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 ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

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 Σ
  - typical use: $m << n$

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 Σ
  - 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 Σ
  - typical use: $m << n$
- Applications:
  - search for a word in a text editor [ctrl-f] ($|\Sigma| \sim 60 - 70$)
  - bioinformatics: DNA ($|\Sigma| = 4$), proteins ($|\Sigma| = 20$)
- 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. `http://www-igm.univ-mlv.fr/~lecroq/string/string.pdf`

Recess 1

Analysis of Algorithms

- Analysis of an algorithm, say A
- Running time of $A \simeq$ number of "elementary operations" executed by A

Recess 1

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

Recess 1

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

Recess 1 (Cont'd)

$O()$ notation

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

Recess 1 (Cont'd)

$O()$ notation

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

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

- $\rightarrow g()$ is an upper bound for $f()$

Recess 1 (Cont'd)

$O()$ notation

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

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

- $\rightarrow g()$ is an upper bound for $f()$
- Roughly: take $f(n)$, keep dominant term, remove multiplicative constant
- Example:
  - $f(n) = 7n^2 + 3n \log n + 12\sqrt{n} - 7$
  - $f(n) = O(n^2)$

Recess 1 (Cont'd)

$O()$ notation

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

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

- $\rightarrow g()$ is an upper bound for $f()$
- Roughly: take $f(n)$, keep dominant term, remove multiplicative constant
- Example:
  - $f(n) = 7n^2 + 3n \log n + 12\sqrt{n} - 7$
  - $f(n) = O(n^2)$
- $O()$ used for worst-case analysis – robustness of algorithm

Recess 1 (Cont'd)

Motif search - naive algorithm (sliding window)

```
void naive(M[0..m-1], T[0..n-1])
for i=0 to n-m do
     j <-- 0;
     while M[j]=T[i+j] && j<= m-1 do
        j <-- j+1;
     endwhile
     if j=m then
        printf(``Motif found at position %d\n'',i);
     endif
endfor
```

Recess 1 (Cont'd)

