Finding a Sparse $k$-Subgraph in Restricted Graph Classes

Rémi Watrigant, Marin Bougeret and Rodolphe Giroudeau

LIRMM, Montpellier, France

Séminaire d’Optimisation Combinatoire
LAMSADE, Université Paris-Dauphine
25/02/2013
Contents

1 Introduction

2 PTAS in Proper Interval Graphs

3 FPT Algorithm in Interval Graphs

4 Open Problems and Future Work
Sparsest $k$-Subgraph Problem ($SkS$)

**Input:** a simple undirected graph $G = (V, E)$, $k \leq |V|$.

**Output:** a set $S \subseteq V$ of size exactly $k$.

**Goal:** minimize $E(S)$ (the number of edges induced by $S$).
Sparsest $k$-Subgraph Problem ($SkS$)

**Input:** a simple undirected graph $G = (V, E)$, $k \leq |V|$.

**Output:** a set $S \subseteq V$ of size exactly $k$.

**Goal:** minimize $E(S)$ (the number of edges induced by $S$)

- Generalization of **INDEPENDENT SET**
  - $SkS$ NP-hard in general graphs (+ W[1]-hard, inapproximable)
Sparsest $k$-Subgraph Problem ($SkS$)

**Input:** a simple undirected graph $G = (V, E)$, $k \leq |V|$.  
**Output:** a set $S \subseteq V$ of size exactly $k$.  
**Goal:** minimize $E(S)$ (the number of edges induced by $S$)

- Generalization of **INDEPENDENT SET**  
  $\Rightarrow SkS$ NP-hard in general graphs (+ W[1]-hard, inapproximable)

- Related problems:
  - maximisation version: **Densest $k$-Subgraph ($DkS$)**
    - exact result for $DkS$ on $\overline{C} \iff$ exact result for $SkS$ on $\overline{C}$
  - dual version: **Maximum Vertex Coverage ($MVC$)**
    - $S \subseteq V$ opt. solution for $SkS \iff V \setminus S$ opt. solution for $MVC(n-k)$

but approximation do not transfer...
Sparsest $k$-Subgraph Problem ($SkS$)

**Input:** a simple undirected graph $G = (V, E)$, $k \leq |V|$.

**Output:** a set $S \subseteq V$ of size exactly $k$.

**Goal:** minimize $E(S)$ (the number of edges induced by $S$)

- Generalization of INDEPENDENT SET
  \[ \Rightarrow SkS \text{ NP-hard in general graphs (}+\text{ W[1]-hard, inapproximable)} \]

- Related problems:
  - maximisation version: Densest $k$-Subgraph ($DkS$)
    - exact result for $DkS$ on $\mathcal{C} \iff$ exact result for $SkS$ on $\overline{\mathcal{C}}$
  - dual version: Maximum Vertex Coverage ($MVC$)
    - $S \subseteq V$ opt. solution for $SkS \iff V \setminus S$ opt. solution for $MVC(n-k)$

  but approximation do not transfer...

- we study $SkS$ in classes where INDEPENDENT SET is polynomial-time solvable
  e.g. perfect graphs and their subclasses
In restricted graphs classes:

<table>
<thead>
<tr>
<th>Densest $k$-Subgraph</th>
<th>Sparsest $k$-Subgraph</th>
</tr>
</thead>
</table>

Watrigant, Bougeret, Giroudeau
In restricted graphs classes:

**Densest \( k \)-Subgraph**
- Chordal graphs:
  - \( NP \)-hard [Corneil, Perl, 1984]
  - 3-approx. [Liazi, Milis, Zissimopoulos, 2008]
  - \( FPT \) ("obvious")

**Sparsest \( k \)-Subgraph**
- Chordal graphs:
  - \( NP \)-hard [Bougeret, Giroudeau, W., 2013]
In restricted graphs classes:

