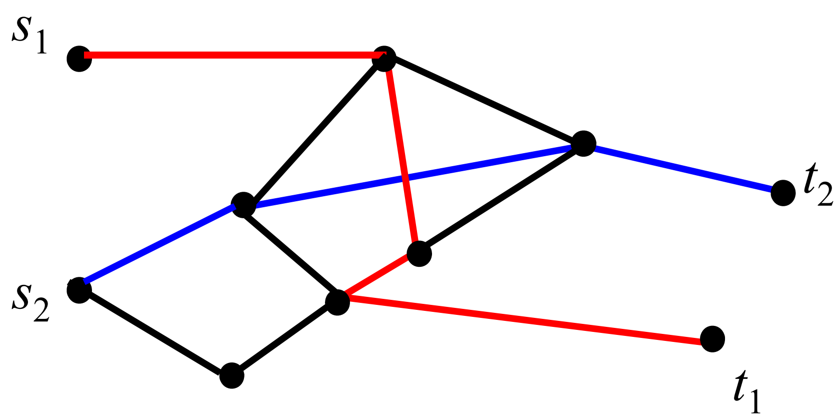
On Shortest Disjoint Paths in Planar Graphs

Yusuke Kobayashi (University of Tokyo)
Christian Sommer (University of Tokyo)

The 20th International Symposium on Algorithms and Computation

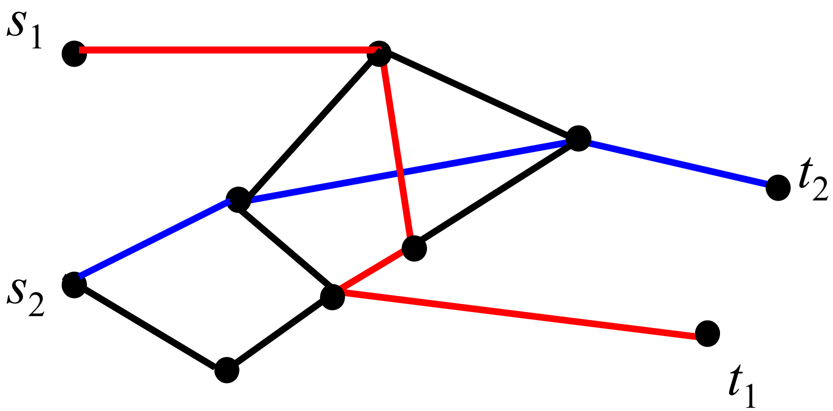
Dec 16, 2009

Disjoint paths problem



- Given vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$
  Find vertex disjoint paths $P_1, \ldots, P_k \quad (P_i : s_i \rightarrow t_i)$
- Many applications (ex. VLSI layout, wireless networks)
- Many variations

Disjoint paths problem



- Given vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$
  Find vertex disjoint paths $P_1, \ldots, P_k \quad (P_i : s_i \rightarrow t_i)$
- Many applications (ex. VLSI layout, wireless networks)
- Many variations
- general k ⇨ NP-hard (Karp, 1975)
- fixed k ⇨ Poly.-time (Robertson-Seymour, 1995)

Shortest disjoint paths problem

Input: vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$

length function l on the edge set

Find: vertex disjoint paths $P_1, \ldots, P_k \quad (P_i : s_i \rightarrow t_i)$

minimizing an objective function

- total length: $\Sigma\, l(P_i)$ (Min-Sum Problem)
- length of the longest path: max $l(P_i)$ (Min-Max Problem)

Shortest disjoint paths problem

Input: vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$

length function l on the edge set

Find: vertex disjoint paths $P_1, \ldots, P_k$ $(P_i : s_i \rightarrow t_i)$

minimizing an objective function

- total length: $\Sigma\, l(P_i)$ (Min-Sum Problem)
- length of the longest path: max $l(P_i)$ (Min-Max Problem)

<u>Our focus</u>

Algorithms for restricted instances of the shortest disjoint paths problem

Our contributions

- Min-Sum Problem

  Poly.-time algorithm for

  - $k = 2$, planar, all terminals are on at most two faces
  - $k = 3$, planar, all terminals are on one face

- Min-Max Problem ($k=2$)
  - tree-width ≥ 3 : NP-hard
  - tree-width ≤ 2 : Poly.-time algorithm

Our contributions

- Min-Sum Problem

  Poly.-time algorithm for

  - $k = 2$, planar, all terminals are on at most two faces
  - $k = 3$, planar, all terminals are on one face

- Min-Max Problem ($k=2$)
  - tree-width ≥ 3 : NP-hard
  - tree-width ≤ 2 : Poly.-time algorithm

Min-Sum problem (known results)

Input: vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$, length function l