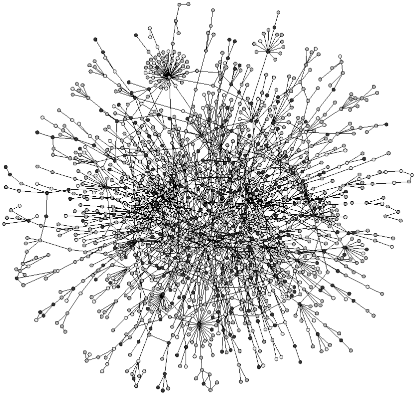
Motif search - naive algorithm (sliding window)

```
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

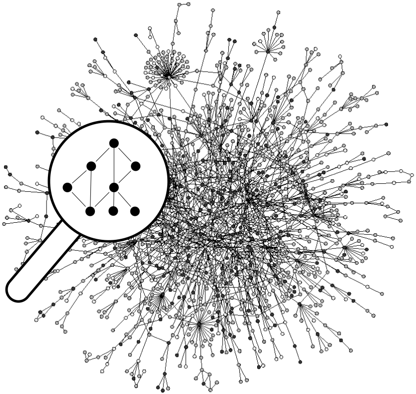
- species: yeast
- vertices $\leftrightarrow$ proteins ($\sim$ 3 500)
- edges $\leftrightarrow$ interactions ($\sim$ 11 000)



Source: http://compbio.pbworks.com/

Motif Search in Graphs

- species: yeast
- vertices $\leftrightarrow$ proteins ($\sim 3\,500$)
- edges $\leftrightarrow$ interactions ($\sim 11\,000$)



Source: http://compbio.pbworks.com/

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

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.

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 $\chi : V \to C$, a motif* M over C

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

Question: Is there an occurrence of M in G ?

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 ?

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

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

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 ?

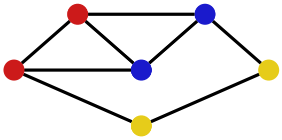
Occurrence = subset $V' \subseteq V$ s.t.

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

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

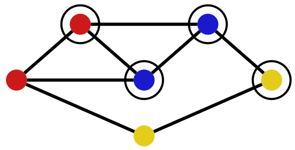
Graph Motif

Example



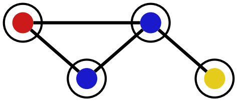
GRAPH MOTIF

Example



Graph Motif

Example



GRAPH MOTIF

Applications

- metabolic networks analysis [LACROIX, FERNANDES & SAGOT, IEEE/ACM TCBB 06]
- PPI networks analysis [BRUCKNER ET AL., J. COMP. BIOL. 10]

GRAPH MOTIF

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]

GRAPH MOTIF

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.

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.

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^* = underlying set of M
- M is colorful if $M^* = M$
- COLORFUL GRAPH MOTIF (or CGM): restriction of GRAPH MOTIF to colorful motifs

Some notations

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

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

- $\mu(G, c)$ = number of vertices having color c 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 ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

Graph Motif IRL

Conclusion

GRAPH MOTIF: first results

Theorem (LACROIX ET AL., IEEE/ACM TCBB 06)

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

Recess 2

Did you say **NP**-complete ?

Algorithmic complexity of Problems

- *Pb*=a problem, n=size of the input

Recess 2

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) $\Rightarrow Pb \in \mathbf{P}$

Recess 2

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**
- *Pb* is intractable if no $O(n^c)$ algo. exists for solving it $\Rightarrow Pb \notin$ **P**

Recess 2

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) $\Rightarrow Pb \in \mathbf{P}$
- *Pb* is intractable if no $O(n^c)$ algo. exists for solving it $\Rightarrow Pb \notin \mathbf{P}$
- very often: we do not know

Recess 2 (Cont'd)

Very often:

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

Recess 2 (Cont'd)

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

Recess 2 (Cont'd)

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

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
- For any instance I' of Pb'

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
- For any instance I' of Pb'
- build in polynomial time a specific instance I of Pb

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
- For any instance I' of Pb'
- build in polynomial time a specific instance *I* of *Pb*
- YES for $I \Leftrightarrow$ YES for I'

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)

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)
- $\Rightarrow$ **NP**-complete = class of hardest such problems