**Densest \( k \)-Subgraph**
- Chordal graphs:  
  \( NP \)-hard [Corneil, Perl, 1984]  
  3-approx. [Liazi, Milis, Zissimopoulos, 2008]  
  \( FPT \) ("obvious")  
- Interval graphs:  
  \( NP \)-h/Poly : open (since [Corneil, Perl, 1984])  
  \( PTAS \) [Nonner, 2011]

**Sparsest \( k \)-Subgraph**
- Chordal graphs:  
  \( NP \)-hard [Bougeret, Giroudeau, W., 2013]  
- Interval graphs:  
  \( NP \)-h/Poly : open  
  2-approx. [Bougeret, Giroudeau, W., 2013]  
  \( FPT \) [Bougeret, Giroudeau, W., 2013]
In restricted graphs classes:

<table>
<thead>
<tr>
<th>Densest $k$-Subgraph</th>
<th>Sparsest $k$-Subgraph</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chordal graphs:</strong></td>
<td><strong>Chordal graphs:</strong></td>
</tr>
<tr>
<td>(NP)-hard [Corneil, Perl, 1984]</td>
<td>(NP)-hard [Bougeret, Giroudeau, W., 2013]</td>
</tr>
<tr>
<td>3-approx. [Liazi, Milis, Zissimopoulos, 2008]</td>
<td></td>
</tr>
<tr>
<td>(FPT) (&quot;obvious&quot;)</td>
<td></td>
</tr>
<tr>
<td><strong>Interval graphs:</strong></td>
<td><strong>Interval graphs:</strong></td>
</tr>
<tr>
<td>(NP)-h/Poly : open (since [Corneil, Perl, 1984])</td>
<td>(NP)-h/Poly : open</td>
</tr>
<tr>
<td>(PTAS) [Nonner, 2011]</td>
<td>(FPT) [Bougeret, Giroudeau, W., 2013]</td>
</tr>
<tr>
<td><strong>Proper interval graphs:</strong></td>
<td><strong>Proper interval graphs:</strong></td>
</tr>
<tr>
<td>(NP)-h/Poly : open</td>
<td>(NP)-h/Poly : open</td>
</tr>
<tr>
<td></td>
<td>(PTAS) [Bougeret, Giroudeau, W., 2013]</td>
</tr>
</tbody>
</table>
In restricted graphs classes:

**Densest \( k \)-Subgraph**
- Chordal graphs: 
  \( NP \)-hard [Corneil, Perl, 1984]
  3-approx. [Liazi, Milis, Zissimopoulos, 2008]
  \( FPT \) ("obvious")
- Interval graphs:
  \( NP \)-h/Poly: open (since [Corneil, Perl, 1984])
  \( PTAS \) [Nonner, 2011]
- Proper interval graphs:
  \( NP \)-h/Poly: open
- Bipartite graphs:
  \( NP \)-h [Corneil, Perl, 1984]

**Sparsest \( k \)-Subgraph**
- Chordal graphs:
  \( NP \)-hard [Bougeret, Giroudeau, W., 2013]
- Interval graphs:
  \( NP \)-h/Poly: open
  2-approx. [Bougeret, Giroudeau, W., 2013]
  \( FPT \) [Bougeret, Giroudeau, W., 2013]
- Proper interval graphs:
  \( NP \)-h/Poly: open
  \( PTAS \) [Bougeret, Giroudeau, W., 2013]
- Bipartite graphs:
  \( NP \)-h [Joret, Vetta, 2012]
In restricted graphs classes:

### Densest $k$-Subgraph

- **Chordal graphs:**
  - \( NP \)-hard [Corneil,Perl,1984]
  - 3-approx. [Liazi,Milis,Zissimopoulos,2008]
  - \( FPT \) ("obvious")

- **Interval graphs:**
  - \( NP \)-h/Poly : open (since [Corneil,Perl,1984])
  - \( PTAS \) [Nonner,2011]

