Tight Approximation and Kernelization Bounds for Vertex-Disjoint Shortest Paths

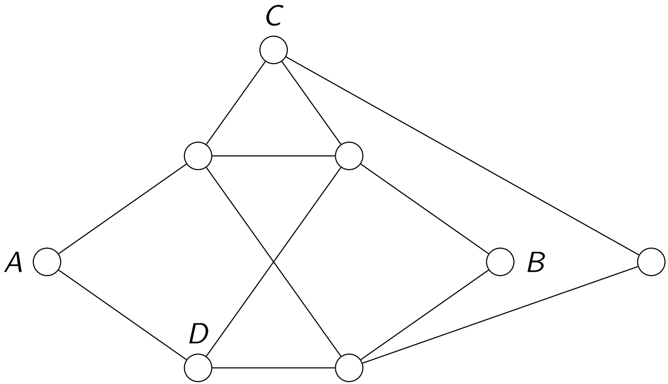
Matthias Bentert Fedor V. Fomin Petr A. Golovach

Univerity of Bergen

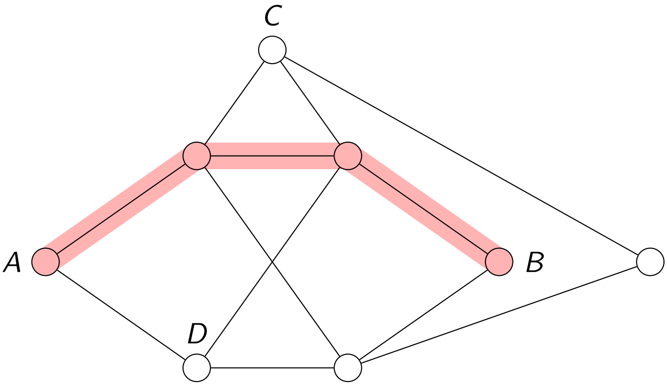
March 05 2025

42nd International Symposium on Theoretical Aspects of Computer Science (STACS)