- problems in **NP**-complete thought not to be polynomial-time solvable
- but remains unknown (cf "**P** =**NP** ?")

GRAPH MOTIF: first results

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

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
- Proof does not hold for COLORFUL GRAPH MOTIF
- Is COLORFUL GRAPH MOTIF any “simpler” ?

GRAPH MOTIF: bad news

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*
- $\mu(G) = 3$

COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- Boolean formula Φ
  - 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 Φ
  - 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 ($\vee$)
  - Φ contains clauses connected by logical AND only ($\wedge$)

Colorful Graph Motif is NP-complete

A detour by SAT

- Boolean formula Φ
  - 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 ($\vee$)
  - Φ contains clauses connected by logical AND only ($\wedge$)
- Example:

$$\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$$

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 $\overline{x_i}$

- $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$

COLORFUL GRAPH MOTIF is NP-complete

A detour by SAT

- variable: x_i
- literal: x_i or $\overline{x_i}$
- $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$
- Goal: satisfy Φ
  - assign TRUE/FALSE to each x_i
  - s.t. Φ 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 Φ 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. Φ is satisfied ?

COLORFUL GRAPH MOTIF is NP-complete

Definition (SAT)

Input: a boolean formula Φ 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. Φ 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 Φ 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.:

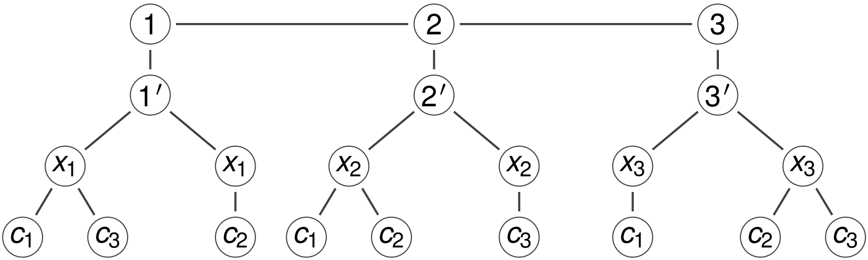
- each clause of Φ contains at most 3 literals, and
- each variable appears in at most 3 clauses, and
- each literal appears in at most 2 clauses

$$\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$$

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



- from $\Phi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$
- 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 ≥ 2 clauses)
- $\mu(G) = 3$ (clause contains ≤ 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 ≥ 2 clauses)
- $\mu(G) = 3$ (clause contains ≤ 3 literals)
- M is colorful

Equivalence YES/NO answer

- ($\Rightarrow$) Pick color x_i corresponding to assignment
- ($\Leftarrow$) Pick vertices x_i and c_j corresponding to occurrence of motif

GRAPH MOTIF: bad news

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*
- $\mu(G) = 3$

GRAPH MOTIF: bad news

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 $\mu(G) \rightarrow$ **NP**-complete
- What if M uses few colors ?

GRAPH MOTIF: more bad news

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*
- $|M^*| = 2$

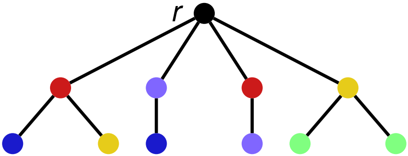
- Reduction from EXACT COVER BY 3-SETS

GRAPH MOTIF: any polynomial case... please ?

Theorem (FELLOWS, F., HERMELIN & VIALETTE, J. COMPUT. SYST. SCI. 07)
GRAPH MOTIF *is in* **P** *whenever G is a tree and $\mu(G) = 2$.*

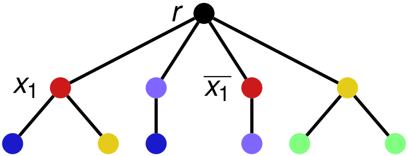
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT



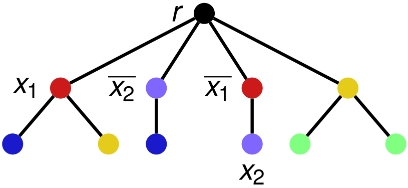
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT



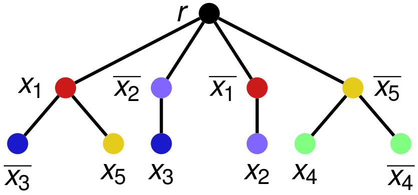
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT



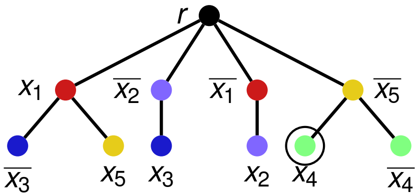
GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT



Graph Motif: a polynomial case

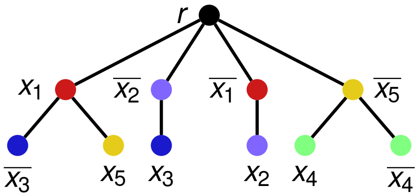
Equivalence with 2-SAT



$$(x_4 \Rightarrow \overline{x_5})$$

GRAPH MOTIF: a polynomial case

Equivalence with 2-SAT



$$(\overline{x_3} \Rightarrow x_1) \wedge (x_5 \Rightarrow x_1) \wedge (x_3 \Rightarrow \overline{x_2}) \wedge (x_2 \Rightarrow \overline{x_1}) \wedge \ldots$$

