Undirected Replacement Paths: Dual Fault Reduces to Single Source

Jakob Nogler[1] **Virginia Vassilevska Williams**[1]

[1]Massachusetts Institute of Technology

UNDIRECTED REPLACEMENT PATHS : DUAL FAULT REDUCES TO SINGLE SOURCE

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E$.

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$.

Note: Output size is $\mathcal{O}(n)$

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	nm		
Directed Graphs	Yes	nm		
	No	nm		

UNDIRECTED REPLACEMENT PATHS: DUAL FAULT REDUCES TO SINGLE SOURCE

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	nm		
	No	nm		

*References can be found on the paper

UNDIRECTED REPLACEMENT PATHS : DUAL FAULT REDUCES TO SINGLE SOURCE

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

Claim: Directed Replacement Path is possible in the same time as All-Pair Shortest Path (APSP)

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3 / \exp(\Omega(\sqrt{\log n}))$		
	No	$n^{2.53\ldots}$		

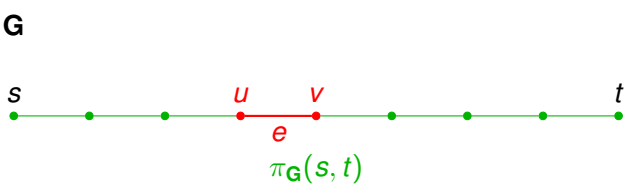
*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

Claim: Directed Replacement Path is possible in the same time as All-Pair Shortest Path (APSP)



$\pi_{\mathbf{G}\setminus e}(s,t)$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$		
	No	$n^{2.53\ldots}$		

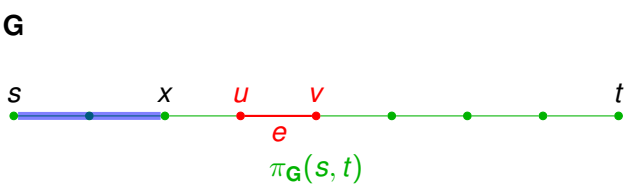
*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

Claim: Directed Replacement Path is possible in the same time as All-Pair Shortest Path (APSP)



$\pi_{\mathbf{G}\setminus e}(s,t) = \pi_{\mathbf{G}}(s,x) \circ$ for $x \preceq u$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3 / \exp(\Omega(\sqrt{\log n}))$		
	No	$n^{2.53\ldots}$		

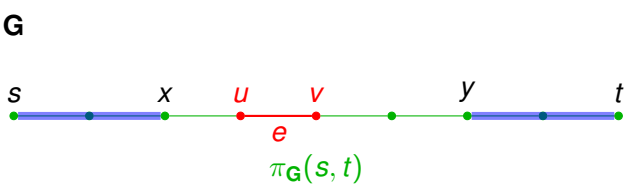
*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

Claim: Directed Replacement Path is possible in the same time as All-Pair Shortest Path (APSP)



$\pi_{\mathbf{G}\setminus e}(s,t) = \pi_{\mathbf{G}}(s,x) \circ \qquad \circ \pi_{\mathbf{G}}(y,t)$ for $x \preceq u$ and $v \preceq y$.

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$		
	No	$n^{2.53\ldots}$		

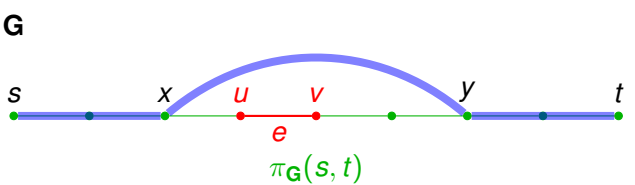
*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

Claim: Directed Replacement Path is possible in the same time as All-Pair Shortest Path (APSP)



$$\pi_{\mathbf{G}\setminus e}(s,t) = \pi_{\mathbf{G}}(s,x) \circ \pi_{\mathbf{G}\setminus \pi_{\mathbf{G}}(s,t)}(x,y) \circ \pi_{\mathbf{G}}(y,t) \quad \text{for } x \preceq u \text{ and } v \preceq y.$$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3 / \exp(\Omega(\sqrt{\log n}))$		
	No	$n^{2.53\ldots}$		

*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3 / \exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	$n^{2.53\ldots}$		

*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3 / \exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	$n^\omega = n^{2.37\ldots}$		

*References can be found on the paper

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	$n^\omega = n^{2.37\ldots}$		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial

*References can be found on the paper

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E$

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E$

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E$.

