Algorismes (no massa) superpolinomials en grafs

Ignasi Sau

CNRS, LIRMM, Montpeller, França

2a JORNADA SCM DE JOVES INVESTIGADORS EN MATEMÀTIQUES
14 d'octubre del 2011

Full de ruta

1. Motivació: complexitat parametritzada
2. 1r paràmetre: amplada de branques/arbre
   - Definicions
   - Programació dinàmica
   - Generalitzacions i conclusions
3. 2n paràmetre: mida de la solució
   - Motivació
   - Trencar carbasses

La propera secció és...

Complexitat parametritzada (Downey i Fellows, 1990's)

- Per mesurar la complexitat d'un problema en grafs, típicament diem que els problemes *difícils* són els NP-durs:

  és *improbable* que puguin ser resolts per un algorisme polinomial en la mida del graf.

  Exemple: trobar la mida mínima d'un VERTEX COVER.

- Idea: expressar el temps d'execució no només en funció de la mida del graf (típicament, el seu número de vèrtexs), sinó també d'un paràmetre addicional associat al graf.

  Exemple: la mida d'un VERTEX COVER que volem trobar.

Complexitat parametritzada (Downey i Fellows, 1990's)

- Per mesurar la complexitat d'un problema en grafs, típicament diem que els problemes difícils són els NP-durs:

  és *improbable* que puguin ser resolts per un algorisme polinomial en la mida del graf.

  Exemple: trobar la mida mínima d'un VERTEX COVER.

- Idea: expressar el temps d'execució no només en funció de la mida del graf (típicament, el seu número de vèrtexs), sinó també d'un paràmetre addicional associat al graf.

  Exemple: la mida d'un VERTEX COVER que volem trobar.

Complexitat parametritzada (Downey i Fellows, 1990's)

- Per mesurar la complexitat d'un problema en grafs, típicament diem que els problemes difícils són els NP-durs:

  és *improbable* que puguin ser resolts per un algorisme polinomial en la mida del graf.

  Exemple: trobar la mida mínima d'un VERTEX COVER.

- Idea: expressar el temps d'execució no només en funció de la mida del graf (típicament, el seu número de vèrtexs), sinó també d'un paràmetre addicional associat al graf.

  Exemple: la mida d'un VERTEX COVER que volem trobar.

Complexitat parametritzada (Downey i Fellows, 1990's)

- Per mesurar la complexitat d'un problema en grafs, típicament diem que els problemes difícils són els NP-durs:

  és *improbable* que puguin ser resolts per un algorisme polinomial en la mida del graf.

  Exemple: trobar la mida mínima d'un VERTEX COVER.

- Idea: expressar el temps d'execució no només en funció de la mida del graf (típicament, el seu número de vèrtexs), sinó també d'un paràmetre addicional associat al graf.

  Exemple: la mida d'un VERTEX COVER que volem trobar.

Complexitat parametritzada (II)

- Donat un problema (NP-dur) amb un *input* de mida n i un paràmetre k, un algorisme "fixed-parameter tractable" (FPT) s'executa en temps

$$f(k) \cdot n^{\mathcal{O}(1)}, \quad \text{per a alguna funció } f.$$

Exemples:

★ Decidir si un graf de mida n té un VERTEX COVER de mida com a molt k es pot fer en temps $1.2738^k \cdot n$.

★ En canvi, el millor algorisme per decidir si un graf de mida n té una CLIQUE de mida k és $\mathcal{O}(n^k)$.

Complexitat parametritzada (II)

- Donat un problema (NP-dur) amb un *input* de mida n i un paràmetre k, un algorisme "fixed-parameter tractable" (FPT) s'executa en temps

$$f(k) \cdot n^{\mathcal{O}(1)}, \quad \text{per a alguna funció } f.$$

Exemples:

★ Decidir si un graf de mida n té un VERTEX COVER de mida com a molt k es pot fer en temps $1.2738^k \cdot n$.

★ En canvi, el millor algorisme per decidir si un graf de mida n té una CLIQUE de mida k és $\mathcal{O}(n^k)$.

Complexitat parametritzada (II)

- Donat un problema (NP-dur) amb un *input* de mida n i un paràmetre k, un algorisme "fixed-parameter tractable" (FPT) s'executa en temps

$$f(k) \cdot n^{\mathcal{O}(1)}, \quad \text{per a alguna funció } f.$$

Exemples:

★ Decidir si un graf de mida n té un VERTEX COVER de mida com a molt k es pot fer en temps $1.2738^k \cdot n$.

★ En canvi, el millor algorisme per decidir si un graf de mida n té una CLIQUE de mida k és $\mathcal{O}(n^k)$.

FPT i algorismes "exponencials simples"

- Teorema de Courcelle (1988):

  Els problemes que es poden expressar en MSOL es poden resoldre en temps $f(k) \cdot n^{\mathcal{O}(1)}$ en graphs G tals que $\mathbf{tw}(G) \leq k$.

- **Problema**: $f(k)$ pot ser enorme!!! (per exemple, $f(k) = 2^{3^{4^{5^{6^{k}}}}}$)

- Un algorisme parametritzat exponencial simple és un algorisme FPT tal que

$$f(k) = 2^{\mathcal{O}(k)}.$$

- Considerarem dos paràmetres naturals:
  - ★ Amplada de branques/arbre.
  - ★ Mida de la solució que volem trobar.

FPT i algorismes "exponencials simples"

- Teorema de Courcelle (1988):

  Els problemes que es poden expressar en MSOL es poden resoldre en temps $f(k) \cdot n^{\mathcal{O}(1)}$ en graphs G tals que $\mathbf{tw}(G) \leq k$.

- **Problema**: $f(k)$ pot ser enorme!!! (per exemple, $f(k) = 2^{3^{4^{5^{6^{k}}}}}$)

- Un algorisme parametritzat exponencial simple és un algorisme FPT tal que

$$f(k) = 2^{\mathcal{O}(k)}.$$

- Considerarem dos paràmetres naturals:
  - ★ Amplada de branques/arbre.
  - ★ Mida de la solució que volem trobar.

FPT i algorismes "exponencials simples"

- Teorema de Courcelle (1988):

  Els problemes que es poden expressar en MSOL es poden resoldre en temps $f(k) \cdot n^{\mathcal{O}(1)}$ en graphs G tals que $\mathbf{tw}(G) \leq k$.

- **Problema**: $f(k)$ pot ser enorme!!! (per exemple, $f(k) = 2^{3^{4^{5^{6^{k}}}}}$)

- Un algorisme parametritzat exponencial simple és un algorisme FPT tal que
$$f(k) = 2^{\mathcal{O}(k)}.$$

- Considerarem dos paràmetres naturals:
  - ★ Amplada de branques/arbre.
  - ★ Mida de la solució que volem trobar.

FPT i algorismes “exponencials simples”

- Teorema de Courcelle (1988):

  Els problemes que es poden expressar en MSOL es poden resoldre en temps $f(k) \cdot n^{\mathcal{O}(1)}$ en graphs G tals que $\mathbf{tw}(G) \leq k$.

- **Problema**: $f(k)$ pot ser enorme!!! (per exemple, $f(k) = 2^{3^{4^{5^{6^{k}}}}}$)

- Un algorisme parametritzat exponencial simple és un algorisme FPT tal que
$$f(k) = 2^{\mathcal{O}(k)}.$$

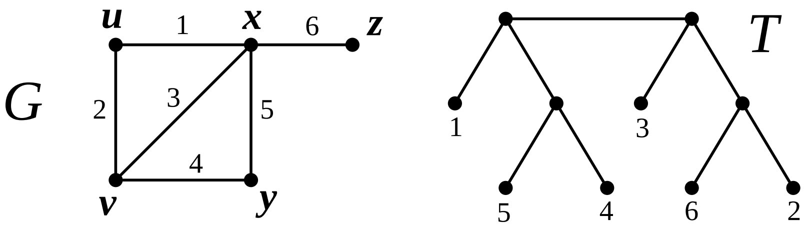
- Considerarem dos paràmetres naturals:
  - ★ Amplada de branques/arbre.
  - ★ Mida de la solució que volem trobar.

La propera secció és...

La propera subsecció és...