2-SAT formula as $(A \Rightarrow B) \Leftrightarrow (\overline{B} \vee A)$

Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
- Colorful Graph Motif and parameter k
- Colorful Graph Motif and parameter ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

Graph Motif IRL

Conclusion

Graph Motif: Coping with hardness

Remarks

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

GRAPH MOTIF: Coping with hardness

Remarks

- motifs tend to be small in practice (compared to the target graph)
- $\rightarrow$ Question 1: algorithm whose running time is
  - polynomial in $n = |V(G)|$ and
  - **exponential** in $k = |M|$?

Graph Motif: Coping with hardness

Remarks

- motifs tend to be small in practice (compared to the target graph)
- $\rightarrow$ Question 1: algorithm whose running time is
  - polynomial in $n = |V(G)|$ and
  - **exponential** in $k = |M|$?
- $\rightarrow$ Question 2: algorithm whose running time is
  - polynomial in $n = |V(G)|$ and
  - **exponential** in $c = |M^*|$?

Graph Motif: Coping with hardness

Remarks

- motifs tend to be small in practice (compared to the target graph)
- $\rightarrow$ Question 1: algorithm whose running time is
  - polynomial in $n = |V(G)|$ and
  - **exponential** in $k = |M|$?
- $\rightarrow$ 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

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

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

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

Parameterized complexity

Definition (Parameterized hierarchy)

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

Parameterized complexity

Definition (Parameterized hierarchy)

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

In a nutshell

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

Parameterized complexity

Definition (Parameterized hierarchy)

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

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

- R.G. Downey, M. R. Fellows – Parameterized Complexity – Springer-Verlag, 1999.
- H. Fernau – Parameterized Algorithmics: A Graph-Theoretic Approach. 2005. Free download at
  `http://www.informatik.uni-trier.de/~fernau/papers/habil.pdf`
- J. Flum and M. Grohe. Parameterized Complexity Theory – Springer-Verlag, 2006.
- R. Niedermeier – Invitation to Fixed-Parameter Algorithms – Oxford University Press, 2006.
- R.G. Downey, M. R. Fellows – Fundamentals of Parameterized Complexity – Springer-Verlag, 2013.
- M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh – Parameterized Algorithms – Springer-Verlag, 2015.

FPT: an ever-growing topic

Monographs

- R.G. Downey, M. R. Fellows – Parameterized Complexity – Springer-Verlag, 1999.
- H. Fernau – Parameterized Algorithmics: A Graph-Theoretic Approach. 2005. Free download at `http://www.informatik.uni-trier.de/~fernau/papers/habil.pdf`
- J. Flum and M. Grohe. Parameterized Complexity Theory – Springer-Verlag, 2006.
- R. Niedermeier – Invitation to Fixed-Parameter Algorithms – Oxford University Press, 2006.
- R.G. Downey, M. R. Fellows – Fundamentals of Parameterized Complexity – Springer-Verlag, 2013.
- M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh – Parameterized Algorithms – Springer-Verlag, 2015.

- Dedicated website `http://fpt.wikidot.com/`

FPT: main techniques

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

FPT: main techniques

- Dynamic Programming (table size and computation exponential in paramater only)
- Bounded Search Tree: test all possible cases, show there are $O(f(k))$ such cases

FPT: main techniques

- Dynamic Programming (table size and computation exponential in paramater 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

FPT: main techniques