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E$

Single Source Replacement Path Problem: Fixed s compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E, t \in V$.

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E$

Single Source Replacement Path Problem: Fixed s compute $d_{\mathbf{G}\setminus e}(s, t)$ for each $e \in E, t \in V$.

Note: Output size is $\mathcal{O}(n^2)$

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$	for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$	for each $e \in E, t \in V$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
Directed Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial

Key Definitions:

	Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source	**Replacement Path Problem:**	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Key Definitions:

	Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source	**Replacement Path Problem:**	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$.

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Dual Fault Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$.

Undirected Replacement Paths: Dual Fault Reduces to Single Source

Key Definitions:

Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$

Single Source Replacement Path Problem: Fixed s compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Dual Fault Replacement Path Problem: Fixed s, t compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$.

Note: Output size is $\mathcal{O}(n^2)$

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	m		
Directed Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$mn^{1/2-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

Some observations:

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

Some observations:

- Dual Fault compared to Single Source has slower algorithms for every setting;

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$

Some observations:

- Dual Fault compared to Single Source has slower algorithms for every setting;
- In directed graphs for combinatorial algorithms: $\tilde{\mathcal{O}}(\sqrt{n}m + n^2)$ (Single Source) **vs** $n^{8/3-o(1)}$ (Dual Fault)

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	n^3	APSP-hard	
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Our Result: UNDIRECTED REPLACEMENT PATHS: DUAL FAULT REDUCES TO SINGLE SOURCE

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Our Result: UNDIRECTED REPLACEMENT PATHS: DUAL FAULT REDUCES TO SINGLE SOURCE

Formally:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n,m,D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n,m,D) + n^2\ )$

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Our Result: UNDIRECTED REPLACEMENT PATHS: DUAL FAULT REDUCES TO SINGLE SOURCE

Formally:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathsf{T}(n,m,D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathsf{T}(n,m,D) + n^2\ )$

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Our Result: UNDIRECTED REPLACEMENT PATHS: DUAL FAULT REDUCES TO SINGLE SOURCE

Directed RP $\iff$ APSP $\iff$ Directed/Undirected SSRP $\iff$ Undirected 2FRP

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Key Definitions:

Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E$
Dual Fault Replacement Path Problem:	Fixed s, t	compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for each $e, f \in E$
Single Source Replacement Path Problem:	Fixed s	compute $d_{\mathbf{G}\setminus e}(s,t)$ for each $e \in E, t \in V$

Our Result: UNDIRECTED REPLACEMENT PATHS: DUAL FAULT REDUCES TO SINGLE SOURCE

Directed RP $\iff$ APSP $\iff$ Directed/Undirected SSRP $\iff$ $\overset{?}{}$ Directed/Undirected 2FRP

	Weighted	Upper Bound $\tilde{\mathcal{O}}(\cdot)$	Lower Bound	Notes
Undirected Graphs	Yes	$n^3/\exp(\Omega(\sqrt{\log n}))$	APSP-hard	
	No	n^ω		
	No	$\sqrt{n}m + n^2$	$mn^{1/2-o(1)}$	Combinatorial
Directed Graphs	Yes	n^3	APSP-hard	
	No	$n^{2.87}$		
	No	$n^{3-1/18}$	$n^{8/3-o(1)}$	Combinatorial

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{O}(\ \mathrm{T}(n, m, D) + n^2\ )$

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s, t)$ for all $e, f \in E$

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

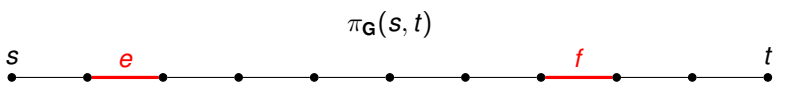
Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$ and $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,t), f \notin \pi_{\mathbf{G}}(s,t)$

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s, t)$ for all $e, f \in \pi_{\mathbf{G}}(s, t)$

G

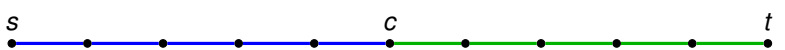


Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Divide and Conquer Strategy:

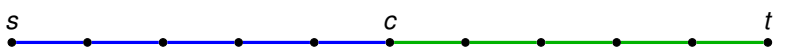
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



