The strict terminal connection problem with a bounded number of routers

Alexsander A. Melo - UFRJ
Celina M. H. de Figueiredo - UFRJ
Uéverton S. Souza - UFF

VII Latin American Workshop on Cliques in Graphs, 2016

Connection tree

- A *connection tree* of a graph G on $W \subseteq V(G)$
  is a subgraph T of G such that:

Connection tree

- A *connection tree* of a graph G on $W \subseteq V(G)$ is a subgraph T of G such that:
  1. T is a tree;

Connection tree

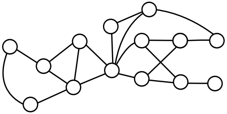
- A *connection tree* of a graph G on $W \subseteq V(G)$ is a subgraph T of G such that:
  1. T is a tree;
  2. $W \subseteq V(T)$;

Connection tree

- A *connection tree* of a graph G on $W \subseteq V(G)$ is a subgraph T of G such that:
  1. T is a tree;
  2. $W \subseteq V(T)$;
  3. $\text{leaves}(T) \subseteq W$.

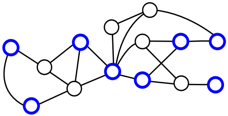
Connection tree

- A *connection tree* of a graph G on $W \subseteq V(G)$ is a subgraph T of G such that:
  1. T is a tree;
  2. $W \subseteq V(T)$;
  3. leaves$(T) \subseteq W$.



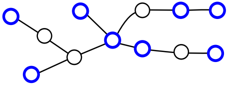
Connection tree

- A *connection tree* of a graph G on $W \subseteq V(G)$ is a subgraph T of G such that:
  1. T is a tree;
  2. $W \subseteq V(T)$;
  3. $\text{leaves}(T) \subseteq W$.



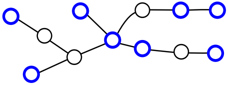
Connection tree

- A *connection tree* of a graph G on $W \subseteq V(G)$ is a subgraph T of G such that:
  1. T is a tree;
  2. $W \subseteq V(T)$;
  3. $\text{leaves}(T) \subseteq W$.



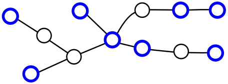
Connection tree: vertices

- In a connection tree T on W, we call the vertices belonging to



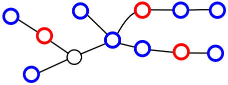
Connection tree: vertices

- In a connection tree T on W, we call the vertices belonging to
  - W of *terminals*;



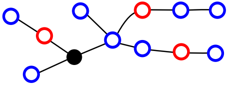
Connection tree: vertices

- In a connection tree T on $\underline{W}$, we call the vertices belonging to
  - W of *terminals*;
  - $V(T) \setminus W$ with degree exactly 2 in T of *linkers*;



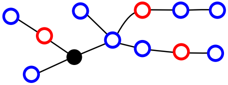
Connection tree: vertices

- In a connection tree T on $\underline{W}$, we call the vertices belonging to
  - W of *terminals*;
  - $V(T) \setminus W$ with degree exactly 2 in T of *linkers*;
  - $V(T) \setminus W$ with degree at least 3 in T of *routers*.



Connection tree: vertices

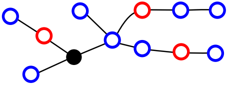
- In a connection tree T on W, we call the vertices belonging to
  - W of *terminals*;
  - $V(T) \setminus W$ with degree exactly 2 in T of *linkers*;
  - $V(T) \setminus W$ with degree at least 3 in T of *routers*.



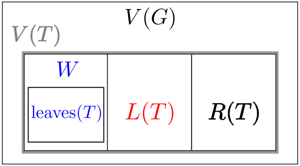
- Thus, there is a partition of $V(T)$ into *terminals*, *linkers* and ***routers***.

Connection tree: vertices

- In a connection tree T on W, we call the vertices belonging to
  - W of *terminals*;
  - $V(T) \setminus W$ with degree exactly 2 in T of *linkers*;
  - $V(T) \setminus W$ with degree at least 3 in T of *routers*.