- Dynamic Programming (table size and computation exponential in paramater 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.

Graph Motif and FPT: which parameters ?

The choice is yours

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

Graph Motif and FPT: which parameters ?

The choice is yours

- Size of the motif $k = |M|$ = 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

Graph Motif and FPT: which parameters ?

The choice is yours

- Size of the motif $k = |M|$ = 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 ℓ (HARTUNG ET AL., JGAA 15)

Graph Motif: parameter c

Reminder: $c = |M^*|$=#colors in M

GRAPH MOTIF: parameter c

Reminder: $c = |M^*|$=#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.*

GRAPH MOTIF: parameter c

Reminder: $c = |M^*|$=#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

GRAPH MOTIF: parameter c

Reminder: $c = |M^*|$=#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

Graph Motif and CGM: FPT issues

Rest of the talk

- We are left with k and ℓ
- 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 ℓ

FPT issues for Graph Motif

Graph Motif and parameter k

Graph Motif and parameter ℓ

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
- $B(v, S)$=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. $|S| = 1$

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

$$B(v,S) = \bigvee_{\substack{u \in N(v) \\ S_1 \uplus S_2 = S \\ \chi(v) \in S_1, \chi(u) \in S_2}} B(v,S_1) \wedge B(u,S_2)$$

$O^*(3^k) \rightarrow$ 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

$P(X)$ = a polynomial built on a set $X = \{x_1, x_2 \dots x_p\}$ 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:

$$P(X) = x_1^2 x_3 x_5 + x_1 x_2 x_4 x_6$$

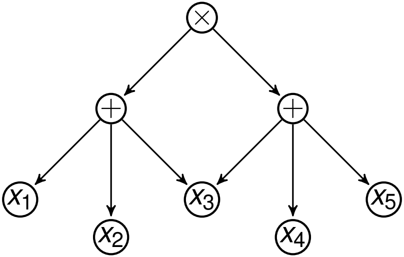
  - $x_1 x_2 x_4 x_6$: multilinear monomial of degree 4
  - $x_1^2 x_3 x_5$: not a multilinear monomial

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: $P(X) = (x_1 + x_2 + x_3)(x_3 + x_4 + x_5)$



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.*

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

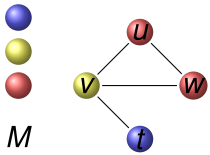
- if circuit size polynomial in k and input size
- then algorithm in $O^*(2^k)$ for CGM

Polynomial P built from G

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

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

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



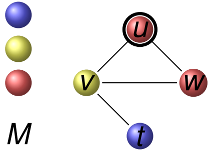
(Partial) computation of $P_{3,u}$ ($k = 3$)

Polynomial P built from G

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

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

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



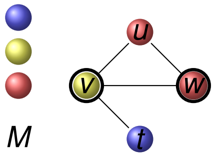
(Partial) computation of $P_{3,u}$ ($k = 3$)
$P_{3,u} = P_{1,u} \cdot (P_{2,v} + P_{2,w}) + \ldots$

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 $P_{3,u}$ ($k = 3$)

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

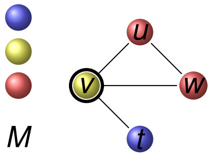
$= x_R \cdot (P_{2,v} + P_{2,w}) + \ldots$

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 $P_{3,u}$ ($k = 3$)

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

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

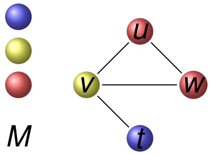
$= x_R \cdot (x_Y \cdot (P_{1,u} + P_{1,w} + P_{1,t}) + P_{2,w}) + \ldots$

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 $P_{3,u}$ ($k = 3$)

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

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

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

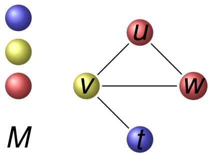
$= x_R \cdot (x_Y \cdot (x_R + x_R + x_B) + P_{2,w}) + \ldots$

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 $P_{3,u}$ ($k = 3$)

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

