Fixed-Parameter Tractability of Maximum Colored Path and Beyond

Fedor V. Fomin, Petr A. Golovach, Tuukka Korhonen,
Kirill Simonov[1], and Giannos Stamoulis[2]

UNIVERSITY OF BERGEN

[1]Hasso Plattner Institute, University of Potsdam [2] LIRMM, Universite de Montpellier, CNRS

SODA 2023

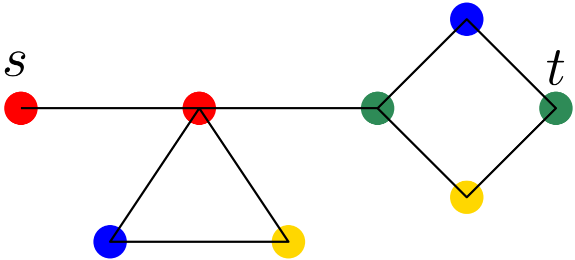
24 January 2023

Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input: Vertex-colored undirected graph, vertices s and t, and an integer k.

Task: Find an s, t-path containing at least k different colors.

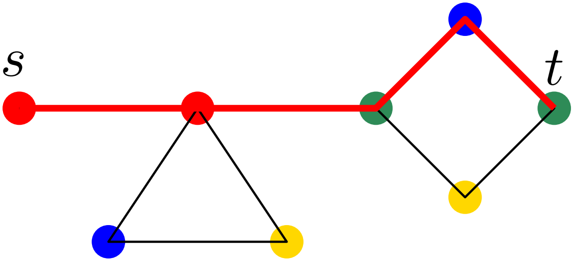


Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input: Vertex-colored undirected graph, vertices s and t, and an integer k.

Task: Find an s, t-path containing at least k different colors.



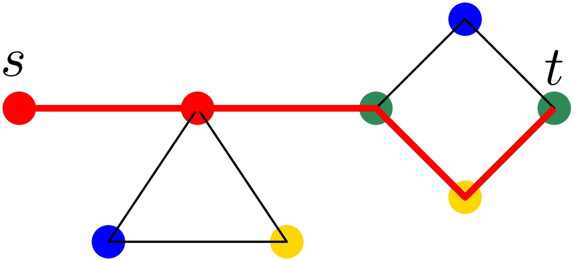
Here, such path exists when $k \leq 3$

Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input: Vertex-colored undirected graph, vertices s and t, and an integer k.

Task: Find an s, t-path containing at least k different colors.

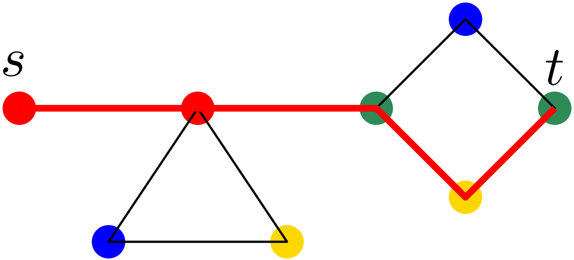


Here, such path exists when $k \leq 3$

Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find an s, t-path containing at least k different colors.



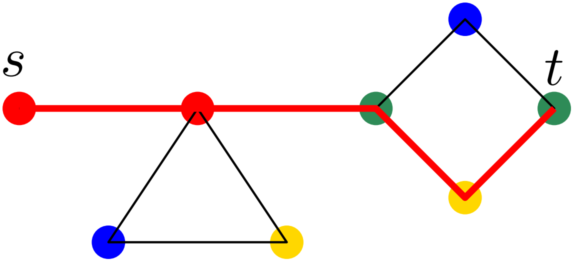
Here, such path exists when $k \leq 3$

- *Path* does not contain repeated vertices

Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find an s, t-path containing at least k different colors.



Here, such path exists when $k \leq 3$

- *Path* does not contain repeated vertices
- Color may repeat multiple times in the path, and it can contain more than k colors

Fixed-parameter tractability of Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input: Vertex-colored undirected graph, vertices s and t, and an integer k.

Task: Find an s, t-path containing at least k different colors.

Theorem

There is a $2^k n^{\mathcal{O}(1)}$ time randomized algorithm for maximum colored s, t-path. Moreover, the algorithm returns the shortest solution.