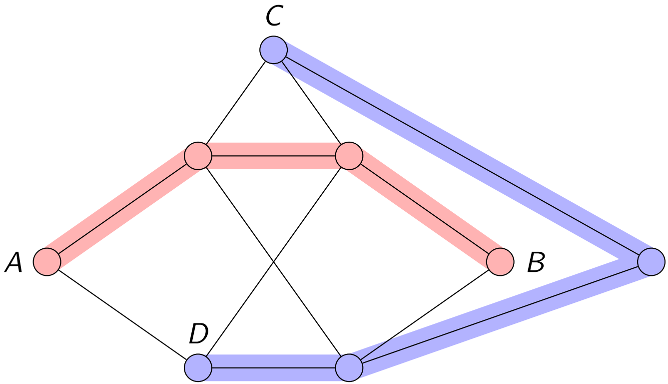
Moving



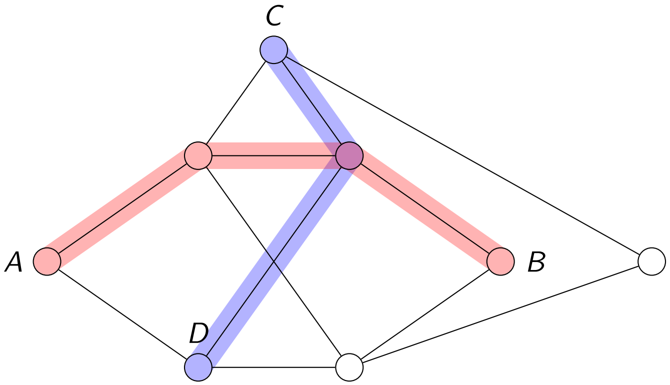
Moving



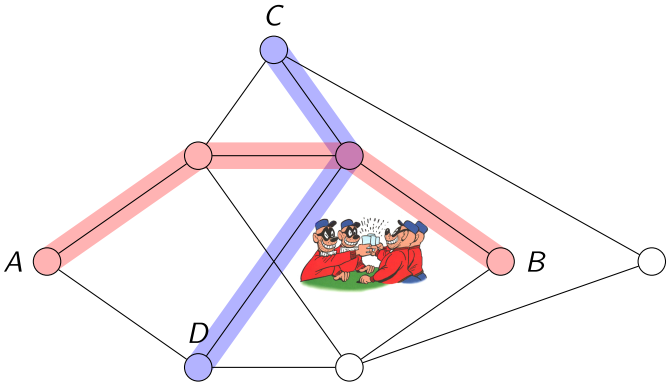
Moving



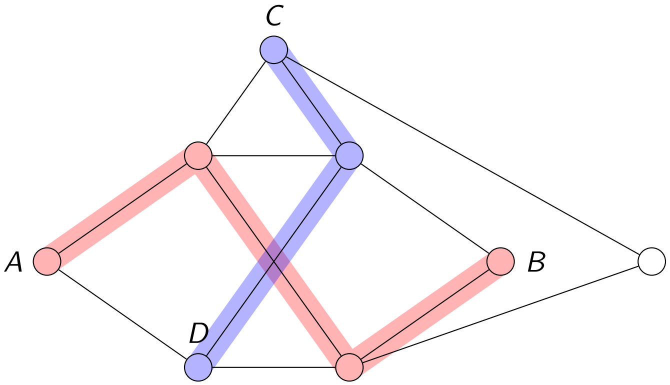
Moving

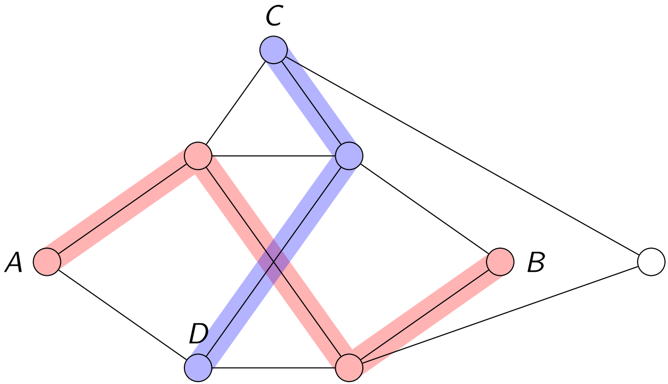


Moving



Moving





Disjoint Shortest Paths

Input: An undirected graph G and vertex pairs $(s_i, t_i)_{i \in \{1,2,\ldots,k\}}$.
Task: Connect as many terminal pairs as possible via vertex-disjoint shortest s_i-t_i-paths.

Related Work

- **2 Disjoint Shortest Paths**

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time

[Eilam-Tzoreff. Discrete Applied Mathematics 1998]

[Akhmedov. CSR 2020]

[**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]

[Akmal, Vassilevska Williams, and Wein. ICALP 2024]

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time
    [Eilam-Tzoreff. Discrete Applied Mathematics 1998]
    [Akhmedov. CSR 2020]
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
    [Akmal, Vassilevska Williams, and Wein. ICALP 2024]
  - $O(nm^5)$ time on directed graphs [Bérczi and Kobayashi. ESA 2017]

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time
    [Eilam-Tzoreff. Discrete Applied Mathematics 1998]
    [Akhmedov. CSR 2020]
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
    [Akmal, Vassilevska Williams, and Wein. ICALP 2024]
  - $O(nm^5)$ time on directed graphs [Bérczi and Kobayashi. ESA 2017]
- **k Disjoint Shortest Paths**

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time

    [Eilam-Tzoreff. Discrete Applied Mathematics 1998]

    [Akhmedov. CSR 2020]

    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]

    [Akmal, Vassilevska Williams, and Wein. ICALP 2024]
  - $O(nm^5)$ time on directed graphs [Bérczi and Kobayashi. ESA 2017]
- **k Disjoint Shortest Paths**
  - solvable in $n^{O(k^{5^k})}$ time on undirected graphs [Lochet. SODA 2021]

