Parameterized Complexity and Approximation Issues for the Colorful Components Problems

Riccardo Dondi[1] Florian Sikora[2]

[1]Universita degli Studi di Bergamo – Italy
[2]LAMSADE, Université Paris Dauphine, CNRS – France

CiE 2016

Parameterized Complexity and Approximation Issues for the Colorful Components Problems

Riccardo Dondi[1] Florian Sikora[2]

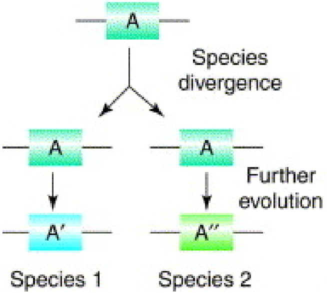
[1]Universita degli Studi di Bergamo – Italy
[2]LAMSADE, Université Paris Dauphine, CNRS – France

CiE 2016

Outline

Motivations from Comparative Genomics.

Find orthologous genes.

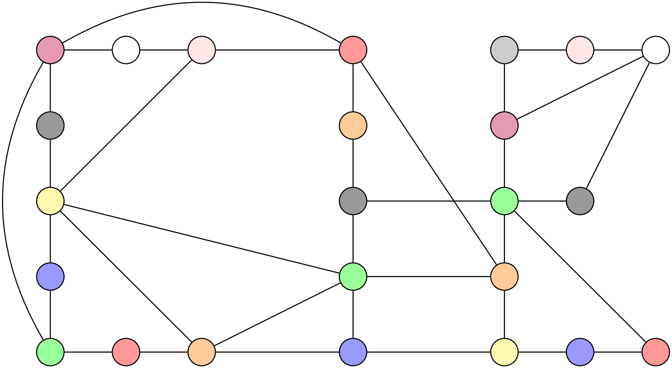


[Trends in Genetics]

Motivations

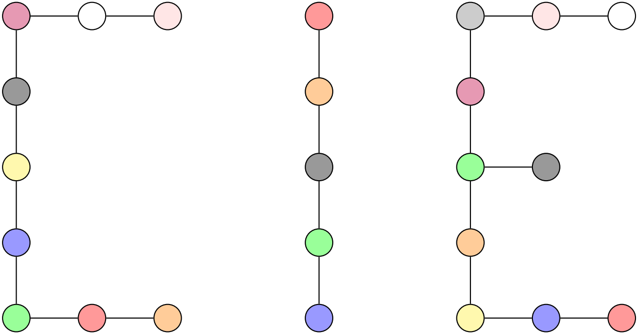
- Graph approach to identify disjoint orthology sets in [WABI'11].
- Partition a **vertex-colored** graph in **colorful** components by removing edges.
- Different optimization measures:
  - Number of components in the solution [LATIN'14].
  - Sum of the number of edges in the transitive closure [WABI'11].

Minimum Colorful Components (MCC)



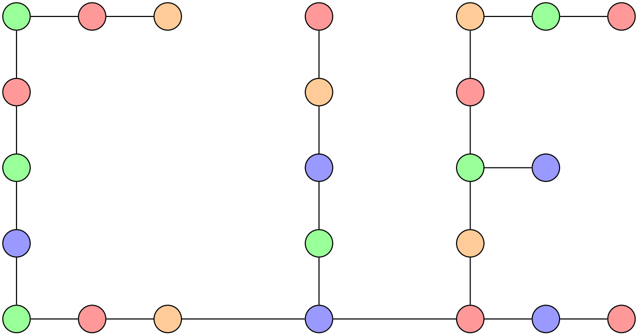
Remove edges to **minimize** the number of **colorful** components.

