An Efficient Dijkstra-Like Algorithm for Finding a Shortest Non-zero Path in Group-Labeled Graphs

<u>**Yutaro Yamaguchi**</u>

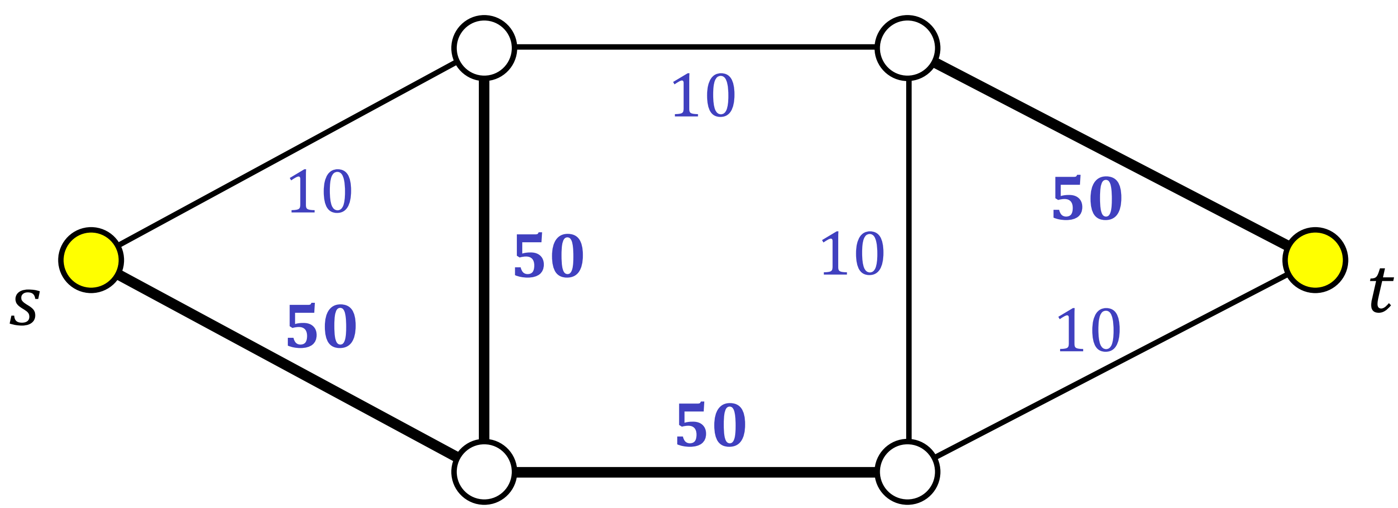
Osaka University, Japan.

HJ 2019 @Tokyo May 27, 2019

Shortest Path Problem

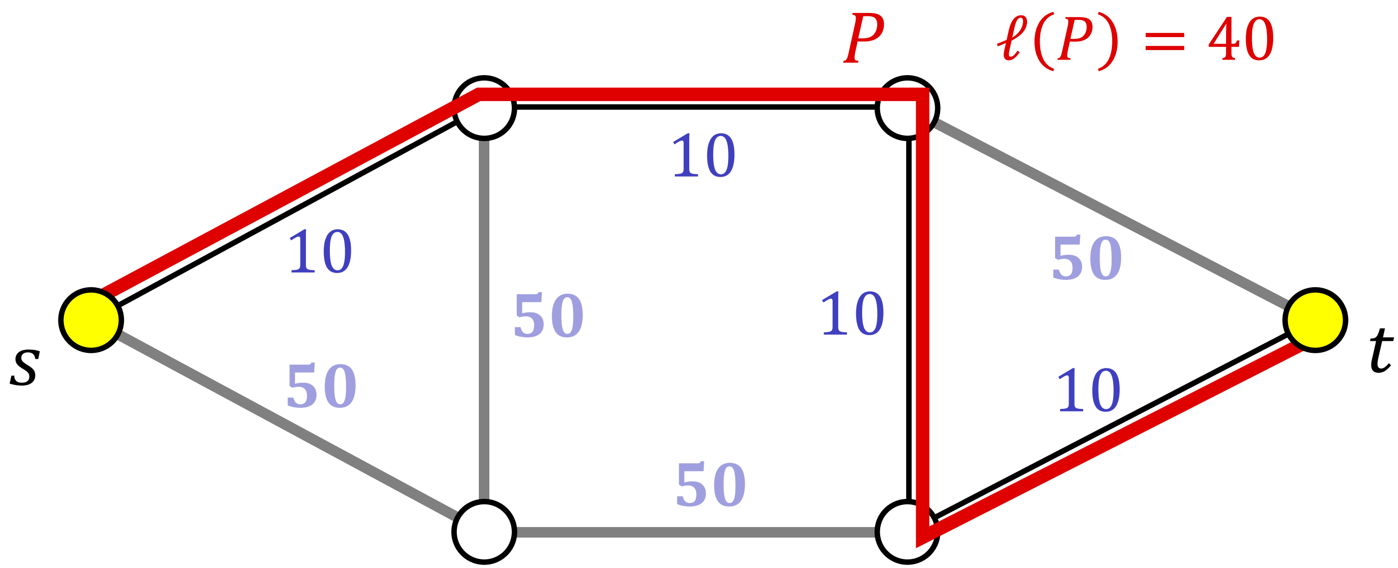
Input $G = (V, E)$: Undirected Graph
$\ell \in \mathbf{R}^{E}_{\geq 0}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest s–t path P in G



Shortest Path Problem

Input	$G = (V, E)$: Undirected Graph
	$\ell \in \mathbf{R}^{E}_{\geq 0}$: Edge Length, $s, t \in V$: Terminals
Goal	Find a shortest s–t path P in G



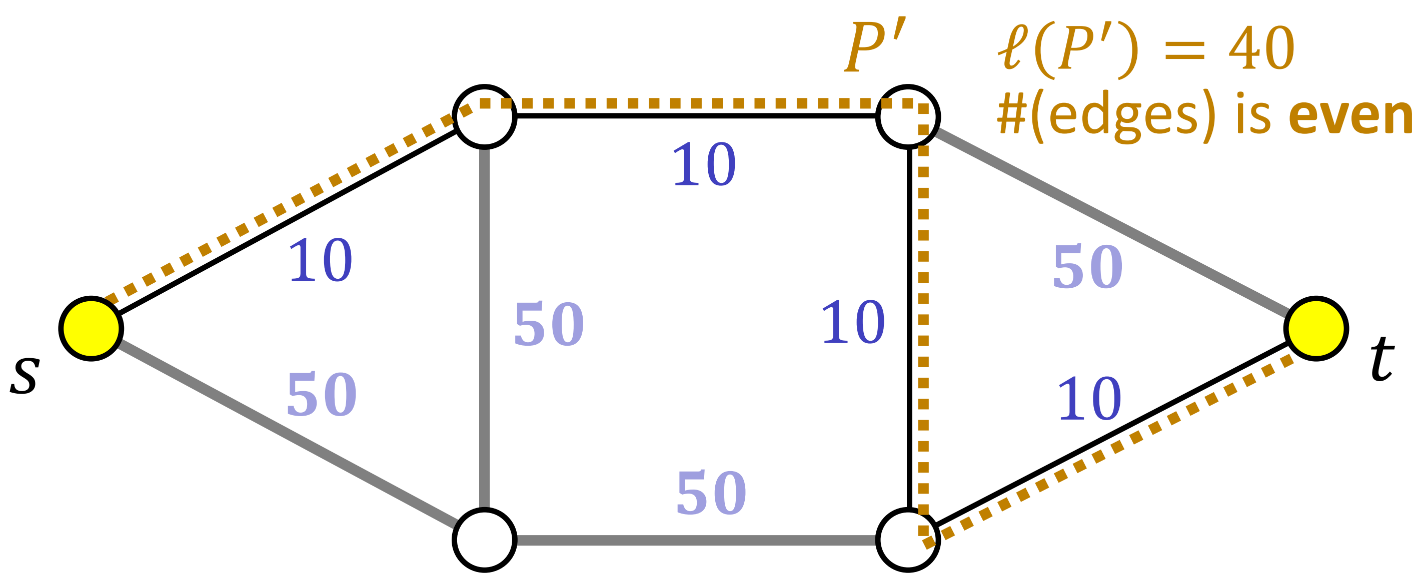
Solved by Dijkstra's Algorithm

Shortest Odd Path Problem

Input $G = (V, E)$: Undirected Graph

$\ell \in \mathbf{R}_{\geq 0}^{E}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **odd** s–t path P in G

