The GRAPH MOTIF problem

Guillaume Fertin
LS2N, Université de Nantes, France
March 2017

Some slides in this talk are courtesy:

- C. Komusiewicz, FS U. Jena
- F. Sikora U. Paris Dauphine
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
Motif Search in Texts

- Goal: search all occurrences of a motif in a text.
  - $T =$ text, of length $n$
  - $M =$ motif, of length $m$
  - $M$ and $T$ built on some alphabet $\Sigma$
  - typical use: $m \ll n$
Motif Search in Texts

- Goal: search all occurrences of a motif in a text.
  - $T =$ text, of length $n$
  - $M =$ motif, of length $m$
  - $M$ and $T$ built on some alphabet $\Sigma$
  - typical use: $m << n$

- Applications:
  - search for a word in a text editor [ctrl-f] ($|\Sigma| \sim 60 - 70$)
  - bioinformatics: DNA ($|\Sigma| = 4$), proteins ($|\Sigma| = 20$)
Motif Search in Texts

- Goal: search all occurrences of a motif in a text.
  - \( T \) = text, of length \( n \)
  - \( M \) = motif, of length \( m \)
  - \( M \) and \( T \) built on some alphabet \( \Sigma \)
  - typical use: \( m \ll n \)

- Applications:
  - search for a word in a text editor [ctrl-f] (\(|\Sigma| \sim 60 - 70\))
  - bioinformatics: DNA (\(|\Sigma| = 4\)), proteins (\(|\Sigma| = 20\))

- Algorithmics:
  - clearly polynomial (naive search w/ sliding window is in \( O(mn) \))
  - nice algorithms back from the 70s (KMP, Boyer-Moore, etc.)
  - see also e.g.
Analysis of Algorithms

- Analysis of an algorithm, say $A$
- Running time of $A \approx$ number of “elementary operations” executed by $A$
Analysis of Algorithms

- Analysis of an algorithm, say $A$
- Running time of $A \approx$ number of “elementary operations” executed by $A$

- Elementary operation:
  - arithmetic operation (+,-,/,*), memory access, assignment, comparison
  - unit cost assumed for each
Analysis of Algorithms

- Analysis of an algorithm, say $A$
- Running time of $A \simeq$ number of "elementary operations" executed by $A$
- Elementary operation:
  - arithmetic operation (+, -, /, *), memory access, assignment, comparison
  - unit cost assumed for each
- Running time = $f(n)$, function of input size $n$ of the instance
O() notation

- Goal: simplify $f(n) \rightarrow g(n)$
O() notation

- Goal: simplify $f(n) \rightarrow g(n)$
- $f(n) = O(g(n))$ if
  $\exists c > 0, n_0$ s.t. $f(n) \leq c \cdot g(n)$ $\forall n \geq n_0$
- $\rightarrow g()$ is an upper bound for $f()$
O() notation

- Goal: simplify $f(n) \rightarrow g(n)$
- $f(n) = O(g(n))$ if
  \[ \exists c > 0, n_0 \text{ s.t. } f(n) \leq c \cdot g(n) \quad \forall n \geq n_0 \]
- $\rightarrow g()$ is an upper bound for $f()$
- Roughly: take $f(n)$, keep dominant term, remove multiplicative constant
- Example:
  - $f(n) = 7n^2 + 3n \log n + 12\sqrt{n} - 7$
  - $f(n) = O(n^2)$
O() notation

▶ Goal: simplify \( f(n) \rightarrow g(n) \)

▶ \( f(n) = O(g(n)) \) if

\[
\exists c > 0, n_0 \text{ s.t. } f(n) \leq c \cdot g(n) \quad \forall n \geq n_0
\]

▶ \( g() \) is an upper bound for \( f() \)

▶ Roughly: take \( f(n) \), keep dominant term, remove multiplicative constant

▶ Example:

\[
\begin{align*}
\text{f}(n) &= 7n^2 + 3n \log n + 12\sqrt{n} - 7 \\
\text{f}(n) &= O(n^2)
\end{align*}
\]

▶ \( O() \) used for worst-case analysis – robustness of algorithm
void naive(M[0..m-1], T[0..n-1])
1. for i=0 to n-m do
2.   j <-- 0;
3.   while M[j]=T[i+j] && j<= m-1 do
4.       j <-- j+1;
5.   endwhile
6.   if j=m then
7.       printf('Motif found at position %d\n',i);
8.   endif
9. endfor
Motif search - naive algorithm (sliding window)

```c
void naive(M[0..m-1], T[0..n-1])
1. for i=0 to n-m do
2.   j <-- 0;
3.   while M[j]=T[i+j] && j<= m-1 do
4.       j <-- j+1;
5.   endwhile
6.   if j=m then
7.       printf(‘‘Motif found at position %d\n’’,i);
8.   endif
9. endfor
```

- each line (individually): constant number of elementary operations
- Lines 3. and 4. most costly: executed at worse $m(n - m)$ times
- $f(n) = O(m(n - m)) = O(nm)$
Motif Search in Graphs

- species: yeast
- vertices ↔ proteins (∼ 3 500)
- edges ↔ interactions (∼ 11 000)

Source: http://compbio.pbworks.com/
Motif Search in Graphs

- species: yeast
- vertices ↔ proteins (∼ 3 500)
- edges ↔ interactions (∼ 11 000)
Motif Search in Graphs

Goal: search one/all occurrence/s of a small graph $H$ in a big graph $G$.

- $G =$ target graph
- $H =$ query graph (motif)
- typical use: $|V(H)| << |V(G)|$
Motif Search in Graphs

Goal: search one/all occurrence/s of a small graph $H$ in a big graph $G$.

- $G$ = target graph
- $H$ = query graph (motif)
- typical use: $|V(H)| << |V(G)|$

Remarks

- $H$: biologically known pathway or a complex of interest
- occurrence = induced subgraph of $G$ isomorphic to $H$
- $\rightarrow$ topology-based approach
Towards topology-free motifs

Two views for Motif Search in Graphs

- **Topological view:**
  - find a small graph in a big graph
  - $\Rightarrow$ subgraph isomorphism problems
Towards topology-free motifs

Two views for Motif Search in Graphs

- **Topological view:**
  - find a small graph in a big graph
  - $\Rightarrow$ subgraph isomorphism problems

- **Functional view:**
  - topology is less important
  - **functionalities** of network vertices $\Rightarrow$ governing principle
  - initiated in Lacroix, Fernandes & Sagot, IEEE/ACM TCBB 06
Topology-free motifs

Applicable in broader scenarios

- motif (pathway or complex) whose topology is not completely known
- noisy networks (missing connections)
- query between well and poorly annotated species
Functional approach

Model

- function $\leftrightarrow$ color
- $\Rightarrow$ graph is vertex-colored (but not properly!)
Functional approach

Model

- function $\leftrightarrow$ color
- $\Rightarrow$ graph is vertex-colored (but not properly!)
- motif (query): multiset of colors
Functional approach

Model

- function \( \leftrightarrow \) color
- \( \Rightarrow \) graph is vertex-colored (but not properly!)
- motif (query): multiset of colors
- motif occurs (and thus “accepted”) if connected in graph
Definition (GRAPH MOTIF – LACROIX ET AL., IEEE/ACM TCBB 06)

Input: A graph $G = (V, E)$, a set of colors $C$, a coloring function $\chi : V \rightarrow C$, a motif* $M$ over $C$

* motif = multiset of colors whose underlying set is $C$. 
**Graph Motif**

**Definition (Graph Motif — Lacroix et al., IEEE/ACM TCBB 06)**

**Input:** A graph $G = (V, E)$, a set of colors $C$, a coloring function $\chi : V \rightarrow C$, a motif* $M$ over $C$

* motif = multiset of colors whose underlying set is $C$.