Minimum Colorful Components (MCC)

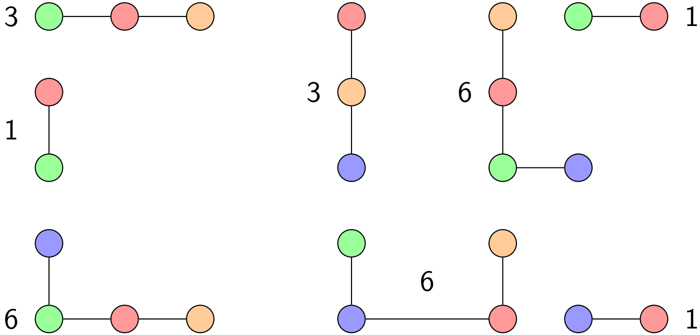


Remove edges to **minimize** the number of **colorful** components.

Maximum Edges in Transitive Closure (MEC)



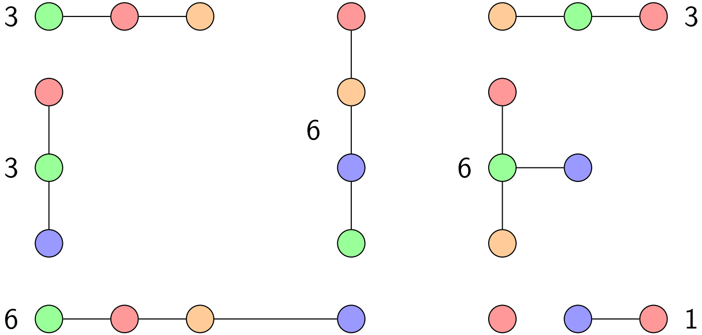
Maximum Edges in Transitive Closure (MEC)



Remove edges to have **colorful** components and **maximize** the number of edges in the transitive closure.

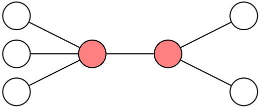
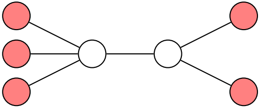
Here: $3 + 1 + 6 + 3 + 6 + 6 + 1 + 1 = 27$

Maximum Edges in Transitive Closure (MEC)



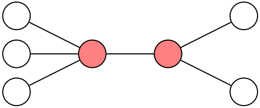
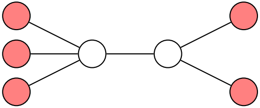
Remove edges to have **colorful** components and **maximize** the number of edges in the transitive closure.
Here: $3 + 3 + 6 + 6 + 3 + 6 + 1 = 28$

Parameterized Complexity

Problem:	VERTEX COVER	INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Cover edges with k vertices	Find k independent vertices
Compl.:		

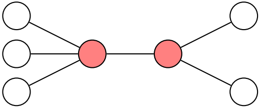
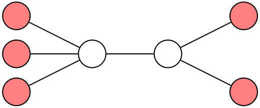
Example from D. Marx.

Parameterized Complexity

Problem:	VERTEX COVER	INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Cover edges with k vertices	Find k independent vertices
Compl.:		
Complexity:	NP-complete	NP-complete

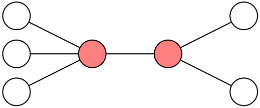
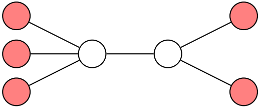
Example from D. Marx.

Parameterized Complexity

Problem:	VERTEX COVER	INDEPENDENT SET
Input:	Graph G, integer k	Graph G, integer k
Question:	Cover edges with k vertices	Find k independent vertices
Compl.:		
Complexity:	NP-complete	NP-complete
Brute-force:	$O(n^k)$ possibilities	$O(n^k)$ possibilities

Example from D. Marx.

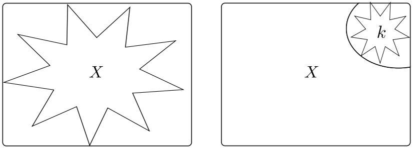
Parameterized Complexity

Problem:	Vertex Cover	Independent Set
Input:	Graph G, integer k	Graph G, integer k
Question:	Cover edges with k vertices	Find k independent vertices
		
Compl.:		
Complexity:	NP-complete	NP-complete
Brute-force:	$O(n^k)$ possibilities	$O(n^k)$ possibilities
Smarter?:	$O(2^k n^2)$ **algorithm**	**No** $f(k)n^{O(1)}$ **algorithm exists**
		

Example from D. Marx.