Find: vertex disjoint paths $P_1, \ldots, P_k$ $(P_i : s_i \rightarrow t_i)$

minimizing the total length: $\Sigma\, l(P_i)$

General graphs

- general k NP-hard
- fixed k OPEN
- k = 2 OPEN

Min-Sum problem (known results)

Input: vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$, length function l

Find: vertex disjoint paths $P_1, \ldots, P_k$ $(P_i : s_i \to t_i)$

minimizing the total length: $\Sigma\, l(P_i)$

General graphs

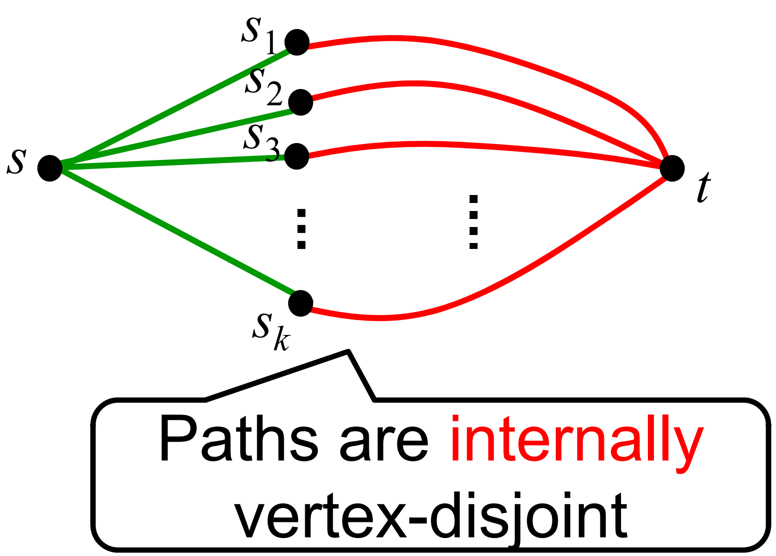
- general k NP-hard
- fixed k OPEN
- k = 2 OPEN

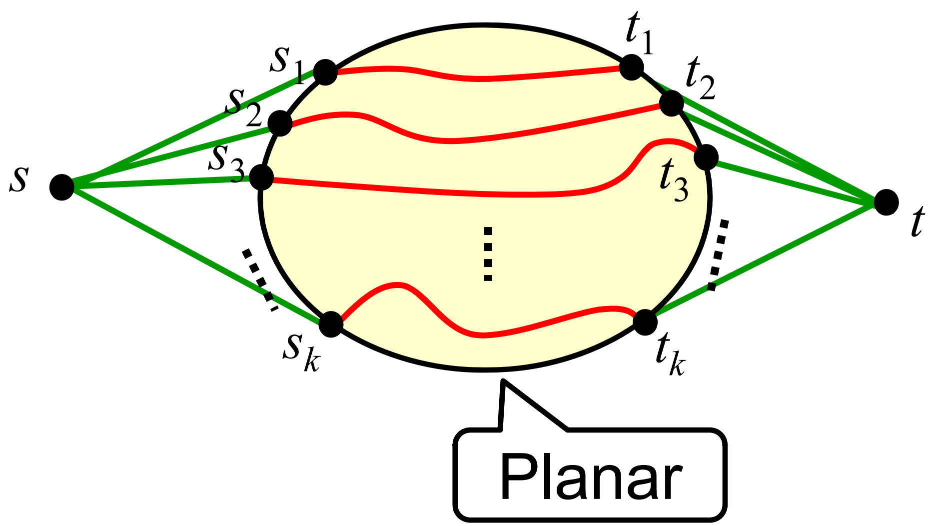
Polynomially solvable cases

- Reducible to the min-cost flow problem
- Planar graph, all s_i's are on one face, and all t_i's are on another face (Colin de Verdière and Schrijver, 2008)

Polynomially solvable cases (Min-Sum)

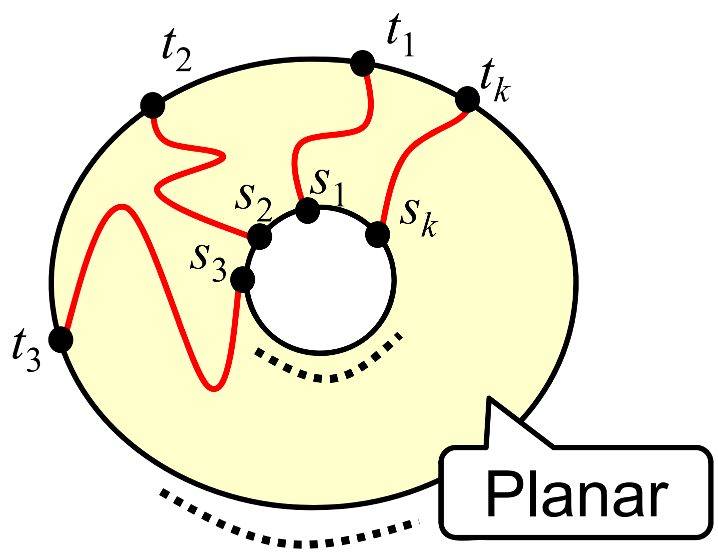
- Reducible to the min-cost flow problem
  - $s_1=s_2=\ldots=s_k$ (and/or) $t_1=t_2=\ldots=t_k$
  - Planar graph, all terminals are on one face, and the ordering is $s_1, s_2, \ldots, s_k, t_k, \ldots, t_2, t_1$