**Question:** Is there an occurrence of $M$ in $G$?
**Definition (GRAPH MOTIF – LACROIX ET AL., IEEE/ACM TCBB 06)**

**Input:** A graph $G = (V, E)$, a set of colors $C$, a coloring function $\chi : V \to C$, a motif* $M$ over $C$

* motif = multiset of colors whose underlying set is $C$.

**Question:** Is there an occurrence of $M$ in $G$?

**Occurrence** = subset $V' \subseteq V$ s.t.

- $\chi(V') = M$, and
- $G[V']$ is connected
**Definition (GRAPH MOTIF – LACROIX ET AL., IEEE/ACM TCBB 06)**

**Input:** A graph $G = (V, E)$, a set of colors $C$, a coloring function $\chi : V \to C$, a motif* $M$ over $C$

* motif = multiset of colors whose underlying set is $C$.

**Question:** Is there an occurrence of $M$ in $G$?

Occurrence = subset $V' \subseteq V$ s.t.

- $\chi(V') = M$, and
- $G[V']$ is connected

Note: if $\chi : V \to C'$ with $C \subseteq C'$, pre-process $G$ by deleting vertices $u \in V(G)$ s.t. $\chi(u) \notin C$
Example
Example
Example
Graph Motif

Applications

- **metabolic networks analysis** [Lacroix, Fernandes & Sagot, IEEE/ACM TCBB 06]
- **PPI networks analysis** [Bruckner et al., J. Comp. Biol. 10]
Applications

- **metabolic networks analysis** [Lacroix, Fernandes & Sagot, IEEE/ACM TCBB 06]
- **PPI networks analysis** [Bruckner et al., J. Comp. Biol. 10]
- **mass spectrometry** (identification of metabolites) [Böcker & Rasche, Bioinformatics 08]
Applications

- **metabolic networks analysis** [*Lacroix, Fernandes & Sagot, IEEE/ACM TCBB 06*]
- **PPI networks analysis** [*Bruckner et al., J. Comp. Biol. 10*]
- **mass spectrometry** (identification of metabolites) [*Böcker & Rasche, Bioinformatics 08*]
- **also study of social networks** [*Pinter-Wollman et al., Behavioral Ecology 14*]
Graph Motif

A well-studied problem

- **Graph Motif** widely studied: ~150 citations for seminal paper in 11 years (source: Google Scholar)
Graph Motif

A well-studied problem

- Graph Motif widely studied: ~150 citations for seminal paper in 11 years (source: Google Scholar)
- Many variants (...too many ?), e.g.:
  - approximate motif
  - connectivity of an occurrence
  - list-colored vertices
Graph Motif

A well-studied problem

- **Graph Motif** widely studied: ~150 citations for seminal paper in 11 years (source: Google Scholar)

- Many variants (...too many ?), e.g.:
  - approximate motif
  - connectivity of an occurrence
  - list-colored vertices

- Several software (a handful): Motus, Torque, GraMoFoNe, PINQ, etc.
A well-studied problem

- **Graph Motif** widely studied: ~150 citations for seminal paper in 11 years (source: Google Scholar)

- Many variants (...too many ?), e.g.:
  - approximate motif
  - connectivity of an occurrence
  - list-colored vertices

- Several software (a handful): Motus, Torque, GraMoFoNe, PINQ, etc.

This talk

- Algorithmic results for **Graph Motif**: a guided tour
- Multiplicity of proof techniques: classical, *ad hoc*, imported from other contexts
Some notations

- $M^*$ = underlying set of $M$
- $M$ is colorful if $M^* = M$
Some notations

- $M^* = \text{underlying set of } M$
- $M$ is \textbf{colorful} if $M^* = M$
- \textbf{COLORFUL GRAPH MOTIF} (or CGM): restriction of GRAPH MOTIF to colorful motifs
Some notations

- $M^* = \text{underlying set of } M$
- $M$ is colorful if $M^* = M$

**COLORFUL GRAPH MOTIF** (or CGM): restriction of GRAPH MOTIF to colorful motifs

- $\mu(G, c) = \text{number of vertices having color } c \text{ in } G$
- $\mu(G) = \max\{\mu(G, c) : c \in C\}$
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
**Theorem** (Lacroix et al., IEEE/ACM TCBB 06)

**Graph Motif** is **NP-complete** even if **G** is a tree.
Did you say \textbf{NP}-complete? \\

Algorithmic complexity of Problems \\

- $Pb=$a problem, $n=$size of the input
Did you say **NP-complete**?

**Algorithmic complexity of Problems**

- $Pb$ = a problem, $n$ = size of the input
  - $Pb$ is **tractable** if solvable in $O(n^c)$ ($c$=constant) $\Rightarrow Pb \in P$
Did you say \textbf{NP}-complete \ ?

\textbf{Algorithmic complexity of Problems}

\begin{itemize}
  \item \textit{Pb}=a problem, \textit{n}=size of the input
  \item \textit{Pb} is \textbf{tractable} if solvable in $O(n^c)$ ($c=$constant) $\Rightarrow \ \textit{Pb} \in \mathbf{P}$
  \item \textit{Pb} is \textbf{intractable} if no $O(n^c)$ algo. exists for solving it  
  $\Rightarrow \ \textit{Pb} \notin \mathbf{P}$
\end{itemize}
Did you say **NP-complete**?

**Algorithmic complexity of Problems**

- $Pb$ is a problem, $n$=size of the input
- $Pb$ is **tractable** if solvable in $O(n^c)$ ($c$=constant) $\Rightarrow Pb \in P$
- $Pb$ is **intractable** if no $O(n^c)$ algo. exists for solving it $\Rightarrow Pb \notin P$
- very often: we do not know
Very often:

- cannot prove $Pb \in \mathbf{P}$
- cannot prove $Pb \notin \mathbf{P}$
Very often:
- cannot prove $Pb \in P$
- cannot prove $Pb \notin P$

Meanwhile...

**New class: NP-complete**

- Idea: identify the *most difficult* such problems
- $Pb$ is NP-complete if *reduction* from another NP-complete problem applies
Very often:
- cannot prove $Pb \in P$
- cannot prove $Pb \notin P$

Meanwhile...

**New class: NP-complete**

- Idea: identify the **most difficult** such problems
- $Pb$ is **NP-complete** if reduction from another **NP-complete** problem applies
- In this talk I will deliberately **not discuss NP-hard vs NP-complete**
Recess 2 (Cont’d)

Reduction – Principle

- Two problems: $Pb$ and $Pb'$
- $Pb$ and $Pb'$ are decision problems (answer: YES/NO)
- $Pb'$ is known to be NP-complete
Reduction – Principle

- Two problems: $Pb$ and $Pb'$
- $Pb$ and $Pb'$ are decision problems (answer: YES/NO)
- $Pb'$ is known to be NP-complete
- For any instance $I'$ of $Pb'$
Reduction – Principle

- Two problems: \( Pb \) and \( Pb' \)
- \( Pb \) and \( Pb' \) are decision problems (answer: Yes/No)
- \( Pb' \) is known to be \( \text{NP} \)-complete
- For any instance \( I' \) of \( Pb' \)
- build in polynomial time a specific instance \( I \) of \( Pb \)
Reduction – Principle

- Two problems: $Pb$ and $Pb'$
- $Pb$ and $Pb'$ are decision problems (answer: YES/NO)
- $Pb'$ is known to be NP-complete
- For any instance $l'$ of $Pb'$
- build in polynomial time a specific instance $l$ of $Pb$
- YES for $l \iff$ YES for $l'$
Recess 2 (Cont’d)