Fixed-Parameter Tractability

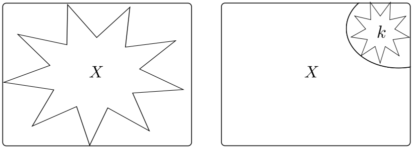
- Problem in FPT: any instance (I, k) solved in $f(k) \cdot |I|^c$.



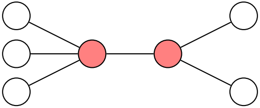
- Examples:
  - Solution of size k in a n-vertices graph.
  - n voters for k candidates.
  - Requests of size k in a n-sized database.
  - ...

Fixed-Parameter Tractability

- Problem in FPT: any instance (I, k) solved in $f(k) \cdot |I|^c$.

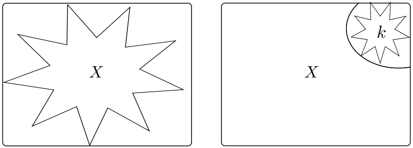


- Examples:
  - Solution of size k in a n-vertices graph.
  - n voters for k candidates.
  - Requests of size k in a n-sized database.
  - ...
- Many way to parameterize.
  - Solution size.



Fixed-Parameter Tractability

- Problem in FPT: any instance (I, k) solved in $f(k) \cdot |I|^c$.



- Examples:
  - Solution of size k in a n-vertices graph.
  - n voters for k candidates.
  - Requests of size k in a n-sized database.
  - ...

- Many way to parameterize.
  - Solution size.
  - Structure of the input.
  - ...

How to obtain FPT algorithm?

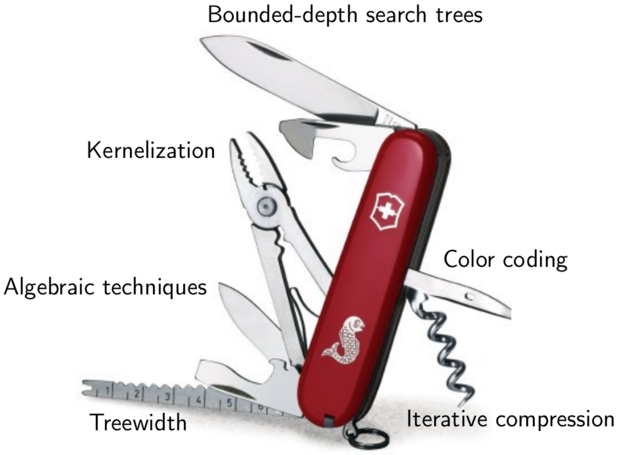
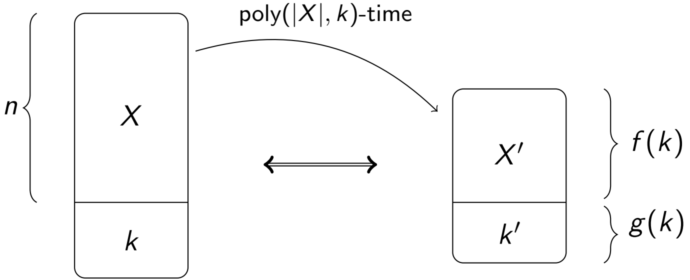


Illustration D. Marx.

Kernelization



Squirrel from [CFKLMPPS'15]

Outline

MCC

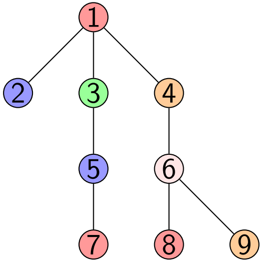
- Known:
  - Not approximable within $O(n^{1/14-\varepsilon})$ [LATIN'14].
  - No $n^{f(k)}$ algorithm (k number of components) [LATIN'14].

MCC

- Known:
  - Not approximable within $O(n^{1/14-\varepsilon})$ [LATIN'14].
  - No $n^{f(k)}$ algorithm (k number of components) [LATIN'14].
- New:
  - On **trees**, not approximable within $1.36 - \varepsilon$ but 2-approximable, **FPT** and polynomial kernel.
  - Polynomial on **paths** but NP-hard for graph with distance to disjoint paths 1.