Fixed-parameter tractability of Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find an s, t-path containing at least k different colors.

Theorem

There is a $2^k n^{\mathcal{O}(1)}$ time randomized algorithm for maximum colored s, t-path. Moreover, the algorithm returns the shortest solution.

("and beyond" will come later)

Fixed-parameter tractability of Maximum Colored Path

MAXIMUM COLORED s,t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find an s,t-path containing at least k different colors.

Theorem

There is a $2^k n^{\mathcal{O}(1)}$ time randomized algorithm for maximum colored s,t-path. Moreover, the algorithm returns the shortest solution.

("and beyond" will come later)

- This is the first FPT algorithm (time complexity $f(k) \cdot n^{\mathcal{O}(1)}$) for this problem

Fixed-parameter tractability of Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find an s, t-path containing at least k different colors.

Theorem

There is a $2^k n^{\mathcal{O}(1)}$ time randomized algorithm for maximum colored s, t-path. Moreover, the algorithm returns the shortest solution.

("and beyond" will come later)

- This is the first FPT algorithm (time complexity $f(k) \cdot n^{\mathcal{O}(1)}$) for this problem
- Assuming set cover conjecture, no $(2 - \varepsilon)^k n^{\mathcal{O}(1)}$ time algorithm for any $\varepsilon > 0$

Fixed-parameter tractability of Maximum Colored Path

MAXIMUM COLORED s, t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find an s, t-path containing at least k different colors.

Theorem

There is a $2^k n^{\mathcal{O}(1)}$ time randomized algorithm for maximum colored s, t-path. Moreover, the algorithm returns the shortest solution.

("and beyond" will come later)

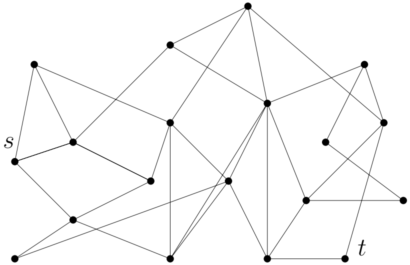
- This is the first FPT algorithm (time complexity $f(k) \cdot n^{\mathcal{O}(1)}$) for this problem
- Assuming set cover conjecture, no $(2 - \varepsilon)^k n^{\mathcal{O}(1)}$ time algorithm for any $\varepsilon > 0$
- NP-hard for directed graphs already when $k = 2$

Application: Longest s, t-Path

LONGEST s, t-PATH

Input: Undirected graph, vertices s and t, and an integer k.

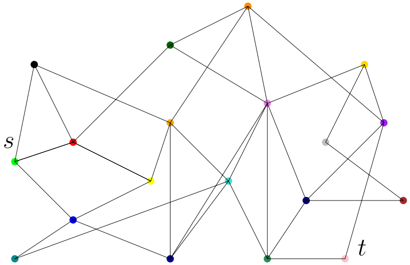
Task: Find an s, t-path of length at least k.



Application: Longest s, t-Path

LONGEST s, t-PATH

Input:	Undirected graph, vertices s and t, and an integer k.
Task:	Find an s, t-path of length at least k.



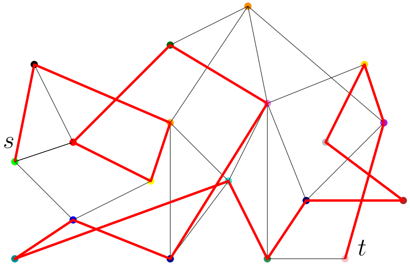
- Longest s, t-path reduces to maximum colored s, t-path by coloring all vertices with different colors

Application: Longest s, t-Path

LONGEST s, t-PATH

Input: Undirected graph, vertices s and t, and an integer k.

Task: Find an s, t-path of length at least k.



- Longest s, t-path reduces to maximum colored s, t-path by coloring all vertices with different colors

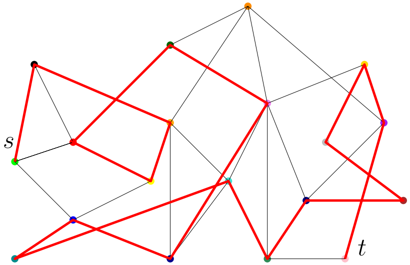
$\Rightarrow$ $2^k n^{\mathcal{O}(1)}$ time algorithm for longest s, t-path

Application: Longest s, t-Path

LONGEST s, t-PATH

Input: Undirected graph, vertices s and t, and an integer k.