1. Motivació: complexitat parametritzada
2. 1r paràmetre: amplada de branques/arbre
   - Definicions
   - Programació dinàmica
   - Generalitzacions i conclusions
3. 2n paràmetre: mida de la solució
   - Motivació
   - Trencar carbasses

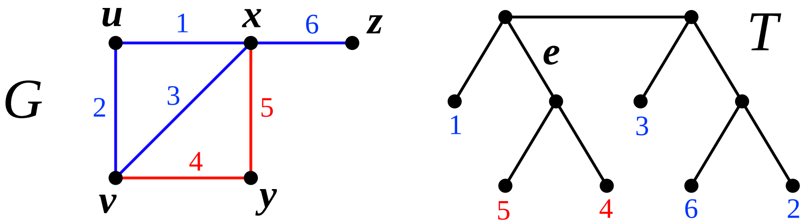
Amplada de branques (branch-width)



Una descomposició en branques d'un graf G és una parella (T, μ):

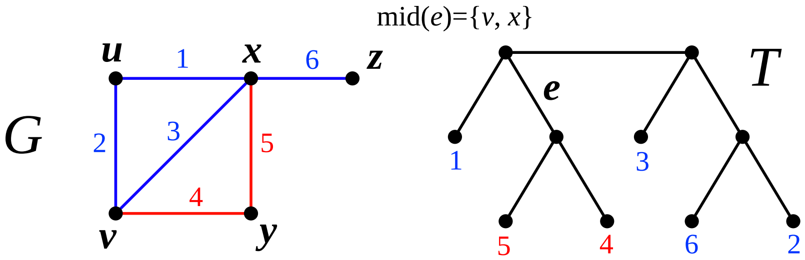
- T és un arbre on tots els vèrtexs interns tenen grau 3.
- μ és una bijecció entre $E(G)$ i les fulles de T.

Amplada de branques (branch-width)



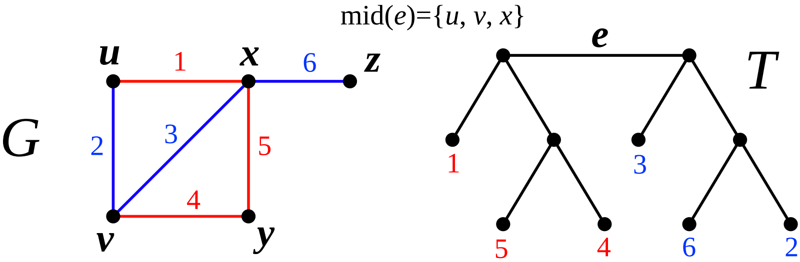
Cada aresta $e \in T$ particiona $E(G)$ en dos conjunts A_e i B_e.

Amplada de branques (branch-width)



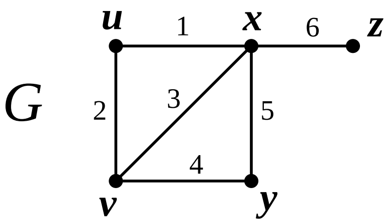
Per a cada $e \in E(T)$, definim **mid**$(e) = V(A_e) \cap V(B_e)$.

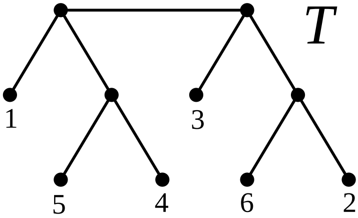
Amplada de branques (branch-width)



L'amplada de (T, μ) és $\max_{e \in E(T)} |\mathbf{mid}(e)|$.

Amplada de branques (branch-width)

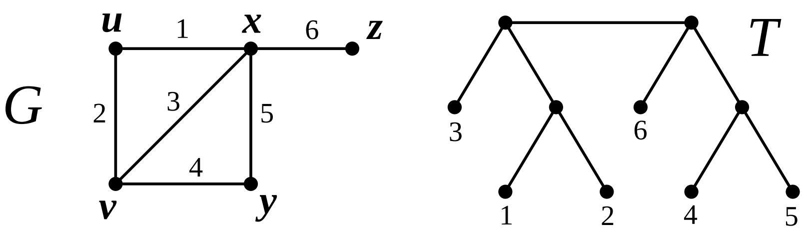




L'amplada de branques d'un graf G, $\mathbf{bw}(G)$, és l'amplada mínima entre totes les descomposicions en branques de G:

$$\mathbf{bw}(G) = \min_{(T,\mu)} \max_{e \in E(T)} |\mathbf{mid}(e)|$$

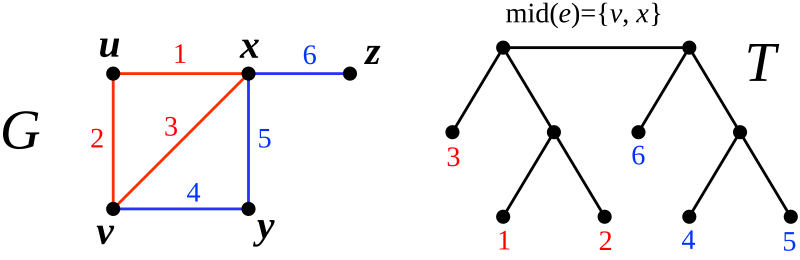
Amplada de branques (branch-width)



L'amplada de branques d'un graf G, **bw**(G), és l'amplada mínima entre totes les descomposicions en branques de G:

$$\mathbf{bw}(G) = \min_{(T,\mu)} \max_{e \in E(T)} |\mathbf{mid}(e)|$$

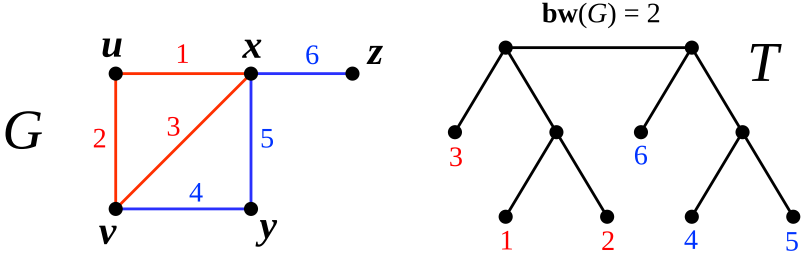
Amplada de branques (branch-width)



L'amplada de branques d'un graf G, $\mathbf{bw}(G)$, és l'amplada mínima entre totes les descomposicions en branques de G:

$$\mathbf{bw}(G) = \min_{(T,\mu)} \max_{e \in E(T)} |\mathbf{mid}(e)|$$

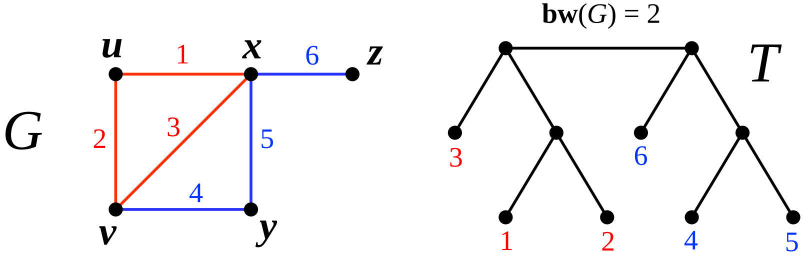
Amplada de branques (branch-width)



L'amplada de branques d'un graf G, **bw**(G), és l'amplada mínima entre totes les descomposicions en branques de G:

$$\mathbf{bw}(G) = \min_{(T,\mu)} \max_{e \in E(T)} |\mathbf{mid}(e)|$$

Amplada de branques (branch-width)



Tenim la següent relació per grafs G tals que $|E(G)| \geq 3$:

$$\mathbf{bw}(G) \;\leq\; \mathbf{tw}(G) + 1 \;\leq\; \frac{3}{2}\mathbf{bw}(G)$$

[Robertson i Seymour. *JCTSB'91*]

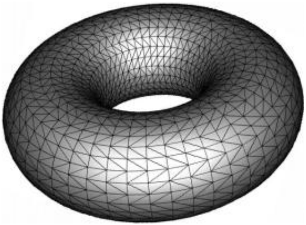
Objectiu d'aquesta part:

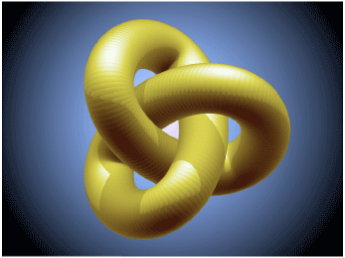
metodologia per obtenir **algorismes parametritzats (pel bw) exponencials simples** per a una classe de problemes NP-durs en **grafs immersos en superfícies**.

[Juanjo Rué, I. S., Dimitrios M. Thilikos. *ICALP'10*]

Superfícies

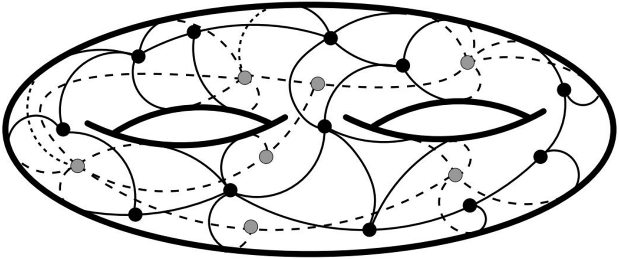
- **SUPERFÍCIE** = ESPAI TOPOLÒGIC, LOCALMENT "PLA"





Grafs en superfícies

Graf immers: graf dibuixat en una superfície, sense creuaments d'arestes



El gènere d'un graf G, $\mathbf{g}(G)$, és el gènere més petit de totes les superfícies en les quals G es pot dibuixar.

La propera subsecció és...