Meaning of all this

► If reduction applies, $Pb$ is at least as hard as $Pb'$
Recess 2 (Cont’d)

Meaning of all this

- If reduction applies, $Pb$ is at least as hard as $Pb'$
- $Pb \in \mathbf{P} \Rightarrow Pb' \in \mathbf{P}$ (using reduction)
Meaning of all this

- If reduction applies, \( Pb \) is at least as hard as \( Pb' \)
- \( Pb \in P \Rightarrow Pb' \in P \) (using reduction)
- \( \Rightarrow \) \textbf{NP}-complete = class of hardest such problems
- problems in \textbf{NP}-complete thought not to be polynomial-time solvable
- but remains unknown (cf “\( P = \text{NP} \) ?”)
Theorem (Lacroix et al., IEEE/ACM TCBB 06)

**GRAPH MOTIF** is **NP-complete** even if **G** is a tree.
**Graph Motif: first results**

**Theorem (Lacroix et al., IEEE/ACM TCBB 06)**

*Graph Motif is NP-complete even if G is a tree.*

- Reduction from Exact Cover by 3-Sets
Theorem (Lacroix et al., IEEE/ACM TCBB 06)

**GRAPH MOTIF** is **NP-complete** even if **G** is a tree.

- Reduction from **EXACT COVER BY 3-SETS**
- Proof **does not hold** for **COLORFUL GRAPH MOTIF**
- **Is** **COLORFUL GRAPH MOTIF** any “simpler”?
**Graph Motif: bad news**


**Colorful Graph Motif is NP-complete even when:**

- \( G \) is a tree and
- \( G \) has maximum degree 3 and
- \( \mu(G) = 3 \)
COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- Boolean formula $\Phi$
  - set $X = \{x_1, x_2 \ldots x_n\}$ of boolean variables
  - clauses $c_1, c_2 \ldots c_m$, each $c_i$ built from $X$
COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- Boolean formula $\Phi$
  - set $X = \{x_1, x_2 \ldots x_n\}$ of boolean variables
  - clauses $c_1, c_2 \ldots c_m$, each $c_i$ built from $X$

- Conjunctive Normal Form (CNF):
  - each clause $c_i$ contains only logical OR ($\lor$)
  - $\Phi$ contains clauses connected by logical AND only ($\land$)
COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- Boolean formula $\Phi$
  - set $X = \{x_1, x_2 \ldots x_n\}$ of boolean variables
  - clauses $c_1, c_2 \ldots c_m$, each $c_i$ built from $X$

- Conjunctive Normal Form (CNF):
  - each clause $c_i$ contains only logical OR ($\lor$)
  - $\Phi$ contains clauses connected by logical AND only ($\land$)

- Example:

$$\Phi = (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3})$$
COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- variable: $x_i$
- literal: $x_i$ or $\overline{x_i}$
COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- variable: $x_i$
- literal: $x_i$ or $\overline{x_i}$

$\Phi = (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3})$
COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- variable: $x_i$
- literal: $x_i$ or $\overline{x_i}$

- $\Phi = (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3})$

- Goal: satisfy $\Phi$
  - assign TRUE/FALSE to each $x_i$
  - s.t. $\Phi$ evaluates to TRUE, i.e.
    - each clause evaluates to TRUE
    - in each clause, at least one literal evaluates to TRUE
COLORFUL GRAPH MOTIF is NP-complete

Definition (SAT)

Input: a boolean formula $\Phi$ in CNF, built on $X = \{x_1, x_2 \ldots x_n\}$.

Question: Is there an assignment TRUE/FALSE of each $x_i$ s.t. $\Phi$ is satisfied?
**COLORFUL GRAPH MOTIF is NP-complete**

**Definition (SAT)**

**Input:** a boolean formula $\Phi$ in CNF, built on $X = \{x_1, x_2 \ldots x_n\}$.

**Question:** Is there an assignment TRUE/FALSE of each $x_i$ s.t. $\Phi$ is satisfied ?

- SAT is **NP**-complete (classical result)
COLORFUL GRAPH MOTIF is NP-complete

3-SAT-x
Many constrained versions of SAT are NP-complete, e.g.:
- each clause of $\Phi$ contains at most 3 literals, and
- each variable appears in at most 3 clauses, and
- each literal appears in at most 2 clauses
**COLORFUL GRAPH MOTIF** is NP-complete

**3-SAT-x**
Many constrained versions of SAT are NP-complete, e.g.:

- each clause of $\Phi$ contains at most 3 literals, and
- each variable appears in at most 3 clauses, and
- each literal appears in at most 2 clauses

$$\Phi = (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3})$$

variable $x_3$, literal $\overline{x_3}$
COLORFUL GRAPH MOTIF is NP-complete

From any instance of 3-SAT-\(x\) to an instance of CGM

\[
\Phi = (x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3}) \land (x_1 \lor \overline{x_2} \lor \overline{x_3})
\]

- from \(\Phi\)
- construct graph \(G\) as above
- \(M = \{1, 2 \ldots n, 1', 2 \ldots n', x_1, x_2 \ldots x_n, c_1, c_2 \ldots c_m\}\)
Reduction from 3-SAT-x to CGM

From any instance of 3-SAT-x to an instance of CGM

- $G$ is a tree of maximum degree 3 (literal appears in $\geq 2$ clauses)
- $\mu(G) = 3$ (clause contains $\leq 3$ literals)
- $M$ is colorful
Reduction from 3-SAT-x to CGM

From any instance of 3-SAT-x to an instance of CGM

- \( G \) is a tree of maximum degree 3 (literal appears in \( \geq 2 \) clauses)
- \( \mu(G) = 3 \) (clause contains \( \leq 3 \) literals)
- \( M \) is colorful

Equivalence \( \text{YES/NO} \) answer

- \( \Rightarrow \) Pick color \( x_i \) corresponding to assignment
- \( \Leftarrow \) Pick vertices \( x_i \) and \( c_j \) corresponding to occurrence of motif
**Theorem** (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)

**Colorful Graph Motif** is \textbf{NP-complete} even when:

- \( G \) is a tree and
- \( G \) has maximum degree 3 and
- \( \mu(G) = 3 \)
Theorem (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)  
**Colorful Graph Motif** is **NP-complete** even when:
- $G$ is a tree and
- $G$ has maximum degree 3 and
- $\mu(G) = 3$

- Restrictions on $G$ and $\mu(G) \rightarrow \text{NP-complete}$
- What if $M$ uses few colors?
**Theorem** (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)

**GRAPH MOTIF** is **NP-complete** even when:

- $G$ is bipartite and
- $G$ has maximum degree 4 and
- $|M^*| = 2$

- Reduction from **EXACT COVER BY 3-SETS**
Theorem (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)

**GRAPH MOTIF** is in P whenever G is a tree and $\mu(G) = 2$. 
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT
**GRAPH MOTIF: a polynomial case**

Equivalence with 2-SAT

[Diagram of a boolean expression tree with variables $x_1$, $\overline{x_2}$, $\overline{x_1}$, and $x_2$, rooted at $r$.]
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT

\[(x_4 \Rightarrow \overline{x_5})\]
**GRAPH MOTIF: a polynomial case**

Equivalence with 2-SAT

\[(\overline{x_3} \Rightarrow x_1) \land (x_5 \Rightarrow x_1) \land (x_3 \Rightarrow \overline{x_2}) \land (x_2 \Rightarrow \overline{x_1}) \land \ldots\]

2-SAT formula as \((A \Rightarrow B) \iff (\overline{B} \lor A)\)
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
Remarks

- motifs tend to be small in practice (compared to the target graph)
Remarks

- motifs tend to be small in practice (compared to the target graph)

  → Question 1: algorithm whose running time is

    - polynomial in $n = |V(G)|$ and
    - exponential in $k = |M|$ ?
Remarks

- motifs tend to be small in practice (compared to the target graph)

- Question 1: algorithm whose running time is
  - polynomial in \( n = |V(G)| \) and
  - exponential in \( k = |M| \) ?

