Parameterized Complexity of the Spanning Tree Congestion Problem

Hans L. Bodlaender[1] Fedor V. Fomin[2]
Petr A. Golovach[3] Yota Otachi[4] Erik Jan van Leeuwen[2]

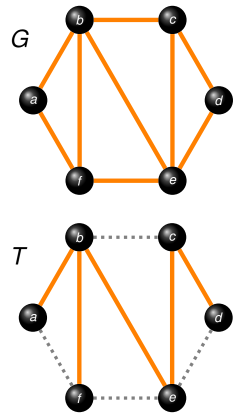
[1]Utrecht University, the Netherlands
[2]University of Bergen, Norway
[3]Durham University, United Kingdom
[4]Tohoku University, Japan

2nd Symposium of ERATO MINATO Discrete Structure Manipulation System Project
Nov. 29–Dec. 1, 2010, Sapporo, Japan

Outline

Outline

Definition: Spanning Tree Congestion

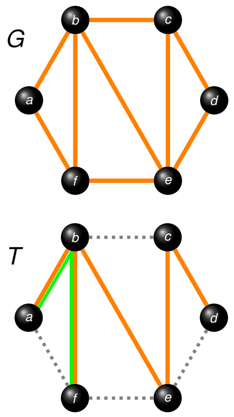


Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Definition: Spanning Tree Congestion

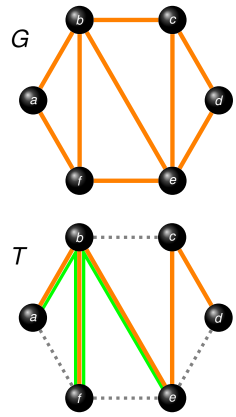


Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Definition: Spanning Tree Congestion

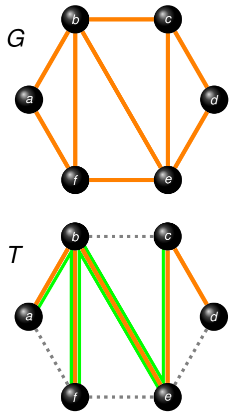


Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Definition: Spanning Tree Congestion



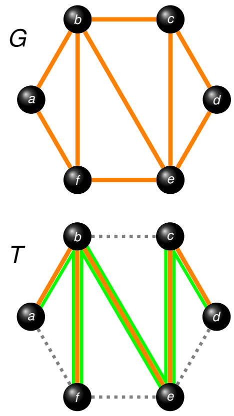
Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Introduction Sketch of proofs Conclusion

Definition: Spanning Tree Congestion



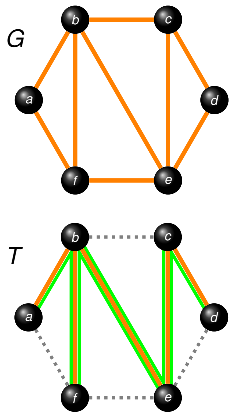
Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Introduction Sketch of proofs Conclusion

Definition: Spanning Tree Congestion



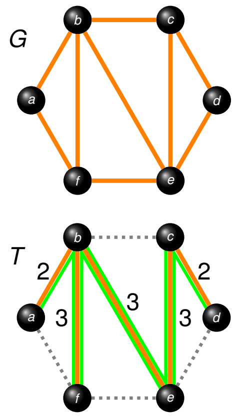
Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Introduction Sketch of proofs Conclusion

Definition: Spanning Tree Congestion



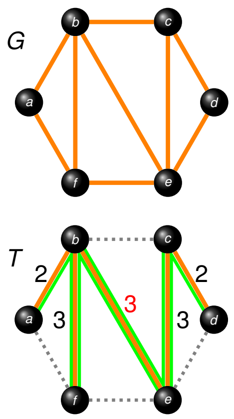
Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Introduction Sketch of proofs Conclusion

Definition: Spanning Tree Congestion



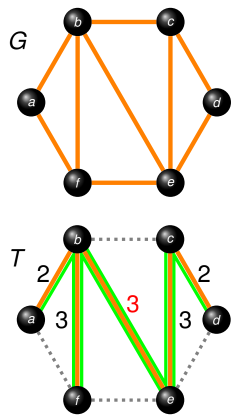
Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

Introduction Sketch of proofs Conclusion

Definition: Spanning Tree Congestion



Definition (Spanning Tree Congestion)

T: a spanning tree of a graph G.