- **Proper interval graphs:**
  - \( NP \)-h/Poly : open

- **Bipartite graphs:**
  - \( NP \)-h [Corneil,Perl,1984]

- **Split graphs:**
  - Polynomial

### Sparsest $k$-Subgraph

- **Chordal graphs:**
  - \( NP \)-hard [Bougeret,Giroudeau,W.,2013]

- **Interval graphs:**
  - \( NP \)-h/Poly : open
  - 2-approx. [Bougeret,Giroudeau,W.,2013]
  - \( FPT \) [Bougeret,Giroudeau,W.,2013]

- **Proper interval graphs:**
  - \( NP \)-h/Poly : open
  - \( PTAS \) [Bougeret,Giroudeau,W.,2013]

- **Bipartite graphs:**
  - \( NP \)-h [Joret,Vetta,2012]

- **Split graphs:**
  - Polynomial
In restricted graphs classes:

<table>
<thead>
<tr>
<th>Densest $k$-Subgraph</th>
<th>Sparsest $k$-Subgraph</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chordal graphs:</strong></td>
<td><strong>Chordal graphs:</strong></td>
</tr>
<tr>
<td>3-approx. [Liazi, Milis, Zissimopoulos, 2008]</td>
<td></td>
</tr>
<tr>
<td>$FPT$ (&quot;obvious&quot;)</td>
<td></td>
</tr>
<tr>
<td><strong>Interval graphs:</strong></td>
<td><strong>Interval graphs:</strong></td>
</tr>
<tr>
<td>$NP$-h/Poly: open (since [Corneil, Perl, 1984])</td>
<td>$NP$-h/Poly: open</td>
</tr>
<tr>
<td><strong>Proper interval graphs:</strong></td>
<td><strong>Proper interval graphs:</strong></td>
</tr>
<tr>
<td>$NP$-h/Poly: open</td>
<td>$NP$-h/Poly: open</td>
</tr>
<tr>
<td>$PTAS$ [Bougeret, Giroudeau, W., 2013]</td>
<td></td>
</tr>
<tr>
<td><strong>Bipartite graphs:</strong></td>
<td><strong>Bipartite graphs:</strong></td>
</tr>
<tr>
<td>$NP$-h [Corneil, Perl, 1984]</td>
<td>$NP$-h [Joret, Vetta, 2012]</td>
</tr>
<tr>
<td><strong>Split graphs:</strong></td>
<td><strong>Split graphs:</strong></td>
</tr>
<tr>
<td>Polynomial</td>
<td>Polynomial</td>
</tr>
<tr>
<td><strong>Planar graphs:</strong></td>
<td><strong>Planar graphs:</strong></td>
</tr>
<tr>
<td>$NP$-h/Poly: open</td>
<td>$NP$-h (from Independent Set)</td>
</tr>
<tr>
<td></td>
<td>$FPT$ (&quot;obvious&quot; linear kernel)</td>
</tr>
</tbody>
</table>

Watrigant, Bougeret, Giroudeau
Finding a Sparse $k$-Subgraph in Restricted Graph Classes
In this talk:

- *PTAS* in Proper Interval graphs.
- *FPT* algorithm in Interval graphs parameterized by the number of edges in the solution (stronger parameterization than by $k$).
Definitions

Polynomial-Time Approximation Scheme (PTAS)

A PTAS for a minimization problem is an algorithm $A_\epsilon$ such that for any fixed $\epsilon > 0$, $A_\epsilon$ runs in polynomial time and outputs a solution of cost $< (1 + \epsilon)OPT$.
Definitions

Polynomial-Time Approximation Scheme (PTAS)
A PTAS for a minimization problem is an algorithm $A_\epsilon$ such that for any fixed $\epsilon > 0$, $A_\epsilon$ runs in polynomial time and outputs a solution of cost $< (1 + \epsilon)OPT$.