- Question 2: algorithm whose running time is
  - polynomial in \( n = |V(G)| \) and
  - exponential in \( c = |M^*| \) ?
Remarks

- **motifs tend to be small** in practice (compared to the target graph)

- **Question 1:** algorithm whose running time is
  - polynomial in \( n = |V(G)| \) and
  - exponential in \( k = |M| \)

- **Question 2:** algorithm whose running time is
  - polynomial in \( n = |V(G)| \) and
  - exponential in \( c = |M^*| \)

- Fixed Parameterized Tractability (FPT) issues
Parameterized complexity

Definition (Fixed-parameter tractability)
A problem $P$ is fixed-parameter tractable (FPT) w.r.t. parameter $k$ if it can be solved in time

$$O(f(k) \cdot poly(n))$$

- $f$: any computable function depending only on $k$
- $n$: size of the input
- $poly(n)$: any polynomial function of $n$
Parameterized complexity

Definition (Fixed-parameter tractability)
A problem $P$ is fixed-parameter tractable (FPT) w.r.t. parameter $k$ if it can be solved in time

$$O(f(k) \cdot \text{poly}(n))$$

- $f$: any computable function depending only on $k$
- $n$: size of the input
- $\text{poly}(n)$: any polynomial function of $n$

- complexity also noted $O^*(f(k))$ (hidden polynomial factor)
- $\rightarrow$ corresponding complexity class: FPT
Parameterized complexity

Definition (Parameterized hierarchy)

\[ \text{FPT} \subseteq \text{W[1]} \subseteq \text{W[2]} \subseteq \ldots \subseteq \text{XP} \]

In a nutshell

- FPT problems: (hopefully) efficiently solvable for small values of parameter
- \text{W[1]}: first class of problems not believed to be in FPT
- \text{W[1]}-complete vs FPT ↔ NP-complete vs P
Parameterized complexity

Definition (Parameterized hierarchy)

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \ldots \subseteq \text{XP}$$

In a nutshell

- **FPT** problems: (hopefully) efficiently solvable for small values of parameter
Parameterized complexity

Definition (Parameterized hierarchy)

\[ \text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \ldots \subseteq \text{XP} \]

In a nutshell

- **FPT** problems: (hopefully) efficiently solvable for small values of parameter
- **W[1]**: first class of problems not believed to be in **FPT**
- **W[1]**-complete vs **FPT** \( \leftrightarrow \) **NP**-complete vs **P**
FPT: an ever-growing topic

Monographs

FPT: an ever-growing topic

Monographs


▶ Dedicated website http://fpt.wikidot.com/
FPT: main techniques

- Dynamic Programming (table size and computation exponential in parameter only)

- Bounded Search Tree: test all possible cases, show there are $O(f(k))$ such cases

- Kernelization: $(I, k) \rightarrow (I', k')$ with same solution, $I'$ solvable in $O(f(k) \cdot \text{poly}(n))$

- Iterative Compression

- Color-Coding

- etc.
FPT: main techniques

- **Dynamic Programming** (table size and computation exponential in parameter only)

- **Bounded Search Tree**: test all possible cases, show there are $O(f(k))$ such cases

- **Kernelization**: $(I, k) \rightarrow (I', k')$ with same solution, $I'$ solvable in $O(f(k) \cdot \text{poly}(n))$
FPT: main techniques

- **Dynamic Programming** (table size and computation exponential in parameter only)
- **Bounded Search Tree**: test all possible cases, show there are $O(f(k))$ such cases
- **Kernelization**: $(I, k) \rightarrow (I', k')$ with same solution, $I'$ solvable in $O(f(k) \cdot \text{poly}(n))$
- **Iterative Compression**
FPT: main techniques

- **Dynamic Programming** (table size and computation exponential in parameter only)
- **Bounded Search Tree**: test all possible cases, show there are $O(f(k))$ such cases
- **Kernelization**: $(I, k) \rightarrow (I', k')$ with same solution, $I'$ solvable in $O(f(k) \cdot poly(n))$
- **Iterative Compression**
- **Color-Coding**
- etc.
The choice is yours

- **Size of the motif** $k = |M| = \text{solution size}$ → classical parameter
The choice is yours

- **Size of the motif** $k = |M| = \text{solution size}
  \rightarrow \text{classical parameter}

- **Number of colors** of the motif $c = |M^*|$
  Remark: $c \leq k$ ($k = c$ for **COLORFUL GRAPH MOTIF**) thus “stronger” than $k$
GRAPH MOTIF and FPT: which parameters?

The choice is yours

- **Size** of the motif $k = |M| = \text{solution size}$
  $\rightarrow$ classical parameter

- **Number of colors** of the motif $c = |M^*|$
  Remark: $c \leq k$ ($k = c$ for COLORFUL GRAPH MOTIF) thus “stronger” than $k$

- **Dual parameter** $\ell = n - k$ (with $n = |V(G)|$)
  Dual = number of vertices *not* in the solution
Did you say dual?

Dual parameter $\ell = n - k$ is probably large... but:

- Reduction rules $\rightarrow$ smaller components in which $\ell \sim k$
- Worst case running time vs experimental running time
- Current-best algorithms for some subgraph mining problems use $\ell$ (HARTUNG ET AL., JGAA 15)
**GRAPH MOTIF: parameter c**

Reminder: $c = |M^*| = \#\text{colors in } M$
**Graph Motif: parameter c**

Reminder: \( c = |M^*| = \text{#colors in } M \)

**Theorem** (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)

*Graph Motif* is \( \text{W[1]} \)-complete when parameterized by \( c \), even in *trees.*
Reminder: $c = |M^*| = \#\text{colors in } M$


**Graph Motif** is **W[1]-complete** when parameterized by $c$, even in *trees*.

- Reduction from **CLIQUE**
**GRAPH MOTIF: parameter c**

Reminder: \( c = |M^*| = \#\text{colors in } M \)

**Theorem (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)\)**

**Graph Motif is \( W[1] \)-complete when parameterized by \( c \), even in trees.**

- Reduction from **CLIQUE**
- \( \Rightarrow \) \( c \) can be discarded for **Graph Motif**
- In proof of theorem, motif \( M \) is **not** colorful
- ... but in **Colorful Graph Motif**: \( c = k \)
- \( \Rightarrow \) \( c \) useless for **Colorful Graph Motif**
Rest of the talk

- We are left with $k$ and $\ell$
- First COLORFUL GRAPH MOTIF (or CGM)
- Then GRAPH MOTIF
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
**Colorful Graph Motif is FPT in** $k = |M|$

**Theorem** (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)

*Colorful Graph Motif* is solvable in $O^*(64^k)$ time.
**COLORFUL GRAPH MOTIF is FPT in** $k = |M|$

**Theorem** (Fellows, F., Hermelin & Vialette, J. Comput. Syst. Sci. 07)

COLORFUL GRAPH MOTIF is solvable in $O^*(64^k)$ time.