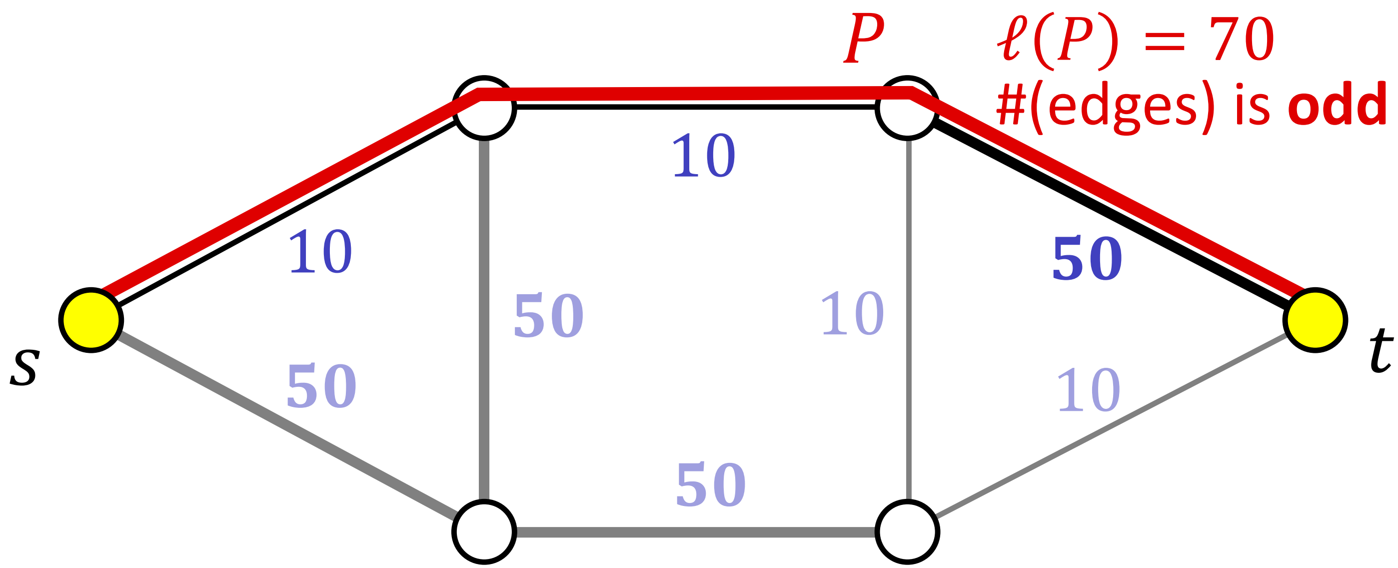


Shortest Odd Path Problem

Input $G = (V, E)$: Undirected Graph

$\ell \in \mathbf{R}_{\geq 0}^{E}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **odd** s–t path P in G



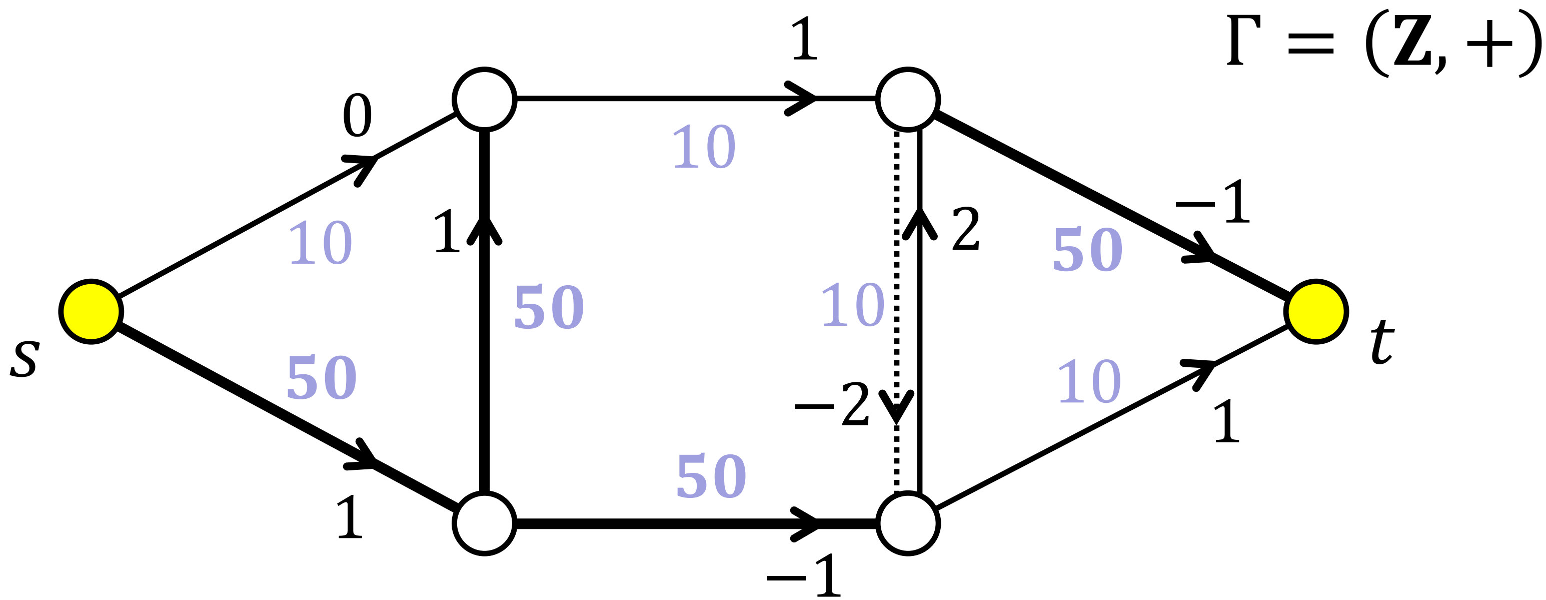
Solved via Weighted Matching

Shortest Non-zero Path Problem

Input $G = (V, E)$: Γ-Labeled Graph (Γ: Group)

$\ell \in \mathbf{R}_{\geq 0}^{E}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **non-zero** s–t path P in G



Shortest Non-zero Path Problem

Input $G = (V, E)$: Γ-Labeled Graph (Γ: Group)

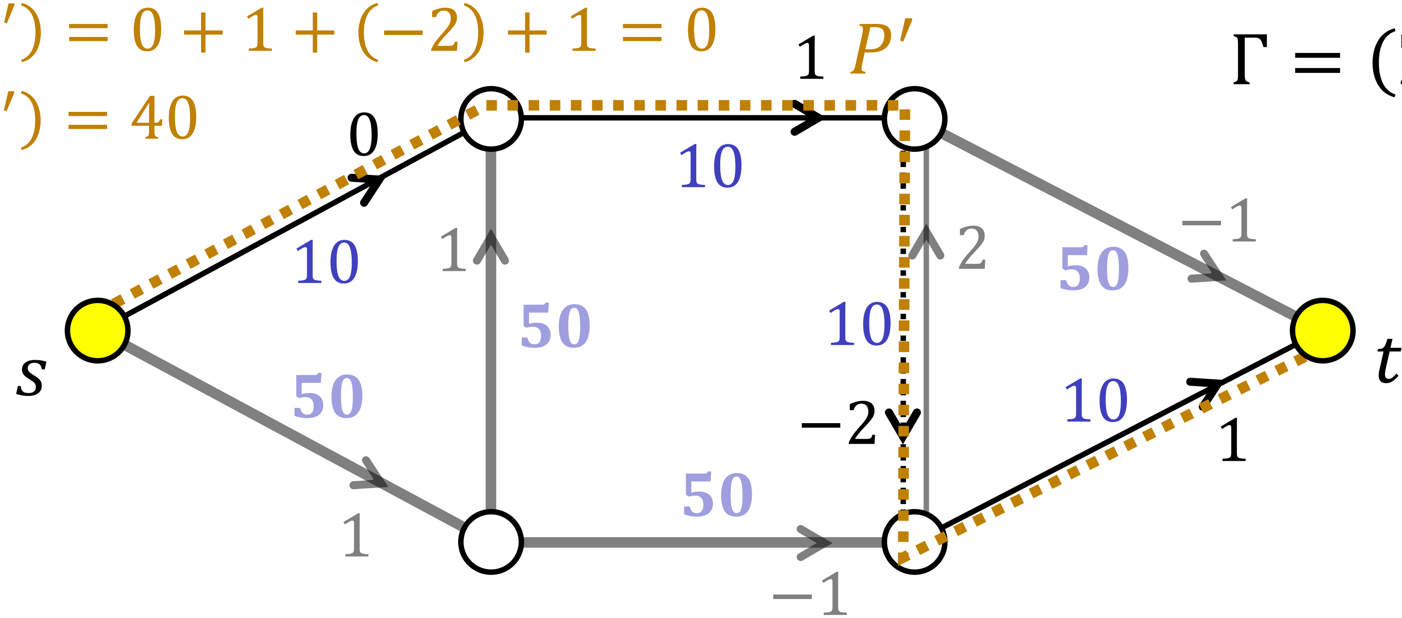
$\ell \in \mathbf{R}_{\geq 0}^{E}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **non-zero** s–t path P in G

$\psi_G(P') = 0 + 1 + (-2) + 1 = 0$

$\ell(P') = 40$

$\Gamma = (\mathbf{Z}, +)$



Shortest Non-zero Path Problem

Input $G = (V, E)$: Γ-Labeled Graph (Γ: Group)