Programació dinàmica (PD)

- Tècnica molt utilitzada per resoldre problemes d'optimització en grafs.
- Es fa "d'abaix a dalt" en una descomposició en branques de l'input graf G.
- Per a cada problema, la PD requereix una definició precisa de les taules, que codifiquen com les potencials solucions (globals) es restringeixen al "middle set" $\mathbf{mid}(e)$.
- La mida de les taules reflecteix la dependència en $|\mathbf{mid}(e)| \leq k$ del temps d'execució de la PD.

Programació dinàmica (PD)

- Tècnica molt utilitzada per resoldre problemes d'optimització en grafs.
- Es fa "d'abaix a dalt" en una descomposició en branques de l'input graf G.
- Per a cada problema, la PD requereix una definició precisa de les taules, que codifiquen com les potencials solucions (globals) es restringeixen al "middle set" $\mathbf{mid}(e)$.
- La mida de les taules reflecteix la dependència en $|\mathbf{mid}(e)| \leq k$ del temps d'execució de la PD.

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   # aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   # possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

   Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos**...

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set $\mathbf{mid}(e)$?

1. Un subconjunt de vèrtexs de $\mathbf{mid}(e)$ (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de $\mathbf{mid}(e)$.
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   # aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de $\mathbf{mid}(e)$ en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   # possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos**...

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   \# aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   \# possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos**...

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   \# aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   \# possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos**...

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   $\#$ aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   $\#$ possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos**...

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   $\#$ aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   $\#$ possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos**...

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: VERTEX COVER, DOMINATING SET.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: LONGEST PATH, CYCLE PACKING, HAMILTONIAN CYCLE.
   $\#$ aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: CONNECTED VERTEX COVER, MAX LEAF SPANNING TREE.
   $\#$ possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

   Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos...**

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: Vertex Cover, Dominating Set.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: Longest Path, Cycle Packing, Hamiltonian Cycle.
   $\#$ aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: Connected Vertex Cover, Max Leaf Spanning Tree.
   $\#$ possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

   Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos...**

Una classificació (parcial) de problemes en grafs

Com podem codificar una solució en un middle set **mid**(e)?

1. Un subconjunt de vèrtexs de **mid**(e) (sense cap condició global).
   Exemples: Vertex Cover, Dominating Set.
   La mida de les taules està fitada per $2^{\mathcal{O}(k)}$.

2. Un *aparellament connex* de vèrtexs de **mid**(e).
   Exemples: Longest Path, Cycle Packing, Hamiltonian Cycle.
   $\#$ aparellaments de k elements és $k^{\Theta(k)} = 2^{\Theta(k \log k)}$...

   OK per grafs planars [Dorn, Penninkx, Bodlaender, Fomin. *ESA'05*];
   OK per grafs en superfícies [Dorn, Fomin, Thilikos. *SWAT'06*].

3. *Agrupament connex* de vèrt. de **mid**(e) en conjunts de mida arbitrària.
   Exemples: Connected Vertex Cover, Max Leaf Spanning Tree.
   $\#$ possibilitats en un conjunt de k elements és $2^{\Theta(k \log k)}$.

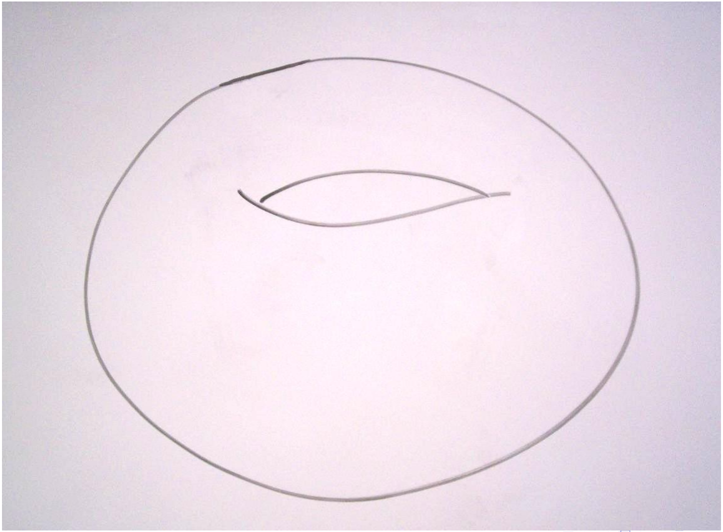
   Cap de les tècniques existents semblava poder-se aplicar a aquesta classe de problemes **codificables per agrupaments connexos...**

Llaços

Sigui G un graf en una superfície Σ. Un llaç és un subespai de Σ homeomorf a $\mathbb{S}^1$ que intersecta G només en vèrtexs.

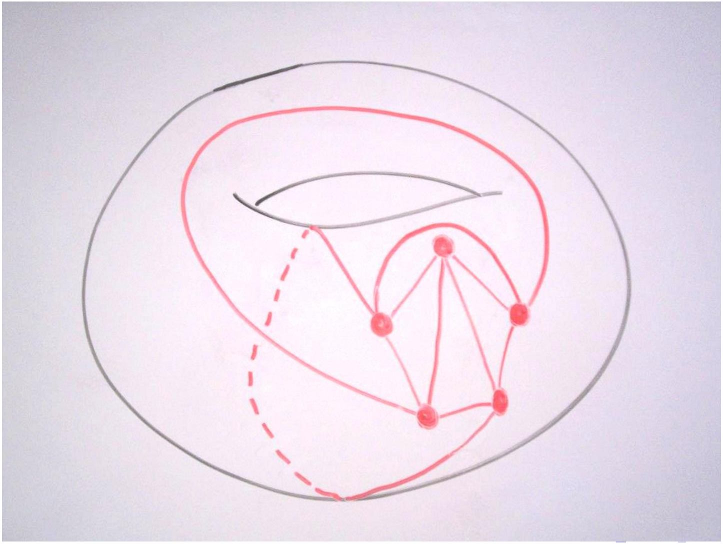
Llaços

Sigui G un graf en una superfície Σ. Un llaç és un subespai de Σ homeomorf a $\mathbb{S}^1$ que intersecta G només en vèrtexs.



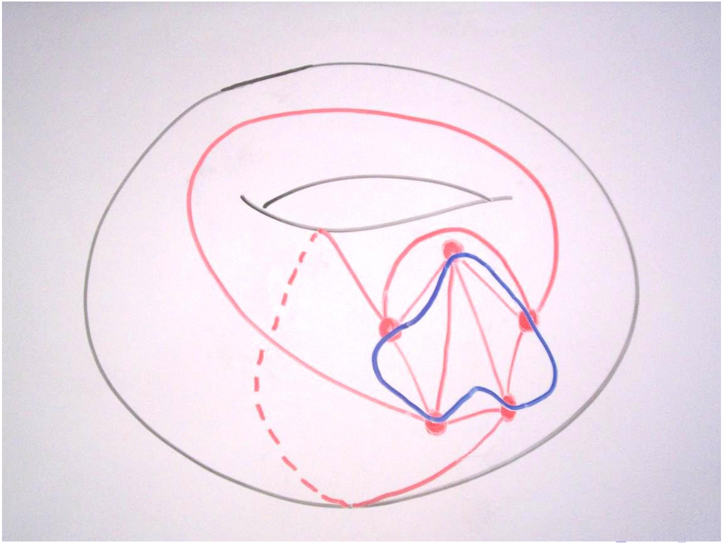
Llaços

Sigui G un graf en una superfície Σ. Un llaç és un subespai de Σ homeomorf a $\mathbb{S}^1$ que intersecta G només en vèrtexs.



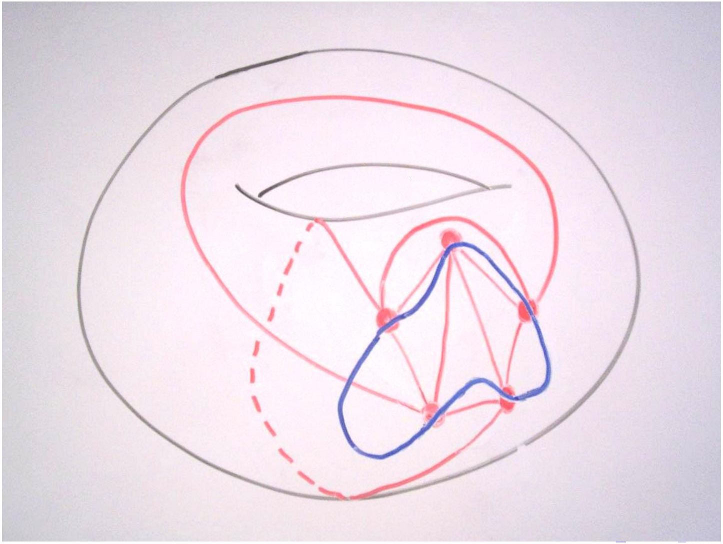
Llaços

Sigui G un graf en una superfície Σ. Un llaç és un subespai de Σ homeomorf a $\mathbb{S}^1$ que intersecta G només en vèrtexs.