**Remarks**

- Deterministic (Dynamic Programming)
- Exponential space
- Proof of concept!
**Colorful Graph Motif is FPT in** $k$

**Theorem** *(Betzler et al., CPM 08)*  
*Colorful Graph Motif* *is solvable in* $O^*(3^k)$ *time.*

**Remarks**

- Simpler (and faster) version of previous result
- Deterministic (Dynamic Programming)
- Exponential space $O^*(2^k)$
- Adapted from [Scott et al., J. Comp. Biol. 06]
COLORFUL GRAPH MOTIF is FPT in $k$

Key elements of Dynamic programming algorithm

- Boolean table $B(v, S)$ with
  - $v$ a vertex of $G$
  - $S$ a subset of $M$

- $B(v, S) = \text{TRUE}$ if there is in $G$ a colorful subtree $T$
  - $v$ is the root of $T$
  - colors of $T$ “agree” with $S$
COLORFUL GRAPH MOTIF is FPT in $k$

Key elements of Dynamic programming algorithm

For any $S$ s.t. $|S| = 1$

$$B(v, S) = \begin{cases} \text{TRUE} & \text{if } S = \{\chi(v)\} \\ \text{FALSE} & \text{otherwise} \end{cases}$$

$$B(v, S) = \bigvee_{u \in N(v)} \big( B(v, S_1) \land B(u, S_2) \big)$$

$O^*(3^k) \rightarrow$ all 3-partitions of a set of size $k$
COLORFUL GRAPH MOTIF is FPT in $k$

**Theorem** *(Guillemot & Sikora, Algorithmica 13)*

COLORFUL GRAPH MOTIF is solvable in $O^*(2^k)$ time.

**Remarks**

- Randomized
- Polynomial space
- Uses the “Multilinear Detection” technique (2010)
A detour by polynomials

\[ P(X) = \text{a polynomial built on a set } X = \{x_1, x_2 \ldots x_p\} \text{ of variables} \]

- a monomial \( m \) in \( P(X) \) is **multilinear** if each variable in \( m \) occurs at most once
- **degree** of a multilinear monomial = number of its variables
- example:

\[ P(X) = x_1^2 x_3 x_5 + x_1 x_2 x_4 x_6 \]

- \( x_1 x_2 x_4 x_6 \): multilinear monomial of degree 4
- \( x_1^2 x_3 x_5 \): not a multilinear monomial
A detour by arithmetic circuits

- arithmetic circuit $C$ over a set $X$ of variables = DAG s.t.
  - internal nodes are the operations $\times$ or $+$,
  - leaves are variables from $X$
- polynomial $P(X) \rightarrow$ arithmetic circuit $C$ over $X$
A detour by arithmetic circuits

- arithmetic circuit $C$ over a set $X$ of variables = DAG s.t.
  - internal nodes are the operations $\times$ or $+$,
  - leaves are variables from $X$

- polynomial $P(X) \rightarrow$ arithmetic circuit $C$ over $X$

- Example: $P(X) = (x_1 + x_2 + x_3)(x_3 + x_4 + x_5)$
Problem $\text{ISML-}k$: given an arithmetic circuit $C$, determine whether $P(X)$ contains a multilinear monomial of degree $k$.

**Theorem (Koutis & Williams, ICALP 09)**

$\text{ISML-}k$ is solvable in $O^*(2^k)$ time using polynomial space.
**Multilinear Detection problem**

Problem ISML-\(k\): given an arithmetic circuit \(C\), determine whether \(P(X)\) contains a multilinear monomial of degree \(k\)

**Theorem (Koutis & Williams, ICALP 09)**

ISML-\(k\) is solvable in \(O^*(2^k)\) time using polynomial space.

**Remarks**

- Randomized algorithm
- If \(C\) is an arithmetic circuit representing \(P\):
  - Running time: poly. factor depends on \#arcs of \(C\)
  - Space: depends on \#internal nodes of \(C\)
$O^*(2^k)$ algorithm for CGM

Build polynomial as follows:

- variables $\leftrightarrow$ colors in $M$
- monomial $\leftrightarrow$ colors in a $k$-node subtree of $G$

$\Rightarrow$ multilinear monomial of degree $k$ $\leftrightarrow$ colorful $k$-node subtree in $G$
$O^*(2^k)$ algorithm for CGM

Build polynomial as follows:

- variables $\leftrightarrow$ colors in $M$
- monomial $\leftrightarrow$ colors in a $k$-node subtree of $G$

$\Rightarrow$ multilinear monomial of degree $k$ $\leftrightarrow$ colorful $k$-node subtree in $G$

- if circuit size polynomial in $k$ and input size
- then algorithm in $O^*(2^k)$ for CGM
Polynomial $P$ built from $G$

\[ P_{1,u} = x_\chi(u) \]

\[ P_{i,u} = \sum_{i'=1}^{i-1} \sum_{v \in N(u)} P_{i',u} P_{i-i',v} \]

\[ P = \sum_{u \in V(G)} P_{k,u} \]

(Partial) computation of $P_{3,u} \ (k = 3)$
Polynomial $P$ built from $G$

\[ P_{1,u} = x_{\chi}(u) \]

\[ P_{i,u} = \sum_{i'=1}^{i-1} \sum_{v \in N(u)} P_{i',u}P_{i-i',v} \]

\[ P = \sum_{u \in V(G)} P_{k,u} \]

(Partial) computation of $P_{3,u}$ ($k = 3$)

\[ P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \ldots \]
Polynomial $P$ built from $G$

\[ P_{1,u} = x_\chi(u) \]

\[ P_{i,u} = \sum_{i'=1}^{i-1} \sum_{v \in N(u)} P_{i',u} P_{i-i',v} \]

\[ P = \sum_{u \in V(G)} P_{k,u} \]

(Partial) computation of $P_{3,u}$ $(k = 3)$

\[ P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \ldots \]

\[ = x_R \cdot (P_{2,v} + P_{2,w}) + \ldots \]
Polynomial $P$ built from $G$

\[ P_{1,u} = x_{\chi(u)} \]

\[ P_{i,u} = \sum_{i'=1}^{i-1} \sum_{v \in N(u)} P_{i',u} P_{i-i',v} \]

\[ P = \sum_{u \in V(G)} P_{k,u} \]

(Partial) computation of $P_{3,u}$ ($k = 3$)

\[ P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \ldots \]

\[ = x_{R} \cdot (P_{2,v} + P_{2,w}) + \ldots \]

\[ = x_{R} \cdot (x_{Y} \cdot (P_{1,u} + P_{1,w} + P_{1,t}) + P_{2,w}) + \ldots \]
Polynomial $P$ built from $G$

$$P_1,u = x_{\chi}(u)$$

$$P_{i,u} = \sum_{i'=1}^{i-1} \sum_{v \in N(u)} P_{i',u}P_{i-i',v}$$

$$P = \sum_{u \in V(G)} P_{k,u}$$

(Partial) computation of $P_{3,u}$ ($k = 3$)

$$P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \cdots$$

$$= x_R \cdot (P_{2,v} + P_{2,w}) + \cdots$$

$$= x_R \cdot (x_Y \cdot (P_{1,u} + P_{1,w} + P_{1,t}) + P_{2,w}) + \cdots$$

$$= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,w}) + \cdots$$
Polynomial $P$ built from $G$

\[ P_{1,u} = x_{\chi(u)} \]

\[ P_{i,u} = \sum_{i'=1}^{i-1} \sum_{v \in N(u)} P_{i',u} P_{i-i',v} \]

\[ P = \sum_{u \in V(G)} P_{k,u} \]