- The *detour* of $\{u, v\} \in E(G)$ is the unique u–v path in T.
- The *congestion of $e \in E(T)$*, $cng_{G,T}(e)$ is the number of edges in G whose detours contain e.
- The *congestion of T*, $cng_G(T)$ is the max. congestion over all its edges.
- The *spanning tree congestion* of G, $stc(G)$ is the min. congestion over all its spanning trees.

The problems

Problem: STC
Instance: Connected graph G, positive integer k.
Question: $stc(G) \leq k$?

Problem: k-STC
Instance: Connected graph G.
Question: $stc(G) \leq k$?
Note: k is a fixed constant.

We investigate the complexity of STC and k-STC.
(k-STC is a parameterized version of STC.)

The problems

Problem: STC
Instance: Connected graph G, positive integer k.
Question: $stc(G) \leq k$?

Problem: k-STC
Instance: Connected graph G.
Question: $stc(G) \leq k$?
Note: k is a fixed constant.

We investigate the complexity of STC and k-STC.
(k-STC is a parameterized version of STC.)

Outline

Previous work

S.T.C. is a relatively new graph parameter.

History

Simonson '87 (implicitly) Bounds for outerplanar graphs
1990's (implicitly) in papers on Tree Spanner problems
Ostrovskii '04 named the parameter S.T.C.
2008–2010 Bounds or exact values for some graphs such as grids, complete k-partite graphs, and hypercubes.

No complexity result (to the best of my knowledge).

Previous work

S.T.C. is a relatively new graph parameter.

History

Simonson '87 (implicitly) Bounds for outerplanar graphs
1990's (implicitly) in papers on Tree Spanner problems
Ostrovskii '04 named the parameter S.T.C.
2008–2010 Bounds or exact values for some graphs such as grids, complete k-partite graphs, and hypercubes.

No complexity result (to the best of my knowledge).

Our results

Theorem (Positive results)
k-STC is linear time solvable for each of the following cases:
1. $k \leq 3$.
2. *input graphs are apex-minor-free.*
3. *input graphs have bounded maximum degree.*

Theorem (Negative results)
k-STC is NP-complete even if the following conditions hold:
- $k \geq 8$,
- *input graphs are K_6-minor-free, and*
- *input graphs have only one vertex of unbounded degree.*

STC is NP-complete for planar graphs.
No PTAS, unless P $=$ NP.

Apex-minor-free graphs

Definition (Apex graphs)
An *apex graph* is a graph that can be made planar by the removal of a single vertex.

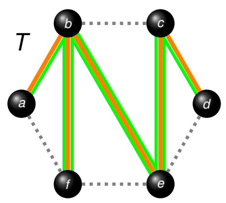
Examples of apex graphs
K_5, $K_{3,n}$ for any n (and of course all planar graphs).

Definition (Apex-minor-free graphs)
A graph class is *apex-minor-free* if it excludes a fixed apex graph as a minor.

Examples of apex-minor-free graphs
Planar graphs, bounded genus graphs, and graphs of bounded treewidth.

Outline

Related problems



stretch = 2
congestion = 3

stretch = 3
congestion = 4

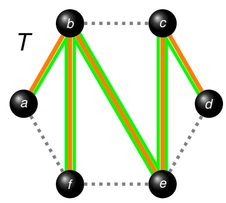
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 2
congestion = 3

stretch = 3
congestion = 4

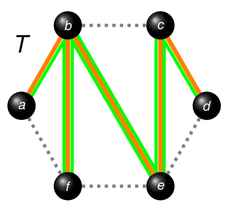
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