Task: Find an s, t-path of length at least k.



- Longest s, t-path reduces to maximum colored s, t-path by coloring all vertices with different colors

$\Rightarrow$ $2^k n^{\mathcal{O}(1)}$ time algorithm for longest s, t-path

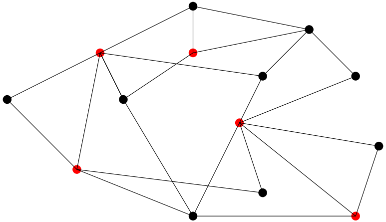
- Previous best algorithm $4^k n^{\mathcal{O}(1)}$ time [Fomin, Lokshtanov, Panolan, Saurabh, Zehavi'18]

Application: T-cycle

T-CYCLE

Input: Undirected graph and a set of terminal vertices T.

Task: Find a cycle that visits each vertex in T.

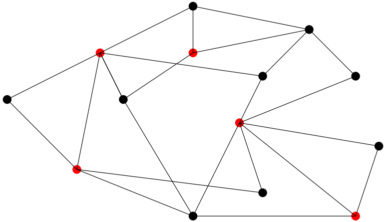


Application: T-cycle

T-CYCLE

Input: Undirected graph and a set of terminal vertices T.

Task: Find a cycle that visits each vertex in T.



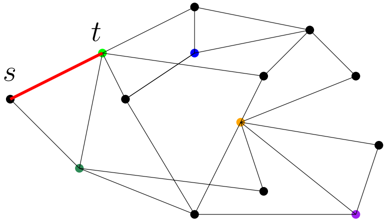
- $2^{|T|} n^{\mathcal{O}(1)}$ time algorithm for T-cycle [Björklund, Husfeldt, Taslaman, SODA'12]

Application: T-cycle

T-CYCLE

Input: Undirected graph and a set of terminal vertices T.

Task: Find a cycle that visits each vertex in T.



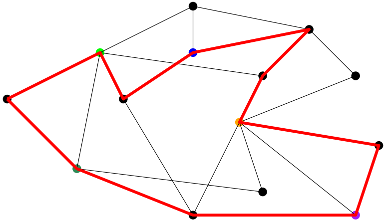
- $2^{|T|}n^{\mathcal{O}(1)}$ time algorithm for T-cycle [Björklund, Husfeldt, Taslaman, SODA'12]
- Reduces to maximum colored s,t-path by coloring the vertices in T with different colors

Application: T-cycle

T-CYCLE

Input: Undirected graph and a set of terminal vertices T.

Task: Find a cycle that visits each vertex in T.



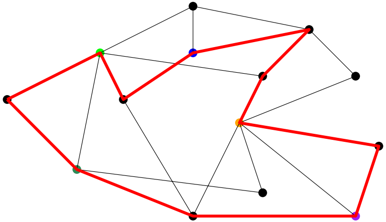
- $2^{|T|} n^{\mathcal{O}(1)}$ time algorithm for T-cycle [Björklund, Husfeldt, Taslaman, SODA'12]
- Reduces to maximum colored s,t-path by coloring the vertices in T with different colors

$\Rightarrow$ $2^{|T|} n^{\mathcal{O}(1)}$ time algorithm also via our result

Application: T-cycle

T-CYCLE

Input: Undirected graph and a set of terminal vertices T.
Task: Find a cycle that visits each vertex in T.