Polynomially solvable cases (Min-Sum)

- Reducible to the min-cost flow problem
  - $s_1=s_2=\ldots=s_k$ (and/or) $t_1=t_2=\ldots=t_k$
  - Planar graph, all terminals are on one face, and the ordering is $s_1, s_2,\ldots, s_k, t_k,\ldots, t_2, t_1$
- Planar graph, all s_i's are on one face, and all t_i's are on another face (Colin de Verdière and Schrijver, 2008)



Polynomially solvable cases (Min-Sum)

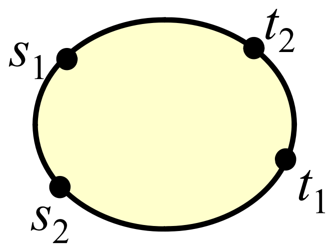
- Reducible to the min-cost flow problem
  - $s_1=s_2=\ldots=s_k$ (and/or) $t_1=t_2=\ldots=t_k$
  - Planar graph, all terminals are on one face, and the ordering is $s_1, s_2,\ldots, s_k, t_k,\ldots, t_2, t_1$
- Planar graph, all s_i's are on one face, and all t_i's are on another face (Colin de Verdière and Schrijver, 2008)

Our results

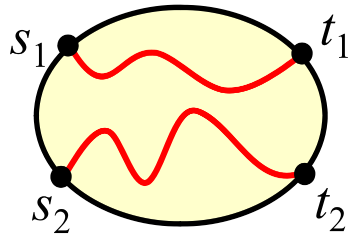
- $k = 2$, planar, all terminals are on at most two faces
- $k = 3$, planar, all terminals are on one face