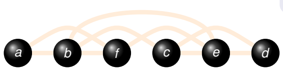
Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 2
congestion = 3



stretch = 3
congestion = 4

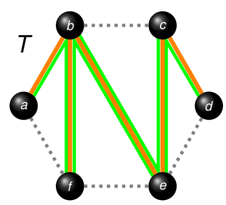
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

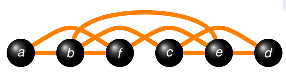
Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 2
congestion = 3



stretch = 3
congestion = 4

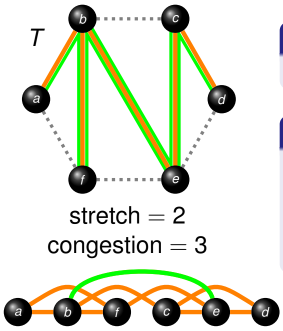
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 3
congestion = 4

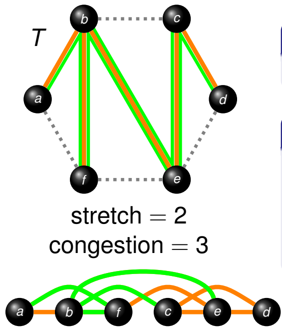
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 3
congestion = 4

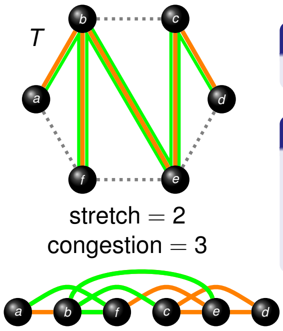
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 3
congestion = 4

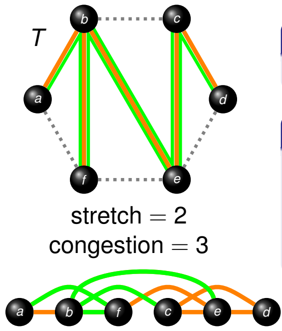
stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Related problems



stretch = 3
congestion = 4

stretch: the length of the longest detour.

Tree spanner problem
Minimize the stretch.

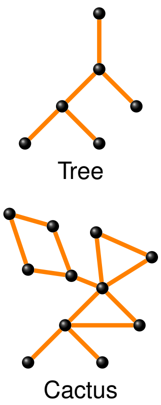
Bandwidth & Cutwidth problems
Embed a graph on a line (or, find a good linear arrangement) so that the stretch (bandwidth) or the congestion (cutwidth) is minimized.

	line	spanning tree
stretch	Bandwidth	Tree Spanner
congestion	Cutwidth	S.T.C.

Outline

1. Introduction
   - Definition of the problem
   - Previous work & Our results
   - Related problems
2. Sketch of proofs
   - k-STC for $k \leq 3$
   - k-STC for bounded degree graphs
3. Conclusion
   - Conclusion & Open problems

k-STC for $k \leq 3$ (1 of 2)



Theorem
$stc(G) = 1 \iff$ *G is a tree*
$stc(G) \leq 2 \iff$ *G is a cactus*

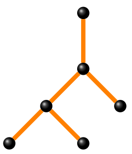
Good characterization for "$stc(G) \leq 3$" ?

Lemma
If $stc(G) \leq 3$, then G is planar.

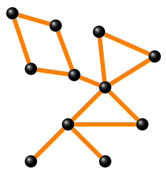
Proof.
1. K_n subdivision $\implies stc \geq n - 1$
2. $K_{n,n}$ subdivision $\implies stc \geq n + 1$

Nonplanarity $\implies K_5$ or $K_{3,3}$ subdivision $\implies stc \geq 4$. □

k-STC for $k \le 3$ (1 of 2)



Tree



Cactus

Theorem

$stc(G) = 1 \iff$ *G is a tree*
$stc(G) \le 2 \iff$ *G is a cactus*

Good characterization for "$stc(G) \le 3$" ?

Lemma

If $stc(G) \le 3$, then G is planar.

Proof.