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time
    [Eilam-Tzoreff. Discrete Applied Mathematics 1998]
    [Akhmedov. CSR 2020]
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
    [Akmal, Vassilevska Williams, and Wein. ICALP 2024]
  - $O(nm^5)$ time on directed graphs [Bérczi and Kobayashi. ESA 2017]
- **k Disjoint Shortest Paths**
  - solvable in $n^{O(k^{5^k})}$ time on undirected graphs [Lochet. SODA 2021]
  - solvable in $n^{O((k+1)!)}$ time and W[1]-hard wrt. k
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time
    [Eilam-Tzoreff. Discrete Applied Mathematics 1998]
    [Akhmedov. CSR 2020]
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
    [Akmal, Vassilevska Williams, and Wein. ICALP 2024]
  - $O(nm^5)$ time on directed graphs [Bérczi and Kobayashi. ESA 2017]
- **k Disjoint Shortest Paths**
  - solvable in $n^{O(k^{5^k})}$ time on undirected graphs [Lochet. SODA 2021]
  - solvable in $n^{O((k+1)!)}$ time and W[1]-hard wrt. k
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
  - no $(2-\varepsilon)$-approximation in $f(k)n^{o(k)}$ time assuming gap-ETH
    [Chitnis, Thomas, and Wirth. WAOA 2024]

Related Work

- **2 Disjoint Shortest Paths**
  - undirected graphs: $O(n^8)$, $O(n^6)$, $O(nm)$, $O(n+m)$ time
    [Eilam-Tzoreff. Discrete Applied Mathematics 1998]
    [Akhmedov. CSR 2020]
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
    [Akmal, Vassilevska Williams, and Wein. ICALP 2024]
  - $O(nm^5)$ time on directed graphs [Bérczi and Kobayashi. ESA 2017]
- **k Disjoint Shortest Paths**
  - solvable in $n^{O(k^{5^k})}$ time on undirected graphs [Lochet. SODA 2021]
  - solvable in $n^{O((k+1)!)}$ time and W[1]-hard wrt. k
    [**B**, Nichterlein, Renken, and Zschoche. SIAM Journal on Discrete Mathematics 2023]
  - no $(2-\varepsilon)$-approximation in $f(k)n^{o(k)}$ time assuming gap-ETH
    [Chitnis, Thomas, and Wirth. WAOA 2024]
  - solvable in polynomial time on directed planar graphs if all terminals are incident to two faces (even min-sum) [Coli n de Verdière and Schrijver. ACM TALG 2011]

Our Results

Parameter	Exact	Approximation
	NP-complete	**$\mathbf{m^{1/2-\varepsilon}}$-approximation is NP-hard**

Our Results

Parameter	Exact	Approximation
	NP-complete	**$\mathbf{m}^{1/2-\varepsilon}$-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no $\mathbf{o(k)}$-approximation in $\mathbf{f(k) \cdot poly(n)}$ time**

Our Results

Parameter	Exact	Approximation
	NP-complete	**$m^{1/2-\varepsilon}$-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no $o(k)$-approximation in $f(k) \cdot \text{poly}(n)$ time**
ℓ	FPT **no poly kernel**	**$\lceil\sqrt{\ell}\rceil$-approximation**

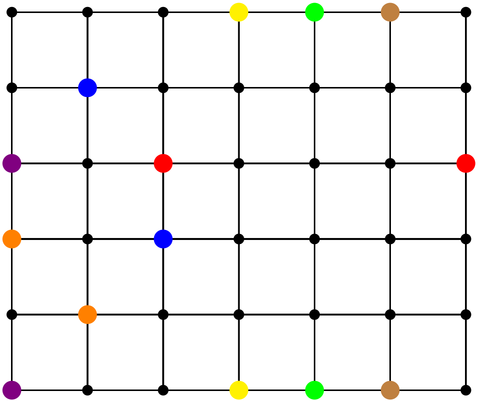
Our Results

Parameter	Exact	Approximation
	NP-complete	**$m^{1/2-\varepsilon}$-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no $o(k)$-approximation in $f(k) \cdot poly(n)$ time**
ℓ	FPT **no poly kernel**	**$\lceil\sqrt{\ell}\rceil$-approximation**

Upper and Lower (FPT) Approximation Bounds

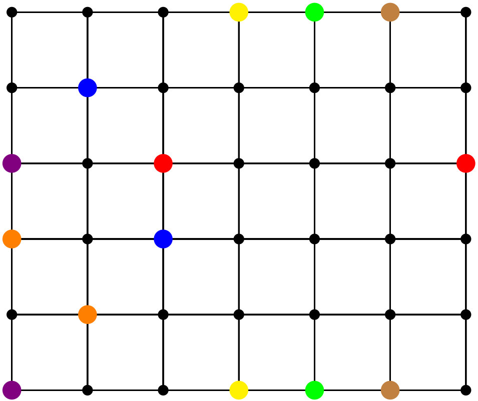
A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance



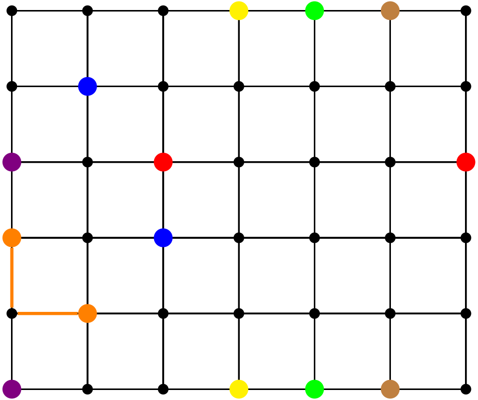
A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path



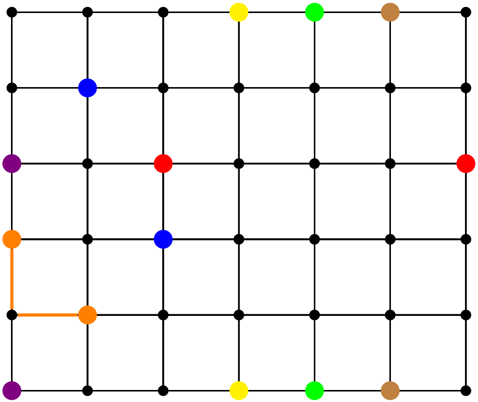
A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path



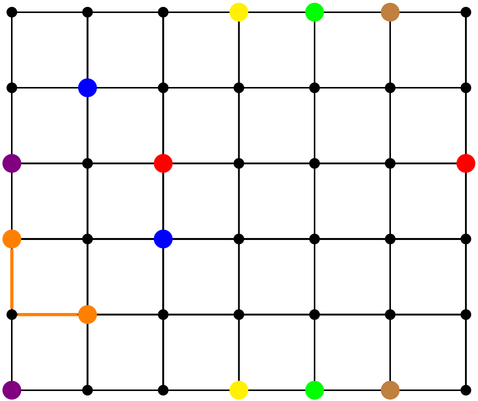
A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases



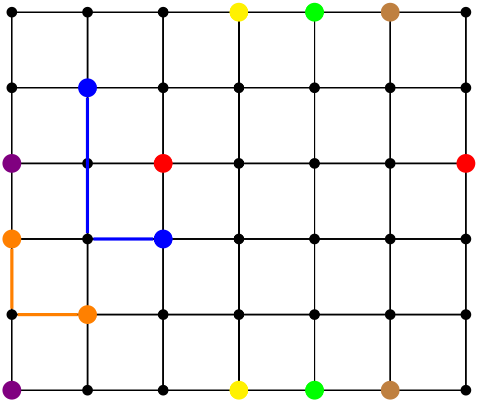
A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases
- repeat until no terminal pair is left



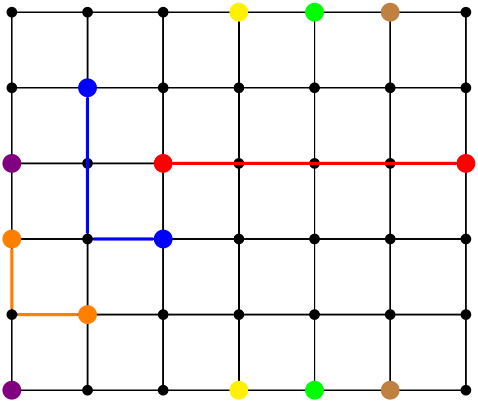
A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases
- repeat until no terminal pair is left



A Polynomial-time Approximation Algorithm

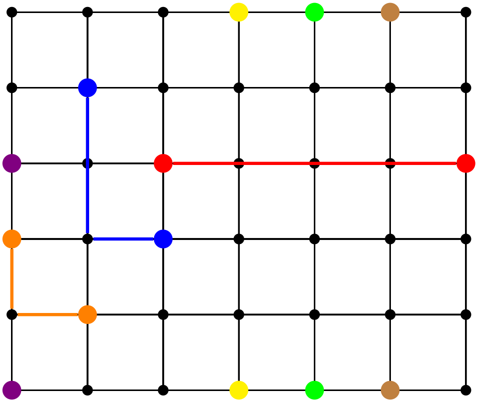
- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases
- repeat until no terminal pair is left