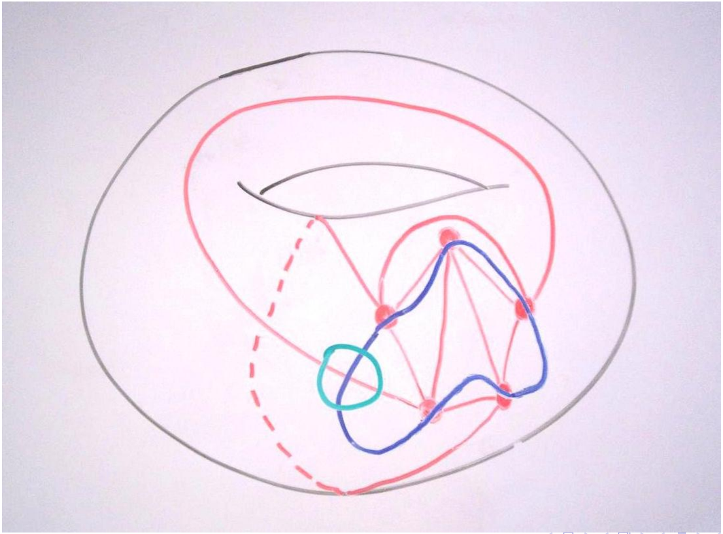
Llaços

Sigui G un graf en una superfície Σ. Un llaç és un subespai de Σ homeomorf a $\mathbb{S}^1$ que intersecta G només en vèrtexs.



Llaços

Sigui G un graf en una superfície Σ. Un llaç és un subespai de Σ homeomorf a $\mathbb{S}^1$ que intersecta G només en vèrtexs.



Descomposicions “sphere cut”

Principal idea per grafs planars [Dorn *et al.* *ESA’05*]:

- Descomposició “sphere cut”: Descomposició en branques on els vèrtexs a cada **mid**(e) estan situats al voltant d’un llaç.
  [Seymour and Thomas. *Combinatorica’94*]

- Recordem que la mida de les taules en la PD depenen de quantes maneres una solució parcial pot intersectar **mid**(e).
- De quantes maneres podem dibuixar polígons disjunts dins d’un cercle tocant el cercle només en els k vèrtexs?

- Exactament el número de *particions sense creuaments* de k elements, que ve donat pels números de Catalan:

$$\mathrm{CN}(k) = \frac{1}{k+1}\binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi}k^{3/2}} \approx 4^k.$$

Descomposicions “sphere cut”

Principal idea per grafs planars [Dorn *et al.* *ESA’05*]:

- Descomposició “sphere cut”: Descomposició en branques on els vèrtexs a cada **mid**(e) estan situats al voltant d’un llaç.
  [Seymour and Thomas. *Combinatorica’94*]
- Recordem que la mida de les taules en la PD depenen de quantes maneres una solució parcial pot intersectar **mid**(e).
- De quantes maneres podem dibuixar polígons disjunts dins d’un cercle tocant el cercle només en els k vèrtexs?
- Exactament el número de *particions sense creuaments* de k elements, que ve donat pels números de Catalan:

$$\mathrm{CN}(k) = \frac{1}{k+1}\binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi}k^{3/2}} \approx 4^k.$$

Descomposicions "sphere cut"

Principal idea per grafs planars [Dorn *et al.* *ESA'05*]:

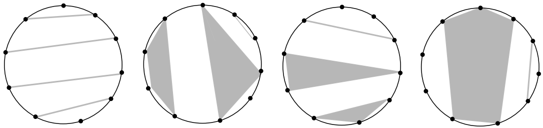
- Descomposició "sphere cut": Descomposició en branques on els vèrtexs a cada **mid**(e) estan situats al voltant d'un llaç. [Seymour and Thomas. *Combinatorica'94*]
- Recordem que la mida de les taules en la PD depenen de quantes maneres una solució parcial pot intersectar **mid**(e).
- De quantes maneres podem dibuixar polígons disjunts dins d'un cercle tocant el cercle només en els k vèrtexs?
- Exactament el número de *particions sense creuaments* de k elements, que ve donat pels números de Catalan:

$$\mathrm{CN}(k) = \frac{1}{k+1}\binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi}k^{3/2}} \approx 4^k.$$

Descomposicions "sphere cut"

Principal idea per grafs planars [Dorn *et al.* *ESA'05*]:

- Descomposició "sphere cut": Descomposició en branques on els vèrtexs a cada **mid**(e) estan situats al voltant d'un llaç. [Seymour and Thomas. *Combinatorica'94*]
- Recordem que la mida de les taules en la PD depenen de quantes maneres una solució parcial pot intersectar **mid**(e).
- De quantes maneres podem dibuixar polígons disjunts dins d'un cercle tocant el cercle només en els k vèrtexs?



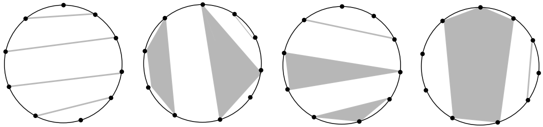
- Exactament el número de *particions sense creuaments* de k elements, que ve donat pels números de Catalan:

$$\mathrm{CN}(k) = \frac{1}{k+1}\binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi}k^{3/2}} \approx 4^k.$$

Descomposicions "sphere cut"

Principal idea per grafs planars [Dorn *et al.* *ESA'05*]:

- Descomposició "sphere cut": Descomposició en branques on els vèrtexs a cada **mid**(e) estan situats al voltant d'un llaç. [Seymour and Thomas. *Combinatorica'94*]
- Recordem que la mida de les taules en la PD depenen de quantes maneres una solució parcial pot intersectar **mid**(e).
- De quantes maneres podem dibuixar polígons disjunts dins d'un cercle tocant el cercle només en els k vèrtexs?



- Exactament el número de *particions sense creuaments* de k elements, que ve donat pels números de Catalan:

$$\mathrm{CN}(k) = \frac{1}{k+1}\binom{2k}{k} \sim \frac{4^k}{\sqrt{\pi}k^{3/2}} \approx 4^k.$$

“Antiga” idea per grafs en superfícies

Principal idea per grafs en superfícies [Dorn *et al.* *SWAT’06*]:

- Planaritzar l’input graf descomposant les solucions potencials en un número de parts que depèn de la superfície.
- Aplicar la tècnica de descomposicions “sphere cut” sobre una versió més complicada del problema, on el número d’aparellaments encara esta fitat per algun número de Catalan.
- Inconvenients d’aquesta tècnica:
  - ★ Depèn fortament de cada problema **particular**.
  - ★ En principi, no es pot aplicar a la classe de problemes **codificables per agrupaments connexos**.

"Antiga" idea per grafs en superfícies

Principal idea per grafs en superfícies [Dorn *et al.* *SWAT'06*]:

- Planaritzar l'input graf descomposant les solucions potencials en un número de parts que depèn de la superfície.
- Aplicar la tècnica de descomposicions "sphere cut" sobre una versió més complicada del problema, on el número d'aparellaments encara esta fitat per algun número de Catalan.
- Inconvenients d'aquesta tècnica:
  - ★ Depèn fortament de cada problema **particular**.
  - ★ En principi, no es pot aplicar a la classe de problemes **codificables per agrupaments connexos**.

Sphere cut $\rightsquigarrow$ surface cut

La nostra estratègia es basa en un nou tipus de descomposició en branques: descomposició "surface cut".

- Les descomposicions "surface cut" per grafs en superfícies **generalitzen** les descomposicions "sphere cut" per grafs planars. [Seymour and Thomas. *Combinatorica'94*]
- És a dir, fem servir directament l'estructura combinatòria de les solucions potencials a la superfície (**sense planarització**).
- Utilitzant les descomposicions "surface cut", presentem d'una manera **unificada** algorismes **exponencials simples** per problemes **codificables per agrupaments connexos**, i amb millor dependència en el **gènere**.

Sphere cut $\rightsquigarrow$ surface cut

La nostra estratègia es basa en un nou tipus de descomposició en branques: descomposició "surface cut".

- Les descomposicions "surface cut" per grafs en superfícies **generalitzen** les descomposicions "sphere cut" per grafs planars. [Seymour and Thomas. *Combinatorica'94*]
- És a dir, fem servir directament l'estructura combinatòria de les solucions potencials a la superfície (**sense planarització**).
- Utilitzant les descomposicions "surface cut", presentem d'una manera **unificada** algorismes **exponencials simples** per problemes **codificables per agrupaments connexos**, i amb millor dependència en el **gènere**.

Sphere cut $\rightsquigarrow$ surface cut

La nostra estratègia es basa en un nou tipus de descomposició en branques: descomposició “surface cut”.