- $k = 2$, planar, all terminals are on at most two faces

<u>One face</u>

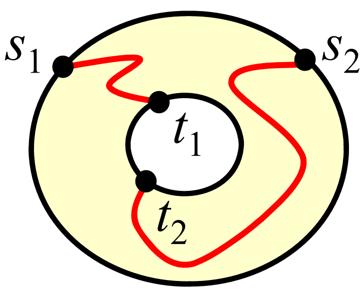


trivially infeasible

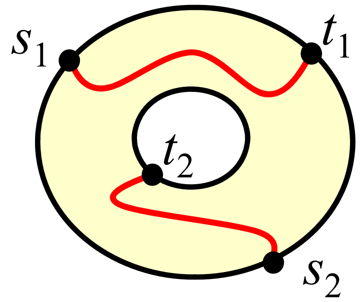


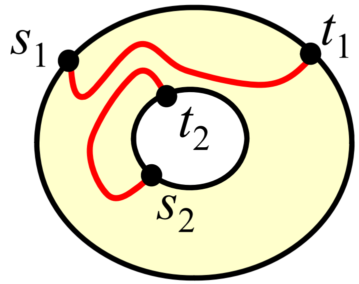
min-cost flow

<u>Two faces</u>



Colin de Verdière-Schrijver

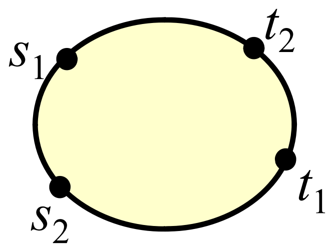




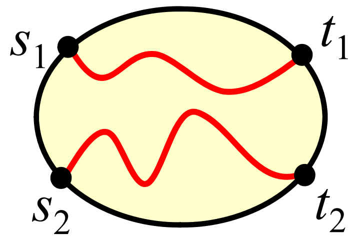
Our results

- $k = 2$, planar, all terminals are on at most two faces

<u>One face</u>

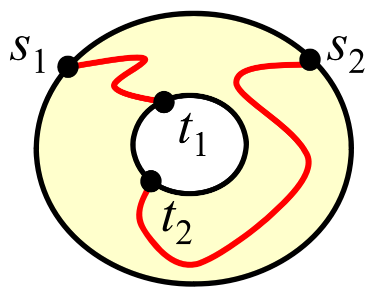


trivially infeasible

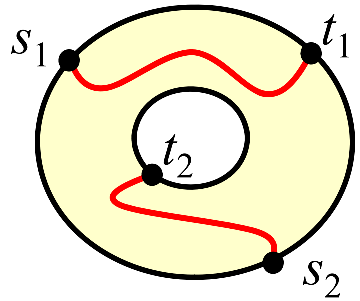


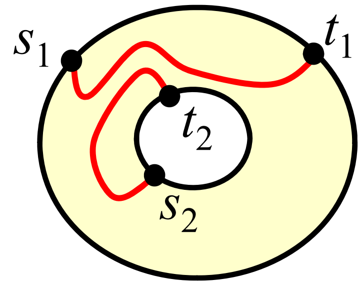
min-cost flow

<u>Two faces</u>



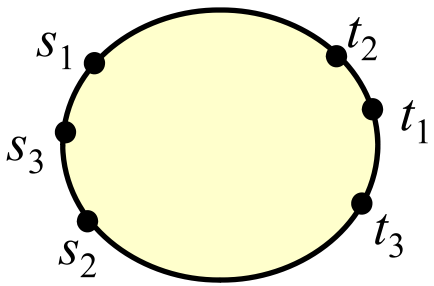
Colin de Verdière-Schrijver



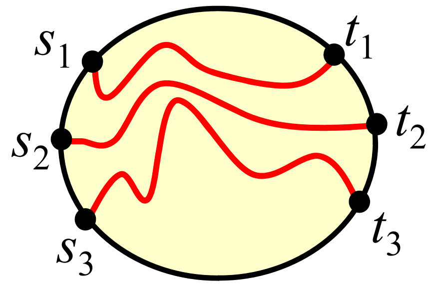


Our results

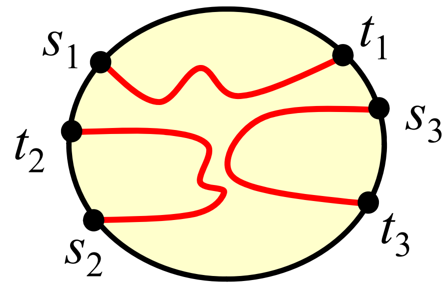
- $k = 3$, planar, all terminals are on one face



trivially infeasible



min-cost flow



Our result

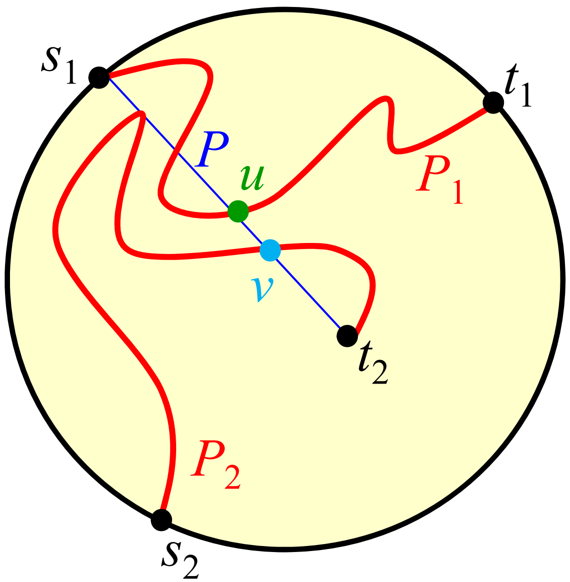
Three terminals on one face

Theorem (Our results)

$k = 2$, planar graph, 3 terminals are on one face

⇨ Min-Sum disjoint paths can be found in poly.-time

- Technique: reduction to the CS's result



(P_1, P_2) : shortest disjoint paths

P : shortest path between s_1 and t_2

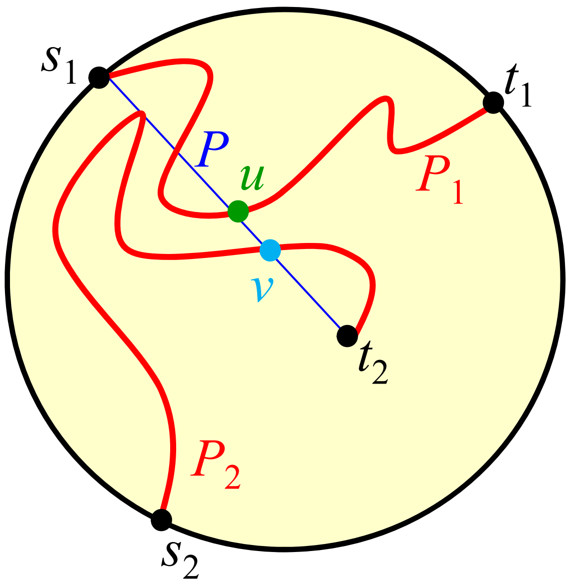
Three terminals on one face

Theorem (Our results)