- Thus, there is a partition of $V(T)$ into *terminals*, *linkers* and ***routers***.



The terminal connection problem

- Motivated by applications in networks, Dourado et al. proposed the TERMINAL CONNECTION PROBLEM:

TERMINAL CONNECTION PROBLEM (TCP)

Instance: A connected graph G, a subset $W \subseteq V(G)$ and two non-negative integers ℓ and r.

Question: Does G admit a connection tree on W with $|L(T)| \leq \ell$ and $|R(T)| \leq r$?

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Design of connection networks with bounded number of non-terminal vertices, **Latin-American Workshop on Clique in Graphs 2012**, *Matemática Contemporânea*, 42, pp. 39–48, 2014.

TCP: computatinal complexity

- The TCP is NP-complete even when either ℓ or r is bounded by a constant.

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Design of connection networks with bounded number of non-terminal vertices, **Latin-American Workshop on Clique in Graphs 2012**, *Matemática Contemporânea*, 42, pp. 39–48, 2014.

TCP: computatinal complexity

- The TCP is NP-complete even when either ℓ or r is bounded by a constant.
- When ℓ and r are both bounded by constants, the TCP is polynomial-time solvable

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Design of connection networks with bounded number of non-terminal vertices, **Latin-American Workshop on Clique in Graphs 2012**, *Matemática Contemporânea*, 42, pp. 39–48, 2014.

TCP: computatinal complexity

- The TCP is NP-complete even when either ℓ or r is bounded by a constant.
- When ℓ and r are both bounded by constants, the TCP is polynomial-time solvable

- **Notation.** ℓ constant: TCP(ℓ); r constant: TCP(r); ℓ and r constants: TCP(ℓ, r).

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Design of connection networks with bounded number of non-terminal vertices, **Latin-American Workshop on Clique in Graphs 2012**, *Matemática Contemporânea*, 42, pp. 39–48, 2014.

The strict terminal connection problem

- By definition, leaves(T) $\subseteq W$;

The strict terminal connection problem

- By definition, leaves$(T) \subseteq W$;   But possibly $W \not\subseteq$ leaves(T).

The strict terminal connection problem

- By definition, leaves(T) $\subseteq W$; But possibly $W \not\subseteq$ leaves(T).
- What happens if we require $W =$ leaves(T)?

The strict terminal connection problem

- By definition, leaves$(T) \subseteq W$;   But possibly $W \not\subseteq$ leaves(T).
- What happens if we require $W =$ leaves(T)?

STRICT TERMINAL CONNECTION PROBLEM (S-TCP)

Instance: A connected graph G, a subset $W \subseteq V(G)$ and two non-negative integers ℓ and r.

Question: Does G admit a **strict** connection tree T on W with $|L(T)| \leq \ell$ and $|R(T)| \leq r$?

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Conexão de terminais com número restrito de roteadores e elos, In Annals of XLVI SBPO, pp. 2965–2976, 2014.

S-TCP: computational complexity

- The S-TCP(ℓ, r) is polynomial-time solvable.
- The S-TCP(ℓ) is NP-complete.

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Conexão de terminais com número restrito de roteadores e elos, In Annals of XLVI SBPO, pp. 2965–2976, 2014.

S-TCP: computational complexity

- The S-TCP(ℓ, r) is polynomial-time solvable.
- The S-TCP(ℓ) is NP-complete.

- We contribute to the complexity of the S-TCP(r).

Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Conexão de terminais com número restrito de roteadores e elos, In Annals of XLVI SBPO, pp. 2965–2976, 2014.

The S-TCP($r = 0$) is polynomial-time solvable

The S-TCP($r = 0$) is polynomial-time solvable

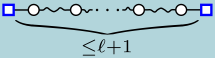
- If $|W| \leq 1$, trivial; Thus, we can suppose $|W| \geq 2$.

The S-TCP($r = 0$) is polynomial-time solvable

- If $|W| \leq 1$, trivial; Thus, we can suppose $|W| \geq 2$.