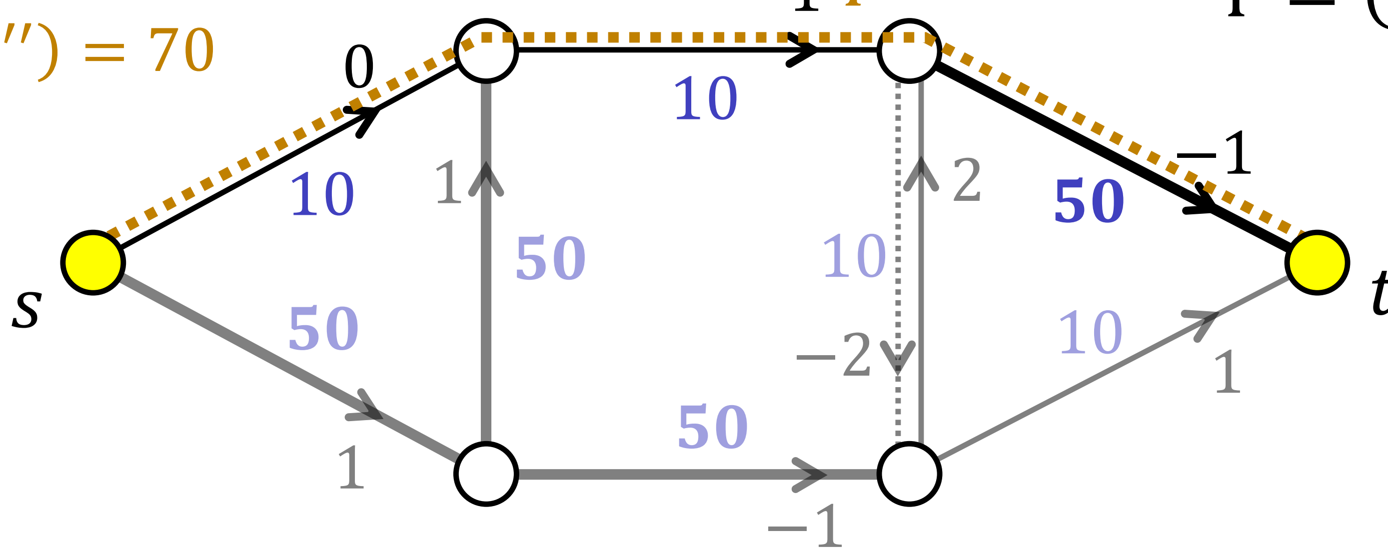
$\ell \in \mathbf{R}_{\geq 0}^{E}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **non-zero** s–t path P in G

$\psi_G(P'') = 0 + 1 + (-1) = 0$

$\ell(P'') = 70$

$\Gamma = (\mathbf{Z}, +)$



Shortest Non-zero Path Problem

Input $G = (V, E)$: Γ-Labeled Graph (Γ: Group)

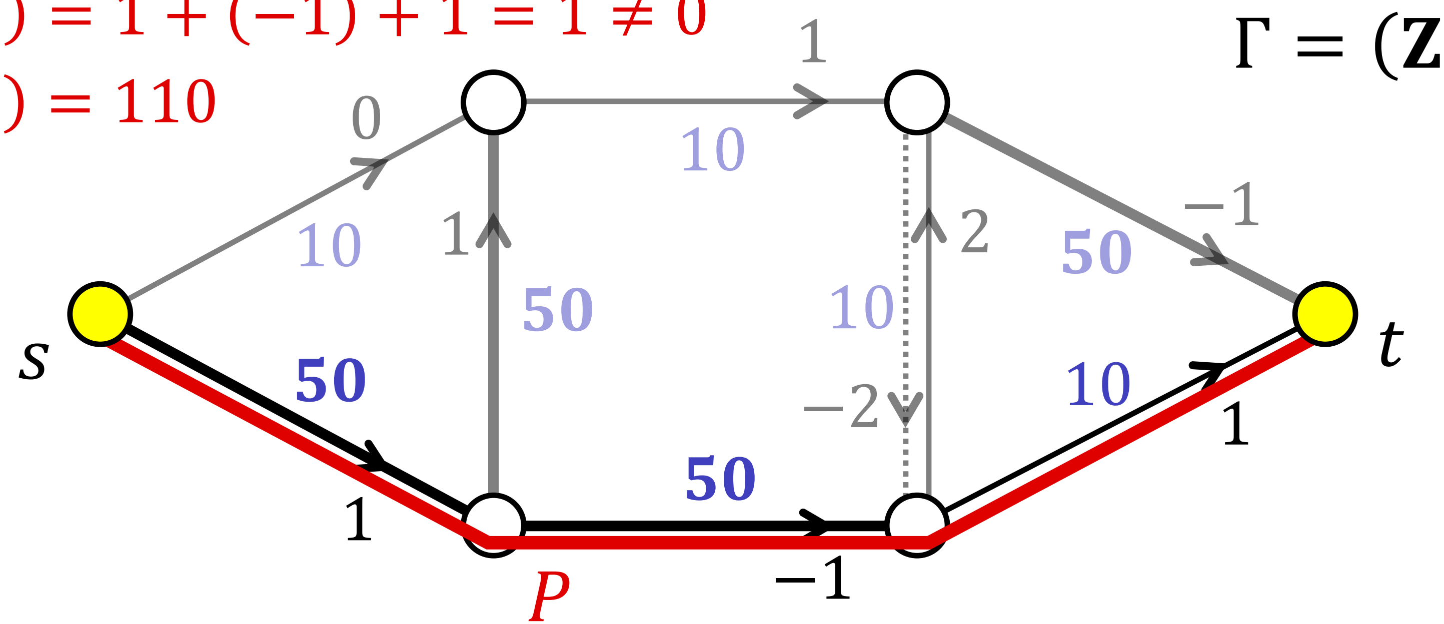
$\ell \in \mathbf{R}^E_{\geq 0}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **non-zero** s–t path P in G

$\psi_G(P) = 1 + (-1) + 1 = 1 \neq 0$

$\ell(P) = 110$

$\Gamma = (\mathbf{Z}, +)$



Shortest Non-zero Path Problem

Input $G = (V, E)$: Γ-Labeled Graph (Γ: Group)

$\ell \in \mathbf{R}^E_{\geq 0}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **non-zero** s–t path P in G

- When $|\Gamma| = 2$, This Problem $\simeq$ **Shortest Odd Path Problem**
- When $\Gamma \simeq \mathbf{Z}_{n_1} \oplus \cdots \oplus \mathbf{Z}_{n_k}$ (i.e., Γ is finite & abelian),
  Randomized Pseudo-Poly via **Permanent Computation**
  [Kobayashi–Toyooka 2017] (HJ 2015)
- When $\Gamma \simeq \mathbf{Z}^r \oplus \mathbf{Z}_{p_1} \oplus \cdots \oplus \mathbf{Z}_{p_k}$ (p_i: prime),
  Deterministic Strongly-Poly via **Weighted Linear Matroid Parity**
  [Y. 2016] + [Iwata–Kobayashi 2017] (JH 2017)
- **Deterministic Strongly-Poly** by **Dijkstra + Recursive Shrinking**
  [Y. 2019] (HJ 2019)

Outline

- Algorithm Framework
  - Basic Idea
  - Auxiliary Problem (Finding a Second Shortest Path)
  - Main Theorem
- Canonical Unbalanced Cycle (CUC)
  - Detour yields a Second Shortest Path (SSP)
  - Shrinking preserves SSP Problem
- Conclusion

Outline

- Algorithm Framework
  - Basic Idea
  - Auxiliary Problem (Finding a Second Shortest Path)
  - Main Theorem
- Canonical Unbalanced Cycle (CUC)
  - Detour yields a Second Shortest Path (SSP)
  - Shrinking preserves SSP Problem
- Conclusion

Basic Idea

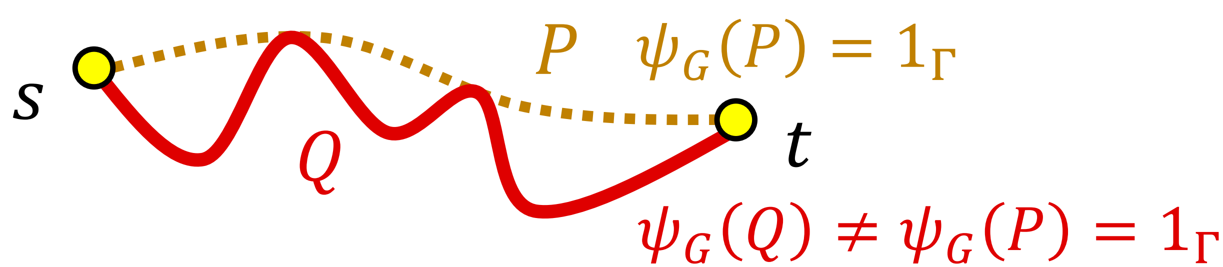
Input	$G=(V,E)$: Γ-Labeled Graph (Γ: Group)
	$\ell \in \mathbf{R}^E_{\geq 0}$: Edge Length, $s,t \in V$: Terminals
Goal	Find a shortest **non-zero** s–t path P in G