- $2^{|T|} n^{\mathcal{O}(1)}$ time algorithm for T-cycle [Björklund, Husfeldt, Taslaman, SODA'12]
- Reduces to maximum colored s, t-path by coloring the vertices in T with different colors

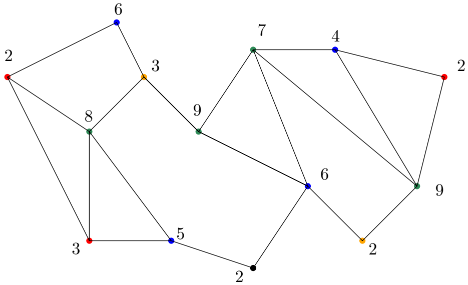
$\Rightarrow$ $2^{|T|} n^{\mathcal{O}(1)}$ time algorithm also via our result

- Allows for generalizations, e.g., need to visit k vertices of large T

Beyond Maximum Colored Path

Input:

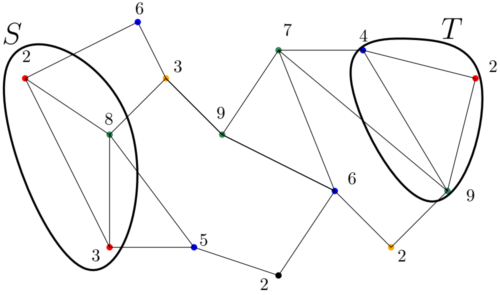
- Colored positive-integer weighted undirected graph



Beyond Maximum Colored Path

Input:

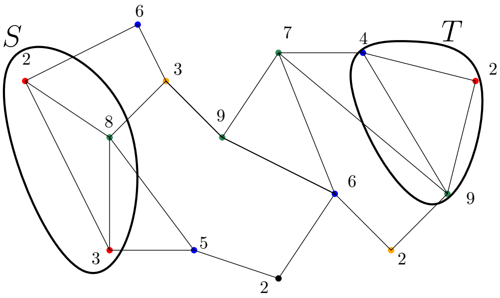
- Colored positive-integer weighted undirected graph
- Two sets of vertices S and T



Beyond Maximum Colored Path

Input:

- Colored positive-integer weighted undirected graph
- Two sets of vertices S and T
- Integers p, k, w



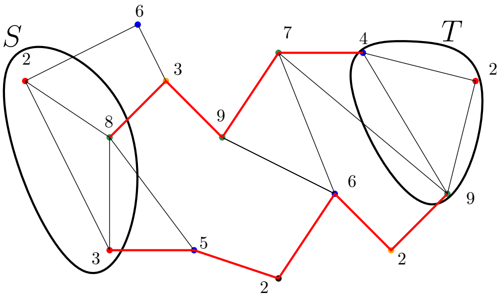
Beyond Maximum Colored Path

Input:

- Colored positive-integer weighted undirected graph
- Two sets of vertices S and T
- Integers p, k, w

Problem:

- Find p vertex-disjoint paths starting in S and ending in T so that



Here, $p = 2$

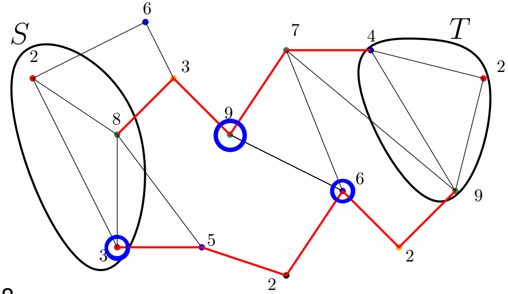
Beyond Maximum Colored Path

Input:

- Colored positive-integer weighted undirected graph
- Two sets of vertices S and T
- Integers p, k, w

Problem:

- Find p vertex-disjoint paths starting in S and ending in T so that
- the vertices of the paths contain a set X of size k, total weight w, and having distinct colors



Here, $p = 2$, $k = 3$, and $w = 18$.

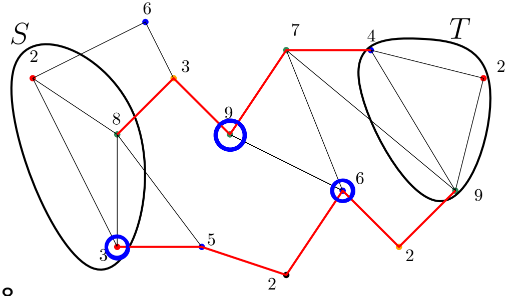
Beyond Maximum Colored Path

Input:

- Colored positive-integer weighted undirected graph
- Two sets of vertices S and T
- Integers p, k, w

Problem:

- Find p vertex-disjoint paths starting in S and ending in T so that
- the vertices of the paths contain a set X of size k, total weight w, and having distinct colors
- While minimizing the total length of the paths



Here, $p = 2$, $k = 3$, and $w = 18$.