A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases
- repeat until no terminal pair is left

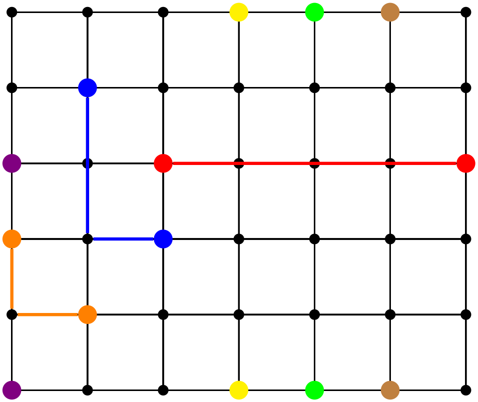
- if $\text{dist}(s_i, t_i) + 1 \leq \sqrt{\ell}$: remove at most $\sqrt{\ell}$ terminal pairs in optimal solution



A Polynomial-time Approximation Algorithm

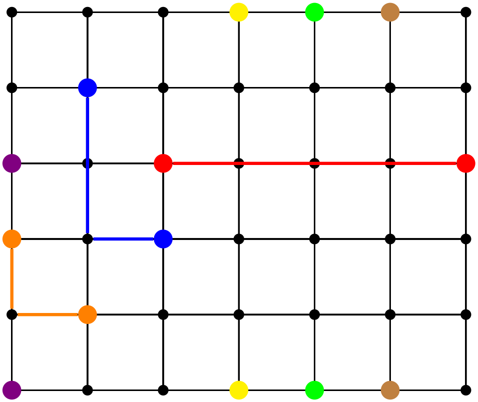
- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases
- repeat until no terminal pair is left

- if $\text{dist}(s_i, t_i) + 1 \leq \sqrt{\ell}$: remove at most $\sqrt{\ell}$ terminal pairs in optimal solution
- if $\text{dist}(s_i, t_i) + 1 > \sqrt{\ell}$: at most $\sqrt{\ell}$ terminal pairs left in optimal solution



A Polynomial-time Approximation Algorithm

- find terminal pair of minimum distance
- connect by any shortest path
- remove all vertices in the path and all terminal pairs whose distance increases
- repeat until no terminal pair is left



- if $\text{dist}(s_i, t_i) + 1 \leq \sqrt{\ell}$: remove at most $\sqrt{\ell}$ terminal pairs in optimal solution
- if $\text{dist}(s_i, t_i) + 1 > \sqrt{\ell}$: at most $\sqrt{\ell}$ terminal pairs left in optimal solution

$\rightsquigarrow$ $\sqrt{\ell}$-approximation in polynomial time

The General Idea

for excluding $m^{1/2-\varepsilon}$-approximations

- Computing a $n^{1-\varepsilon}$-approximation for Clique is NP-hard.

The General Idea

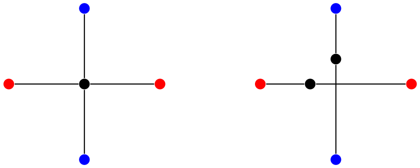
for excluding $m^{1/2-\varepsilon}$-approximations

- Computing a $n^{1-\varepsilon}$-approximation for Clique is NP-hard.
- Reduction to Vertex-Disjoint Shortest Paths such that the optimal solution corresponds to a largest clique and $m' \in O(n^2)$

The General Idea

for excluding $m^{1/2-\varepsilon}$-approximations

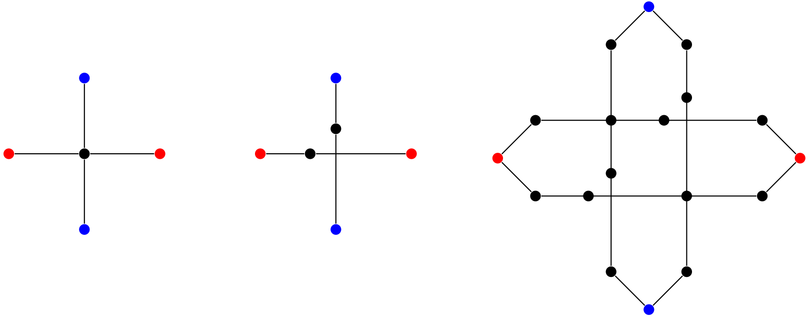
- Computing a $n^{1-\varepsilon}$-approximation for CLIQUE is NP-hard.
- Reduction to VERTEX-DISJOINT SHORTEST PATHS such that the optimal solution corresponds to a largest clique and $m' \in O(n^2)$