1. Find a shortest s–t path P in G by Dijkstra's Algorithm
2. If P is non-zero ($\psi_G(P) \neq 1_\Gamma$), then return P
3. Otherwise, find and return an s–t path Q in G s.t. $\ell(Q)$ is minimized subject to $\psi_G(Q) \neq \psi_G(P)$

Basic Idea

Input	$G = (V, E)$: Γ-Labeled Graph (Γ: Group)
	$\ell \in \mathbf{R}_{\geq 0}^{E}$: Edge Length, $s, t \in V$: Terminals
Goal	Find a shortest **non-zero** s–t path P in G

1. Find a shortest s–t path P in G by Dijkstra's Algorithm
2. If P is non-zero ($\psi_G(P) \neq 1_\Gamma$), then return P
3. Otherwise, find and return an s–t path Q in G s.t. $\ell(Q)$ is minimized subject to $\psi_G(Q) \neq \psi_G(P)$



Auxiliary Problem for Main Task

Input $(G = (V, E), \ell, s, t)$: Original Input

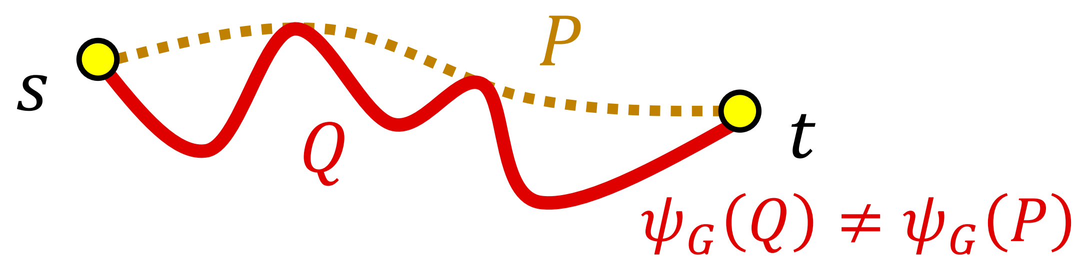
P: Shortest s–t Path in G

Goal Find an s–t path Q in G **second shortest** to P

Def.

1. Find a shortest s–t path P in G by Dijkstra's Algorithm
2. If P is non-zero $(\psi_G(P) \neq 1_\Gamma)$, then return P
3. Otherwise, find and return an s–t path Q in G s.t.

$\ell(Q)$ is minimized subject to $\psi_G(Q) \neq \psi_G(P)$



Auxiliary Problem for Main Task

Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

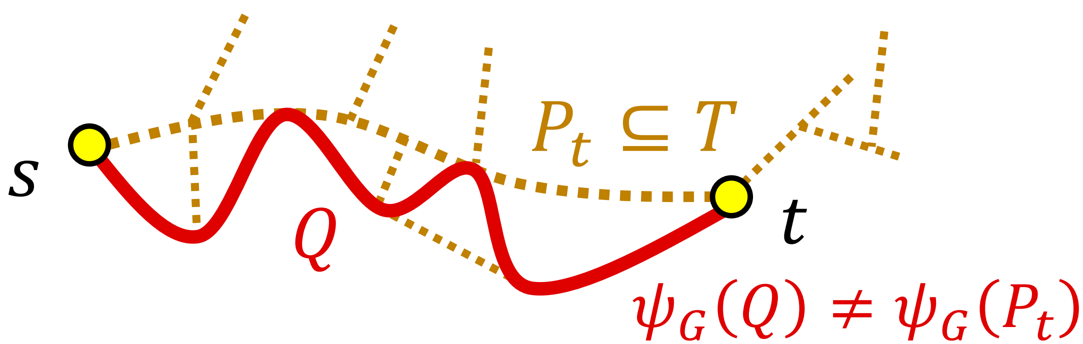
Goal Find an s–t path Q in G **second shortest** to P_t

1. Find a shortest s–t path P in G by **Dijkstra's Algorithm**

Output

Def.

A **Tree** in which each s–v path P_v is shortest in G



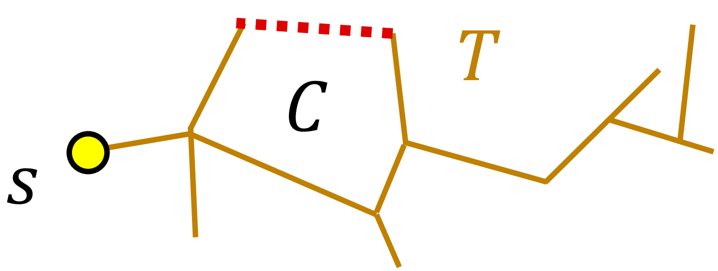
Idea to Find a Second Shortest Path (SSP)

Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t

1. Find a "nice" unbalanced cycle C $(\psi_G(C) \neq 1_\Gamma)$
2. If t is on C, then return a **Detour** Q from P_t around C
3. Otherwise, shrink C into a single vertex b, and recursively solve a small instance



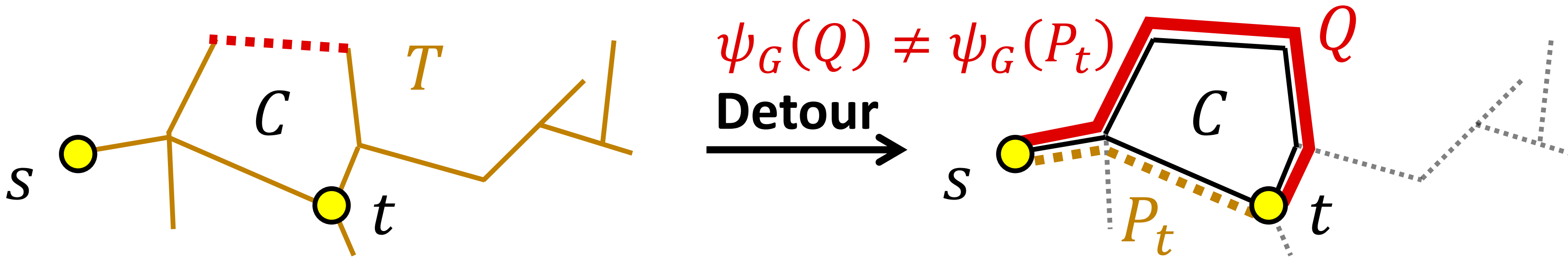
Idea to Find a Second Shortest Path (SSP)

Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t

1. Find a "nice" unbalanced cycle C $(\psi_G(C) \neq 1_\Gamma)$
2. If t is on C, then return a **Detour** Q from P_t around C
3. Otherwise, shrink C into a single vertex b, and recursively solve a small instance



Idea to Find a Second Shortest Path (SSP)

Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t

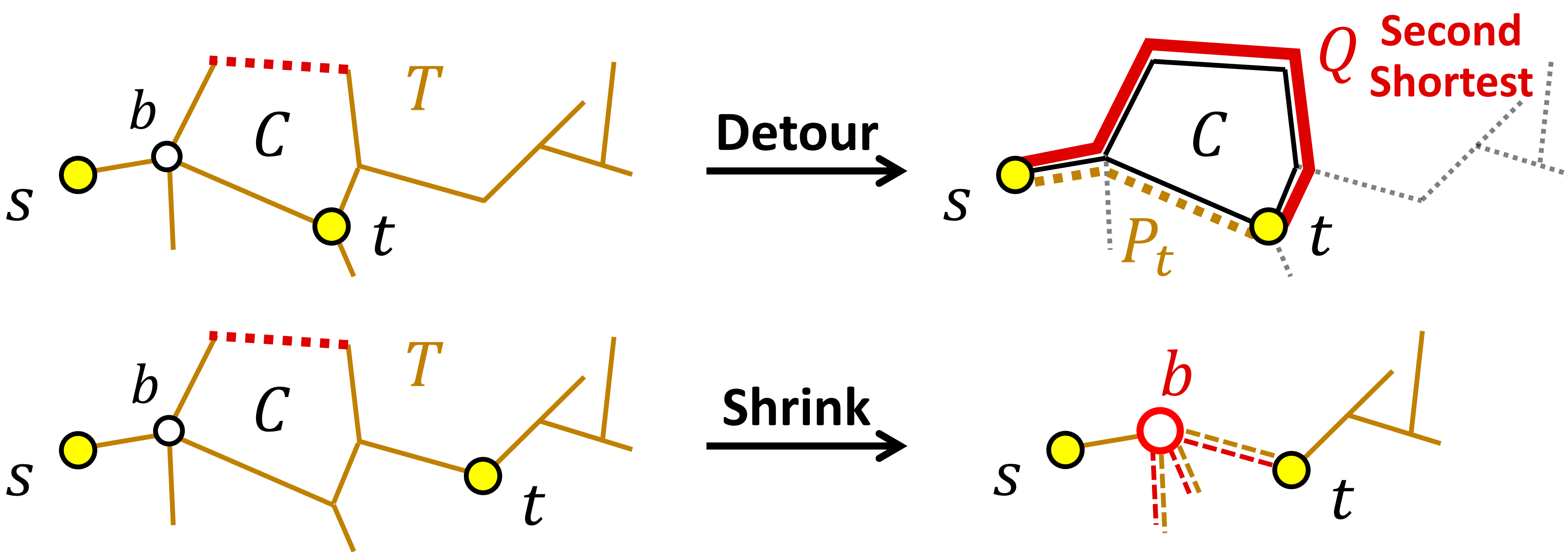
1. Find a "nice" unbalanced cycle C $(\psi_G(C) \neq 1_\Gamma)$
2. If t is on C, then return a **Detour** Q from P_t around C
3. Otherwise, shrink C into a single vertex b, and recursively solve a small instance

Main Theorem (Informal)

Thm. $\exists C$: Unbalanced Cycle with $b \in V(C)$ s.t.