(Partial) computation of $P_{3,u}$ ($k = 3$)

\[ P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \ldots \]
\[ = x_R \cdot (P_{2,v} + P_{2,w}) + \ldots \]
\[ = x_R \cdot (x_Y \cdot (P_{1,u} + P_{1,w} + P_{1,t}) + P_{2,w}) + \ldots \]
\[ = x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,w}) + \ldots \]
\[ = x_R \cdot (x_Y \cdot x_R + x_Y \cdot x_R + x_Y \cdot x_B + P_{2,w}) + \ldots \]
Polynomial $P$ built from $G$

$$P_{1,u} = x_{\chi}(u)$$

$$P_{i,u} = \sum_{i' = 1}^{i-1} \sum_{v \in N(u)} P_{i',u} P_{i-i',v}$$

$$P = \sum_{u \in V(G)} P_{k,u}$$

(Partial) computation of $P_{3,u}$ ($k = 3$)

$$P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \cdots$$

$$= x_R \cdot (P_{2,v} + P_{2,w}) + \cdots$$

$$= x_R \cdot (x_Y \cdot (P_{1,u} + P_{1,w} + P_{1,t}) + P_{2,w}) + \cdots$$

$$= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,w}) + \cdots$$

$$= x_R \cdot (x_Y \cdot x_R + x_Y \cdot x_R + x_Y \cdot x_B + P_{2,w}) + \cdots$$

$$= x_R x_Y x_R + x_R x_Y x_R + x_R x_Y x_B + \cdots$$
CGM w.r.t. $k$: a tight lower bound

Can we do better than $O^*(2^k)$?
CGM w.r.t. $k$: a tight lower bound

Can we do better than $O^*(2^k)$?

**Theorem** (Björklund et al., Algorithmica 15)

*Under SeCoCo*, **COLORFUL GRAPH MOTIF** cannot be solved in $O^*((2 - \epsilon)^k)$ time, $\epsilon > 0$.

*SeCoCo = SET COVER Conjecture [Cygan et al., CCC 12]:

if $P \neq NP$, for any $\epsilon > 0$, **SET COVER** cannot be solved in $O^*((2 - \epsilon)^p)$ where $p = |U|$ is the size of the universe.
CGM w.r.t. $k$: a tight lower bound

Reduction

- **SET COVER:**
  - $U = \{x_1, x_2 \ldots x_n\}$
  - $S = \{S_1, S_2 \ldots S_m\}$
  - integer $t$
CGM w.r.t. $k$: a tight lower bound

Reduction

- **SET COVER:**
  - $U = \{x_1, x_2 \ldots x_n\}$
  - $S = \{S_1, S_2 \ldots S_m\}$
  - integer $t$

- **CGM:**
  - Graph $G$
    - $V(G) = \{r\} \cup U \cup \{s_i^j : i \in [m], j \in [t]\}$
    - $r$ connected to every $s_i^j$, $x_p$ connected to all $s_i^j$ s.t. $x_p \in S_i$
    - colors: $x_i \rightarrow c_i$, $r \rightarrow c_{n+1}$, $s_i^j = c_{n+1+j}$ ($i \in [m], j \in [t]$)
  - Motif $M = \{c_1, c_2 \ldots c_{n+t+1}\}$ (thus $k = n + t + 1$)

$O^*((2 - \epsilon)^k)$ for CGM $\Rightarrow$ $O^*((2 - \epsilon)^{n+t})$ for SET COVER

[CYGAN ET AL., CCC 12]:

$O^*((2 - \epsilon)^{n+t})$ for SET COVER $\Rightarrow$ $O^*((2 - \epsilon')^n)$ for SET COVER
## Summary: COLORFUL GRAPH MOTIF w.r.t. $k$

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Technique</th>
<th>Algorithm</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O^*(64^k)$</td>
<td>Dyn. Prog.</td>
<td>Det.</td>
<td>Exp.</td>
</tr>
<tr>
<td>no $O^*((2 - \epsilon)^k)$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
CGM is FPT in $\ell$

Reminder: $\ell = n - k$ ($=$#nodes not kept in solution)

**Theorem (Betzler et al., IEEE/ACM TCBB 11)**

CGM *is solvable in* $O^*(2^\ell)$ *time.*

Bounded Search Tree
Branching Rule: if there exists two vertices $u, v$ s.t. $\chi(u) = \chi(v)$, remove either $u$ or $v$ from the graph.
CGM is FPT in $\ell$

**Branching Rule:** if there exists two vertices $u, v$ s.t. $\chi(u) = \chi(v)$, remove either $u$ or $v$ from the graph.
CGM is FPT in $\ell$

**Branching Rule:** if there exists two vertices $u, v$ s.t. $\chi(u) = \chi(v)$, remove either $u$ or $v$ from the graph.
**Branching Rule:** if there exists two vertices $u, v$ s.t. $\chi(u) = \chi(v)$, remove either $u$ or $v$ from the graph
CGM is FPT in $\ell$

**Algorithm Analysis**

- at least 1 vertex removed at each step
- $\rightarrow$ height of tree at most $\ell$
- 2 choices per step
- $\rightarrow 2^\ell$ possibilities
- each leaf: colorful graph
- if one such graph is of order $k$ and connected, return Yes, otherwise No
CGM is FPT in $\ell$

**Algorithm Analysis**

- at least 1 vertex removed at each step
- $\rightarrow$ height of tree at most $\ell$
- 2 choices per step
- $\rightarrow 2^\ell$ possibilities
- each leaf: colorful graph
- if one such graph is of order $k$ and connected, return **YES**, otherwise **NO**

Can we do better?
FPT lower bound for CGM and $\ell$

**Theorem (F. & Komusiewicz, CPM’16)**

Under SETH*, CGM **cannot be solved in** $O^*((2 - \epsilon)^{\ell})$ **time**, $\epsilon > 0$.

* SETH = Strong Exponential Time Hypothesis [Impagliazzo et al., JCSS 01]:

if $P \neq NP$, for any $\epsilon > 0$, CNF-SAT cannot be solved in $O^*((2 - \epsilon)^p)$, with $p=$number of variables of CNF formula
FPT lower bound for CGM and $\ell$

Reduction from CNF-SAT with $\ell = p$

$$F = (x \lor \overline{y} \lor z) \land (y \lor \overline{z})$$

![Graph Motif problem](image-url)
FPT lower bound for CGM and $\ell$

Reduction from CNF-SAT with $\ell = p$

$$F = (x \lor y \lor z) \land (y \lor z)$$

\begin{figure}
\centering
\includegraphics[width=\textwidth]{graph.png}
\end{figure}

$X \lor \bar{X} \lor Y \lor \bar{Y} \lor Z \lor \bar{Z}$

$(x \lor \bar{y} \lor z) \land (y \lor \bar{z})$
FPT lower bound for CGM and $\ell$

Reduction from CNF-SAT with $\ell = p$

$$F = (x \lor \overline{y} \lor z) \land (y \lor \overline{z})$$
FPT lower bound for CGM and $\ell$

Reduction from CNF-SAT with $\ell = p$

$$F = (x \lor \neg y \lor z) \land (y \lor \neg z)$$
CGM and \( \ell \) for trees

**Theorem (F. & Komusiewicz, CPM’16)**

CGM in trees is solvable in \( O^*(\sqrt{2^\ell}) \) time.
A kernel for CGM in trees

Kernelization

- Use reduction rules
- Instance \((T, M) \rightarrow (T', M')\) with same answer \text{YES}/\text{NO}
- Reduced instance \((T', M')\) called \textit{kernel}
- If size of kernel = \(O(f(\ell))\) then FPT in \(\ell\)
A kernel for CGM in trees

Kernelization

- Use reduction rules
- Instance \((T, M) \rightarrow (T', M')\) with same answer YES/NO
- Reduced instance \((T', M')\) called kernel
- If size of kernel = \(O(f(\ell))\) then FPT in \(\ell\)

**Theorem** \((F. \& \text{ Komusiewicz}, \text{ CPM'16})\)
CGM in trees admits a kernel of size \(2\ell + 1\).
A kernel for CGM in trees

$T =$ the input tree

**Definition**
A vertex is **unique** if no other vertex has the same color in $T$

**Observation**: at most $2 \ell$ vertices are not unique in $T$. 
A kernel for CGM in trees

\( T = \) the input tree

**Definition**

A vertex is **unique** if no other vertex has the same color in \( T \)

**Observation**: at most \( 2\ell \) vertices are not unique in \( T \).

- \( C^+ = \) set of colors occurring more than once in \( C \); \(|C^+| = c^+\)
- \( n^+ = \sum_{c \in C^+} \mu(T, c) \); \( n^- = \) # non-unique vertices
A kernel for CGM in trees

T = the input tree

Definition
A vertex is unique if no other vertex has the same color in T

Observation: at most \(2\ell\) vertices are not unique in T.