Lemma

If $|W| = 2$, then $\mathcal{I} = (G, W, \ell)$ is a Yes *of the* S-TCP*($r = 0$) if and only if the distance in G between the two vertices of W is at most $\ell + 1$.*

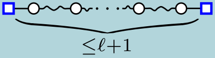


The S-TCP($r = 0$) is polynomial-time solvable

- If $|W| \leq 1$, trivial; Thus, we can suppose $|W| \geq 2$.

Lemma

If $|W| = 2$, then $\mathcal{I} = (G, W, \ell)$ is a YES *of the* S-TCP*($r = 0$) if and only if the distance in G between the two vertices of W is at most $\ell + 1$.*



Lemma

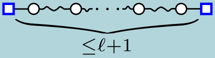
Let T be a tree. If $|\text{leaves}(T)| > 2$, then $\Delta(T) \geq 3$.

The S-TCP($r = 0$) is polynomial-time solvable

- If $|W| \leq 1$, trivial; Thus, we can suppose $|W| \geq 2$.

Lemma

If $|W| = 2$, then $\mathcal{I} = (G, W, \ell)$ is a YES *of the* S-TCP*($r = 0$) if and only if the distance in G between the two vertices of W is at most $\ell + 1$.*



Lemma

Let T be a tree. If $|\text{leaves}(T)| > 2$, then $\Delta(T) \geq 3$.

Corollary

The S-TCP*($r = 0$) is polynomial-time solvable.*

S-TCP($r = 1$)

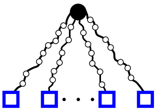


Figure : The topology.

S-TCP($r = 1$): MIN-SUM st−DISJOINT PATHS

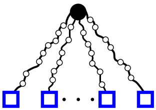


Figure : The topology.

- We prove that the S-TCP($r = 1$) is polynomial-time solvable by a Turing reduction to MIN-SUM st−DISJOINT PATHS:

MIN-SUM st−DISJOINT PATHS (st-DP)

Instance: A graph G, two distinct vertices $s, t \in V(G)$ and two non-negative integers k and x.

Question: Are there k vertex-disjoint paths between s and t in G whose sum of their lenghts is at most x?

S-TCP($r = 1$): the instance $f(\mathcal{I}, \rho)$

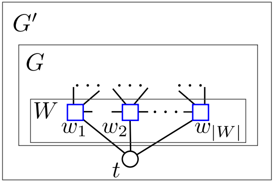
- Let $\mathcal{I} = (G, W, \ell)$ be an instance of the S-TCP($r = 1$) and let $\rho \in V(G) \setminus W$.

S-TCP($r = 1$): the instance $f(\mathcal{I}, \rho)$

- Let $\mathcal{I} = (G, W, \ell)$ be an instance of the S-TCP($r = 1$) and let $\rho \in V(G) \setminus W$.
- Let $f(\mathcal{I}, \rho) = (G', s, t, k, x)$ be the instance of $st-$DP defined as follows:

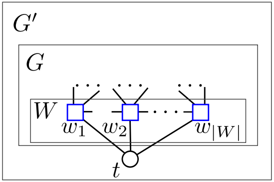
S-TCP($r = 1$): the instance $f(\mathcal{I}, \rho)$

- Let $\mathcal{I} = (G, W, \ell)$ be an instance of the S-TCP($r = 1$) and let $\rho \in V(G) \setminus W$.
- Let $f(\mathcal{I}, \rho) = (G', s, t, k, x)$ be the instance of st−DP defined as follows:
  - $V(G') = V(G) \cup \{t\}$;
  - $E(G') = E(G) \cup \{wt \mid w \in W\}$;



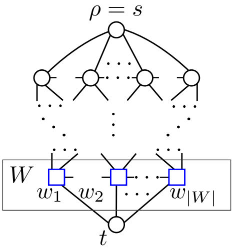
- Let $\mathcal{I} = (G, W, \ell)$ be an instance of the S-TCP($r = 1$) and let $\rho \in V(G) \setminus W$.
- Let $f(\mathcal{I}, \rho) = (G', s, t, k, x)$ be the instance of $st-$DP defined as follows:

  - $V(G') = V(G) \cup \{t\}$;
  - $E(G') = E(G) \cup \{wt \mid w \in W\}$;
  - $s = \rho$, $k = |W|$ e $x = \ell + 2k$.