- For a vertex in $C - b$, a detour Q around C is an **SSP**
- After shrinking C into b, **corresponding SSPs** remain

Moreover, such C can be found in $O(|E|)$ time



On Computational Time

Thm. $\exists C$: Unbalanced Cycle with $b \in V(C)$ s.t.

- For a vertex in $C - b$, a detour Q around C is an **SSP**
- After shrinking C into b, **corresponding SSPs** remain

Moreover, such C can be found in $\mathrm{O}(|E|)$ time

Obs. Shrinking occurs at most $(|V| - 2)$ times

Cor. An SSP can be found in $\mathrm{O}(|V| \cdot |E|)$ time (if exists)

#(vertices) is reduced

Outline

- Algorithm Framework
  - Basic Idea
  - Auxiliary Problem (Finding a Second Shortest Path)
  - Main Theorem
- Canonical Unbalanced Cycle (CUC)
  - Detour yields a Second Shortest Path (SSP)
  - Shrinking preserves SSP Problem
- Conclusion

Canonical Unbalanced Cycle (CUC)

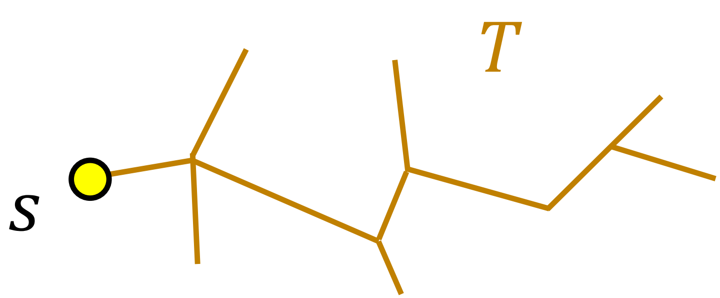
Def. $T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

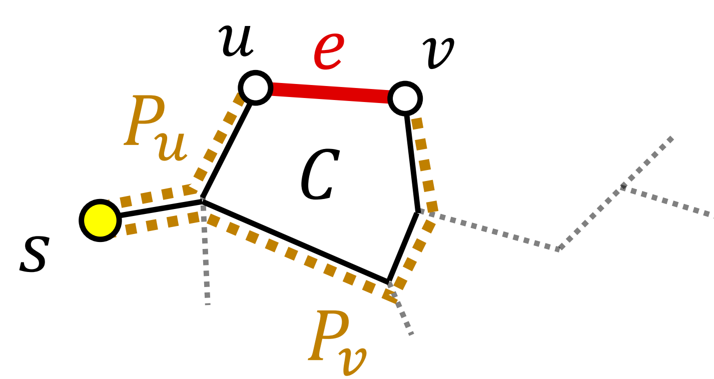
C is a **Canonical Unbalanced Cycle**

$$\Updownarrow$$

- $C \setminus T = \{e\}$
- $e \in \underset{e' = \{u,v\}}{\arg\min} \left\{ \ell(P_u) + \ell(P_v) + \ell(e') \mid \psi_G\left(P_u * e' * \overline{P_v}\right) \neq 1_\Gamma \right\}$

$P_u * e * \overline{P_v}$ is a Shortest Non-zero s–s Walk in G





Detour around a CUC yields an SSP

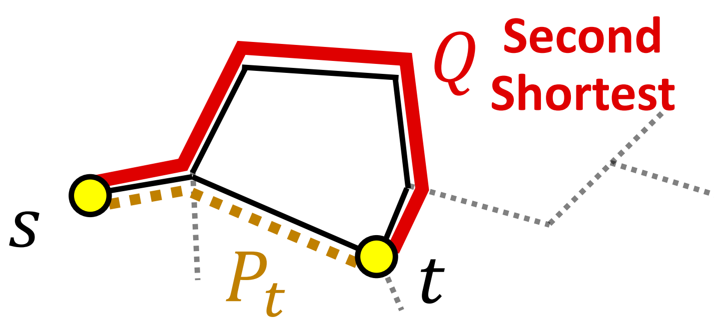
Def. $T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

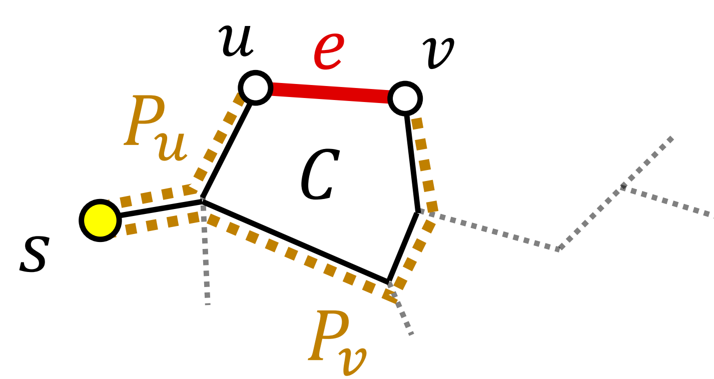
C is a **Canonical Unbalanced Cycle**

$$\Updownarrow$$

- $C \setminus T = \{e\}$
- $e \in \underset{e' = \{u,v\}}{\arg\min} \left\{ \ell(P_u) + \ell(P_v) + \ell(e') \mid \psi_G\left(P_u * e' * \overline{P_v}\right) \neq 1_\Gamma \right\}$

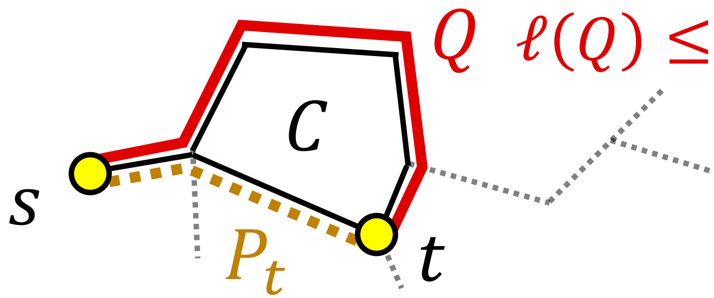
Lem. On a CUC C, every detour Q from T is an **SSP**

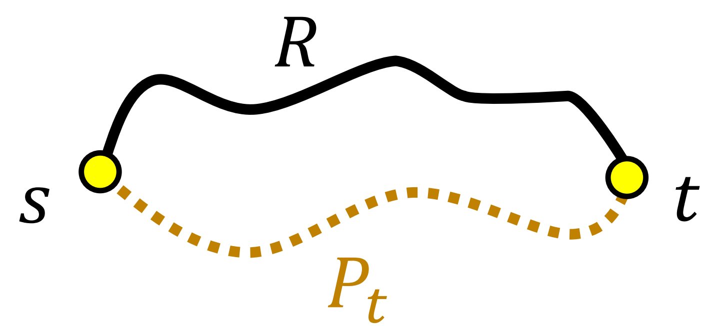




Detour around a CUC yields an SSP

Lem. On a CUC C, every detour Q from T is an **SSP**



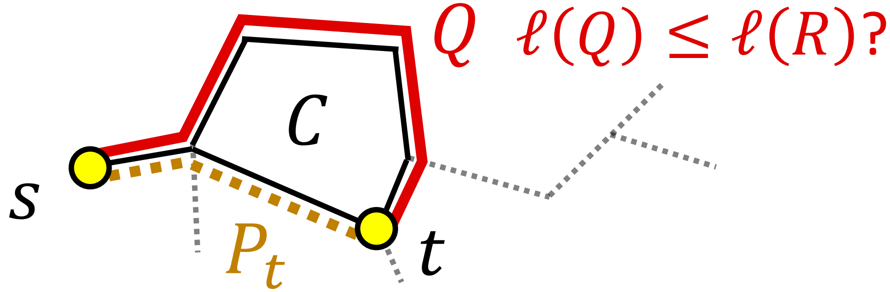


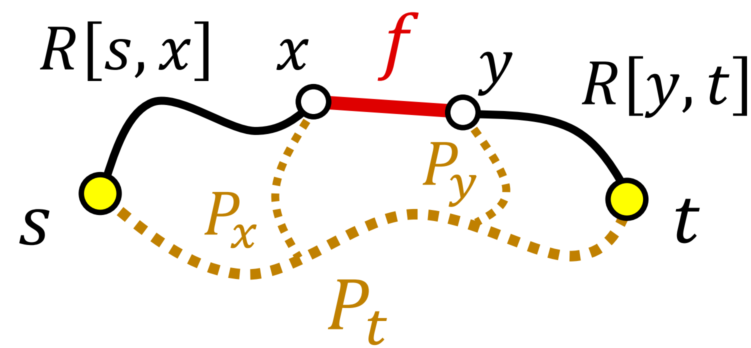
Fix an s–t path R with $\psi_G(R) \neq \psi_G(P_t)$