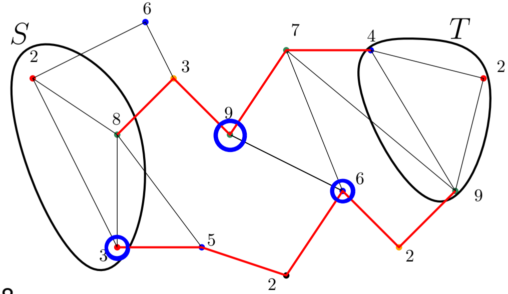
Beyond Maximum Colored Path

Input:

- Colored positive-integer weighted undirected graph
- Two sets of vertices S and T
- Integers p, k, w

Problem:

- Find p vertex-disjoint paths starting in S and ending in T so that
- the vertices of the paths contain a set X of size k, total weight w, and having distinct colors
- While minimizing the total length of the paths



Here, $p = 2$, $k = 3$, and $w = 18$.

Main theorem: Randomized algorithm with time complexity $2^{k+p} n^{\mathcal{O}(1)} w$.

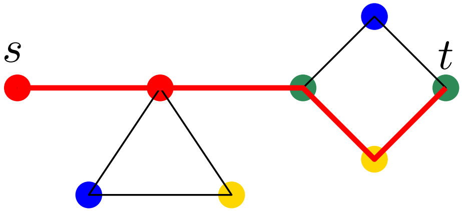
The Algorithm

In this talk, we focus on:

MAXIMUM COLORED s, t-PATH

Input: Vertex-colored undirected graph, vertices s and t, and an integer k.

Task: Find a k-colored s, t-path. (s, t-path with at least k different colors)

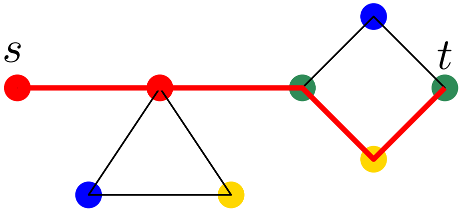


The Algorithm

In this talk, we focus on:

MAXIMUM COLORED s,t-PATH

Input:	Vertex-colored undirected graph, vertices s and t, and an integer k.
Task:	Find a k-colored s,t-path. (s,t-path with at least k different colors)



- Based on algebraic approach, extending ideas that were developed by [Björklund, Husfeldt, Taslaman'12], for T-cycle [Björklund'14] for Hamiltonicity, and [Björklund, Husfeldt, Kaski, Koivisto'17] for k-path

Algebraic approach

General idea:

Algebraic approach

General idea:

- Design a multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq 4n$ over GF($2^{3+\lceil \log n \rceil}$) (finite field of order $\geq 8n$ and characteristic 2) so that

Algebraic approach

General idea:

- Design a multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq 4n$ over $\mathrm{GF}(2^{3+\lceil \log n \rceil})$ (finite field of order $\geq 8n$ and characteristic 2) so that

  1. yes-instance $\rightarrow p(x_1, \ldots, x_m)$ is a non-zero polynomial

Algebraic approach

General idea:

- Design a multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq 4n$ over $\mathrm{GF}(2^{3+\lceil \log n \rceil})$ (finite field of order $\geq 8n$ and characteristic 2) so that
  1. yes-instance $\rightarrow p(x_1, \ldots, x_m)$ is a non-zero polynomial
  2. no-instance $\rightarrow p(x_1, \ldots, x_m)$ is identically zero polynomial

Algebraic approach

General idea:

- Design a multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq 4n$ over $\mathrm{GF}(2^{3+\lceil \log n \rceil})$ (finite field of order $\geq 8n$ and characteristic 2) so that
    1. yes-instance $\rightarrow p(x_1, \ldots, x_m)$ is a non-zero polynomial
    2. no-instance $\rightarrow p(x_1, \ldots, x_m)$ is identically zero polynomial
    3. Given $x_1, \ldots, x_m$, $p(x_1, \ldots, x_m)$ can be evaluated in $2^k n^{\mathcal{O}(1)}$ time

Algebraic approach

General idea:

- Design a multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq 4n$ over GF($2^{3+\lceil \log n \rceil}$) (finite field of order $\geq 8n$ and characteristic 2) so that
  1. yes-instance $\rightarrow p(x_1, \ldots, x_m)$ is a non-zero polynomial
  2. no-instance $\rightarrow p(x_1, \ldots, x_m)$ is identically zero polynomial
  3. Given $x_1, \ldots, x_m$, $p(x_1, \ldots, x_m)$ can be evaluated in $2^k n^{\mathcal{O}(1)}$ time

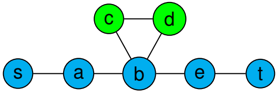
- By DeMillo–Lipton–Schwartz–Zippel lemma, the problem is then solved in $2^k n^{\mathcal{O}(1)}$ time by evaluating $p(x_1, \ldots, x_m)$ for random $x_1, \ldots, x_m$.

Algebraic approach

General idea:

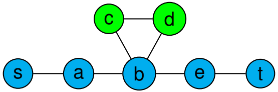
- Design a multivariate polynomial $p(x_1, \ldots, x_m)$ of total degree $\leq 4n$ over $\mathrm{GF}(2^{3+\lceil \log n \rceil})$ (finite field of order $\geq 8n$ and characteristic 2) so that
  1. yes-instance $\rightarrow p(x_1, \ldots, x_m)$ is a non-zero polynomial
  2. no-instance $\rightarrow p(x_1, \ldots, x_m)$ is identically zero polynomial
  3. Given $x_1, \ldots, x_m$, $p(x_1, \ldots, x_m)$ can be evaluated in $2^k n^{\mathcal{O}(1)}$ time
- By DeMillo–Lipton–Schwartz–Zippel lemma, the problem is then solved in $2^k n^{\mathcal{O}(1)}$ time by evaluating $p(x_1, \ldots, x_m)$ for random $x_1, \ldots, x_m$.
- Characteristic 2?
  - $x + x = 0$ for any x

Polynomial over labeled walks



$k = 2$

Polynomial over labeled walks

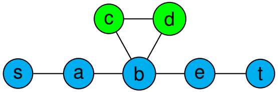


$k = 2$

Definition:

- For each edge uv associate variable $f_e(uv)$
- For each vertex w associate variable $f_v(w)$
- For each color-label pair (x, y) associate variable $f_c(x, y)$

Polynomial over labeled walks



$k = 2$

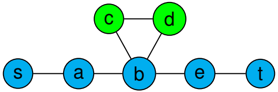
Definition:

- For each edge uv associate variable $f_e(uv)$
- For each vertex w associate variable $f_v(w)$
- For each color-label pair (x, y) associate variable $f_c(x, y)$

For a labeled walk W, associate monomial $f(W)$ that is product of edge variables, vertex variables of labeled vertices, and color-label pair variables corresponding to labeled vertices

$$f(sa\overset{1}{b}c\overset{2}{d}bet) = f_e(sa)f_e(ab)f_e(bc)f_e(cd)f_e(db)f_e(be)f_e(et)f_v(b)f_v(d)f_c(\textcolor{blue}{\bullet}, 1)f_c(\textcolor{green}{\bullet}, 2)$$

Polynomial over labeled walks



$k = 2$

Definition:

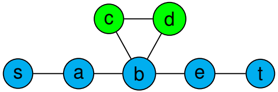
- For each edge uv associate variable $f_e(uv)$
- For each vertex w associate variable $f_v(w)$
- For each color-label pair (x, y) associate variable $f_c(x, y)$

For a labeled walk W, associate monomial $f(W)$ that is product of edge variables, vertex variables of labeled vertices, and color-label pair variables corresponding to labeled vertices

$$f(sa\overset{1}{b}c\overset{2}{d}bet) = f_e(sa)f_e(ab)f_e(bc)f_e(cd)f_e(db)f_e(be)f_e(et)f_v(b)f_v(d)f_c({\color{blue}\bullet}, 1)f_c({\color{green}\bullet}, 2)$$

For integer ℓ, let $\mathcal{C}_\ell$ be the family of labeled-digon-free labeled (s, t)-walks of length ℓ

Polynomial over labeled walks



$k = 2$

Definition:

- For each edge uv associate variable $f_e(uv)$
- For each vertex w associate variable $f_v(w)$
- For each color-label pair (x, y) associate variable $f_c(x, y)$

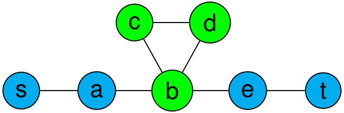
For a labeled walk W, associate monomial $f(W)$ that is product of edge variables, vertex variables of labeled vertices, and color-label pair variables corresponding to labeled vertices

$$f({\color{red} sab\overset{1}{c}\overset{2}{d}bet}) = f_e(sa)f_e(ab)f_e(bc)f_e(cd)f_e(db)f_e(be)f_e(et)f_v(b)f_v(d)f_c({\color{blue}\bullet}, 1)f_c({\color{green}\bullet}, 2)$$

For integer ℓ, let $\mathcal{C}_\ell$ be the family of labeled-digon-free labeled (s, t)-walks of length ℓ

- Define $f(\mathcal{C}_\ell) = \sum_{W \in \mathcal{C}_\ell} f(W)$

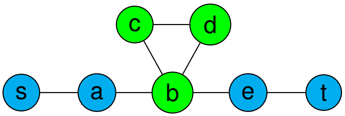
Yes-instances



$k = 2$

(1) Exists a k-colored (s, t)-path of length ℓ $\Rightarrow$ $f(\mathcal{C}_\ell)$ non-zero

Yes-instances

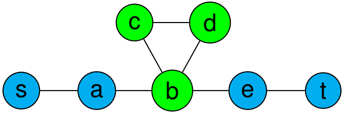


$k = 2$

(1) Exists a k-colored (s, t)-path of length ℓ $\Rightarrow$ $f(\mathcal{C}_\ell)$ non-zero

- Labeled walk $s\overset{1}{a}\overset{2}{b}et \in \mathcal{C}_5$ contributes the monomial $f_e(sa)f_e(ab)f_e(be)f_e(et)f_v(a)f_v(b)f_c(\bullet, 1)f_c(\bullet, 2)$

Yes-instances

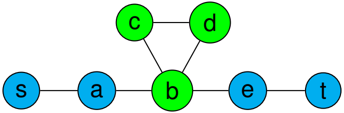


$k = 2$

(1) Exists a k-colored (s, t)-path of length ℓ $\quad\Rightarrow\quad$ $f(\mathcal{C}_\ell)$ non-zero

- Labeled walk $s\overset{1}{a}\overset{2}{b}et \in \mathcal{C}_5$ contributes the monomial $f_e(sa)f_e(ab)f_e(be)f_e(et)f_v(a)f_v(b)f_c(\bullet, 1)f_c(\bullet, 2)$
- No other labeled walk in $\mathcal{C}_5$ contributes the same monomial

Yes-instances



$k = 2$

(1) Exists a k-colored (s, t)-path of length ℓ $\Rightarrow$ $f(\mathcal{C}_\ell)$ non-zero

- Labeled walk $\overset{1}{s}\overset{2}{a}bet \in \mathcal{C}_5$ contributes the monomial $f_e(sa)f_e(ab)f_e(be)f_e(et)f_v(a)f_v(b)f_c(\bullet, 1)f_c(\bullet, 2)$
- No other labeled walk in $\mathcal{C}_5$ contributes the same monomial

$\Rightarrow$ $f(\mathcal{C}_5)$ is non-zero

No-instances

(2) No k-colored (s, t)-path of length $\leq \ell \quad \Rightarrow \quad f(\mathcal{C}_\ell)$ identically zero

No-instances

(2) No k-colored (s, t)-path of length $\leq \ell$ $\Rightarrow$ $f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

No-instances

(2) No k-colored (s, t)-path of length $\leq \ell$ $\Rightarrow$ $f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

$\Rightarrow$ Labeled walks in $\mathcal{C}_\ell$ can be paired as $\{W, \phi(W)\}$, implying everything cancels out over fields of characteristic 2

No-instances

(2) No k-colored (s, t)-path of length $\leq \ell \quad \Rightarrow \quad f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

$\Rightarrow$ Labeled walks in $\mathcal{C}_\ell$ can be paired as $\{W, \phi(W)\}$, implying everything cancels out over fields of characteristic 2