- Les descomposicions “surface cut” per grafs en superfícies **generalitzen** les descomposicions “sphere cut” per grafs planars. [Seymour and Thomas. *Combinatorica'94*]
- És a dir, fem servir directament l'estructura combinatòria de les solucions potencials a la superfície (**sense planarització**).
- Utilitzant les descomposicions “surface cut”, presentem d'una manera **unificada** algorismes **exponencials simples** per problemes **codificables per agrupaments connexos**, i amb millor dependència en el **gènere**.

Descomposicions "surface cut" (versió simplificada)

Sigui G un graf immens en una superfície Σ, amb $\mathbf{g}(\Sigma) = \mathbf{g}$.

Una descomposició "surface cut" de G és una descomposició en branques (T, μ) de G i un conjunt $A \subseteq V(G)$, amb $|A| = \mathcal{O}(\mathbf{g})$, tal que $\forall e \in E(T)$:

- o bé $|\mathbf{mid}(e) \setminus A| \leq 2$,
- o
  - els vèrtexs a $\mathbf{mid}(e) \setminus A$ estan situats en un conjunt $\mathcal{N}$ de $\mathcal{O}(\mathbf{g})$ llaços;
  - aquests llaços intersecten en $\mathcal{O}(\mathbf{g})$ vèrtexs;
  - $\Sigma \setminus \bigcup_{N \in \mathcal{N}} N$ té exactament 2 components connexes.

Descomposicions "surface cut" (versió simplificada)

Sigui G un graf immens en una superfície Σ, amb $\mathbf{g}(\Sigma) = \mathbf{g}$.

Una descomposició "surface cut" de G és una descomposició en branques (T, μ) de G i un conjunt $A \subseteq V(G)$, amb $|A| = \mathcal{O}(\mathbf{g})$, tal que $\forall e \in E(T)$:

- o bé $|\mathbf{mid}(e) \setminus A| \leq 2$,
- o
  - els vèrtexs a $\mathbf{mid}(e) \setminus A$ estan situats en un conjunt $\mathcal{N}$ de $\mathcal{O}(\mathbf{g})$ llaços;
  - aquests llaços intersecten en $\mathcal{O}(\mathbf{g})$ vèrtexs;
  - $\Sigma \setminus \bigcup_{N \in \mathcal{N}} N$ té exactament 2 components connexes.

Descomposicions "surface cut" (versió simplificada)

Sigui G un graf immens en una superfície Σ, amb $\mathbf{g}(\Sigma) = \mathbf{g}$.

Una descomposició "surface cut" de G és una descomposició en branques (T, μ) de G i un conjunt $A \subseteq V(G)$, amb $|A| = \mathcal{O}(\mathbf{g})$, tal que $\forall e \in E(T)$:

- o bé $|\mathbf{mid}(e) \setminus A| \leq 2$,
- o
  - ★ els vèrtexs a $\mathbf{mid}(e) \setminus A$ estan situats en un conjunt $\mathcal{N}$ de $\mathcal{O}(\mathbf{g})$ llaços;
  - ★ aquests llaços intersecten en $\mathcal{O}(\mathbf{g})$ vèrtexs;
  - ★ $\Sigma \setminus \bigcup_{N \in \mathcal{N}} N$ té exactament 2 components connexes.

Descomposicions “surface cut” (versió simplificada)

Sigui G un graf immens en una superfície Σ, amb $\mathbf{g}(\Sigma) = \mathbf{g}$.

Una descomposició “surface cut” de G és una descomposició en branques (T, μ) de G i un conjunt $A \subseteq V(G)$, amb $|A| = \mathcal{O}(\mathbf{g})$, tal que $\forall e \in E(T)$:

- o bé $|\mathbf{mid}(e) \setminus A| \leq 2$,
- o
  - ★ els vèrtexs a $\mathbf{mid}(e) \setminus A$ estan situats en un conjunt $\mathcal{N}$ de $\mathcal{O}(\mathbf{g})$ llaços;
  - ★ aquests llaços intersecten en $\mathcal{O}(\mathbf{g})$ vèrtexs;
  - ★ $\Sigma \setminus \bigcup_{N \in \mathcal{N}} N$ té exactament 2 components connexes.

Descomposicions "surface cut" (versió simplificada)

Sigui G un graf immens en una superfície Σ, amb $\mathbf{g}(\Sigma) = \mathbf{g}$.

Una descomposició "surface cut" de G és una descomposició en branques (T, μ) de G i un conjunt $A \subseteq V(G)$, amb $|A| = \mathcal{O}(\mathbf{g})$, tal que $\forall e \in E(T)$:

- o bé $|\mathbf{mid}(e) \setminus A| \leq 2$,
- o
  - els vèrtexs a $\mathbf{mid}(e) \setminus A$ estan situats en un conjunt $\mathcal{N}$ de $\mathcal{O}(\mathbf{g})$ llaços;
  - aquests llaços intersecten en $\mathcal{O}(\mathbf{g})$ vèrtexs;
  - $\Sigma \setminus \bigcup_{N \in \mathcal{N}} N$ té exactament 2 components connexes.

Resultats principals

1. Una descomposició "surface cut" es pot calcular eficientment:

Theorem (Rué, S., i Thilikos)

Donat G amb n vèrtexs immers en una superfície de gènere $\mathbf{g}$, *amb* $\mathbf{bw}(G) \leq k$, *es pot construir en temps* $2^{3k+\mathcal{O}(\log k)} \cdot n^3$ *una descomposició "surface cut"* (T, μ) *de G d'amplada* $\leq 27k + \mathcal{O}(\mathbf{g})$.

2. Si la PD es fa en una descomposició "surface cut", llavors l'algorisme depèn del **bw** d'una manera exponencial simple:

Theorem (Rué, S., i Thilikos)

Donat un problema P codificable per agrupaments connexos en un graf G immers en una superfície de gènere $\mathbf{g}$, *amb* $\mathbf{bw}(G) \leq k$, *la mida de les taules de la PD per resoldre P en una descomposició "surface cut" de G està fitada per* $2^{\mathcal{O}(\log \mathbf{g} \cdot k + \log k \cdot \mathbf{g})}$.

Resultats principals

1. Una descomposició "surface cut" es pot calcular eficientment:

Theorem (Rué, S., i Thilikos)

Donat G amb n vèrtexs immers en una superfície de gènere $\mathbf{g}$*, amb* $\mathbf{bw}(G) \leq k$*, es pot construir en temps* $2^{3k+\mathcal{O}(\log k)} \cdot n^3$ *una descomposició "surface cut"* (T, μ) *de G d'amplada* $\leq 27k + \mathcal{O}(\mathbf{g})$*.*

2. Si la PD es fa en una descomposició "surface cut", llavors l'algorisme depèn del **bw** d'una manera exponencial simple:

Theorem (Rué, S., i Thilikos)

Donat un problema P codificable per agrupaments connexos en un graf G immers en una superfície de gènere $\mathbf{g}$*, amb* $\mathbf{bw}(G) \leq k$*, la mida de les taules de la PD per resoldre P en una descomposició "surface cut" de G està fitada per* $2^{\mathcal{O}(\log \mathbf{g} \cdot k + \log k \cdot \mathbf{g})}$*.*

Com fer servir aquesta metodologia

- Hem presentat una metodologia per obtenir algorismes de PD per grafs en superfícies amb temps $2^{\mathcal{O}(\mathbf{bw})} \cdot n^{\mathcal{O}(1)}$.

- Com fer servir aquesta metodologia?

  1. Sigui **P** un problema **codificable per agrupaments connexos** en un graf G immers en una superfície.
  2. En una etapa de **preprocessament**, construïm una **descomposició "surface cut"** de G, fent servir el 1r Teorema.
  3. Apliquem un algorisme de **PD** *intel.ligent* per resoldre **P** en la descomposició "surface cut".
  4. El temps d'execució **exponencial simple** de l'algorisme és una conseqüència del 2n Teorema.

Com fer servir aquesta metodologia

- Hem presentat una metodologia per obtenir algorismes de PD per grafs en superfícies amb temps $2^{\mathcal{O}(\mathbf{bw})} \cdot n^{\mathcal{O}(1)}$.
- Com fer servir aquesta metodologia?

1. Sigui **P** un problema **codificable per agrupaments connexos** en un graf G immers en una superfície.
2. En una etapa de **preprocessament**, construïm una **descomposició "surface cut"** de G, fent servir el 1r Teorema.
3. Apliquem un algorisme de **PD** *intel.ligent* per resoldre **P** en la descomposició "surface cut".
4. El temps d'execució **exponencial simple** de l'algorisme és una conseqüència del 2n Teorema.

Com fer servir aquesta metodologia

- Hem presentat una metodologia per obtenir algorismes de PD per grafs en superfícies amb temps $2^{\mathcal{O}(\mathbf{bw})} \cdot n^{\mathcal{O}(1)}$.

- Com fer servir aquesta metodologia?