- \(C^+\) = set of colors occurring more than once in C; \(|C^+| = c^+\)
- \(n^+ = \sum_{c \in C^+} \mu(T, c)\); \(n^- = \#\) non-unique vertices
  - \(n = n^+ + n^-\)
  - \(|M| = c^+ + n^-\)
  - \(\ell = n - |M| \Rightarrow \ell = n^+ - c^+\)
A kernel for CGM in trees

\( T = \) the input tree

**Definition**
A vertex is **unique** if no other vertex has the same color in \( T \)

**Observation**: at most \( 2\ell \) vertices are not unique in \( T \).

- \( C^+ = \) set of colors occurring more than once in \( C ; |C^+| = c^+ \)
- \( n^+ = \sum_{c \in C^+} \mu(T, c) ; n^- = \# \) non-unique vertices
  - \( n = n^+ + n^- \)
  - \( |M| = c^+ + n^- \)
  - \( \ell = n - |M| \implies \ell = n^+ - c^+ \)
- \( n^+ \geq 2c^+ \implies \ell \geq \frac{n^+}{2} \)
A kernel for CGM in trees

- root $T$ at arbitrary unique vertex $r$
- if all vertices non-unique $\rightarrow \ell \geq \frac{n}{2}$ and kernel already exists
A kernel for CGM in trees

- root $T$ at arbitrary unique vertex $r$
- if all vertices non-unique $\rightarrow \ell \geq \frac{n}{2}$ and kernel already exists

**Definition**

- **pendant** subtree of root $v$: contains all descendants of $v$.  
- **pendant non-unique subtrees**: maximal pendant subtrees in which no vertex is unique
A kernel for CGM in trees

- Left: input instance w/ pendant non-unique subtrees
- Middle: after Phase I, all vertices on paths between unique vertices are contracted into \( r \).
- Right: after Phase II, all vertices with a color that was removed in Phase I are removed together with their descendants.
CGM and $\ell$ for trees

- Phases I and II: reduction rules
- After application: root $r$ + non-unique vertices only
CGM and $\ell$ for trees

- Phases I and II: reduction rules
- After application: root $r$ + non-unique vertices only
- by Observation, \# non-unique vertices $\leq 2\ell$
- $\Rightarrow$ new tree with $\leq 2\ell + 1$ vertices
<table>
<thead>
<tr>
<th>General graphs</th>
<th>Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O^*(2^\ell)$</td>
<td>$O^*(\sqrt{2}^\ell)$</td>
</tr>
<tr>
<td>no $O^*((2 - \epsilon)^\ell)$</td>
<td></td>
</tr>
<tr>
<td>no poly. kernel</td>
<td>$(2\ell + 1)$-vertex kernel</td>
</tr>
</tbody>
</table>
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
From **COLORFUL GRAPH MOTIF** to **GRAPH MOTIF**

- 2 results can be transferred from CGM to **GRAPH MOTIF**
- Price to pay:
  - Increased time complexity (but still exp. in $k$ only)
  - Randomized algorithm
- Secret ingredient: the **Color-Coding** technique
Color-Coding for GRAPH MOTIF

For a color $c$ in $M$, $\text{occ}_M(c) = \#\text{occurrences of } c\text{ in } M$

**Color Coding: General Idea**

- for each color $c \in C$ s.t. $\text{occ}_M(c) \geq 2$
  - create $\text{occ}_M(c)$ new colors
  - replace $c$ in $M$ by these colors → new motif is colorful
  - randomly recolor vertices of $G$ with color $c$ with one of new colors
- colorful motif → use your favorite CGM algorithm!
Color-Coding for Graph Motif
Color-Coding for **GRAPH MOTIF**

\[ M \Rightarrow \]

\[ G \]

G. Fertin The Graph Motif problem
Color-Coding for **GRAPH MOTIF**

$M \Rightarrow$ 

$G \Rightarrow$
Color-Coding for GRAPH MOTIF

\[ M \implies \]

\[ G \implies \]
Running-time increase

- random coloring: a “good” solution may not be colorful
  - may lead to false negatives
- repeat process until probability of success is $1 - \epsilon$ ($\epsilon > 0$)
- probability of a good coloring of $G$: $\frac{k!}{k^k} \geq e^{-k}$
- needs $|\ln(\epsilon)|e^k$ iterations (i.e., random colorings of $G$)
In a nutshell:

- Fellows et al. 2007: \( O^*(64^k) \rightarrow O^*(87^k) \)
- Betzler et al. 2008: \( O^*(3^k) \rightarrow O^*(4.32^k) \)
Adapting MLD to GRAPH MOTIF

$O^*(2^k)$ algorithm by Guillemot & Sikora 2013

- works only for CGM
- if $M \neq M^*$, solution is not a multilinear monomial
- previous construction needs to be adapted
- introduction of variables for each vertex of $G$
Adapting MLD to GRAPH MOTIF

- One variable $x_u$ per vertex $u$ of $G$
- Each color $c$ that appears $m$ times in $M \rightarrow$ variables $y_{c,1}, y_{c,2}, \ldots, y_{c,m}$
- Circuit is modified: $P_{u,1} = x_u \cdot (y_{c,1} + y_{c,2} + \ldots + y_{c,m})$
  - Variables $x_u \rightarrow$ a node of $G$ is used only once
  - Variables $y_j \rightarrow$ right #colors required by $M$
- Solution: multilinear monomial of degree $k' = 2k$ ($k$ nodes + $k$ colors)
- Complexity $O^*(2^{k'}) \rightarrow O^*(4^k)$
Adapting MLD to GRAPH MOTIF – Example

\[ x_u(y_{R,1} + y_{R,2}) \cdot x_v y_{Y,1} \cdot x_w(y_{R,1} + y_{R,2}) \cdot x_t y_{B,1} + \ldots \]
Adapting MLD to \textbf{GRAPH MOTIF} – Example

\[ x_u(y_{R,1} + y_{R,2}) \cdot x_v y_{Y,1} \cdot x_w(y_{R,1} + y_{R,2}) \cdot x_t y_{B,1} + \cdots \]

\[ = x_u y_{R,1} \cdot x_v y_{Y,1} \cdot x_w y_{R,1} \cdot x_t y_{B,1} + \cdots \]
Adapting MLD to \textbf{G}raph \textbf{M}otif – Example

\begin{align*}
x_u(y_{R,1} + y_{R,2}) \cdot x_v y_{Y,1} \cdot x_w(y_{R,1} + y_{R,2}) \cdot x_t y_{B,1} + \ldots \\
= x_u y_{R,1} \cdot x_v y_{Y,1} \cdot x_w y_{R,1} \cdot x_t y_{B,1} + \\
x_u y_{R,1} \cdot x_v y_{Y,1} \cdot x_w y_{R,2} \cdot x_t y_{B,1} + \ldots
\end{align*}
Adapting MLD to GRAPH MOTIF – Example

\[ x_u(y_{R,1} + y_{R,2}) \cdot x_v y_{Y,1} \cdot x_w(y_{R,1} + y_{R,2}) \cdot x_t y_{B,1} + \ldots = x_u y_{R,1} \cdot x_v y_{Y,1} \cdot x_w y_{R,1} \cdot x_t y_{B,1} + \]

\[ x_u y_{R,1} \cdot x_v y_{Y,1} \cdot x_w y_{R,2} \cdot x_t y_{B,1} + \ldots \]

- solution: a multilinear monomial of degree

\[ 2k = 8 \]


**Graph Motif is FPT in $k$**

Previous results superseded by following theorem

**Theorem (Björklund, Kaski & Kowalik, Algorithmica 15)**

Graph Motif is solvable in $O^*(2^k)$ time using polynomial space.