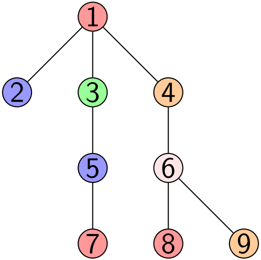
MCC in trees is Multicut in trees - Positive

- From a MCC instance on trees...



MCC in trees is Multicut in trees - Positive

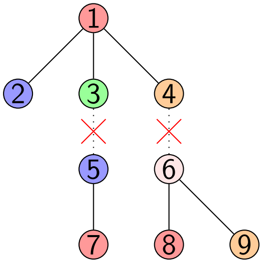
- From a MCC instance on trees...



- ... create a instance of Multicut with the same tree and:
  - $(1,7),(1,8),(7,8),(2,5),(4,9)$ to separate.

MCC in trees is Multicut in trees - Positive

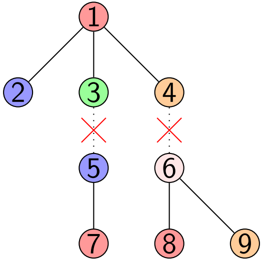
- From a MCC instance on trees...



- ... create a instance of Multicut with the same tree and:
  - $(1,7),(1,8),(7,8),(2,5),(4,9)$ to separate.

MCC in trees is Multicut in trees - Positive

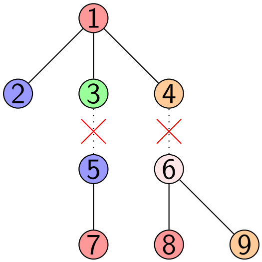
- From a MCC instance on trees...



- ... create a instance of Multicut with the same tree and:
  - $(1,7),(1,8),(7,8),(2,5),(4,9)$ to separate.
- k edges to cut $\leftrightarrow$ $k+1$ colorful components in the tree.

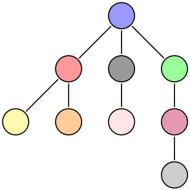
MCC in trees is Multicut in trees - Positive

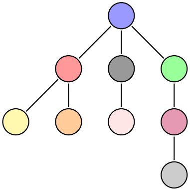
- From a MCC instance on trees...



- ... create a instance of Multicut with the same tree and:
  - $(1,7),(1,8),(7,8),(2,5),(4,9)$ to separate.
- k edges to cut $\leftrightarrow$ $k+1$ colorful components in the tree.
  - k^3 (bi)-kernel.
  - Solvable in $O^*(1.56^k)$.
  - 2-approximable.

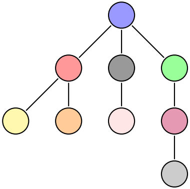
MCC in trees is Multicut in trees – Negative



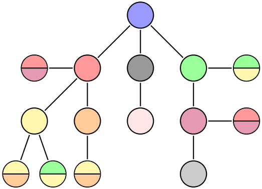


MCC

MCC in trees is Multicut in trees – Negative

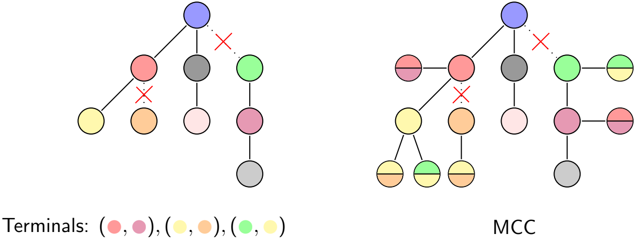


Terminals: (●, ●), (●, ●), (●, ●)



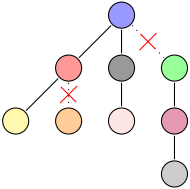
MCC

MCC in trees is Multicut in trees – Negative

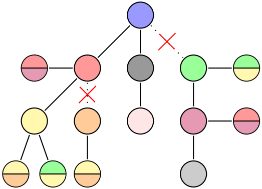


- k edges to cut $\leftrightarrow$ $k + 1$ colorful components

MCC in trees is Multicut in trees – Negative



Terminals: (●, ●), (●, ●), (●, ●)

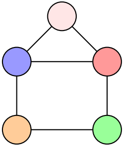


MCC

- k edges to cut $\leftrightarrow$ $k+1$ colorful components
  - MCC not approximable within $1.36-\varepsilon$ unless P=NP.