- $\exists f = \{x, y\} \in R$ s.t. $\psi_G\left(P_x * f * \overline{P_y}\right) \neq 1_\Gamma$
  $\rightarrow \ell(P_x) + \ell\left(P_y\right) + \ell(f) \geq \ell(P_t) + \ell(Q)$ (C is a CUC)
- $\ell(R[s, x]) \geq \ell(P_x)$ (P_x is shortest)
- $\ell(R[y, t]) \geq \left|\ell\left(P_y\right) - \ell(P_t)\right|$ (o/w, $\exists$shortcut for T)

Detour around a CUC yields an SSP

Lem. On a CUC C, every detour Q from T is an **SSP**





Fix an s–t path R with $\boxed{\psi_G(R) \neq \psi_G(P_t)}$

- $\exists f = \{x,y\} \in R$ s.t. $\psi_G\left(P_x * f * \overline{P_y}\right) \neq 1_\Gamma$

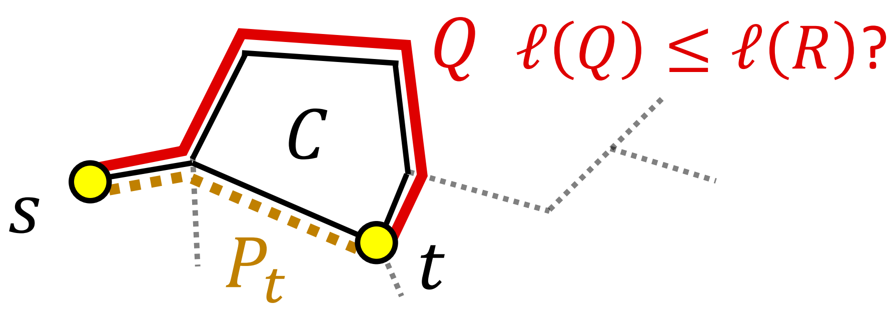
  o/w, $\psi_G(R) = \prod_{f=\{x,y\}\in R} \psi_G(P_x)^{-1} \cdot \psi_G\left(P_y\right)$

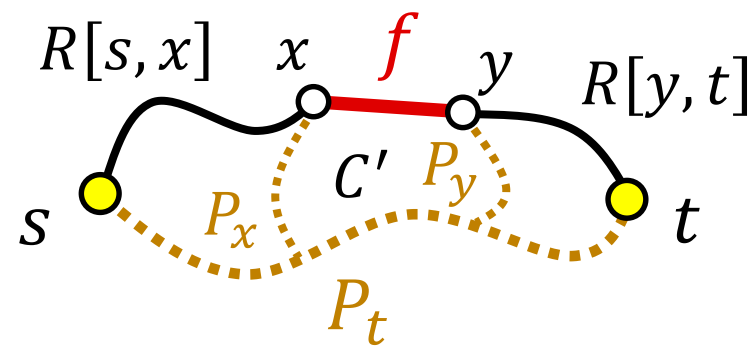
  $= \psi_G(P_s)^{-1} \cdot \psi_G(P_t) \boxed{= \psi_G(P_t)}$

Contradiction!

Detour around a CUC yields an SSP

Lem. On a CUC C, every detour Q from T is an **SSP**



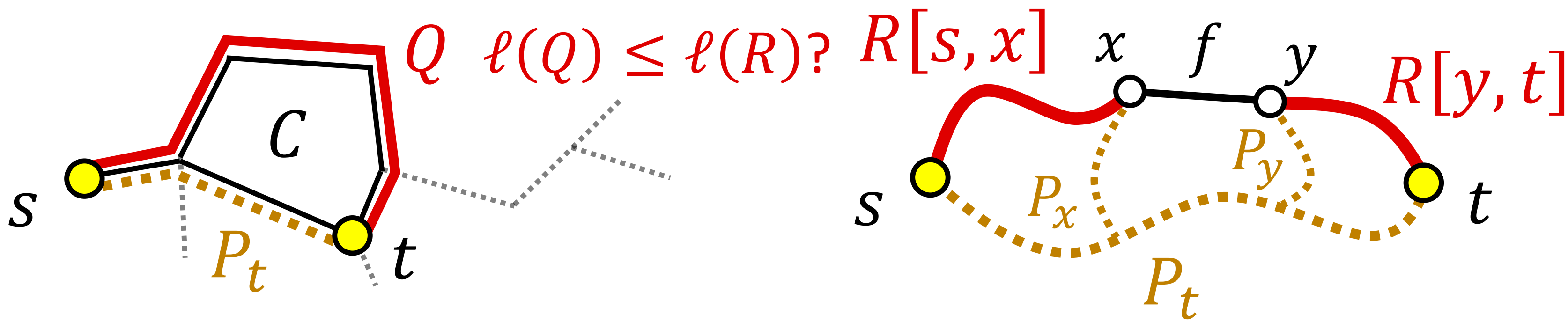


Fix an s–t path R with $\psi_G(R) \neq \psi_G(P_t)$

- $\exists f = \{x, y\} \in R$ s.t. $\psi_G\left(P_x * f * \overline{P_y}\right) \neq 1_\Gamma$
  $\rightarrow \ell(P_x) + \ell\left(P_y\right) + \ell(f) \geq \ell(P_t) + \ell(Q)$ (C is a CUC)

Detour around a CUC yields an SSP

Lem. On a CUC C, every detour Q from T is an **SSP**

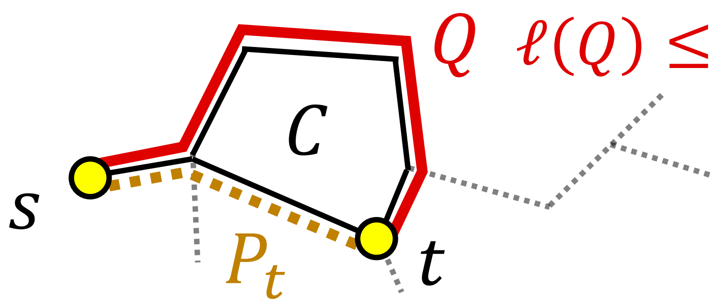


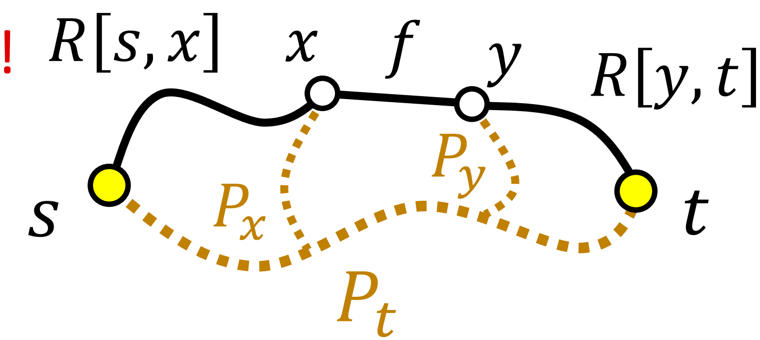
Fix an s–t path R with $\psi_G(R) \neq \psi_G(P_t)$

- $\exists f = \{x, y\} \in R$ s.t. $\psi_G\left(P_x * f * \overline{P_y}\right) \neq 1_\Gamma$
  $\rightarrow \ \ell(P_x) + \ell\left(P_y\right) + \ell(f) \geq \ell(P_t) + \ell(Q)$ (C is a CUC)
- $\ell(R[s,x]) \geq \ell(P_x)$ (P_x is shortest)
- $\ell(R[y,t]) \geq \left|\ell\left(P_y\right) - \ell(P_t)\right|$ (o/w, ∃shortcut for T)

Detour around a CUC yields an SSP

<u>**Lem.**</u> On a CUC C, every detour Q from T is an **SSP**





$$\boxed{\ell(R) \geq \ell(P_x) + \ell(f) + \ell\left(P_y\right) - \ell(P_t) \geq \ell(Q)}$$

$$\ell(P_x) + \ell\left(P_y\right) + \ell(f) \geq \ell(P_t) + \ell(Q) \quad (C \text{ is a CUC})$$