Fixed-Parameter Tractable (FPT)
An FPT algorithm for a parameterized problem is an algorithm that exactly solves the problem in $O(f(k).\text{poly}(n))$ where $n$ is the size of the instance and $k$ the parameter of the instance.
Definitions

Interval graphs = intersection graphs of intervals on the real line.
Definitions

Interval graphs = intersection graphs of intervals on the real line.

proper interval graph = no interval contains properly another one = unit interval graphs
PTAS in Proper Interval Graphs

Idea of the algorithm:
PTAS in Proper Interval Graphs

Idea of the algorithm:
- sorting intervals according to their right endpoints
PTAS in Proper Interval Graphs

Idea of the algorithm:

- sorting intervals according to their right endpoints
- greedy decomposition of the graph into a path of separators
PTAS in Proper Interval Graphs

Idea of the algorithm:
- sorting intervals according to their right endpoints
- greedy decomposition of the graph into a path of separators
- re-structuration of an optimal solution into a near optimal solution such that all near optimal solutions can be enumerated in polynomial time
PTAS in Proper Interval Graphs

Idea of the algorithm:

- sorting intervals according to their right endpoints
- greedy decomposition of the graph into a path of separators
- re-structuration of an optimal solution into a near optimal solution such that all near optimal solutions can be enumerated in polynomial time
- dynamic programming processes the graph through the decomposition, enumerating all possible solutions.
PTAS in Proper Interval Graphs

The decomposition:
PTAS in Proper Interval Graphs

The decomposition:

\[ I_{m_1} \]
PTAS in Proper Interval Graphs

The decomposition:

\[ R_1 \xrightarrow{I_{m_1}} B_1 \]
PTAS in Proper Interval Graphs

The decomposition:
PTAS in Proper Interval Graphs

The decomposition:

\[ R_1 \quad \quad I_{m_1} \quad \quad B_1 \quad \quad L_2 \quad \quad I_{m_2} \]
The decomposition:
The only edges between blocks $B_i$ and $B_{i+1}$ are between $R_i$ and $L_{i+1}$.

Given $S \subseteq I$ we have:

$$E(S) = \sum_{i=1}^{a} E(B_i \cap S) + \sum_{i=1}^{a-1} E(R_i \cap S, L_{i+1} \cap S)$$
Compaction

Let $S \subseteq \mathcal{I}$ be a solution, and $S^c = \text{comp}(S) \subseteq \mathcal{I}$ such that for each block $i \in \{1, \ldots, a\}$:

- for all $I \in S \cap L_i$, \text{comp}(I) \in L_i \text{ and is at the right of } I$
- for all $I \in S \cap R_i$, \text{comp}(I) \in R_i \text{ and is at the left of } I$
Compaction

Let $S \subseteq \mathcal{I}$ be a solution, and $S^c = \text{comp}(S) \subseteq \mathcal{I}$ such that for each block $i \in \{1, \ldots, a\}$:

- for all $I \in S \cap L_i$, $\text{comp}(I) \in L_i$ and is at the right of $I$
- for all $I \in S \cap R_i$, $\text{comp}(I) \in R_i$ and is at the left of $I$
Compaction

Let $S \subseteq \mathcal{I}$ be a solution, and $S^c = \text{comp}(S) \subseteq \mathcal{I}$ such that for each block $i \in \{1, \ldots, a\}$:

- for all $I \in S \cap L_i$, $\text{comp}(I) \in L_i$ and is at the right of $I$
- for all $I \in S \cap R_i$, $\text{comp}(I) \in R_i$ and is at the left of $I$
Compaction

Let $S \subseteq \mathcal{I}$ be a solution, and $S^c = \text{comp}(S) \subseteq \mathcal{I}$ such that for each block $i \in \{1, \ldots, a\}$:

- for all $I \in S \cap L_i$, $\text{comp}(I) \in L_i$ and is at the right of $I$
- for all $I \in S \cap R_i$, $\text{comp}(I) \in R_i$ and is at the left of $I$
Compaction