S-TCP($r = 1$): the instance $f(\mathcal{I}, \rho)$

- Let $\mathcal{I} = (G, W, \ell)$ be an instance of the S-TCP($r = 1$) and let $\rho \in V(G) \setminus W$.
- Let $f(\mathcal{I}, \rho) = (G', s, t, k, x)$ be the instance of $st-$DP defined as follows:
  - $V(G') = V(G) \cup \{t\}$;
  - $E(G') = E(G) \cup \{wt \mid w \in W\}$;
  - $s = \rho$, $k = |W|$ e $x = \ell + 2k$.



S-TCP($r = 1$): relation between $\mathcal{I}$ and $f(\mathcal{I}, \rho)$

Lemma

An instance $\mathcal{I} = (G, W, \ell)$ of the S-TCP($r = 1$) is a YES *instance if and only if there is $\rho \in V(G) \setminus W$, such that $f(\mathcal{I}, \rho)$ is a* YES *instance of st-DP.*

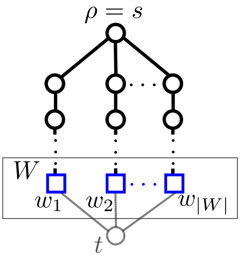
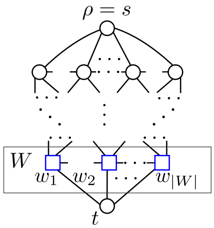


Figure : $k = |W|$ e $x = \ell + 2k$

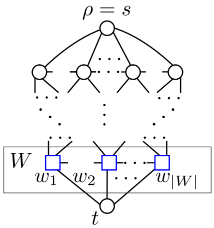
The S-TCP($r = 1$) is polynomial-time solvable

- For all $\rho \in V(G) \setminus W$:
  - Construct the instance $f(\mathcal{I}, \rho)$;
  - Verify if $f(\mathcal{I}, \rho)$ is a YES instance of $st-$DP;
  - If so: return $\mathcal{I}$ is a YES instance of the S-TCP($r = 1$);
- If for every $\rho \in V(G) \setminus W$, $f(\mathcal{I}, \rho)$ is a NO instance of $st-$DP: return $\mathcal{I}$ is a NO instance of the S-TCP($r = 1$).



The S-TCP($r = 1$) is polynomial-time solvable

- For all $\rho \in V(G) \setminus W$:
  - Construct the instance $f(\mathcal{I}, \rho)$;
  - Verify if $f(\mathcal{I}, \rho)$ is a YES instance of $st-$DP;
  - If so: return $\mathcal{I}$ is a YES instance of the S-TCP($r = 1$);
- If for every $\rho \in V(G) \setminus W$, $f(\mathcal{I}, \rho)$ is a NO instance of $st-$DP: return $\mathcal{I}$ is a NO instance of the S-TCP($r = 1$).

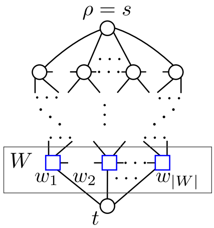


Corollary

The S-TCP($r = 1$) is polynomial-time solvable.

The S-TCP($r = 1$) is polynomial-time solvable

- For all $\rho \in V(G) \setminus W$:
  - Construct the instance $f(\mathcal{I}, \rho)$;
  - Verify if $f(\mathcal{I}, \rho)$ is a YES instance of $st-$DP;
  - If so: return $\mathcal{I}$ is a YES instance of the S-TCP($r = 1$);
- If for every $\rho \in V(G) \setminus W$, $f(\mathcal{I}, \rho)$ is a NO instance of $st-$DP: return $\mathcal{I}$ is a NO instance of the S-TCP($r = 1$).



Corollary

The S-TCP($r = 1$) is polynomial-time solvable.

Corollary

The S-TCP($r = 2$) when the two routers are neighbours is polynomial-time solvable.

