - $m \cdot k = \left(1 + \lceil \frac{n}{f(n)} \rceil\right)(1 + f(n)) \geq \frac{n}{f(n)} \cdot f(n) \in \Omega(n)$
  - $m \cdot k \leq \left(2 + \lceil \frac{n}{f(n)} \rceil\right)(1 + f(n)) \leq 2 + 2f(n) + 2n \in \mathcal{O}(n)$

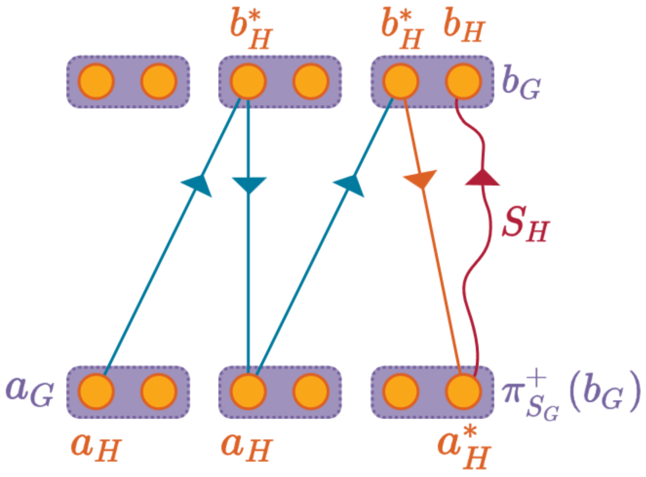
Composed Graphs—Spanners

- tensor product of outer (G) and inner (H) graph
- compose spanner S of size $|S_G| n_H + |S_H| n_G$



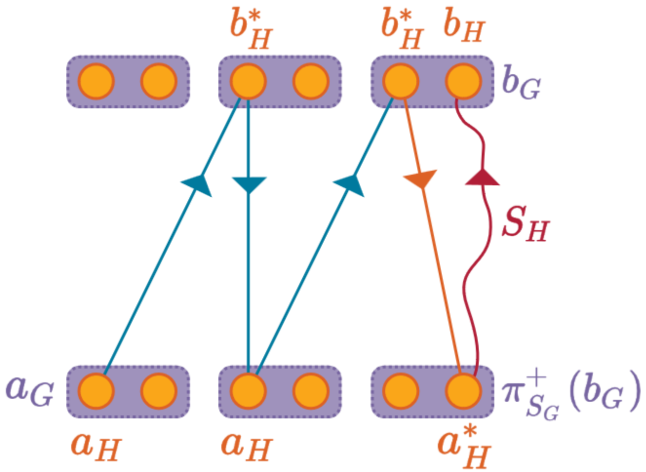
Composed Graphs—Spanners

- tensor product of outer (G) and inner (H) graph
- compose spanner S of size $|S_G|n_H + |S_H|n_G$
  - choose $b^*_H \in B_H, a^*_H \in A_H$ such that $\lambda_H(\{a^*_H, b^*_H\})$ is minimal
  - for $(a_G, b_G) \in S_G$ for all $a_h \in A_H$, add $\{(a_G, a_H), (b_G, b^*_H)\}$ to S
  - for $(a_H, b_H) \in S_H$ for all $b_G \in B_G$, add $\{(\pi^+_{S_G}(b_G), a_H), (b_G, b_H)\}$ to S



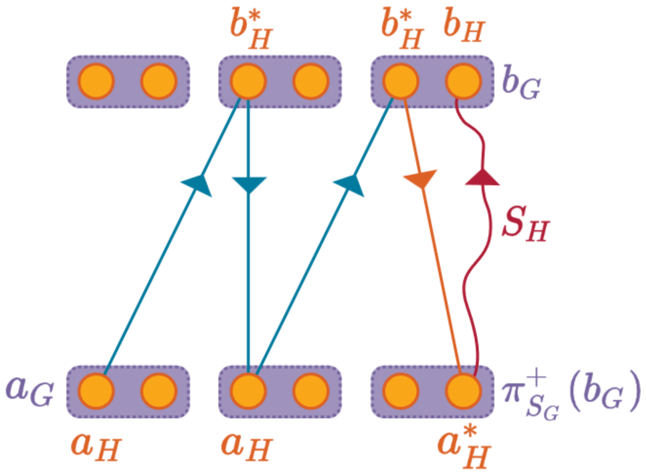
Composed Graphs—Spanners

- tensor product of outer (G) and inner (H) graph
- compose spanner S of size $|S_G|n_H + |S_H|n_G$
  - choose $b_H^* \in B_H, a_H^* \in A_H$ such that $\lambda_H(\{a_H^*, b_H^*\})$ is minimal
  - for $(a_G, b_G) \in S_G$ for all $a_h \in A_H$, add $\{(a_G, a_H), (b_G, b_H^*)\}$ to S
  - for $(a_H, b_H) \in S_H$ for all $b_G \in B_G$, add $\{(\pi^+_{S_G}(b_G), a_H), (b_G, b_H)\}$ to S
- path from $(a_G, a_H) \in A_G \times A_H$ to $(b_G, b_H) \in B_G \times B_H$
  - use path $a_G \to b_G$ from S_G, second last in $(a'_G, a_H) \in A$
  - if $a'_G \neq \pi^+_{S_G}(b_G)$ then next $(b_G, b_H^*), (\pi^+_{S_G}(b_G), a_H^*)$
  - in bags $\pi^+_{S_G}(b_G), b_G$, use spanner S_H



Composed Graphs—Spanners

- tensor product of outer (G) and inner (H) graph
- compose spanner S of size $|S_G|n_H + |S_H|n_G$
  - choose $b_H^* \in B_H, a_H^* \in A_H$ such that $\lambda_H(\{a_H^*, b_H^*\})$ is minimal
  - for $(a_G, b_G) \in S_G$ for all $a_h \in A_H$, add $\{(a_G, a_H), (b_G, b_H^*)\}$ to S
  - for $(a_H, b_H) \in S_H$ for all $b_G \in B_G$, add $\{(\pi^+_{S_G}(b_G), a_H), (b_G, b_H)\}$ to S



- path from $(a_G, a_H) \in A_G \times A_H$ to $(b_G, b_H) \in B_G \times B_H$
  - use path $a_G \to b_G$ from S_G, second last in $(a'_G, a_H) \in A$
  - if $a'_G \neq \pi^+_{S_G}(b_G)$ then next $(b_G, b_H^*), (\pi^+_{S_G}(b_G), a_H^*)$
  - in bags $\pi^+_{S_G}(b_G), b_G$, use spanner S_H
- works for arbitrary composed graphs, adapt b_H^*, a_H^* per bag