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

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

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

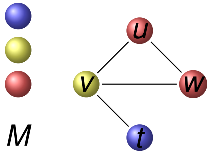
$= x_R \cdot (x_Y \cdot x_R + x_Y \cdot x_R + x_Y \cdot x_B + P_{2,w}) + \dots$

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 $P_{3,u}$ ($k = 3$)

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

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

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

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

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

$= x_R x_Y x_R + x_R x_Y x_R + x_R x_Y x_B + \ldots$

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

CGM w.r.t. k: a tight lower bound

Reduction

- SET COVER:
  - $U = \{x_1, x_2 \dots x_n\}$
  - $\mathcal{S} = \{S_1, S_2 \dots S_m\}$
  - 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_j^i, x_p connected to all s_j^i s.t. $x_p \in S_i$
    - colors: $x_i \longrightarrow c_i$, $r \longrightarrow c_{n+1}$, $s_i^j = c_{n+1+j}$ $(i \in [m], j \in [t])$
  - Motif $M = \{c_1, c_2 \dots c_{n+t+1}\}$ (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

Complexity	Technique	Algorithm	Space
$O^*(64^k)$	Dyn. Prog.	Det.	Exp.
$O^*(3^k)$	Dyn. Prog.	Det.	Exp.
$O^*(2^k)$ no $O^*((2-\epsilon)^k)$	Multilinear Det.	Random	Poly.

Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
- Colorful Graph Motif and parameter k
- Colorful Graph Motif and parameter ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

Graph Motif IRL

Conclusion

CGM is FPT in ℓ

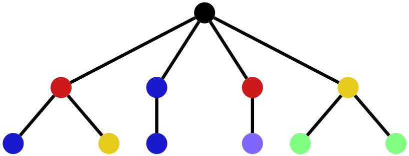
Reminder: $\ell = n - k$ (=#nodes not kept in solution)

Theorem (Betzler et al., IEEE/ACM TCBB 11)
CGM *is solvable in* $O^*(2^\ell)$ *time.*

Bounded Search Tree

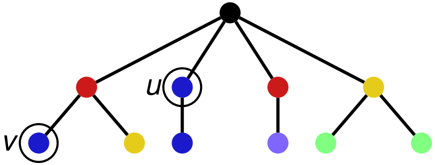
CGM is FPT in ℓ

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



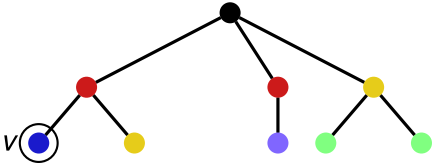
CGM is FPT in ℓ

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



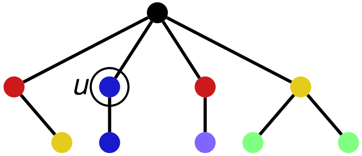
CGM is FPT in ℓ

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 ℓ

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 ℓ

Algorithm Analysis

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

CGM is FPT in ℓ

Algorithm Analysis

- at least 1 vertex removed at each step
- $\rightarrow$ height of tree at most ℓ
- 2 choices per step
- $\rightarrow$ 2^ℓ 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 ℓ

Theorem (F. & KOMUSIEWICZ, CPM'16)
Under SETH, CGM cannot be solved in $O^*((2-\epsilon)^\ell)$ time, $\epsilon > 0$.*

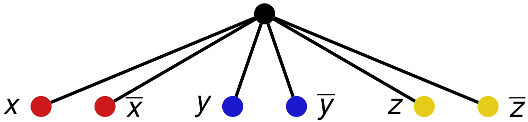
* SETH = Strong Exponential Time Hypothesis [IMPAGLIAZZO ET AL., JCSS 01]:

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 ℓ

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

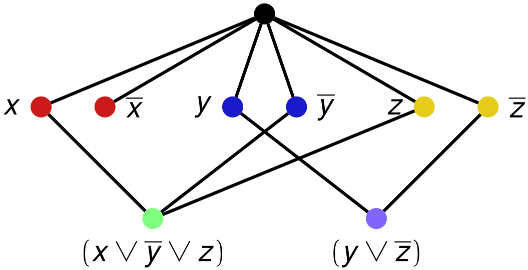
$$F = (x \vee \overline{y} \vee z) \wedge (y \vee \overline{z})$$



FPT lower bound for CGM and ℓ

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

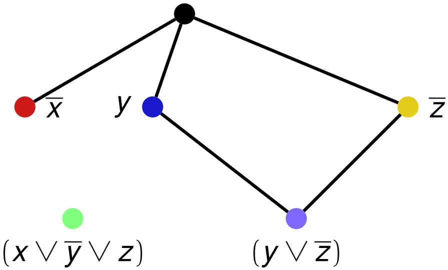
$$F = (x \vee \overline{y} \vee z) \wedge (y \vee \overline{z})$$



FPT lower bound for CGM and ℓ

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

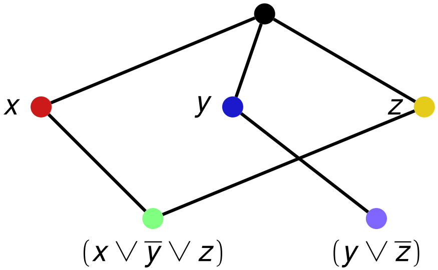
$$F = (x \vee \overline{y} \vee z) \wedge (y \vee \overline{z})$$



FPT lower bound for CGM and ℓ

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

$$F = (x \vee \overline{y} \vee z) \wedge (y \vee \overline{z})$$



CGM and ℓ 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 YES/NO
- Reduced instance (T', M') called kernel
- If size of kernel = $O(f(\ell))$ then FPT in ℓ

A kernel for CGM in trees

Kernelization

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