1. K_n subdivision $\implies stc \ge n-1$
2. $K_{n,n}$ subdivision $\implies stc \ge n+1$

Nonplanarity $\implies K_5$ or $K_{3,3}$ subdivision $\implies stc \ge 4$. □

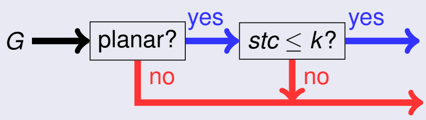
k-STC for $k \le 3$ (2 of 2)

Theorem

k-STC is solvable in linear time if $k \le 3$.

Proof.

1. Check the planarity of G in linear time.
2. If the answer is NO, then $stc(G) \ge 4$.
3. Otherwise, use our linear time algorithm for planar graphs. (Or, we can use another *practical* algorithm.)



□

Outline

k-STC for graphs of bounded degree (1 of 2)

We need the following two lemmas. ($\Delta(G)$ is the max degree)

Lemma

$tw(G) \le \Delta(G)(stc(G) - 1)/2$.

Lemma

k-STC is linear time solvable for graphs of bounded treewidth.

Proof of the second lemma.

k-STC can be expressed in MSO logic. Courcelle's Theorem implies the lemma. □

MSO logic expression for k-STC

$$\begin{aligned}
\mathrm{Deg1}(v_1, E_1) &:= (\exists e_1 \in E_1)(\forall e_2 \in E_1)(e_1 = e_2 \iff \mathbf{inc}(v_1, e_2))\\
\mathrm{Part}(V_1, V_2, V_3) &:= V_2 \neq \emptyset \wedge V_3 \neq \emptyset \wedge (V_2 \cup V_3 = V_1) \wedge (V_2 \cap V_3 = \emptyset)\\
\mathrm{Adj}(v_1, v_2, E_1) &:= v_1 \neq v_2 \wedge (\exists e_1 \in E_1)(\mathbf{inc}(v_1, e) \wedge \mathbf{inc}(v_2, e))\\
E_1 = \mathrm{Ind}(V_1) &:= (\forall e_1)(e_1 \in E_1 \iff (\exists v_1, v_2 \in V_1)(v_1 \neq v_2 \wedge \mathbf{inc}(v_1, e_1) \wedge \mathbf{inc}(v_2, e_1)))\\
E_1 = \mathrm{Inc}_{\mathrm{E}}(v_1) &:= (\forall e_1)(e_1 \in E_1 \iff \mathbf{inc}(v_1, e_1))\\
V_1 = \mathrm{Inc}_{\mathrm{V}}(E_1) &:= (\forall v_1)(v_1 \in V_1 \iff (\exists e_1 \in E_1)(\mathbf{inc}(v_1, e_1)))\\
\mathrm{Conn}(E_1) &:= (\forall V_2, V_3)(\mathrm{Part}(\mathrm{Inc}_{\mathrm{V}}(E_1), V_2, V_3) \implies (\exists v_2 \in V_2, v_3 \in V_3)(\mathrm{Adj}(v_2, v_3, E_1)))\\
\mathrm{BiConn}(E_1) &:= (\exists v_1, v_2, v_3 \in \mathrm{Inc}_{\mathrm{V}}(E_1))(v_i \neq v_j)(1 \le i < j \le 3) \wedge (\forall v_4)(\mathrm{Conn}(E_1 \setminus \mathrm{Inc}_{\mathrm{E}}(v_4)))\\
\mathrm{Forest}(E_1) &:= (\forall V_1 \subseteq \mathrm{Inc}_{\mathrm{V}}(E_1))(\neg \mathrm{BiConn}(\mathrm{Ind}(V_1) \cap E_1))\\
\mathrm{Tree}(E_1) &:= \mathrm{Forest}(E_1) \wedge \mathrm{Conn}(E_1)\\
\mathrm{Path}(v_1, v_2, E_1) &:= \mathrm{Tree}(E_1) \wedge (\forall v_3 \in \mathrm{Inc}_{\mathrm{V}}(E_1))(\mathrm{Deg1}(v_3, E_1) \iff v_3 = v_1 \vee v_3 = v_2)\\
\mathrm{SpnTree}(E_1) &:= \mathrm{Tree}(E_1) \wedge (\forall v)(v \in \mathrm{Inc}_{\mathrm{V}}(E_1))\\
\mathrm{Detour}(e_1, E_1) &:= (\exists v_1, v_2)(v_1 \neq v_2 \wedge \mathbf{inc}(v_1, e_1) \wedge \mathbf{inc}(v_2, e_1) \wedge \mathrm{Path}(v_1, v_2, E_1))\\
\mathrm{Cong}_k(e_0, E_0) &:= \neg(\exists e_1, \ldots, e_k)((e_i \notin E_0)(1 \le i \le k) \wedge e_i \neq e_j(1 \le i < j \le k)\\
&\qquad \wedge (\exists E_i)(e_0 \in E_i \wedge E_i \subseteq E_0 \wedge \mathrm{Detour}(e_i, E_i))(1 \le i \le k))
\end{aligned}$$