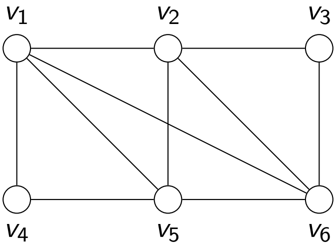
The General Idea

for excluding $m^{1/2-\varepsilon}$-approximations

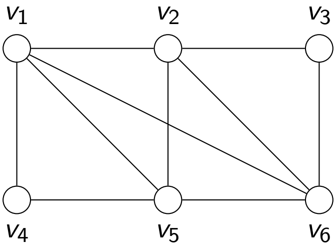
- Computing a $n^{1-\varepsilon}$-approximation for Clique is NP-hard.
- Reduction to Vertex-Disjoint Shortest Paths such that the optimal solution corresponds to a largest clique and $m' \in O(n^2)$

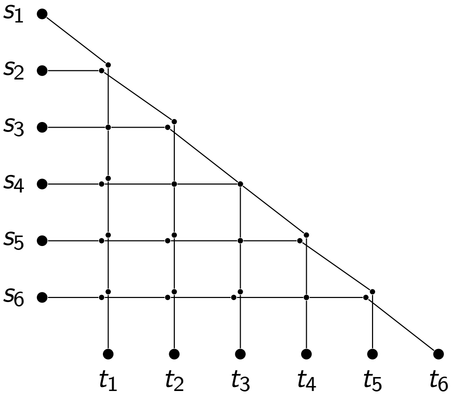


The Reduction

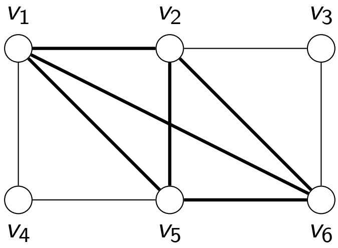


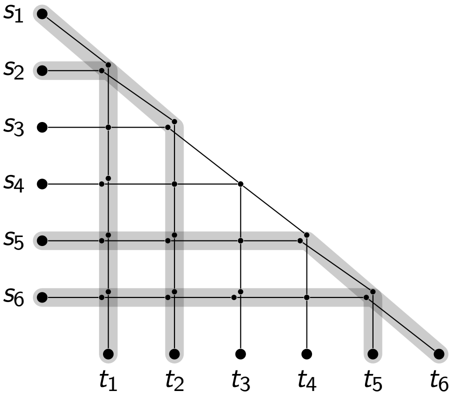
The Reduction





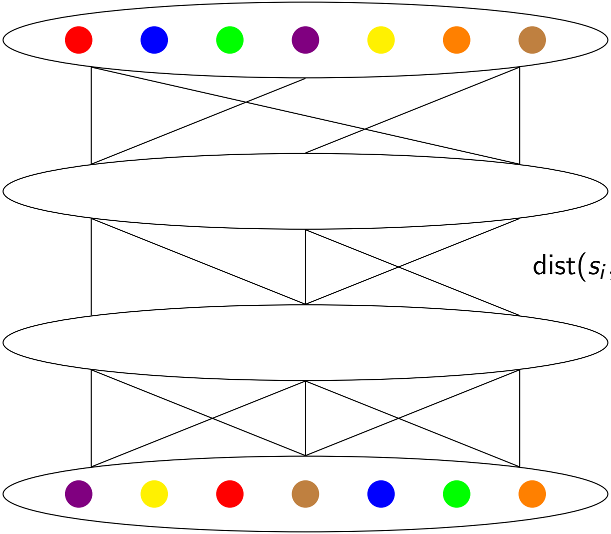
The Reduction





Excluding Polynomial Kernels

A Special Case



dist(s_i, t_i) = # layers - 1

Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance

Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance
- $|I| \leq \text{poly}(\sum_{i=1}^{t} |I_i|)$
- $\ell \leq \text{poly}(\max_{i=1}^{t} |I_i| + \log(t))$

Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance
- $|I| \leq \text{poly}(\sum_{i=1}^{t} |I_i|)$
- $\ell \leq \text{poly}(\max_{i=1}^{t} |I_i| + \log(t))$