  1. Sigui **P** un problema **codificable per agrupaments connexos** en un graf G immers en una superfície.
  2. En una etapa de **preprocessament**, construïm una **descomposició "surface cut"** de G, fent servir el 1r Teorema.
  3. Apliquem un algorisme de **PD** *intel.ligent* per resoldre **P** en la descomposició "surface cut".
  4. El temps d'execució **exponencial simple** de l'algorisme és una conseqüència del 2n Teorema.

Com fer servir aquesta metodologia

- Hem presentat una metodologia per obtenir algorismes de PD per grafs en superfícies amb temps $2^{\mathcal{O}(\mathbf{bw})} \cdot n^{\mathcal{O}(1)}$.

- Com fer servir aquesta metodologia?

  1. Sigui **P** un problema **codificable per agrupaments connexos** en un graf G immers en una superfície.
  2. En una etapa de **preprocessament**, construïm una **descomposició "surface cut"** de G, fent servir el 1r Teorema.
  3. Apliquem un algorisme de **PD** *intel.ligent* per resoldre **P** en la descomposició "surface cut".
  4. El temps d'execució **exponencial simple** de l'algorisme és una conseqüència del 2n Teorema.

Com fer servir aquesta metodologia

- Hem presentat una metodologia per obtenir algorismes de PD per grafs en superfícies amb temps $2^{\mathcal{O}(\mathbf{bw})} \cdot n^{\mathcal{O}(1)}$.

- Com fer servir aquesta metodologia?

    1. Sigui **P** un problema **codificable per agrupaments connexos** en un graf G immers en una superfície.
    2. En una etapa de **preprocessament**, construïm una **descomposició "surface cut"** de G, fent servir el 1r Teorema.
    3. Apliquem un algorisme de **PD** *intel.ligent* per resoldre **P** en la descomposició "surface cut".
    4. El temps d'execució **exponencial simple** de l'algorisme és una conseqüència del 2n Teorema.

La propera subsecció és...

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: *H*-MINOR CONTAINMENT

   ★ *H*-MINOR CONTAINMENT quan l'input graf *G* està en una superfície.
   [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

   Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.
   ($h = |V(H)|$, $k = \mathbf{bw}(G)$, $n = |V(G)|$)

   ★ Algorisme exponencial simple per input grafs planars.
   [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

   Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

   Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

[Rué, S., Thilikos. *2011*]

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: *H*-MINOR CONTAINMENT

   - ★ *H*-MINOR CONTAINMENT quan l'input graf G està en una superfície.
     [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

     Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.

     $(h = |V(H)|, k = \mathbf{bw}(G), n = |V(G)|)$

   - ★ Algorisme exponencial simple per input grafs planars.
     [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

     Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

     Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

[Rué, S., Thilikos. *2011*]

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: H-MINOR CONTAINMENT

   - ★ H-MINOR CONTAINMENT quan l'input graf G està en una superfície.
     [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

     Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.

     $(h = |V(H)|, k = \mathbf{bw}(G), n = |V(G)|)$

   - ★ Algorisme exponencial simple per input grafs planars.
     [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

     Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

     Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

[Rué, S., Thilikos. *2011*]

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: H-MINOR CONTAINMENT

   - H-MINOR CONTAINMENT quan l'input graf G està en una superfície.
     [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

     Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.

     $(h = |V(H)|, k = \mathbf{bw}(G), n = |V(G)|)$

   - Algorisme exponencial simple per input grafs planars.
     [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

     Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

     Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

[Rué, S., Thilikos. *2011*]

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: H-MINOR CONTAINMENT

   ★ H-MINOR CONTAINMENT quan l'input graf G està en una superfície.
   [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

   Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.
   $(h = |V(H)|, k = \mathbf{bw}(G), n = |V(G)|)$

   ★ Algorisme exponencial simple per input grafs planars.
   [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

   Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

   Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

2. Es pot aplicar aquesta metodologia a **grafs més generals**?

   Resultat recent: grafs "minor-free" [Rué, S., Thilikos. *2011*]

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: H-MINOR CONTAINMENT

   ★ H-MINOR CONTAINMENT quan l'input graf G està en una superfície.
   [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

   Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.
   $(h = |V(H)|,\ k = \mathbf{bw}(G),\ n = |V(G)|)$

   ★ Algorisme exponencial simple per input grafs planars.
   [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

   Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

   Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

2. Es pot aplicar aquesta metodologia a **grafs més generals**?

   Resultat recent: grafs "minor-free" [Rué, S., Thilikos. *2011*]

Generalitzacions

1. Es pot aplicar aquesta metodologia a **problemes més complicats**?

   Problema fonamental: *H*-MINOR CONTAINMENT

   ★ *H*-MINOR CONTAINMENT quan l'input graf *G* està en una superfície.
   [Adler, Dorn, Fomin, S., Thilikos. *SWAT'10*]

   Algo amb temps $2^{\mathcal{O}(k)} \cdot h^{2k} \cdot 2^{\mathcal{O}(h)} \cdot n$.
   $(h = |V(H)|, k = \mathbf{bw}(G), n = |V(G)|)$

   ★ Algorisme exponencial simple per input grafs planars.
   [Adler, Dorn, Fomin, S., Thilikos. *ESA'10*]

   Realment exponencial simple: $2^{\mathcal{O}(h)} \cdot n$.

   Es pot generalitzar a grafs immersos en una superfície arbitrària?

   Problemes de grafs arrelats: CAMINS DISJUNTS

2. Es pot aplicar aquesta metodologia a **grafs més generals**?

   Resultat recent: grafs "minor-free" [Rué, S., Thilikos. *2011*]

Altres resultats recents

1. Algorismes aleatoris per problemes codificables per agrupaments connexos en grafs generals, amb temps $2^{\mathcal{O}(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

   [Cygan, Nederlof, (Pilipczuk)^2, van Rooij, Wojtaszczyk. *FOCS'11*]

   - Introdueixen una tècnica de PD anomenada Cut&Count.
     (Es basa en un resultat de Probabilitat: Lema d'isolació.)
   - Es poden desaleatoritzar aquests algorismes?

   [Lokshtanov, Marx, Saurabh. *SODA'11*]
   Si 3SAT no es pot resoldre en $2^{o(n)}$, llavors CAMINS DISJUNTS no es pot resoldre en $2^{o(\mathbf{tw}\log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ en grafs generals.

   - HAMILTONIAN PATH, FVS, CONNECTED VERTEX COVER,...
     És $2^{\mathcal{O}(\mathbf{tw}\log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ el millor possible?

Altres resultats recents

1. Algorismes aleatoris per problemes codificables per agrupaments connexos en grafs generals, amb temps $2^{\mathcal{O}(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

   [Cygan, Nederlof, (Pilipczuk)^2, van Rooij, Wojtaszczyk. *FOCS'11*]

   - Introdueixen una tècnica de PD anomenada Cut&Count.
     (Es basa en un resultat de Probabilitat: Lema d'isolació.)
   - Es poden desaleatoritzar aquests algorismes?

   [Lokshtanov, Marx, Saurabh. *SODA'11*]
   Si 3SAT no es pot resoldre en $2^{o(n)}$, llavors CAMINS DISJUNTS no es pot resoldre en $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ en grafs generals.

   - HAMILTONIAN PATH, FVS, CONNECTED VERTEX COVER,...
     És $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ el millor possible?

Altres resultats recents

1. Algorismes aleatoris per problemes codificables per agrupaments connexos en grafs generals, amb temps $2^{\mathcal{O}(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

   [Cygan, Nederlof, (Pilipczuk)^2, van Rooij, Wojtaszczyk. *FOCS'11*]

   - Introdueixen una tècnica de PD anomenada Cut&Count.
     (Es basa en un resultat de Probabilitat: Lema d'isolació.)
   - Es poden desaleatoritzar aquests algorismes?

   [Lokshtanov, Marx, Saurabh. *SODA'11*]
   Si 3SAT no es pot resoldre en $2^{o(n)}$, llavors CAMINS DISJUNTS no es pot resoldre en $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ en grafs generals.

   - HAMILTONIAN PATH, FVS, CONNECTED VERTEX COVER,...
     És $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ el millor possible?

Altres resultats recents

1. Algorismes aleatoris per problemes codificables per agrupaments connexos en grafs generals, amb temps $2^{\mathcal{O}(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

   [Cygan, Nederlof, (Pilipczuk)^2, van Rooij, Wojtaszczyk. *FOCS'11*]

   - Introdueixen una tècnica de PD anomenada Cut&Count.
     (Es basa en un resultat de Probabilitat: Lema d'isolació.)
   - Es poden desaleatoritzar aquests algorismes?

2. Per a un problema FPT, és sempre possible que existeixin algorismes amb temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$?

   [Lokshtanov, Marx, Saurabh. *SODA'11*]
   Si 3SAT no es pot resoldre en $2^{o(n)}$, llavors Camins Disjunts no es pot resoldre en $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ en grafs generals.

   - Hamiltonian Path, FVS, Connected Vertex Cover,...
     És $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ el millor possible?