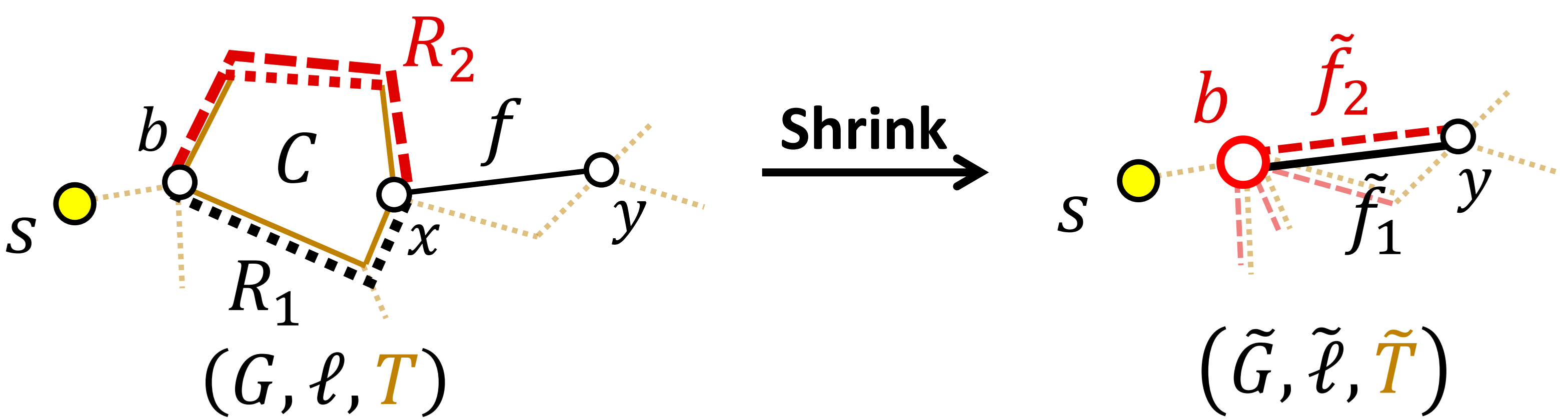
- $\ell(R[s,x]) \geq \ell(P_x)$ (P_x is shortest)
- $\ell(R[y,t]) \geq \left|\ell\left(P_y\right) - \ell(P_t)\right|$ (o/w, ∃shortcut for T)

Shrinking preserves SSP Problem

Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t



$$\psi_{\tilde{G}}(\tilde{f}_i; b \to y) \coloneqq \psi_G(R_i * f)$$

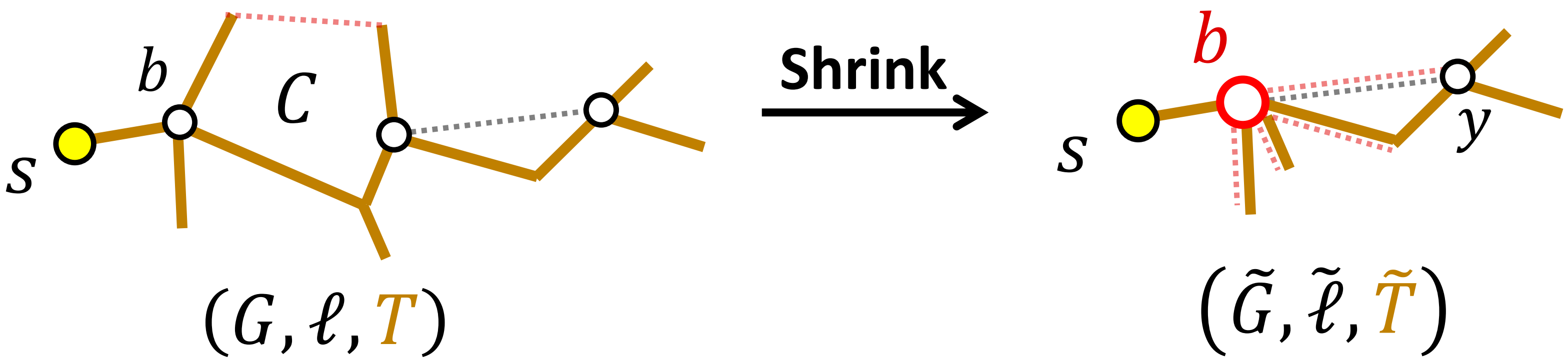
$$\tilde{\ell}(\tilde{f}_i) \coloneqq \ell(R_i) + \ell(f)$$

Shrinking preserves SSP Problem

Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t



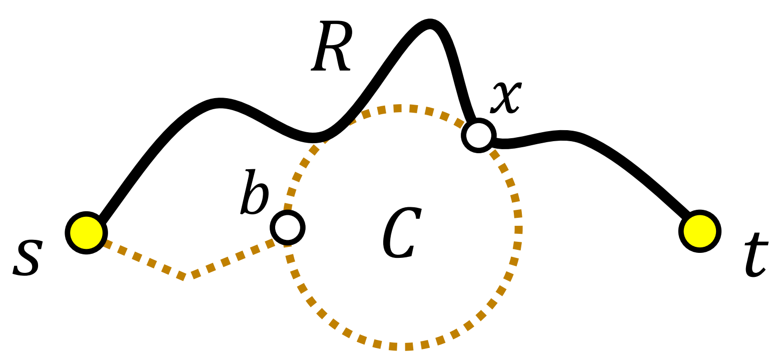
$$\psi_{\tilde{G}}(\tilde{f}_i; b \to y) := \psi_G(R_i * f)$$
$$\tilde{\ell}(\tilde{f}_i) := \ell(R_i) + \ell(f)$$

Shrinking preserves SSP Problem

Input	$(G = (V, E), \ell, s, t)$: Original Input
	$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s
Goal	Find an s–t path Q in G **second shortest** to P_t

R: s–t path, x: Last Vertex on R intersecting $P_b \cup C$

Case 1. $x \in C - b$



Shrinking preserves SSP Problem

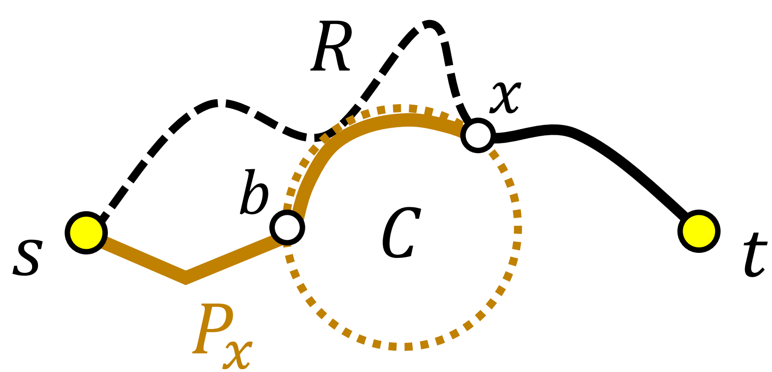
Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t

R: s–t path, x: Last Vertex on R intersecting $P_b \cup C$

Case 1. $x \in C - b$



$$\psi_G(R[s,x]) = \psi_G(P_x)$$
$$\Downarrow$$
$$\ell(R) \geq \ell(P_x) + \ell(R[x,t])$$

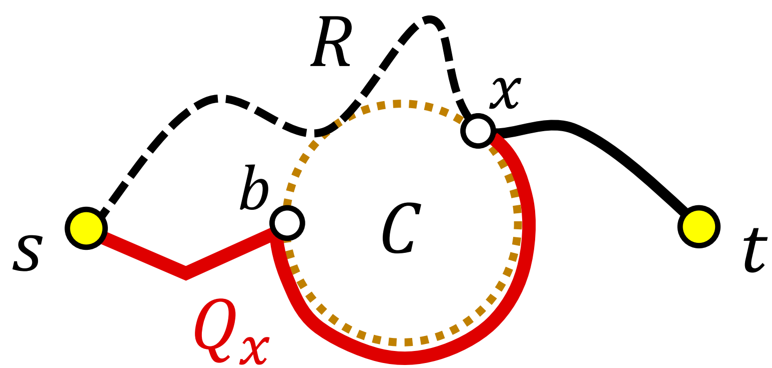
P_x is enough instead of $R[s,x]$

Shrinking preserves SSP Problem

Input	$(G=(V,E),\ell,s,t)$: Original Input
	$T=\bigcup_{v\in V}P_v$: Shortest Path Tree of G rooted at s
Goal	Find an s–t path Q in G **second shortest** to P_t

R: s–t path, x: Last Vertex on R intersecting $P_b\cup C$

Case 1. $x\in C-b$



$$\psi_G(R[s,x])\neq\psi_G(P_x)$$
$$\Downarrow$$
$$\psi_G(Q_x)\neq\psi_G(P_x)$$
$$\ell(R)\geq\ell(Q_x)+\ell(R[x,t])$$