Let $S \subseteq \mathcal{I}$ be a solution, and $S^c = \text{comp}(S) \subseteq \mathcal{I}$ such that for each block $i \in \{1, \ldots, a\}$:

- for all $I \in S \cap L_i$, $\text{comp}(I) \in L_i$ and is at the right of $I$

- for all $I \in S \cap R_i$, $\text{comp}(I) \in R_i$ and is at the left of $I$

Lemma

If $\text{comp}$ is a compaction of a solution $S$ such that for all block $i \in \{1, \ldots, a\}$, we have

$$E(\text{comp}(S \cap B_i)) \leq \rho E(S \cap B_i)$$

Then $\text{comp}(S)$ is a $\rho$-approximation of $S$. 
PTAS in Proper Interval Graphs
Re-structuration of optimal solutions

Let us built a compaction that yields a \((1 + \frac{4}{P})\)-approximation for any fixed \(P\).
PTAS in Proper Interval Graphs

Re-structuration of optimal solutions

Let us built a compaction that yields a $(1 + \frac{4}{P})$-approximation for any fixed $P$. Let $X \subseteq B_i$ be a solution. We note $X = X_L \cup X_R$. Set sizes are in lowercase.
PTAS in Proper Interval Graphs
Re-structuration of optimal solutions

Let us built a compaction that yields a \((1 + \frac{4}{P})\)-approximation for any fixed \(P\).
Let \(X \subseteq B_i\) be a solution. We note \(X = X_L \cup X_R\). Set sizes are in lowercase.

- we divide \(X_L\) into \(P\) consecutive subsets of same size \(q_L \rightarrow X_1^L, \ldots, X_P^L\)
- we divide \(X_R\) into \(P\) consecutive subsets of same size \(q_R \rightarrow X_1^R, \ldots, X_P^R\)

Then define the compaction: for any \(t \in \{1, \ldots, P\}\)
PTAS in Proper Interval Graphs
Re-structuration of optimal solutions

Let us built a compaction that yields a \((1 + \frac{4}{P})\)-approximation for any fixed \(P\).

Let \(X \subseteq B_i\) be a solution. We note \(X = X_L \cup X_R\). Set sizes are in lowercase.

- we divide \(X_L\) into \(P\) consecutive subsets of same size \(q_L \rightarrow X_1^L, \ldots, X_P^L\)
- we divide \(X_R\) into \(P\) consecutive subsets of same size \(q_R \rightarrow X_1^R, \ldots, X_P^R\)

Then define the compaction: for any \(t \in \{1, \ldots, P\}\)

- \(Y_t^L\) are the \(q_L\) rightmost intervals at the left of the rightmost interval of \(X_t^L\)
- \(Y_t^R\) are the \(q_R\) leftmost intervals at the right of the leftmost interval of \(X_t^R\)
PTAS in Proper Interval Graphs

Re-structuration of optimal solutions

What do we need to construct such a solution?
PTAS in Proper Interval Graphs

Re-structuration of optimal solutions

What do we need to construct such a solution?

- the leftmost interval of $X^L_t$ for $t \in \{1, \ldots, P\}$
- the rightmost interval of $X^R_t$ for $t \in \{1, \ldots, P\}$
- $x_R, x_L$ (plus remainders of divisions by $P$...)