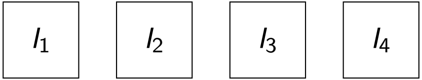
approach: iteratively combine two instances such that number of layers increase by $O(k)$

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance
- $|I| \leq \text{poly}(\sum_{i=1}^{t} |I_i|)$
- $\ell \leq \text{poly}(\max_{i=1}^{t} |I_i| + \log(t))$

approach: iteratively combine two instances such that number of layers increase by $O(k)$



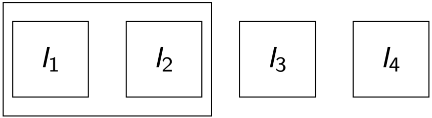
Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance
- $|I| \leq \text{poly}(\sum_{i=1}^{t} |I_i|)$
- $\ell \leq \text{poly}(\max_{i=1}^{t} |I_i| + \log(t))$

approach: iteratively combine two instances such that number of layers increase by $O(k)$



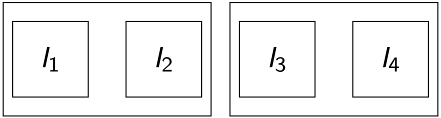
Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance
- $|I| \leq \text{poly}(\sum_{i=1}^{t} |I_i|)$
- $\ell \leq \text{poly}(\max_{i=1}^{t} |I_i| + \log(t))$

approach: iteratively combine two instances such that number of layers increase by $O(k)$



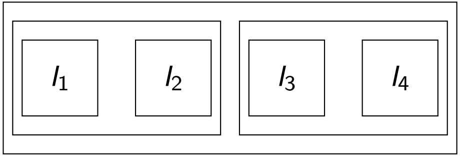
Or Composition

input: t instances $I_1, I_2, \ldots, I_t$ of special case all with same number of layers and terminal pairs

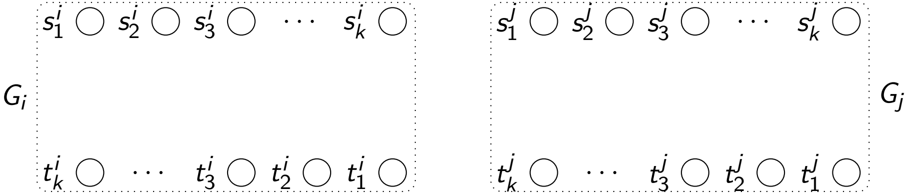
output: one instance (I, ℓ) of vertex-disjoint shortest paths such that

- I is a yes-instance $\Leftrightarrow$ at least one input instance is a yes-instance
- $|I| \leq \text{poly}(\sum_{i=1}^{t} |I_i|)$
- $\ell \leq \text{poly}(\max_{i=1}^{t} |I_i| + \log(t))$

approach: iteratively combine two instances such that number of layers increase by $O(k)$