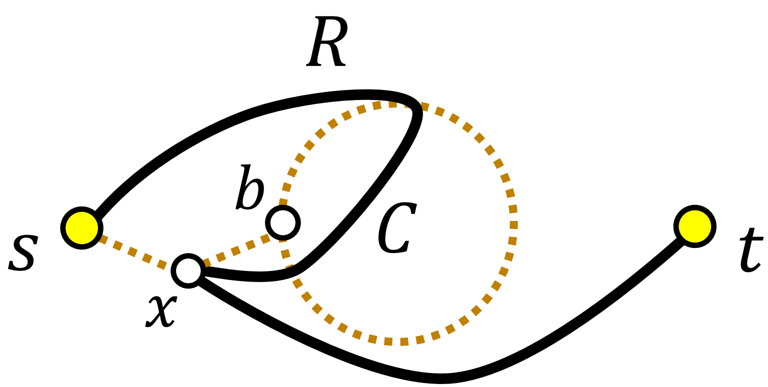
Q_x is enough instead of $R[s,x]$

Shrinking preserves SSP Problem

Input	$(G = (V, E), \ell, s, t)$: Original Input
	$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s
Goal	Find an s–t path Q in G **second shortest** to P_t

R: s–t path, x: Last Vertex on R intersecting $P_b \cup C$

Case 2. $x \in P_b$

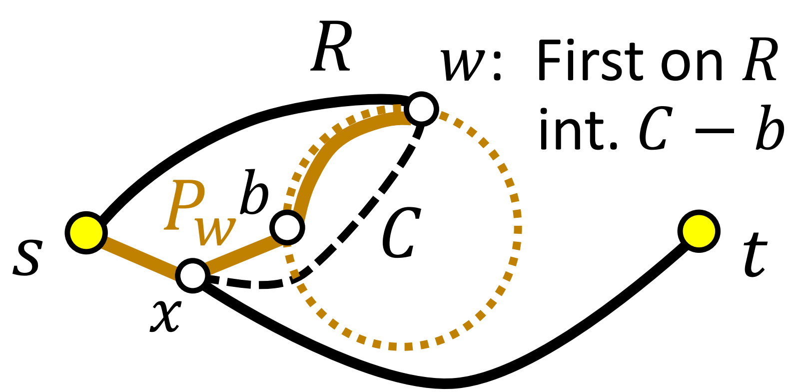


Shrinking preserves SSP Problem

Input	$(G = (V, E), \ell, s, t)$: Original Input
	$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s
Goal	Find an s–t path Q in G **second shortest** to P_t

R: s–t path, x: Last Vertex on R intersecting $P_b \cup C$

Case 2. $x \in P_b$



$$\psi_G(R[w,x]) = \psi_G(\overline{P_w}[w,x])$$
$$\Downarrow$$
$$\ell(R[w,x]) \geq \ell(\overline{P_w}[w,x])$$

$\overline{P_w}[w,x]$ is enough instead of $R[w,x]$

Shrinking preserves SSP Problem

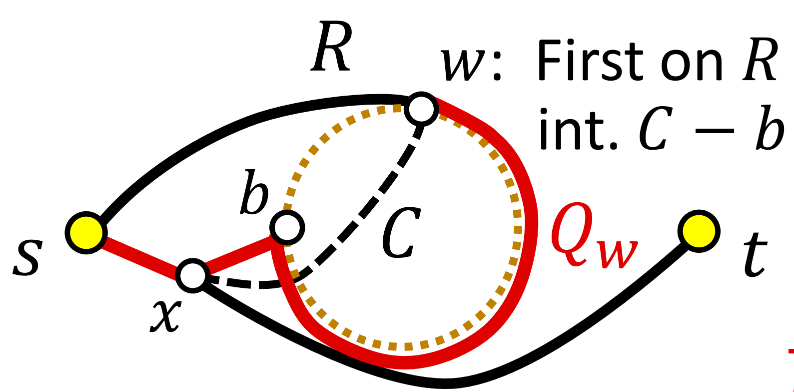
Input $(G = (V, E), \ell, s, t)$: Original Input

$T = \bigcup_{v \in V} P_v$: Shortest Path Tree of G rooted at s

Goal Find an s–t path Q in G **second shortest** to P_t

R: s–t path, x: Last Vertex on R intersecting $P_b \cup C$

Case 2. $x \in P_b$



$$\psi_G(R[w,x]) \neq \psi_G(\overline{P_w}[w,x])$$
$$\Downarrow$$
$$\psi_G(\overline{Q_w}[w,x]) \neq \psi_G(\overline{P_w}[w,x])$$
$$\ell(R[w,x]) \geq \ell(\overline{Q_w}[w,x])$$

$\overline{Q_w}[w,x]$ is enough instead of $R[w,x]$

Outline

- Algorithm Framework
  - Basic Idea
  - Auxiliary Problem (Finding a Second Shortest Path)
  - Main Theorem
- Canonical Unbalanced Cycle (CUC)
  - Detour yields a Second Shortest Path (SSP)
  - Shrinking preserves SSP Problem
- Conclusion

Conclusion

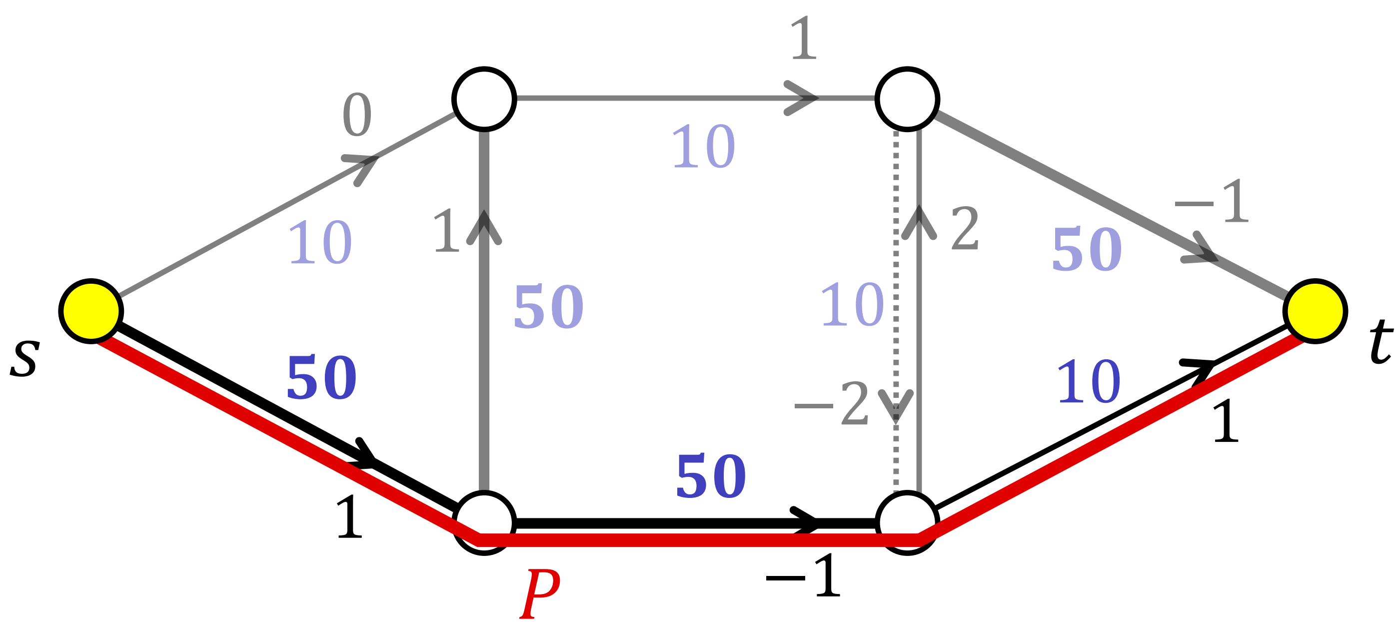
Input $G = (V, E)$: Γ-Labeled Graph (Γ: Group)

$\ell \in \mathbf{R}^E_{\geq 0}$: Edge Length, $s, t \in V$: Terminals

Goal Find a shortest **non-zero** s–t path P in G

Thm. Solved by $O(|V| \cdot |E|)$ elementary operations

[Y. 2019]



Conclusion

Input	$G = (V, E)$: Γ-Labeled Graph (Γ: Group)
	$\ell \in \mathbf{R}^{E}_{\boxed{\geq 0}}$: Edge Length, $s, t \in V$: Terminals
Goal	Find a shortest **non-zero** s–t path P in G

Thm. Solved by $\mathrm{O}(|V| \cdot |E|)$ elementary operations

[Y. 2019]

- Dijkstra + Shrinking **Canonical Unbalanced Cycles**
- Depending heavily on **Nonnegativity of Edge Length**

Q. How about a general input "without **Negative Cycle**"?

[Unconstrained] **Strongly-Poly** via **Weighted Matching**

[Parity Constrained] Open... (as long as I know)