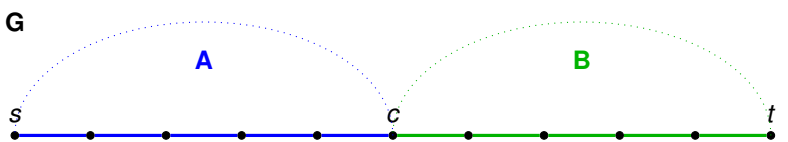
Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$



Divide and Conquer Strategy:

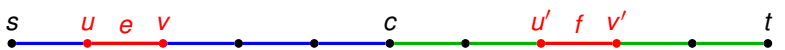
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Divide and Conquer Strategy:

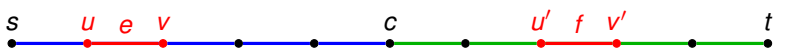
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

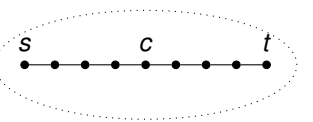
Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



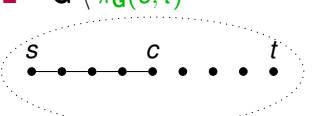
Compute SSRP four times:

G



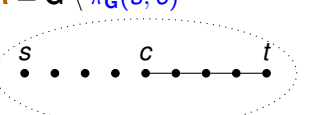
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



Source: from t

Divide and Conquer Strategy:

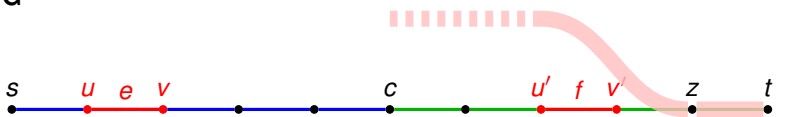
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \Bigg\{$$

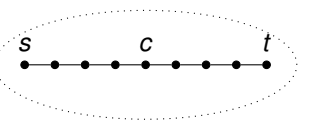
$\pi_{\mathbf{G}\setminus e,f}(s,t)$ visits no node $c \preceq x \preceq u'$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

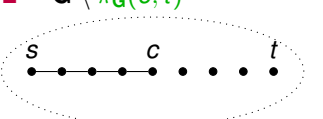
Compute SSRP four times:

G



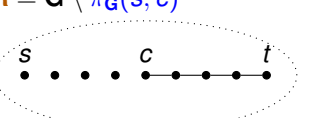
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



Source: from t

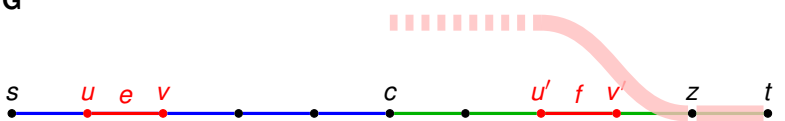
Proof Overview:

$\Longrightarrow$ $\exists$ algorithm for SSRP on undirected graphs with weights from domain D in time $\mathrm{T}(n, m, D)$

$\Longrightarrow$ —— " —— for 2FRP —————————— " ————————— in time $\tilde{\mathcal{O}}(\,\mathrm{T}(n, m, D) + n^2\,)$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



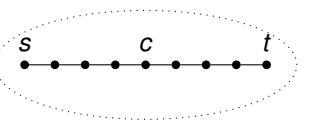
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \vdots \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

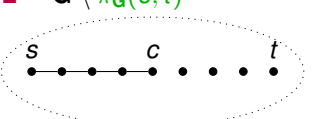
Compute SSRP four times:

G



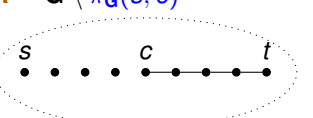
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



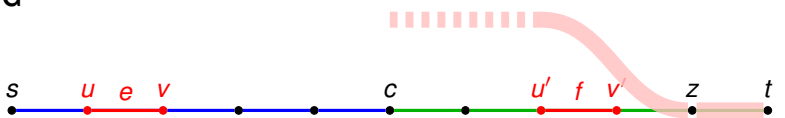
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



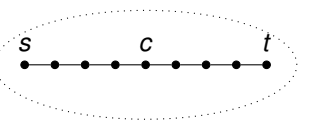
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} \min_{v' \preceq z \preceq t} \left\{ d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