$$stc(G) \le k \iff G \models (\exists E_0)(\mathrm{SpnTree}(E_0) \wedge (\forall e_0 \in E_0)(\mathrm{Cong}_k(e_0, E_0))).$$

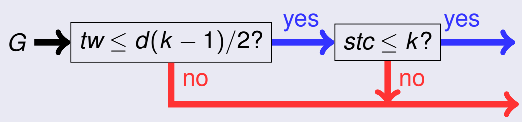
k-STC for graphs of bounded degree (2 of 2)

Theorem

k-STC is linear time solvable on graphs of bounded degree.

Proof.

1. Check $tw(G) \le d(k-1)/2$ in linear time. ($d := \Delta(G)$)
2. If the answer is NO, then $stc(G) > k$.
3. O.w., use a linear time algorithm for bounded treewidth.



We use Bodlaender's algorithm to check bounded tw. □

Outline

1. Introduction
   - Definition of the problem
   - Previous work & Our results
   - Related problems
2. Sketch of proofs
   - k-STC for $k \leq 3$
   - k-STC for bounded degree graphs
3. Conclusion
   - Conclusion & Open problems

Our results

Theorem (Positive results)

k-STC is linear time solvable for each of the following cases:

1. $k \leq 3$.
2. *input graphs are apex-minor-free.*
3. *input graphs have bounded maximum degree.*

Theorem (Negative results)

k-STC is NP-complete even if the following conditions hold:

- $k \geq 8$,
- *input graphs are K_6-minor-free, and*
- *input graphs have only one vertex of unbounded degree.*

STC is NP-complete for planar graphs.
No PTAS, unless $\mathrm{P} = \mathrm{NP}$.

Open problems

- Complexity of k-STC for $k \in \{4, 5, 6, 7\}$.
  - I think 4-STC might be NP-complete.
- Complexity of STC and k-STC for some graph classes.
  - Recent progress with Y. Okamoto, R. Uehara, and T. Uno: STC is NP-complete for split graphs and chordal bipartite graphs, and linear time solvable for trivially perfect graphs.
- Approximation. ($O(\log n)$-factor approximation?)
  - Constant factor approximation might be NP-hard.
- Is STC $\in$ FPT for k-outerplanar graphs? (parameter is k)
  - If $k = 1$, i.e., for outerplanar graphs, STC can be solved in linear time. (with Bodlaender, Kozawa, Matsushima)

Thank you for your attention!

Introduction Sketch of proofs Conclusion

Open problems

- Complexity of k-STC for $k \in \{4, 5, 6, 7\}$.
  - I think 4-STC might be NP-complete.
- Complexity of STC and k-STC for some graph classes.
  - Recent progress with Y. Okamoto, R. Uehara, and T. Uno: STC is NP-complete for split graphs and chordal bipartite graphs, and linear time solvable for trivially perfect graphs.
- Approximation. ($O(\log n)$-factor approximation?)
  - Constant factor approximation might be NP-hard.
- Is STC $\in$ FPT for k-outerplanar graphs? (parameter is k)
  - If $k = 1$, i.e., for outerplanar graphs, STC can be solved in linear time. (with Bodlaender, Kozawa, Matsushima)

Thank you for your attention!

21 / 21