The PARTITIONED S-TCP($r = 2$)

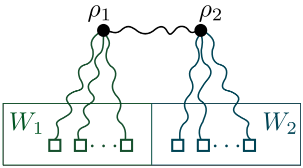
PARTITIONED STRICT TERMINAL CONNECTION PROBLEM

Instance: A connected graph G, a subset $W \subseteq V(G)$, a bipartition $W = W_1 \cup W_2$ and a non-negative integer ℓ.

Question: Does G admit a strict connection tree T on W with $|L(T)| \leq \ell$ and $|R(T)| = 2$, such that:

- $\text{dist}_T(\rho_1, w) < \text{dist}_T(\rho_2, w), \forall\, w \in W_1$;
- $\text{dist}_T(\rho_2, w') < \text{dist}_T(\rho_1, w'), \forall\, w' \in W_2$;

where $R(T) = \{\rho_1, \rho_2\}$?

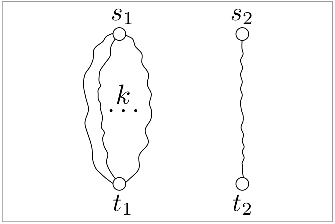


$k+1$DP

$k+1$ VERTEX-DISJOINT PATHS BETWEEN TWO PAIRS OF VERTICES ($k+1$DP)

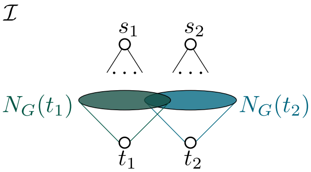
Instance: A graph G, four vertices $s_1, t_1, s_2, t_2 \in V(G)$ and an integer $k \geq 0$.

Question: Are there $k+1$ vertex-disjoint paths in G, such that k of these paths are between s_1 and t_1 and the remaining path is between s_2 and t_2?



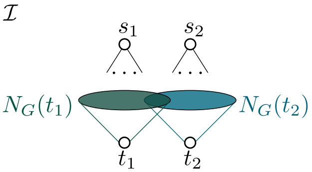
PARTITIONED S-TCP($r = 2$): the instance $f(\mathcal{I})$

- Let $\mathcal{I} = (G, (s_1, t_1), (s_2, t_2), k)$ be an instance of $k + 1$DP;



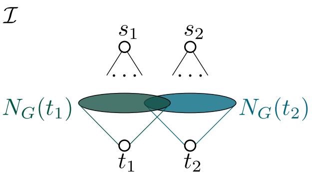
Partitioned S-TCP($r = 2$): the instance $f(\mathcal{I})$

- Let $\mathcal{I} = (G, (s_1, t_1), (s_2, t_2), k)$ be an instance of $k + 1$DP;
- Let $f(\mathcal{I}) = (G', W_1 \cup W_2, \ell)$ be the instance of the Partitioned S-TCP($r = 2$) defined as follows:



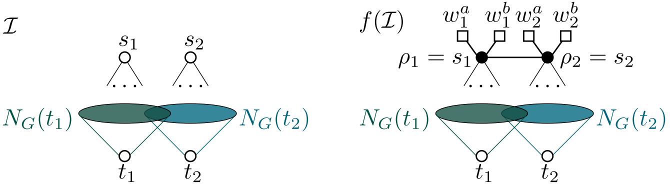
PARTITIONED S-TCP($r = 2$): the instance $f(\mathcal{I})$

- Let $\mathcal{I} = (G, (s_1, t_1), (s_2, t_2), k)$ be an instance of $k + 1$DP;
- Let $f(\mathcal{I}) = (G', W_1 \cup W_2, \ell)$ be the instance of the PARTITIONED S-TCP($r = 2$) defined as follows:



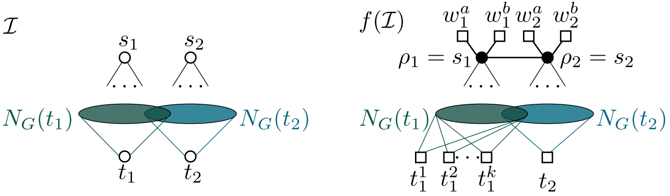
PARTITIONED S-TCP($r = 2$): the instance $f(\mathcal{I})$