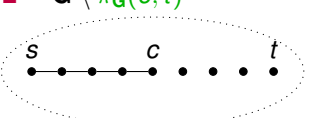
Compute SSRP four times:

G



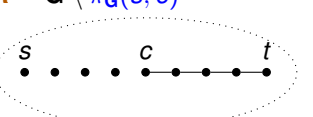
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



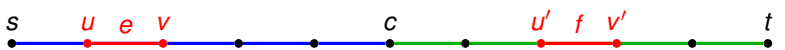
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



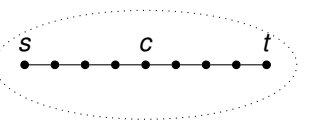
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} \min_{v' \preceq z \preceq t} \left\{ d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \min_{s \preceq z \preceq u} \left\{ d_{\mathbf{G}}(s,z) + d_{\mathbf{R}\setminus f}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } v \preceq y \preceq c \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

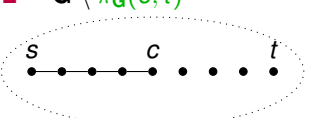
Compute SSRP four times:

G



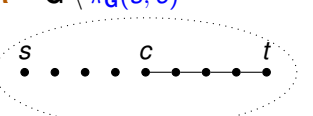
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



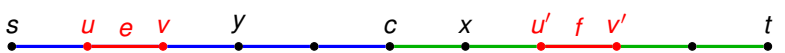
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



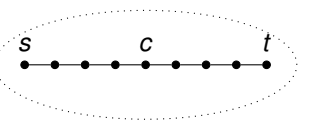
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} \min_{v' \preceq z \preceq t} \left\{ d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \min_{s \preceq z \preceq u} \left\{ d_{\mathbf{G}}(s,z) + d_{\mathbf{R}\setminus f}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } v \preceq y \preceq c \\ & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits such nodes } x, y \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

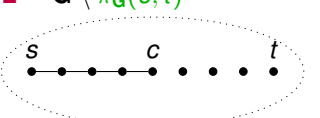
Compute SSRP four times:

G



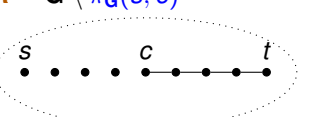
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



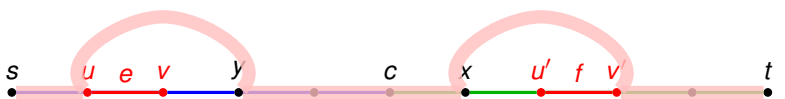
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	— " —	for 2FRP	— " —	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



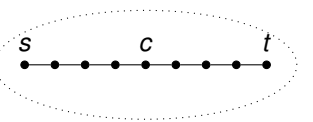
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} \min_{v' \preceq z \preceq t} \left\{ d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \min_{s \preceq z \preceq u} \left\{ d_{\mathbf{G}}(s,z) + d_{\mathbf{R}\setminus f}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } v \preceq y \preceq c \\ & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits such nodes } x, y \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

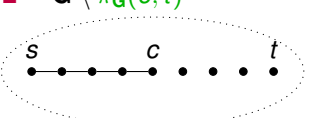
Compute SSRP four times:

G



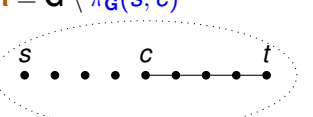
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



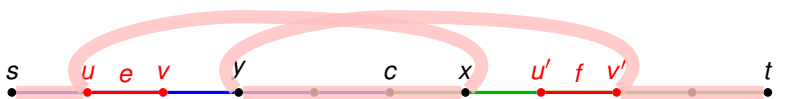
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



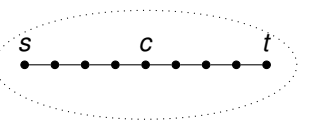
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} \min_{v' \preceq z \preceq t} \left\{ d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \min_{s \preceq z \preceq u} \left\{ d_{\mathbf{G}}(s,z) + d_{\mathbf{R}\setminus f}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } v \preceq y \preceq c \\ & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits such nodes } x, y \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

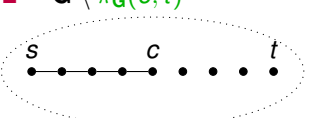
Compute SSRP four times:

G



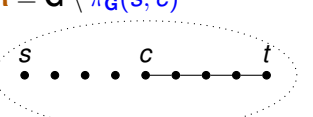
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



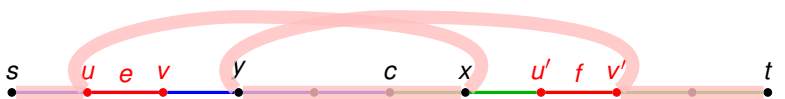
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



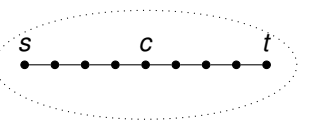
$$d_{\mathbf{G}\setminus e,f}(s,t) = \min \begin{cases} \min_{v' \preceq z \preceq t} \left\{ d_{\mathbf{L}\setminus e}(s,z) + d_{\mathbf{G}}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } c \preceq x \preceq u' \\ \min_{s \preceq z \preceq u} \left\{ d_{\mathbf{G}}(s,z) + d_{\mathbf{R}\setminus f}(z,t) \right\} & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits no node } v \preceq y \preceq c \\ d_{\mathbf{G}\setminus e}(s,c) + d_{\mathbf{G}\setminus f}(c,t) & \pi_{\mathbf{G}\setminus e,f}(s,t) \text{ visits such nodes } x, y \end{cases}$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

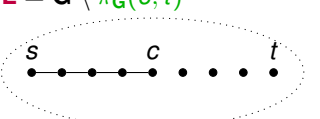
Compute SSRP four times:

G



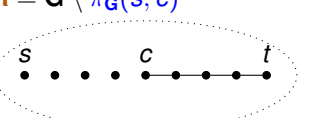
Source(s): from s and t

$\mathbf{L} = \mathbf{G} \setminus \pi_{\mathbf{G}}(c,t)$



Source: from s

$\mathbf{R} = \mathbf{G} \setminus \pi_{\mathbf{G}}(s,c)$



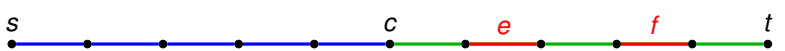
Source: from t

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Divide and Conquer Strategy:

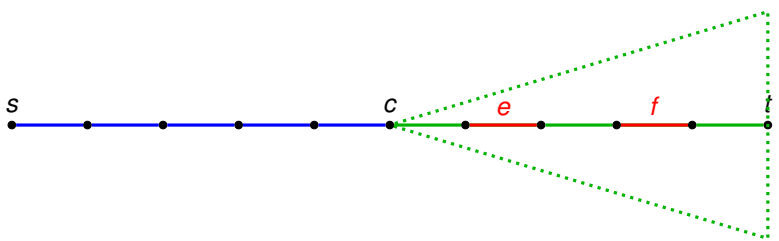
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;

Divide and Conquer Strategy:

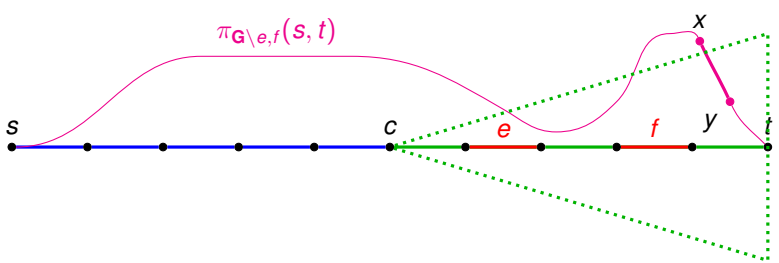
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;

Divide and Conquer Strategy:

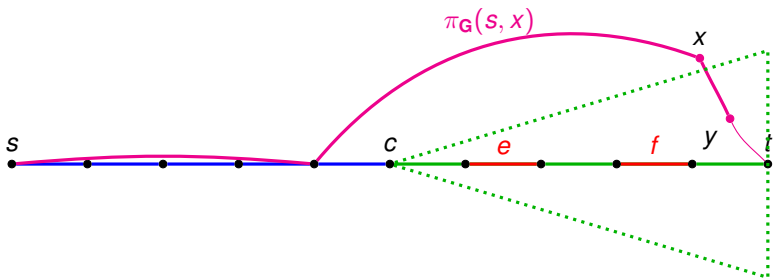
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;