Altres resultats recents

1. Algorismes aleatoris per problemes codificables per agrupaments connexos en grafs generals, amb temps $2^{\mathcal{O}(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

   [Cygan, Nederlof, (Pilipczuk)^2, van Rooij, Wojtaszczyk. *FOCS'11*]

   - Introdueixen una tècnica de PD anomenada Cut&Count. *(Es basa en un resultat de Probabilitat: Lema d'isolació.)*
   - Es poden desaleatoritzar aquests algorismes?

2. Per a un problema FPT, és sempre possible que existeixin algorismes amb temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$?

   [Lokshtanov, Marx, Saurabh. *SODA'11*]
   Si 3SAT no es pot resoldre en $2^{o(n)}$, llavors CAMINS DISJUNTS no es pot resoldre en $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ en grafs generals.

   - HAMILTONIAN PATH, FVS, CONNECTED VERTEX COVER,...
     És $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ el millor possible?

Altres resultats recents

1. Algorismes aleatoris per problemes codificables per agrupaments connexos en grafs generals, amb temps $2^{\mathcal{O}(\mathbf{tw})} \cdot n^{\mathcal{O}(1)}$.

   [Cygan, Nederlof, (Pilipczuk)^2, van Rooij, Wojtaszczyk. *FOCS'11*]

   - Introdueixen una tècnica de PD anomenada Cut&Count.
     (Es basa en un resultat de Probabilitat: Lema d'isolació.)
   - Es poden desaleatoritzar aquests algorismes?

2. Per a un problema FPT, és sempre possible que existeixin algorismes amb temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$?

   [Lokshtanov, Marx, Saurabh. *SODA'11*]
   Si 3SAT no es pot resoldre en $2^{o(n)}$, llavors CAMINS DISJUNTS no es pot resoldre en $2^{o(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ en grafs generals.

   - HAMILTONIAN PATH, FVS, CONNECTED VERTEX COVER,...
     És $2^{\mathcal{O}(\mathbf{tw} \log \mathbf{tw})} \cdot n^{\mathcal{O}(1)}$ el millor possible?

La propera secció és...

1. Motivació: complexitat parametritzada
2. 1r paràmetre: amplada de branques/arbre
   - Definicions
   - Programació dinàmica
   - Generalitzacions i conclusions
3. 2n paràmetre: mida de la solució
   - Motivació
   - Trencar carbasses

La propera subsecció és...

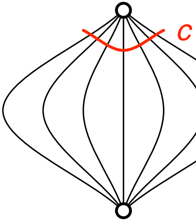
Carbasses 

("graf" = multigraf)

[G. Joret, C. Paul, I. S., S. Saurabh, S. Thomassé. *ESA'11*]

Carbasses 

c-carbassa:



(“graf” = multigraf)

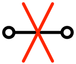
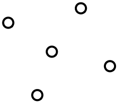
[G. Joret, C. Paul, I. S., S. Saurabh, S. Thomassé. *ESA’11*]

Grafs sense la c-carbassa com a menor

- $c = 1$: grafs buits
- $c = 2$: boscos
- $c = 3$: no hi ha dos cicles compartint una aresta
- etc.

Grafs sense la c-carbassa com a menor

- $c = 1$: grafs buits

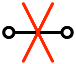
- $c = 2$: boscos

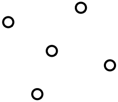
- $c = 3$: no hi ha dos cicles compartint una aresta

- etc.

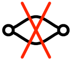
Grafs sense la c-carbassa com a menor

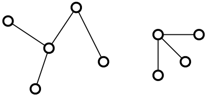
- $c = 1$: grafs buits





- $c = 2$: boscos



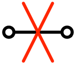
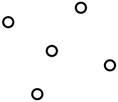


- $c = 3$: no hi ha dos cicles compartint una aresta

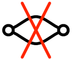
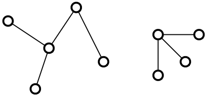
- etc.

Grafs sense la c-carbassa com a menor

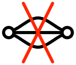
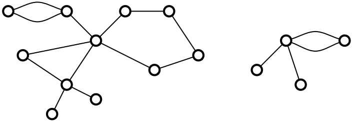
- $c = 1$: grafs buits

- $c = 2$: boscos

- $c = 3$: no hi ha dos cicles compartint una aresta

- etc.

La propera subsecció és...

1. Motivació: complexitat parametritzada
2. 1r paràmetre: amplada de branques/arbre
   - Definicions
   - Programació dinàmica
   - Generalitzacions i conclusions
3. 2n paràmetre: mida de la solució
   - Motivació
   - Trencar carbasses

Trencar carbasses

A partir d'ara:

- $c \geq 1$ un enter fixat
- G input graf
- $n := |V(G)|$

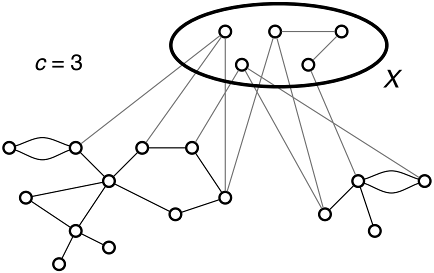
Conjunt trencador de c-carbasses (hitting set):
subconjunt $X \subseteq V(G)$ tal que $G - X$ no té c-carbasses com a menor

Trencar carbasses

A partir d'ara:

- $c \geq 1$ un enter fixat
- G input graf
- $n := |V(G)|$

Conjunt trencador de c-carbasses (hitting set):
subconjunt $X \subseteq V(G)$ tal que $G - X$ no té c-carbasses com a menor



El problema en qüestió

Ens interessem pel següent problema parametritzat:

k-CONJUNT TRENCADOR DE c-CARBASSES:
Input: un graf $G = (V, E)$.
Paràmetre: k.
Objectiu: decidir si $\exists$ conjunt trencador de c-carbasses de mida $\leq k$.

La versió d'optimització és NP-difícil $\forall c \geq 1$.

Alguns casos particulars i el resultat principal

- $c = 1$: VERTEX COVER
  Algorisme FPT en temps $\mathcal{O}(1.2738^k + kn)$ [Chen, Kanj, Xia '10]
- $c = 2$: FEEDBACK VERTEX SET
  Algorisme FPT en temps $\mathcal{O}(3.83^k \cdot k \cdot n^2)$ [Cao, Chen, Liu '10]
- $c \geq 3$:
  Algorisme FPT $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

Pregunta: Hi ha un algorisme FPT en temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ $\forall c \geq 1$?
(és a dir, un algorisme FPT **exponencial simple**)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

Hi ha un algorisme FPT exponencial simple $\forall c \geq 1$

Nota: no hi ha algorisme FPT en temps $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ assumint la ETH.

Alguns casos particulars i el resultat principal

- $c = 1$: VERTEX COVER
  Algorisme FPT en temps $\mathcal{O}(1.2738^k + kn)$ [Chen, Kanj, Xia '10]

- $c = 2$: FEEDBACK VERTEX SET
  Algorisme FPT en temps $\mathcal{O}(3.83^k \cdot k \cdot n^2)$ [Cao, Chen, Liu '10]

- $c \geq 3$:
  Algorisme FPT $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

Pregunta: Hi ha un algorisme FPT en temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ $\forall c \geq 1$?
(és a dir, un algorisme FPT **exponencial simple**)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

Hi ha un algorisme FPT exponencial simple $\forall c \geq 1$

Nota: no hi ha algorisme FPT en temps $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ assumint la ETH

Alguns casos particulars i el resultat principal

- $c = 1$: VERTEX COVER
  Algorisme FPT en temps $\mathcal{O}(1.2738^k + kn)$ [Chen, Kanj, Xia '10]

- $c = 2$: FEEDBACK VERTEX SET
  Algorisme FPT en temps $\mathcal{O}(3.83^k \cdot k \cdot n^2)$ [Cao, Chen, Liu '10]

- $c \geq 3$:
  Algorisme FPT $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

Pregunta: Hi ha un algorisme FPT en temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ $\forall c \geq 1$?
(és a dir, un algorisme FPT **exponencial simple**)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

Hi ha un algorisme FPT exponencial simple $\forall c \geq 1$

Nota: no hi ha algorisme FPT en temps $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ assumint la ETH

Alguns casos particulars i el resultat principal

- $c = 1$: Vertex Cover
  Algorisme FPT en temps $\mathcal{O}(1.2738^k + kn)$ [Chen, Kanj, Xia '10]
- $c = 2$: Feedback Vertex Set
  Algorisme FPT en temps $\mathcal{O}(3.83^k \cdot k \cdot n^2)$ [Cao, Chen, Liu '10]
- $c \geq 3$:
  Algorisme FPT $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

Pregunta: Hi ha un algorisme FPT en temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ $\forall c \geq 1$?
(és a dir, un algorisme FPT **exponencial simple**)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

Hi ha un algorisme FPT exponencial simple $\forall c \geq 1$

Nota: no hi ha algorisme FPT en temps $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ assumint la ETH

Alguns casos particulars i el resultat principal

- $c = 1$: VERTEX COVER
  Algorisme FPT en temps $\mathcal{O}(1.2738^k + kn)$ [Chen, Kanj, Xia '10]
- $c = 2$: FEEDBACK VERTEX SET
  Algorisme FPT en temps $\mathcal{O}(3.83^k \cdot k \cdot n^2)$ [Cao, Chen, Liu '10]
- $c \geq 3$:
  Algorisme FPT $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

Pregunta: Hi ha un algorisme FPT en temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ $\forall c \geq 1$?
(és a dir, un algorisme FPT **exponencial simple**)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

Hi ha un algorisme FPT exponencial simple $\forall c \geq 1$

Nota: no hi ha algorisme FPT en temps $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ assumint la ETH.