Theorem (F. & KOMUSIEWICZ, 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 2ℓ 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 2ℓ vertices are not unique in T.

- C^+= set of colors occuring 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 2ℓ vertices are not unique in T.

- C^+= set of colors occuring 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 2ℓ vertices are not unique in T.

- C^+= set of colors occuring more than once in C ; $|C^+| = c^+$
- $n^+ = \sum_{c \in C^+} \mu(T, c)$; n^-= # non-unique vertices
  - $n = n^+ + n^-$
  - $|M| = c^+ + n^-$
  - $\ell = n - |M| \Rightarrow \ell = n^+ - c^+$
- $n^+ \geq 2c^+ \Rightarrow \ell \geq \frac{n^+}{2}$

A kernel for CGM in trees

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

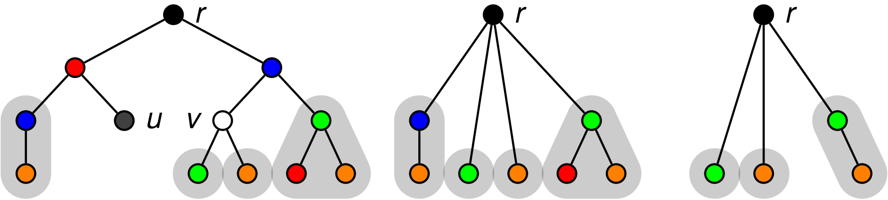
A kernel for CGM in trees

- root T at arbitray 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 ℓ for trees

- Phases I and II: reduction rules
- After application: root r + non-unique vertices only

CGM and ℓ 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

Summary: Colorful Graph Motif w.r.t. ℓ

General graphs	Trees
$O^*(2^\ell)$ no $O^*((2-\epsilon)^\ell)$	$O^*(\sqrt{2}^\ell)$
no poly. kernel	$(2\ell+1)$-vertex kernel

Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
- Colorful Graph Motif and parameter k
- Colorful Graph Motif and parameter ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

Graph Motif IRL

Conclusion

From Colorful Graph Motif to Graph Motif

- 2 results can be transfered 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

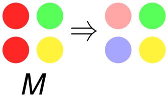
For a color c in M, $occ_M(c)$=#occurrences of c in M

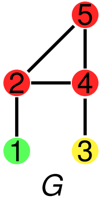
Color Coding: General Idea

- for each color $c \in C$ s.t. $occ_M(c) \geq 2$
  - create $occ_M(c)$ new colors
  - replace c in M by these colors $\rightarrow$ new motif is colorful
  - randomly recolor vertices of G with color c with one of new colors
- colorful motif $\rightarrow$ use your favorite CGM algorithm!

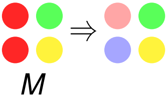
Color-Coding for GRAPH MOTIF

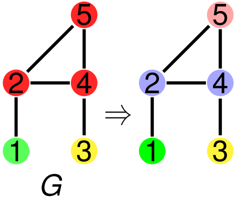
Color-Coding for GRAPH MOTIF



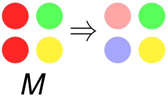


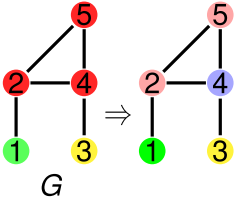
Color-Coding for GRAPH MOTIF





Color-Coding for GRAPH MOTIF





Color-Coding for GRAPH MOTIF

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)

From COLORFUL GRAPH MOTIF to GRAPH MOTIF

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

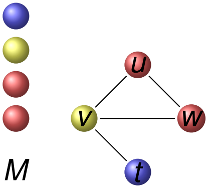
$O^*(2^k)$ **algorithm by Guillemot & Sikora 2013**

- 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

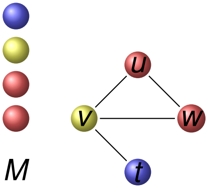
- One variable x_u per vertex u of G
- Each color c that appears m times in $M \to$ variables $y_{c,1}, y_{c,2}, \ldots, y_{c,m}$
- Circuit is modified: $P_{u,1} = x_u \cdot (y_{c,1} + y_{c,2} + \ldots + y_{c,m})$
  - Variables $x_u \to$ a node of G is used only once
  - Variables $y_j \to$ right #colors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

Adapting MLD to GRAPH MOTIF – Example



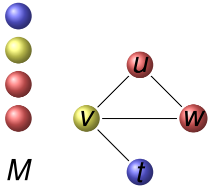
$$x_u(y_{R,1}+y_{R,2})\cdot x_v y_{Y,1}\cdot x_w(y_{R,1}+y_{R,2})\cdot x_t y_{B,1}+\dots$$

Adapting MLD to GRAPH MOTIF – Example



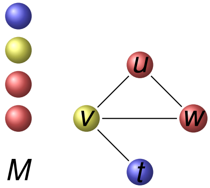
$$x_u(y_{R,1}+y_{R,2})\cdot x_v y_{Y,1}\cdot x_w(y_{R,1}+y_{R,2})\cdot x_t y_{B,1}+\dots$$
$$= x_u y_{R,1}.x_v y_{Y,1}.x_w y_{R,1}.x_t y_{B,1}+$$

Adapting MLD to GRAPH MOTIF – Example



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

Adapting MLD to Graph Motif – Example



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

- 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

Complexity	Technique	Algorithm	Space
$O^*(87^k)$	Dyn. Prog. + Color-Coding	Random	Exp.