Three different cancellation arguments as building blocks for ϕ:

No-instances

(2) No k-colored (s, t)-path of length $\leq \ell \quad \Rightarrow \quad f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

$\Rightarrow$ Labeled walks in $\mathcal{C}_\ell$ can be paired as $\{W, \phi(W)\}$, implying everything cancels out over fields of characteristic 2

Three different cancellation arguments as building blocks for ϕ:

- Bijective-labeling based cancellation (from [Björklund'14], [Björklund, Husfeldt, Kaski, Koivisto'17])

No-instances

(2) No k-colored (s, t)-path of length $\leq \ell \quad \Rightarrow \quad f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

$\Rightarrow$ Labeled walks in $\mathcal{C}_\ell$ can be paired as $\{W, \phi(W)\}$, implying everything cancels out over fields of characteristic 2

Three different cancellation arguments as building blocks for ϕ:

- Bijective-labeling based cancellation (from [Björklund'14], [Björklund, Husfeldt, Kaski, Koivisto'17])
- Cycle-reversal based cancellation (from [Björklund, Husfeldt, Taslaman'12])

No-instances

(2) No k-colored (s,t)-path of length $\leq \ell \quad \Rightarrow \quad f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

$\Rightarrow$ Labeled walks in $\mathcal{C}_\ell$ can be paired as $\{W, \phi(W)\}$, implying everything cancels out over fields of characteristic 2

Three different cancellation arguments as building blocks for ϕ:

- Bijective-labeling based cancellation (from [Björklund'14], [Björklund, Husfeldt, Kaski, Koivisto'17])
- Cycle-reversal based cancellation (from [Björklund, Husfeldt, Taslaman'12])
- Label-swap based cancellation

No-instances

(2) No k-colored (s,t)-path of length $\leq \ell \quad \Rightarrow \quad f(\mathcal{C}_\ell)$ identically zero

Goal: Design a function $\phi : \mathcal{C}_\ell \to \mathcal{C}_\ell$ so that for all $W \in \mathcal{C}_\ell$

- $f(W) = f(\phi(W))$
- $\phi(W) \neq W$
- $\phi(\phi(W)) = W$

$\Rightarrow$ Labeled walks in $\mathcal{C}_\ell$ can be paired as $\{W, \phi(W)\}$, implying everything cancels out over fields of characteristic 2

Three different cancellation arguments as building blocks for ϕ:

- Bijective-labeling based cancellation (from [Björklund'14], [Björklund, Husfeldt, Kaski, Koivisto'17])
- Cycle-reversal based cancellation (from [Björklund, Husfeldt, Taslaman'12])
- Label-swap based cancellation

Combination of arguments: 18 cases and 14 pages

Conclusion

- We gave a $2^k n^{\mathcal{O}(1)}$ time algorithm for finding an (s, t)-path with at least k different colors

Conclusion

- We gave a $2^k n^{\mathcal{O}(1)}$ time algorithm for finding an (s, t)-path with at least k different colors
  - (and generalizations)

Conclusion

- We gave a $2^k n^{\mathcal{O}(1)}$ time algorithm for finding an (s,t)-path with at least k different colors
  - (and generalizations)
- The proof that if no solution exists then $f(\mathcal{C}_\ell)$ is identically zero is the most complicated part

Conclusion

- We gave a $2^k n^{\mathcal{O}(1)}$ time algorithm for finding an (s, t)-path with at least k different colors
  - (and generalizations)
- The proof that if no solution exists then $f(\mathcal{C}_\ell)$ is identically zero is the most complicated part
- Open problem: Is there an FPT-algorithm for maximum colored two disjoint paths?

Conclusion

- We gave a $2^k n^{\mathcal{O}(1)}$ time algorithm for finding an (s, t)-path with at least k different colors
  - (and generalizations)
- The proof that if no solution exists then $f(\mathcal{C}_\ell)$ is identically zero is the most complicated part
- Open problem: Is there an FPT-algorithm for maximum colored two disjoint paths?
- Open problem: Is there a $1.99^k n^{\mathcal{O}(1)}$ time algorithm for longest (s, t)-path?

Thank you!