MCC hardness

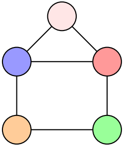
Vertex Cover:



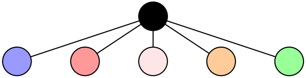
MCC:

MCC hardness

VERTEX COVER:

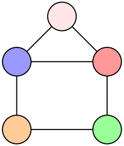


MCC:

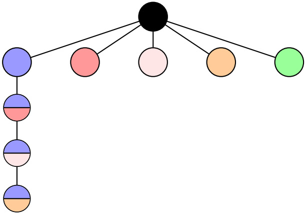


MCC hardness

Vertex Cover:

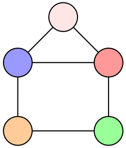


MCC:

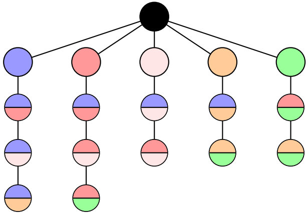


MCC hardness

Vertex Cover:

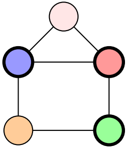


MCC:

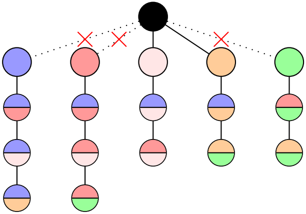


MCC hardness

VERTEX COVER:



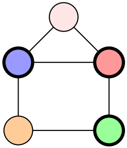
MCC:



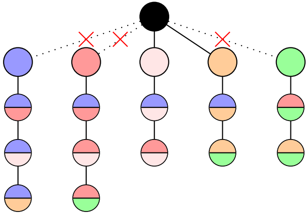
- vertex cover of size $k \leftrightarrow k + 1$ colorful components.

MCC hardness

Vertex Cover:



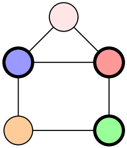
MCC:



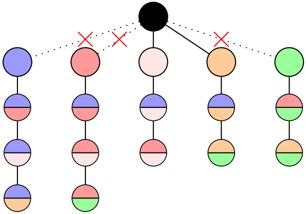
- vertex cover of size $k \leftrightarrow k+1$ colorful components.
- NP-hard for graphs with distance to disjoint paths 1.

MCC hardness

VERTEX COVER:



MCC:



- vertex cover of size $k \leftrightarrow k+1$ colorful components.
- NP-hard for graphs with distance to disjoint paths 1.
  - No $O(n^{f(k)})$ algorithm for $k=$
    - distance to disjoint paths (but poly on paths via DP),
    - pathwidth,
    - distance to interval graphs.

Outline

Introduction

MCC

MEC

MEC

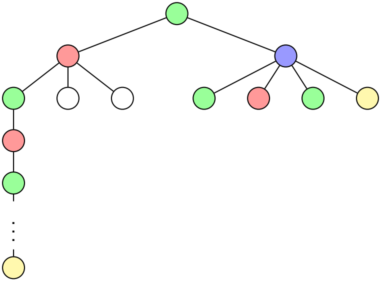
- Known:
  - APX-hard even with 3 colors [Iwoca'14]
  - Not approximable within $O(n^{1/3-\varepsilon})$, even for trees where each color appears twice [Iwoca'14].
  - $\sqrt{2 \cdot OPT}$ approximable [Iwoca'14].