**Remarks**

- Randomized
- *Constrained* Multilinear Detection
- Result independently published in [Pinter, Zehavi - 2016]
**Summary: GRAPH MOTIF w.r.t. $k$**

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Technique</th>
<th>Algorithm</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O^*(2.54k)$</td>
<td>Constrained Multilinear Det.</td>
<td>Random</td>
<td>Exp.</td>
</tr>
<tr>
<td>no $O^*((2 - \epsilon)^k)$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: best **deterministic** algorithm in $O^*(5.22k)$ [PINTER ET AL., DAM 16]
Theorem (BETZLER ET AL., IEEE/ACM TCBB 11)

**Graph Motif** is $W[1]$-complete when parameterized by $\ell$. 
**Theorem** *(Betzler et al., IEEE/ACM TCBB 11)*

**Graph Motif** is \(W[1]\)-complete when parameterized by \(\ell\).

**Remarks**

- reduction from *Independent Set*
- \(M\) has only 2 colors
**Example**

\[ u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \]

\[ n = 5, \, m = 5, \, p = 3 \]
**Graph Motif** is $W[1]$-complete w.r.t. $\ell$

**Example**

$n = 5$, $m = 5$, $p = 3$
GRAPH MOTIF is W[1]-complete w.r.t. $\ell$

Example

$n = 5, m = 5, p = 3$
**Example**

\[ n = 5, m = 5, p = 3 \]
Graph Motif is $W[1]$-complete w.r.t. $\ell$

Example

$n = 5, m = 5, p = 3$
**Graph Motif** is $\mathcal{W}[1]$-complete w.r.t. $\ell$

Example

$n = 5$, $m = 5$, $p = 3$

$M = \{ n - p; \quad m + 1 \}$
**Graph Motif** is \( W[1] \)-complete w.r.t. \( \ell \)

**Example**

\[
\begin{align*}
n &= 5, \\
m &= 5, \\
p &= 3
\end{align*}
\]

\[
M = \{v^{*}, \text{red nodes} \setminus \{v^{*}\}, \text{blue nodes} \}
\]
**Graph Motif** is $W[1]$-complete w.r.t. $\ell$

Example

$n = 5, m = 5, p = 3$

$M = \{n-p; m+1\}$
**Example**

$n = 5$, $m = 5$, $p = 3$

$M = \{\text{red: } n-p; \text{ blue: } m+1\}$
**Example**

$G. Fertin$ The Graph Motif problem

$\text{GRAPH MOTIF is W[1]-complete w.r.t. } \ell$

$n = 5, m = 5, p = 3$

$M = \{ \text{red circles: } n-p; \text{ blue circles: } m+1 \}$
**Theorem (F. & Komusiewicz, CPM 16)**

Graph Motif is solvable in $O^*(4^\ell)$ time when $G$ is a tree.

→ Dynamic Programming
Summary: **GRAPH MOTIF w.r.t.** $\ell$

<table>
<thead>
<tr>
<th>General graphs</th>
<th>Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W[1]$-complete</td>
<td>$O^*(4^\ell)$</td>
</tr>
<tr>
<td></td>
<td>no poly. kernel</td>
</tr>
</tbody>
</table>
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
GRAPH MOTIF and variants: practical issues

- **Motus** [Lacroix et al., Bioinformatics 06]
- **Torque** [Bruckner, Hüffner, Karp, Shamir & Sharan, Bruckner et al., J. Comp. Biol. 10]
- **GraMoFoNe** [Blin, Sikora & Vialette, BICoB 10]
- **RANGI** [Rudi et al., IEEE ACM/TCBB 13]
- **SIMBio** [Rubert et al., BIBE 15]
- **CeFunMo** [Kouhsar et al., Computers in Biology and Medicine 16]
A focus on GraMoFoNe

- cytoscape plugin (open-source java platform, popular in bioinfo)
- supports queries up to 20–25 proteins
- colorful and multiset motifs
- can report all solutions
- deals with approx. solutions (insertions, deletions)
- also deals list-coloring
- technique: Pseudo-Boolean programming
Querying biological networks

Example

- **Query**: Mouse DNA synthesome complex (13 proteins)
- **Target**: Yeast network (∼ 5 300 proteins, ∼ 40 000 interactions)
- **Output**: match consists of 12 proteins with 2 insertions and 3 deletions
Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
  Colorful Graph Motif and parameter $k$
  Colorful Graph Motif and parameter $\ell$

FPT issues for Graph Motif
  Graph Motif and parameter $k$
  Graph Motif and parameter $\ell$

Graph Motif IRL

Conclusion
About GRAPH MOTIF

Quick Summary

► Biologically motivated problem (also applies in other contexts)
► Very large literature (∼140 citations in 10 years)
► Survey ? Work in progress! (with J. Fradin, G. Jean and F. Sikora)
► Multiple improvements over the time (see parameter $k$)
► Recent, sometimes involved techniques
  ▶ SeCoCo (2012)
  ▶ MLD (2010) and constrained versions
  ▶ mixed techniques
► Many variants
► Several software
Open Questions?

- Yes and no!
- Yes: many questions, many variants
- No(t so much) if (COLORFUL) GRAPH MOTIF general case and parameter $k$...
- ...unless you require deterministic algorithms! $\rightarrow$ beat current-best solutions
- Yes:
  - further study parameter $\ell$
  - specific case of trees + inquire about treewidth
A larger view 1/2

From Biology to Computer Science

- Biologically motivated problems become more “interesting”
  - discrete data structures
  - more and more “complicated” graphs (e.g. metagenomics)
  - more and more complicated structures (e.g. sequences with intergene sizes)
  - → more and more intricate (thus interesting) problems
A larger view 1/2

From Biology to Computer Science

- Biologically motivated problems become more “interesting”
  - discrete data structures
  - more and more “complicated” graphs (e.g. metagenomics)
  - more and more complicated structures (e.g. sequences with intergene sizes)
  - → more and more intricate (thus interesting) problems

- FPT well-adapted
  - together with data reduction rules (complexity often collapses on real data)
  - allows to “advertise” new FPT techniques
  - sometimes initiate new techniques
From Computer Science to Bioinfo

- FPT + data reduction rules should be advertised and used
- see the different GRAPh MOTIF software
- how can we convince potential users?
- e.g. why relatively fast exact rather than very fast heuristic?
From Computer Science to Bioinfo

▶ FPT + data reduction rules should be advertised and used
▶ see the different GRAPH MOTIF software
▶ how can we convince potential users?
▶ e.g. why relatively fast exact rather than very fast heuristic?

Thank you for your attention