$k = 2$, planar graph, 3 terminals are on one face

⇨ Min-Sum disjoint paths can be found in poly.-time

- Technique: reduction to the CS's result



(P_1, P_2) : shortest disjoint paths

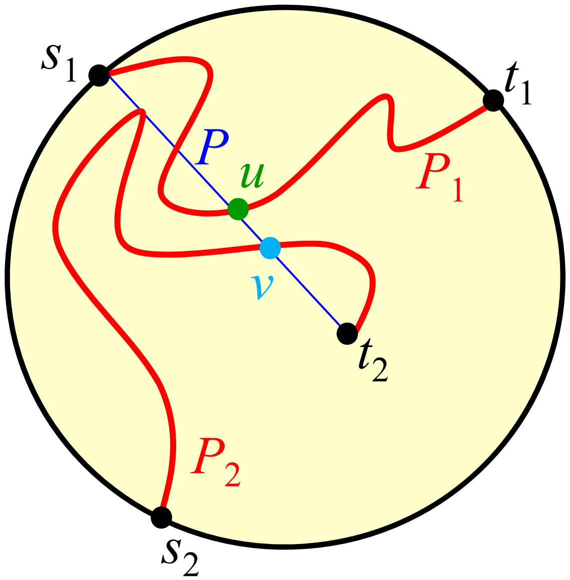
P : shortest path between s_1 and t_2

u : vertex in $P \cap P_1$ closest to t_2

v : vertex in $\underline{P^{[u,t2]}} \cap P_2$ closest to u

subpath of P between u and t_2

Observations



(P_1, P_2) : shortest disjoint paths

P : shortest path between s_1 and t_2

u : vertex in $P \cap P_1$ closest to t_2

v : vertex in $P^{[u,\, t2]} \cap P_2$ closest to u

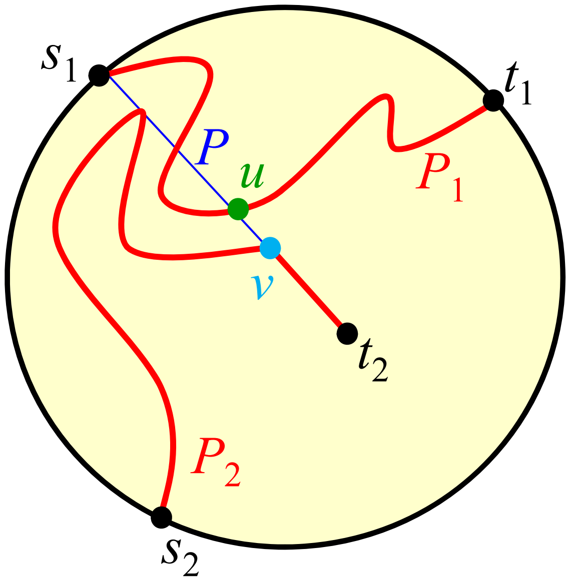
Obs. 1

Internal vertices of $P^{[u,\, v]}$ are not used in (P_1, P_2)

Obs. 2

We may assume that P_2 contains $P^{[v,\, t2]}$

Observations



(P_1, P_2) : shortest disjoint paths

P : shortest path between s_1 and t_2

u : vertex in $P \cap P_1$ closest to t_2

v : vertex in $P^{[u, t2]} \cap P_2$ closest to u

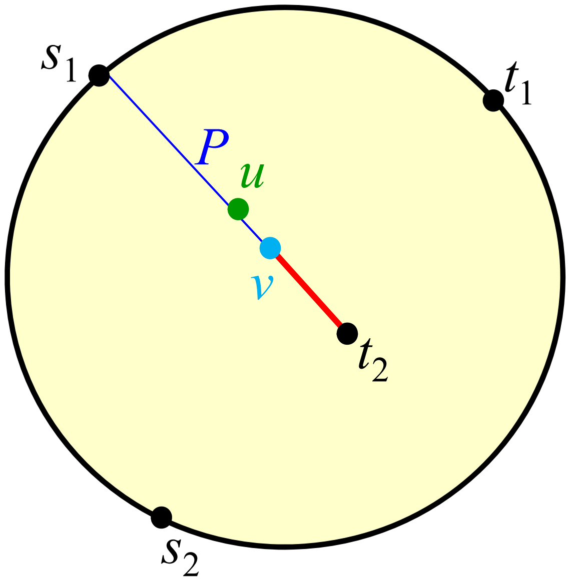
Obs. 1

Internal vertices of $P^{[u, v]}$ are not used in (P_1, P_2)

Obs. 2

We may assume that P_2 contains $P^{[v, t2]}$

How can we find (P_1, P_2) ?