Alguns casos particulars i el resultat principal

- $c = 1$: VERTEX COVER
  Algorisme FPT en temps $\mathcal{O}(1.2738^k + kn)$ [Chen, Kanj, Xia '10]

- $c = 2$: FEEDBACK VERTEX SET
  Algorisme FPT en temps $\mathcal{O}(3.83^k \cdot k \cdot n^2)$ [Cao, Chen, Liu '10]

- $c \geq 3$:
  Algorisme FPT $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ [Fomin, Lokshtanov, Misra, Philip, Saurabh '11]

Pregunta: Hi ha un algorisme FPT en temps $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ $\forall c \geq 1$?
(és a dir, un algorisme FPT **exponencial simple**)

Theorem (Joret, Paul, S., Saurabh, Thomassé)

Hi ha un algorisme FPT exponencial simple $\forall c \geq 1$

Nota: no hi ha algorisme FPT en temps $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ assumint la ETH

Algorisme FPT

Objectiu: Algorisme FPT exponencial simple.

Utilitzem la tècnica de la compressió iterativa.

Ens concentrem en la versió "DISJUNTA" del problema:

Input: G, conjunt trencador X amb $|X| \leq k+1$.

Pregunta: hi ha un conjunt trencador X', amb $|X'| \leq k$, que sigui **disjunt** d'X?

Algorisme FPT

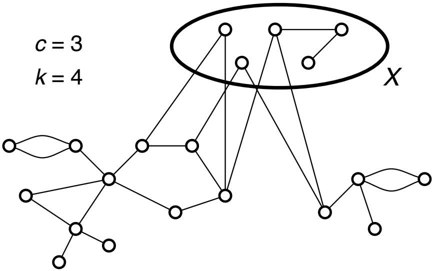
Objectiu: Algorisme FPT exponencial simple.

Utilitzem la tècnica de la compressió iterativa.

Ens concentrem en la versió "DISJUNTA" del problema:

Input: G, conjunt trencador X amb $|X| \leq k + 1$.

Pregunta: hi ha un conjunt trencador X', amb $|X'| \leq k$, que sigui **disjunt** d'X?



Algorisme FPT

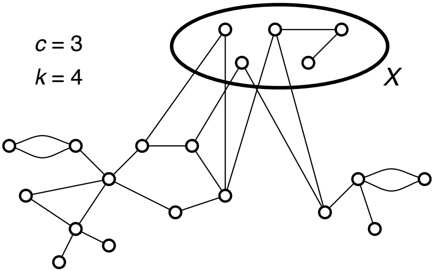
Objectiu: Algorisme FPT exponencial simple.

Utilitzem la tècnica de la compressió iterativa.

Ens concentrem en la versió "DISJUNTA" del problema:

Input: G, conjunt trencador X amb $|X| \leq k + 1$.

Pregunta: hi ha un conjunt trencador X', amb $|X'| \leq k$, que sigui **disjunt** d'X?



Algorisme FPT

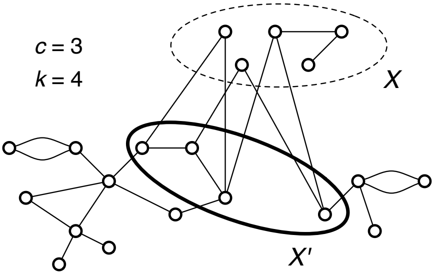
Objectiu: Algorisme FPT exponencial simple.

Utilitzem la tècnica de la compressió iterativa.

Ens concentrem en la versió "DISJUNTA" del problema:

Input: G, conjunt trencador X amb $|X| \leq k+1$.

Pregunta: hi ha un conjunt trencador X', amb $|X'| \leq k$, que sigui **disjunt** d'X?



Lemma

Algo $d^k \cdot n^{\mathcal{O}(1)}$ *per la versió* DISJUNTA $\Rightarrow$
Algo $(d+1)^k \cdot n^{\mathcal{O}(1)}$ *per* k-CONJUNT TRENCADOR DE c-CARBASSES

Ingredients principals de l'algorisme FPT per la versió DISJUNTA:

- ★ Regles de reducció ad hoc (polinomials)
- ★ Regla de reducció basada en "protrusions"
- ★ Regla de "branching" (amb # de subproblemes exponencial simple)
- ★ "kernel" lineal en un cas especial

- ★ Podem evitar fer servir les "protrusions"?
- ★ Problema difícil: eliminar $\leq k$ vèrtexs d'un graf donat per obtenir un graf amb tw $\leq c$, per $c \geq 1$?

(OK, exponencial simple per $c = 2$) [Kim, Paul, Philip'11]

Lemma

Algo $d^k \cdot n^{\mathcal{O}(1)}$ *per la versió* DISJUNTA $\Rightarrow$
Algo $(d+1)^k \cdot n^{\mathcal{O}(1)}$ *per* k-CONJUNT TRENCADOR DE c-CARBASSES

Ingredients principals de l'algorisme FPT per la versió DISJUNTA:

- ★ Regles de reducció ad hoc (polinomials)
- ★ Regla de reducció basada en "protrusions"
- ★ Regla de "branching" (amb $\#$ de subproblemes exponencial simple)
- ★ "kernel" lineal en un cas especial

- ★ Podem evitar fer servir les "protrusions"?
- ★ Problema difícil: eliminar $\leq k$ vèrtexs d'un graf donat per obtenir un graf amb **tw** $\leq c$, per $c \geq 1$?

(OK, exponencial simple per $c = 2$) [Kim, Paul, Philip'11]

Lemma

Algo $d^k \cdot n^{\mathcal{O}(1)}$ per la versió DISJUNTA $\Rightarrow$
Algo $(d+1)^k \cdot n^{\mathcal{O}(1)}$ per k-CONJUNT TRENCADOR DE *c*-CARBASSES

Ingredients principals de l'algorisme FPT per la versió DISJUNTA:

- ★ Regles de reducció ad hoc (polinomials)
- ★ Regla de reducció basada en "protrusions"
- ★ Regla de "branching" (amb $\#$ de subproblemes exponencial simple)
- ★ "kernel" lineal en un cas especial

- ★ Podem evitar fer servir les "protrusions"?
- ★ Problema difícil: eliminar $\leq k$ vèrtexs d'un graf donat per obtenir un graf amb **tw** $\leq c$, per $c \geq 1$?

(OK, exponencial simple per $c = 2$) [Kim, Paul, Philip'11]

Lemma

Algo $d^k \cdot n^{\mathcal{O}(1)}$ *per la versió* DISJUNTA $\Rightarrow$
Algo $(d+1)^k \cdot n^{\mathcal{O}(1)}$ *per* k-CONJUNT TRENCADOR DE c-CARBASSES

Ingredients principals de l'algorisme FPT per la versió DISJUNTA:

- ★ Regles de reducció ad hoc (polinomials)
- ★ Regla de reducció basada en "protrusions"
- ★ Regla de "branching" (amb # de subproblemes exponencial simple)
- ★ "kernel" lineal en un cas especial

- ★ Podem evitar fer servir les "protrusions"?
- ★ Problema difícil: eliminar $\leq k$ vèrtexs d'un graf donat per obtenir un graf amb **tw** $\leq c$, per $c \geq 1$?

(OK, exponencial simple per $c = 2$) [Kim, Paul, Philip'11]

Lemma

Algo $d^k \cdot n^{\mathcal{O}(1)}$ *per la versió* DISJUNTA $\Rightarrow$
Algo $(d+1)^k \cdot n^{\mathcal{O}(1)}$ *per* k-CONJUNT TRENCADOR DE c-CARBASSES

Ingredients principals de l'algorisme FPT per la versió DISJUNTA:

- ★ Regles de reducció ad hoc (polinomials)
- ★ Regla de reducció basada en "protrusions"
- ★ Regla de "branching" (amb $\#$ de subproblemes exponencial simple)
- ★ "kernel" lineal en un cas especial

- ★ Podem evitar fer servir les "protrusions"?
- ★ Problema difícil: eliminar $\leq k$ vèrtexs d'un graf donat per obtenir un graf amb **tw** $\leq c$, per $c \geq 1$?

(OK, exponencial simple per $c = 2$) [Kim, Paul, Philip'11]

Gràcies!