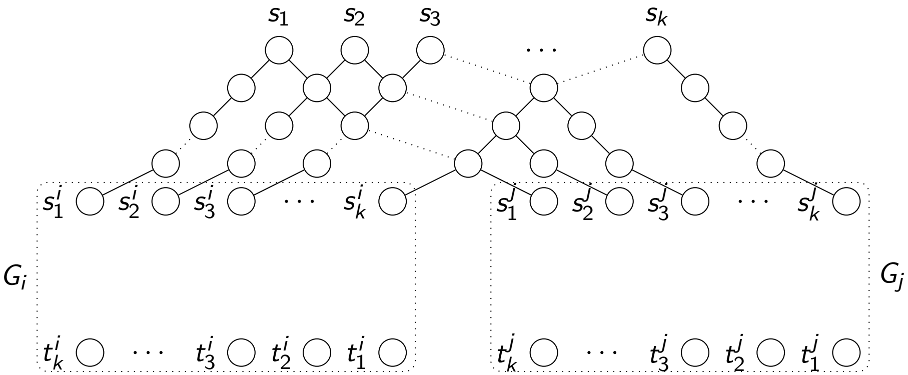


The Gadget



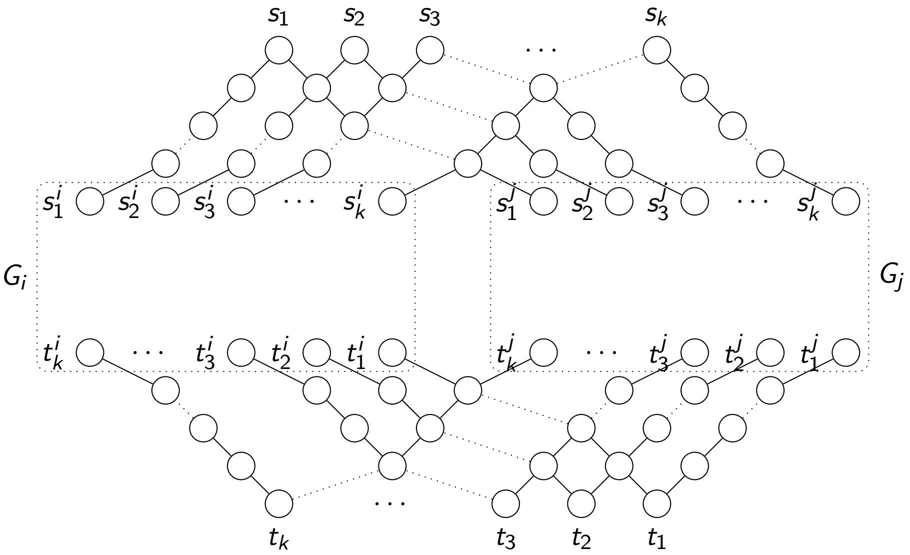
G_i
s_1^i s_2^i s_3^i $\cdots$ s_k^i
t_k^i $\cdots$ t_3^i t_2^i t_1^i
G_j
s_1^j s_2^j s_3^j $\cdots$ s_k^j
t_k^j $\cdots$ t_3^j t_2^j t_1^j

The Gadget



s_1 s_2 s_3 … s_k
s_1^i s_2^i s_3^i … s_k^i
s_1^j s_2^j s_3^j … s_k^j
G_i
G_j
t_k^i … t_3^i t_2^i t_1^i
t_k^j … t_3^j t_2^j t_1^j

The Gadget


s_1
s_2
s_3
$\cdots$
s_k
s_1^i
s_2^i
s_3^i
$\cdots$
s_k^i
s_1^j
s_2^j
s_3^j
$\cdots$
s_k^j
G_i
G_j
t_k^i
$\cdots$
t_3^i
t_2^i
t_1^i
t_k^j
$\cdots$
t_3^j
t_2^j
t_1^j
t_k
$\cdots$
t_3
t_2
t_1

Conclusion

Parameter	Exact	Approximation
	NP-complete	$\mathbf{m^{1/2-\varepsilon}}$**-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no** $\mathbf{o(k)}$**-approximation in** $\mathbf{f(k) \cdot poly(n)}$ **time**
ℓ	FPT **no poly kernel**	$\lceil\sqrt{\ell}\rceil$**-approximation**

Conclusion

Parameter	Exact	Approximation
	NP-complete	**$m^{1/2-\varepsilon}$-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no $o(k)$-approximation in $f(k) \cdot poly(n)$ time**
ℓ	FPT **no poly kernel**	**$\lceil\sqrt{\ell}\rceil$-approximation**

- FPT approximation for structural parameters?

Conclusion

Parameter	Exact	Approximation
	NP-complete	$\mathbf{m^{1/2-\varepsilon}}$**-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no $\mathbf{o(k)}$-approximation in $\mathbf{f(k) \cdot poly(n)}$ time**
ℓ	FPT **no poly kernel**	$\lceil\sqrt{\ell}\rceil$**-approximation**

- FPT approximation for structural parameters?
- poly(ℓ)-size lossy kernels with approximation factors in $o(\sqrt{\ell})$ (or even constant)?

Conclusion

Parameter	Exact	Approximation
	NP-complete	**$m^{1/2-\varepsilon}$-approximation is NP-hard**
k	XP and W[1]-hard	$c \cdot k$-approximation for any c **no $o(k)$-approximation in $f(k) \cdot poly(n)$ time**
ℓ	FPT **no poly kernel**	**$\lceil \sqrt{\ell} \rceil$-approximation**

- FPT approximation for structural parameters?
- poly(ℓ)-size lossy kernels with approximation factors in $o(\sqrt{\ell})$ (or even constant)?

Thank you.