P : shortest path between s_1 and t_2

u, v : fix two candidate vertices

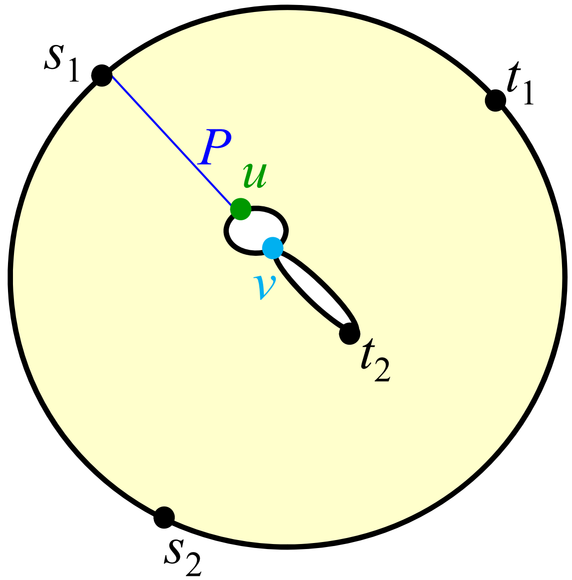
Obs. 1

Internal vertices of $P^{[u,v]}$ are not used in (P_1, P_2)

Obs. 2

We may assume that P_2 contains $P^{[v,t2]}$

How can we find (P_1, P_2) ?



P : shortest path between s_1 and t_2

u, v : fix two candidate vertices

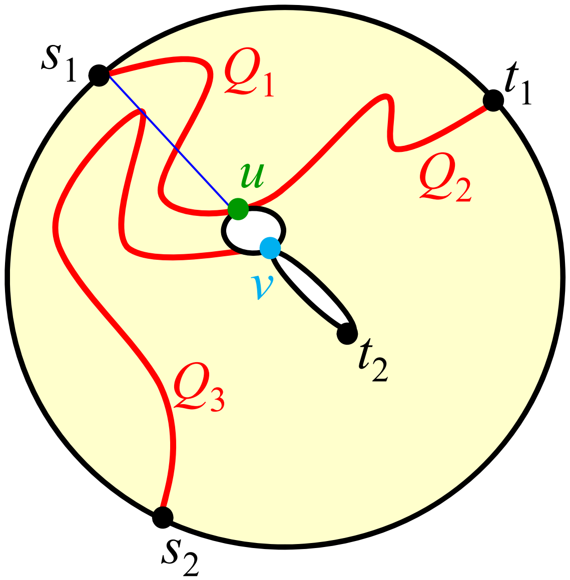
1. Remove vertices in $P^{[u,\, t2]}$ except for v

2. We find three disjoint paths

   $Q_1 : s_1 \rightarrow u \qquad Q_2 : t_1 \rightarrow u \qquad Q_3 : s_2 \rightarrow v$

   with minimum total length

How can we find (P_1, P_2) ?



P : shortest path between s_1 and t_2

u, v : fix two candidate vertices

at most $|V|^2$ possibilities

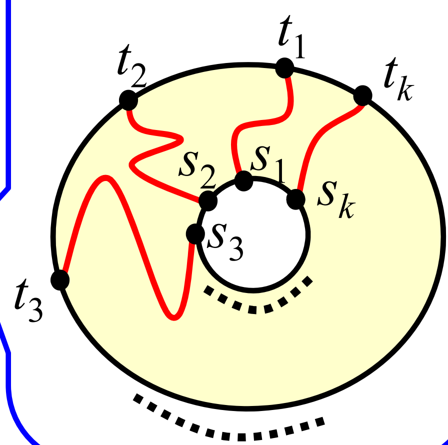
1. Remove vertices in $P^{[u,\, t2]}$ except for v

2. We find three disjoint paths

   $Q_1 : s_1 \to u$ $Q_2 : t_1 \to u$ $Q_3 : s_2 \to v$

   with minimum total length

Algorithm of Colin de Verdière and Schrijver



Our contributions

- Min-Sum Problem

  Poly.-time algorithm for

  - $k = 2$, planar, all terminals are on at most two faces
  - $k = 3$, planar, all terminals are on one face

- Min-Max Problem (k=2)
  - tree-width ≥ 3 : NP-hard
  - tree-width ≤ 2 : Poly.-time algorithm

Min-Max problem

Input: vertex pairs $(s_1, t_1), \ldots, (s_k, t_k)$, length function l

Find: vertex disjoint paths $P_1, \ldots, P_k$ $(P_i : s_i \rightarrow t_i)$

minimizing the length of the longest path: max $l(P_i)$

<u>Observations</u>

- Most cases are NP-hard (ex. k=2, s_1=s_2 , t_1=t_2)