$O^*(4.32^k)$	Dyn. Prog. + Color-Coding	Random	Exp.
$O^*(4^k)$	Multilinear Det.	Random	Poly.
$O^*(2.54^k)$	Constrained Multilinear Det.	Random	Exp.
$O^*(2^k)$ Björklund et al. no $O^*((2-\epsilon)^k)$	Constrained Multilinear Det.	Random	Poly.

Note: best deterministic algorithm in $O^*(5.22^k)$ [Pinter et al., DAM 16]

GRAPH MOTIF w.r.t. ℓ: bad news

Theorem (BETZLER ET AL., IEEE/ACM TCBB 11)

GRAPH MOTIF *is* **W[1]***-complete when parameterized by* ℓ.

GRAPH MOTIF w.r.t. ℓ: bad news

Theorem (BETZLER ET AL., IEEE/ACM TCBB 11)

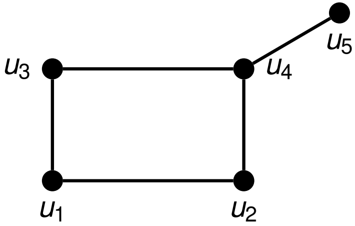
GRAPH MOTIF *is* **W[1]***-complete when parameterized by* ℓ.

Remarks

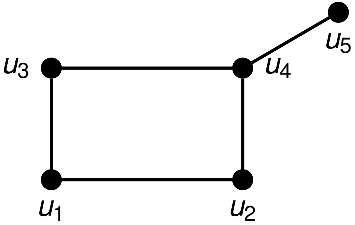
- reduction from INDEPENDENT SET
- M has only 2 colors

GRAPH MOTIF is W[1]-complete w.r.t. ℓ

Example

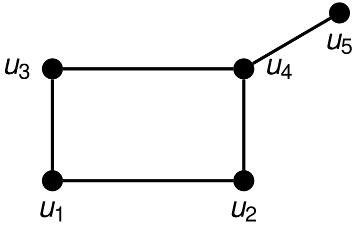


$n = 5, m = 5, p = 3$

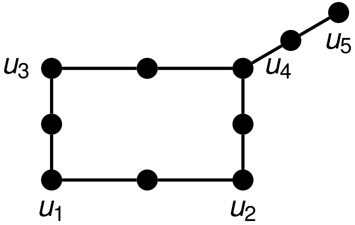


GRAPH MOTIF is W[1]-complete w.r.t. ℓ

Example

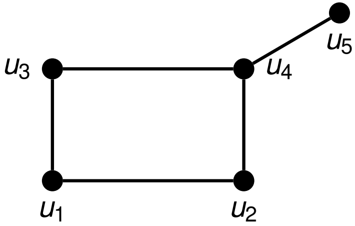


$n = 5, m = 5, p = 3$

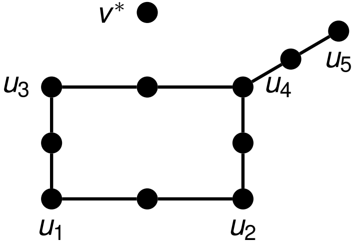


Graph Motif is W[1]-complete w.r.t. ℓ

Example

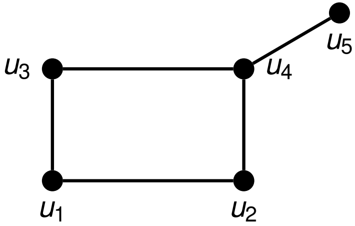


$n = 5, m = 5, p = 3$

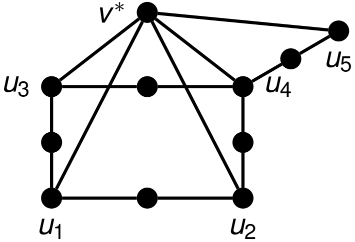


GRAPH MOTIF is W[1]-complete w.r.t. ℓ

Example

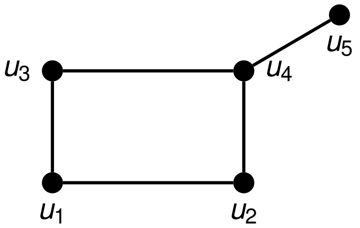


$n = 5, m = 5, p = 3$

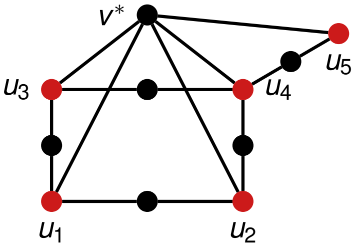


GRAPH MOTIF is W[1]-complete w.r.t. ℓ

Example

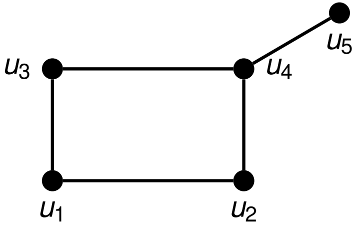


$n = 5, m = 5, p = 3$

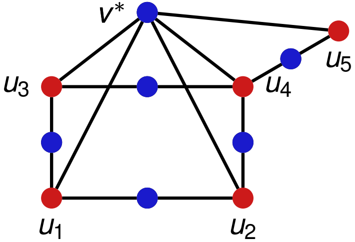


GRAPH MOTIF is W[1]-complete w.r.t. ℓ

Example



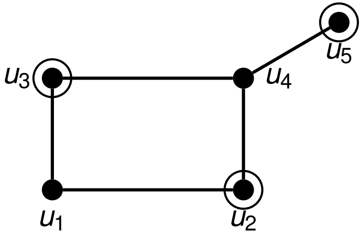
$n = 5, m = 5, p = 3$



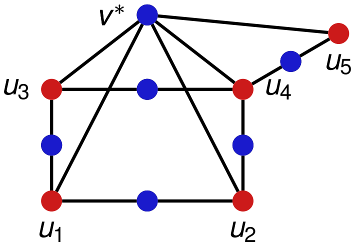
$M = \{ \bullet^{n-p}; \bullet^{m+1} \}$

Graph Motif is W[1]-complete w.r.t. ℓ

Example



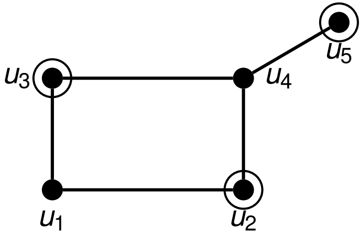
$n = 5, m = 5, p = 3$



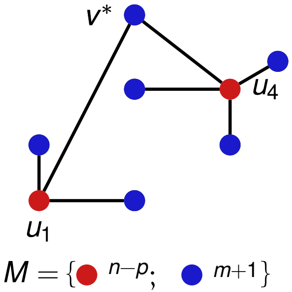
$M = \{$ ● $^{n-p}$; ● $^{m+1}\}$

Graph Motif is W[1]-complete w.r.t. ℓ

Example

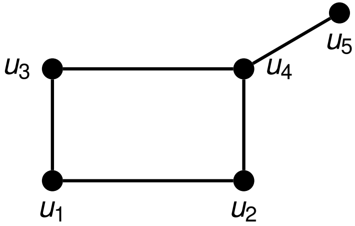


$n = 5, m = 5, p = 3$

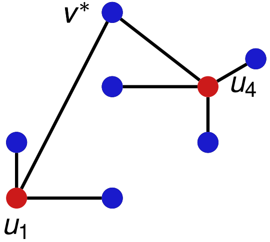


Graph Motif is W[1]-complete w.r.t. ℓ

Example



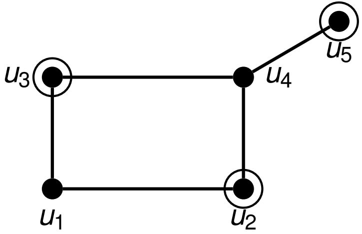
$n = 5, m = 5, p = 3$



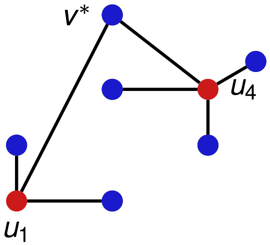
$M = \{$ ● $^{n-p}$; ● $^{m+1}\}$

GRAPH MOTIF is W[1]-complete w.r.t. ℓ

Example



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



$M = \{ \bullet^{n-p}; \bullet^{m+1} \}$

GRAPH MOTIF w.r.t. ℓ in trees ?

Theorem (F. & KOMUSIEWICZ, CPM 16)

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

$\rightarrow$ Dynamic Programming

Summary: Graph Motif w.r.t. ℓ

General graphs	Trees
W[1]-complete	$O^*(4^\ell)$
	no poly. kernel

Outline

Introduction

First Results

FPT issues

FPT issues for Colorful Graph Motif
- Colorful Graph Motif and parameter k
- Colorful Graph Motif and parameter ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

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]

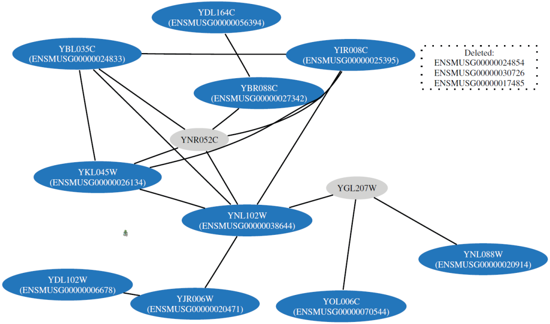
A focus on GraMoFoNe

- `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 ℓ

FPT issues for Graph Motif
- Graph Motif and parameter k
- Graph Motif and parameter ℓ

Graph Motif IRL

Conclusion

About GRAPH MOTIF

Quick Summary

- Biologically motivated problem (also applies in other contexts)
- Very large literature ($\sim$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! $\longrightarrow$ beat current-best solutions
- Yes:
  - further study parameter ℓ
  - 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)
  - $\longrightarrow$ 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)
  - $\longrightarrow$ 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

A larger view 2/2

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?

A larger view 2/2

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