- Let $\mathcal{I} = (G, (s_1, t_1), (s_2, t_2), k)$ be an instance of $k + 1$DP;
- Let $f(\mathcal{I}) = (G', W_1 \cup W_2, \ell)$ be the instance of the PARTITIONED S-TCP($r = 2$) defined as follows:

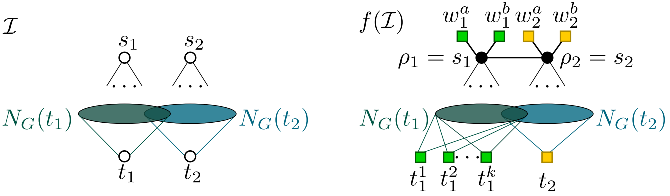


Partitioned S-TCP($r = 2$): the instance $f(\mathcal{I})$

- Let $\mathcal{I} = (G, (s_1, t_1), (s_2, t_2), k)$ be an instance of $k + 1$DP;
- Let $f(\mathcal{I}) = (G', W_1 \cup W_2, \ell)$ be the instance of the PARTITIONED S-TCP($r = 2$) defined as follows:



- Let $\mathcal{I} = (G, (s_1, t_1), (s_2, t_2), k)$ be an instance of $k + 1$DP;
- Let $f(\mathcal{I}) = (G', W_1 \cup W_2, \ell)$ be the instance of the PARTITIONED S-TCP($r = 2$) defined as follows:



- $W_1 = \{t_1^i \mid 1 \leq i \leq k\} \cup \{w_1^a, w_1^b\}$ e $W_2 = \{w_2^a, w_2^b, t_2\}$; $\ell = |V(G) \setminus W|$.

The PARTITIONED S-TCP($r = 2$) is NP-complete

Lemma

An instance $\mathcal{I}$ of $k+1$DP is a YES *instance if and only if $f(\mathcal{I})$ is a* YES *instance of the* PARTITIONED S-TCP*($r = 2$).*

The PARTITIONED S-TCP($r = 2$) is NP-complete

Lemma

An instance $\mathcal{I}$ of $k+1$DP is a YES *instance if and only if $f(\mathcal{I})$ is a* YES *instance of the* PARTITIONED S-TCP*($r = 2$).*

Corollary

The PARTITIONED S-TCP*($r = 2$) is NP-complete.*

The PARTITIONED S-TCP($r = 2$) is NP-complete

Lemma

An instance $\mathcal{I}$ of $k+1$DP is a YES *instance if and only if $f(\mathcal{I})$ is a* YES *instance of the* PARTITIONED S-TCP*($r = 2$).*

Corollary

The PARTITIONED S-TCP*($r = 2$) is NP-complete.*

Corollary

*For all $r \geq 2$, the r-*PARTITIONED S-TCP*(r) is NP-complete.*

Conclusion

Results:

- The S-TCP($r = 0$) , S-TCP($r = 1$) and the S-TCP($r = 2$) when is required that the two router are neighbours are polynomial-time solvable;
- The r-PARTITIONED S-TCP(r) is NP-complete, for all $r \geq 2$.

Open problem:

- The computational complexity of the S-TCP(r), for $r \geq 2$.

Thank you for your attention!

References I

- Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Design of connection networks with bounded number of non-terminal vertices, Latin-American Workshop on Clique in Graphs 2012, *Matemática Contemporânea*, 42, pp. 39–48, 2014.
- Dourado C. M., Oliveira A. R., Protti, F., and Souza, S. U.: Conexão de terminais com número restrito de roteadores e elos, In Annals of XLVI SBPO, pp. 2965–2976, 2014.
- Even S., Itai A., and Shamir A.: On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM Journal on Computing*, 5(4), pp. 691–703, 1976.
- Shiloach Y., and Perl Y.: Finding two disjoint paths between two pairs of vertices in a graph. *Journal of the ACM (JACM)*, 25(1), pp. 1–9, 1978.
- Suurballe, W. J.: Disjoint paths in network. *Networks*, 4(2), pp. 125–145, 1974.