<u>Our results</u>

When $k = 2$,

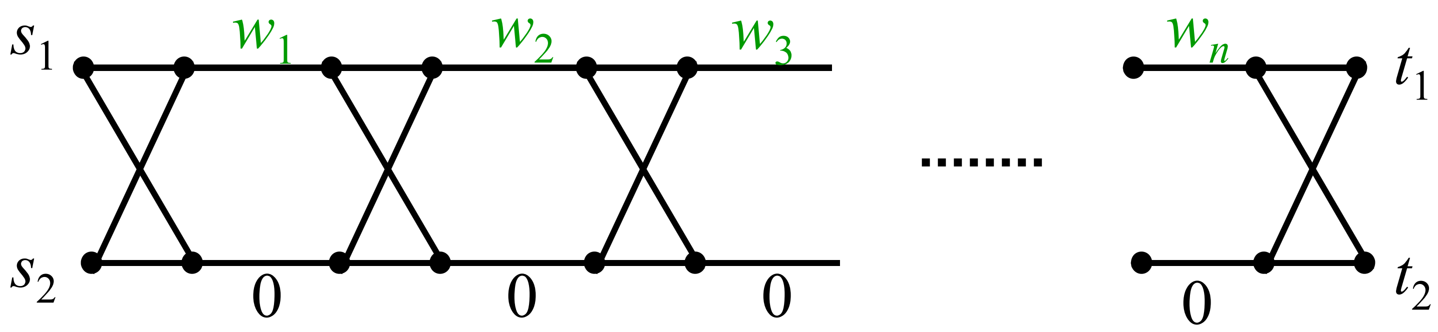
- tree-width ≥ 3 : NP-hard
- tree-width ≤ 2 : Poly.-time algorithm

NP-hardness (tw ≥ 3, k=2)

- Reduction from NP-hard problem "Partition"

  (Divide given integers w_1, w_2, … , w_n into two sets so that the sum of the numbers in each set is equal)

- Consider the following Min-Max Problem

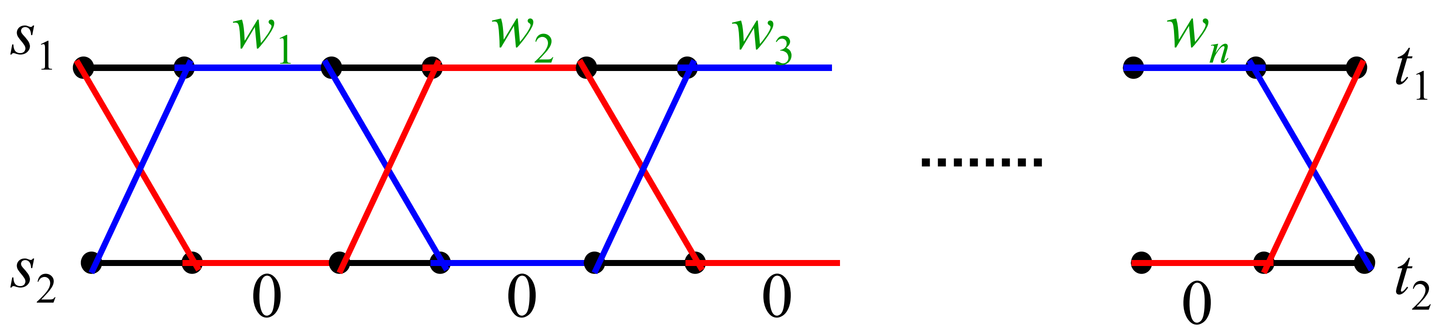


NP-hardness (tw ≥ 3, k=2)

- Reduction from NP-hard problem "Partition"

  Divide given integers $w_1, w_2, \ldots, w_n$ into two sets so that the sum of the numbers in each set is equal