MEC

- Known:
  - APX-hard even with 3 colors [Iwoca'14]
  - Not approximable within $O(n^{1/3-\varepsilon})$, even for trees where each color appears twice [Iwoca'14].
  - $\sqrt{2 \cdot OPT}$ approximable [Iwoca'14].
- New:
  - **FPT** w.r.t. size of the solution.
  - Kernel polynomial on **trees**, exponential on graphs.
  - Polynomial on **paths** but NP-hard for graph with distance to disjoint paths 1.

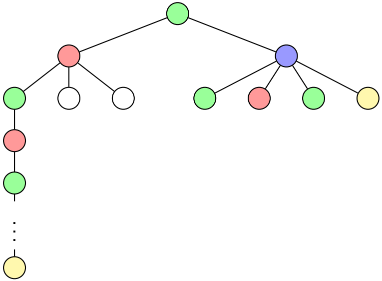
A poly-kernel for trees

- Suppose there is no 2 adjacent nodes with same color.



A poly-kernel for trees

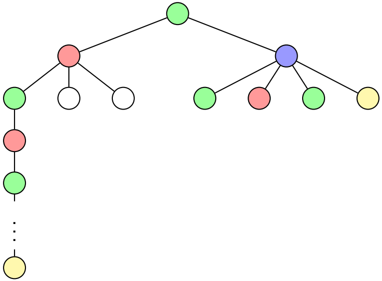
- Suppose there is no 2 adjacent nodes with same color.



- If **path of length** $2k$ from root to leaf: trivial yes (matching).

A poly-kernel for trees

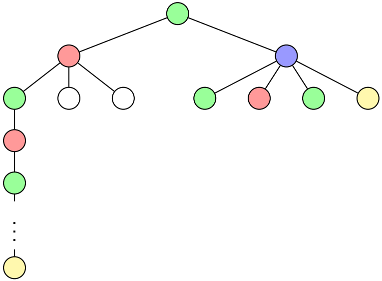
- Suppose there is no 2 adjacent nodes with same color.



- If **path of length** $2k$ from root to leaf: trivial yes (matching).
- If k **ancestors of leafs**: trivial yes (matching).

A poly-kernel for trees

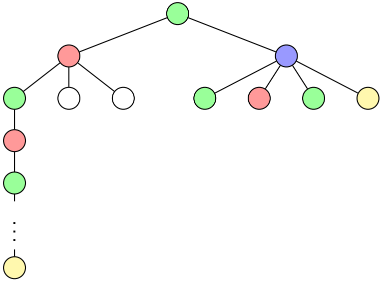
- Suppose there is no 2 adjacent nodes with same color.



- If **path of length** $2k$ from root to leaf: trivial yes (matching).
- If k **ancestors of leafs**: trivial yes (matching).
- If an ancestor with $\sqrt{2k}$ **leafs of different colors**: done.

A poly-kernel for trees

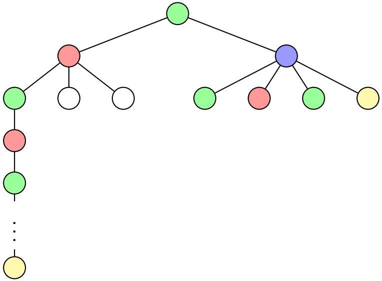
- Suppose there is no 2 adjacent nodes with same color.



- If **path of length** $2k$ from root to leaf: trivial yes (matching).
- If k **ancestors of leafs**: trivial yes (matching).
- If an ancestor with $\sqrt{2k}$ **leafs of different colors**: done.
  - Remove **all but one** occurrence of each color for an ancestor.

A poly-kernel for trees

- Suppose there is no 2 adjacent nodes with same color.



- If **path of length** $2k$ from root to leaf: trivial yes (matching).
- If k **ancestors of leafs**: trivial yes (matching).
- If an ancestor with $\sqrt{2k}$ **leafs of different colors**: done.
  - Remove **all but one** occurrence of each color for an ancestor.
- Tree with $O(k^2)$ **nodes**.

A kernel for graphs

- Can be extended to general graphs.
- Use a DFS giving a spanning tree.
- But gives exponential kernel.

Open questions

- Kernel lower bound?
- Fine grained complexity?
- More structural results?
- Parameterization above guarantee?

Merci!