Divide and Conquer Strategy:

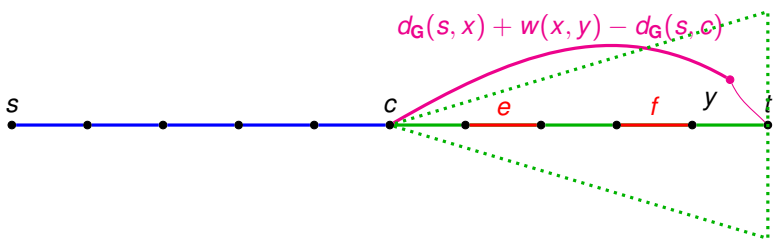
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;
- For each node $y \in T_c$ add an edge $\{c, y\}$ of weight

$$\min_{x \notin T_c} \{d_{\mathbf{G}}(s,x) + w(x,y)\} - d_{\mathbf{G}}(c,t)$$

Divide and Conquer Strategy:

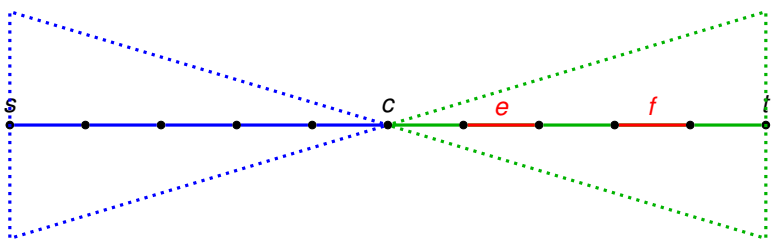
1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

G



Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;
- For each node $y \in T_c$ add an edge $\{c, y\}$ of weight

$$\min_{x \notin T_c} \{d_{\mathbf{G}}(s,x) + w(x,y)\} - d_{\mathbf{G}}(c,t)$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

Problematic: Added edges have weight potentially not in domain D anymore.

Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;
- For each node $y \in T_c$ add an edge $\{c, y\}$ of weight

$$\min_{x \notin T_c} \{d_{\mathbf{G}}(s,x) + w(x,y)\} - d_{\mathbf{G}}(c,t)$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

Problematic: Added edges have weight potentially not in domain D anymore.

Workaround: We can get time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$ algorithm for SSRP with arbitrary weights from source

Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;
- For each node $y \in T_c$ add an edge $\{c, y\}$ of weight

$$\min_{x \notin T_c} \{d_{\mathbf{G}}(s,x) + w(x,y)\} - d_{\mathbf{G}}(c,t)$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Proof Overview:

	$\exists$ algorithm	for SSRP	on undirected graphs with weights from domain D	in time $\mathrm{T}(n, m, D)$
$\Longrightarrow$	—— " ——	for 2FRP	———————— " ————————	in time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$

Want to compute: $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,t)$

Problematic: Added edges have weight potentially not in domain D anymore.

Workaround: We can get time $\tilde{\mathcal{O}}(\ \mathrm{T}(n, m, D) + n^2\ )$ algorithm for SSRP with arbitrary weights from source + other tricks to accommodate change

Construct **B** as follows:

- Add the subtree T_c rooted at c from the shortest path tree of s;
- For each node $y \in T_c$ add an edge $\{c, y\}$ of weight

$$\min_{x \notin T_c} \{d_{\mathbf{G}}(s,x) + w(x,y)\} - d_{\mathbf{G}}(c,t)$$

Divide and Conquer Strategy:

1. "Wisely" choose "midpoint" $c \in \pi_{\mathbf{G}}(s,t)$ and split $\pi_{\mathbf{G}}(s,t) = \pi_{\mathbf{G}}(s,c) \circ \pi_{\mathbf{G}}(c,t)$;
2. Via SSRP algorithm: compute $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e \in \pi_{\mathbf{G}}(s,c)$ and $f \in \pi_{\mathbf{G}}(c,t)$;
3. Create "wisely" two graphs **A** and **B** such that:
   - $d_{\mathbf{A}\setminus e,f}(s,c)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(s,c)$;
   - $d_{\mathbf{B}\setminus e,f}(c,t)$ allows us to infer $d_{\mathbf{G}\setminus e,f}(s,t)$ for all $e, f \in \pi_{\mathbf{G}}(c,t)$.

Thanks!