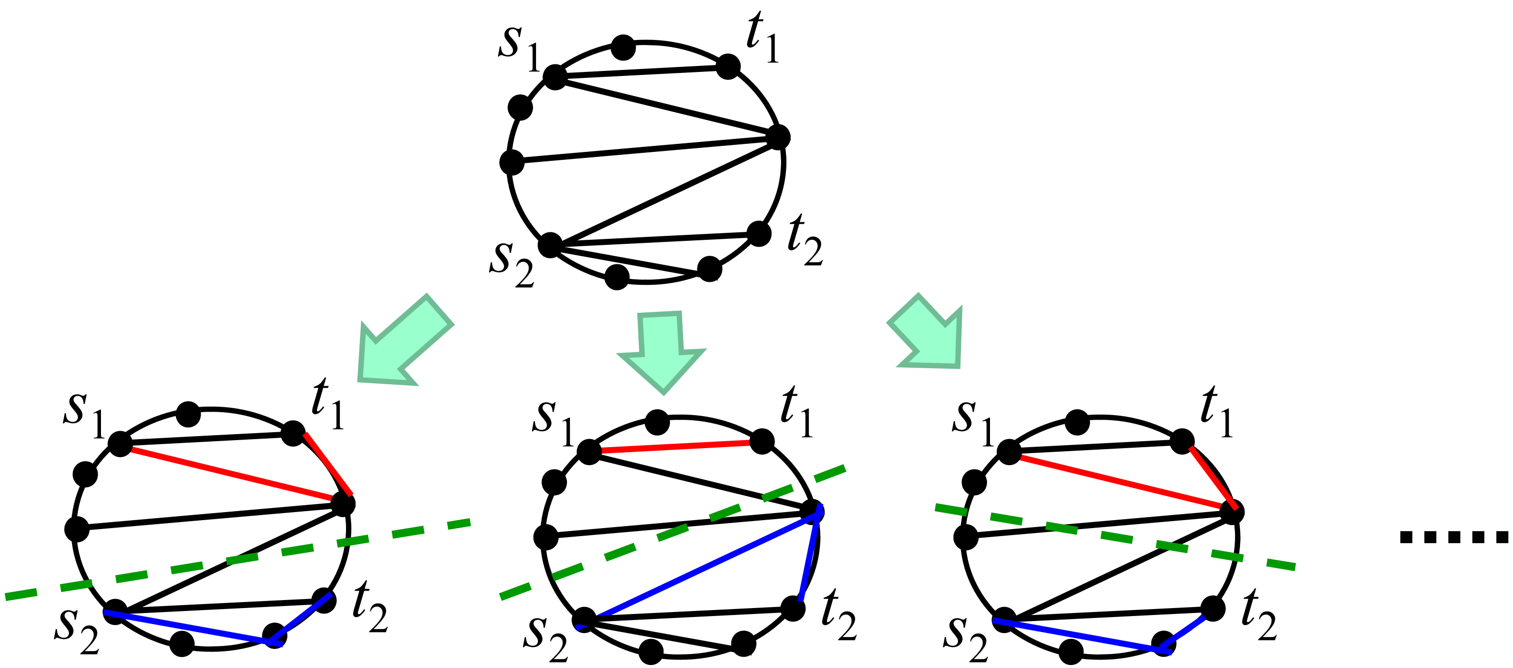
- Consider the following Min-Max Problem



Paths with $l(P_1) = l(P_2)$ exist ⟺ Partition exists

Poly.-time algorithm (tw ≤ 2, k=2)

- Outerplanar graph ⇨ easy to solve
  - consider every partition of the graph
  - find a shortest path in each part

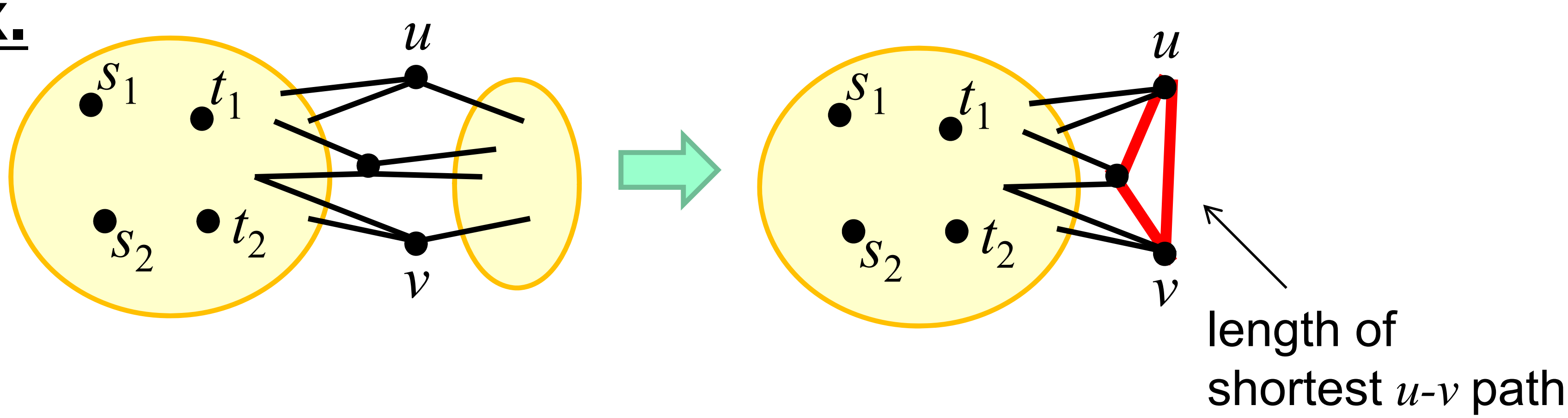


Poly.-time algorithm (tw ≤ 2, $k=2$)

- Outerplanar graph ⇨ easy to solve
  - consider every partition of the graph
  - find a shortest path in each part

- tw ≤ 2 ⇨ Reduction to outerplanar case

ex.



Summary

- Min-Sum Problem

  Poly.-time algorithm for

  - $k = 2$, planar, all terminals are on at most two faces
  - $k = 3$, planar, all terminals are on one face

- Min-Max Problem (k=2)
  - tree-width ≥ 3 : NP-hard
  - tree-width ≤ 2 : Poly.-time algorithm

- Many Open Problems

  ex. Min-Sum problem in $\left\{\begin{array}{l}\text{general}\\ \text{planar}\end{array}\right\}$ graphs for fixed k