$\Rightarrow 2P + O(1)$ variables ranging in $\{0, \ldots, n\}$
PTAS in Proper Interval Graphs

Sketch of proof of the $\left(1 + \frac{4}{P}\right)$ approximation ratio:
PTAS in Proper Interval Graphs

Sketch of proof of the \((1 + \frac{4}{P})\) approximation ratio:

- \(OPT = \binom{x_L}{2} + \binom{x_R}{2} + \sum_{t=1}^{a} \sum_{u=1}^{a} E(X^L_t, X^R_u)\)
- \(SOL = \binom{x_L}{2} + \binom{x_R}{2} + \sum_{t=1}^{a} \sum_{u=1}^{a} E(Y^L_t, Y^R_u)\)
Sketch of proof of the \((1 + \frac{4}{P})\) approximation ratio:

- **OPT** = \((\binom{x_L}{2}) + (\binom{x_R}{2}) + \sum_{t=1}^{a} \sum_{u=1}^{a} E(X^L_t, X^R_u)\)
- **SOL** = \((\binom{x_L}{2}) + (\binom{x_R}{2}) + \sum_{t=1}^{a} \sum_{u=1}^{a} E(Y^L_t, Y^R_u)\)

But:
PTAS in Proper Interval Graphs

Sketch of proof of the $(1 + \frac{4}{P})$ approximation ratio:

- $OPT = (x^L_2) + (x^R_2) + \sum_{t=1}^{a} \sum_{u=1}^{a} E(X^L_t, X^R_u)$
- $SOL = (x^L_2) + (x^R_2) + \sum_{t=1}^{a} \sum_{u=1}^{a} E(Y^L_t, Y^R_u)$

But:

- if some intervals of $Y^L_t$ overlap some intervals of $Y^R_u$

Then:

- all intervals of $X^L_{t+1}$ overlap all intervals of $\bigcup_{i=1}^{u-1} X^R_i$
Sketch of proof of the \( (1 + \frac{4}{p}) \) approximation ratio:

- \( \text{OPT} = \binom{x_L}{2} + \binom{x_R}{2} + \sum_{t=1}^{a} \sum_{u=1}^{a} E(X^L_t, X^R_u) \)
- \( \text{SOL} = \binom{Y_L}{2} + \binom{Y_R}{2} + \sum_{t=1}^{a} \sum_{u=1}^{a} E(Y^L_t, Y^R_u) \)

But:

- if some intervals of \( Y^L_t \) overlap some intervals of \( Y^R_u \)

Then:

- all intervals of \( X^L_{t+1} \) overlap all intervals of \( \bigcup_{i=1}^{u-1} X^R_i \)

Finally, we can prove that \( \frac{\text{SOL}}{\text{OPT}} \leq 1 + \frac{4}{p} \)
Conclusion:

Theorem

For any $P$, the previous algorithm outputs a $(1 + \frac{4}{P})$-approximation for the $k$-Sparsest Subgraph in Proper Interval graphs in $O(n^{O(P)})$
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:

- we sort intervals according to their right endpoints
Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:

- we sort intervals according to their right endpoints
- parameters of the dynamic programming:
  $k' \leftarrow k$, $C' \leftarrow C^*$, $s \leftarrow$ left endpoint of the leftmost interval
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:
- we sort intervals according to their right endpoints
- parameters of the dynamic programming:
  $k' \leftarrow k$, $C' \leftarrow C^*$, $s \leftarrow$ left endpoint of the leftmost interval
- given the parameters, we construct all subsets $T$ s.t.
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:

- we sort intervals according to their right endpoints
- parameters of the dynamic programming:
  $k' \leftarrow k$, $C' \leftarrow C^*$, $s \leftarrow$ left endpoint of the leftmost interval
- given the parameters, we construct all subsets $T$ s.t.
  
  (i) $T$ is connected
  (ii) $T$ starts after $s$ (i.e. to the right of $s$)
  (iii) $|T| \leq k'$ and $E(T) \leq C'$
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:

- we sort intervals according to their right endpoints
- parameters of the dynamic programming:
  $k' \leftarrow k$, $C' \leftarrow C^*$, $s \leftarrow$ left endpoint of the leftmost interval
- given the parameters, we construct all subsets $T$ s.t.
  (i) $T$ is connected
  (ii) $T$ starts after $s$ (i.e. to the right of $s$)
  (iii) $|T| \leq k'$ and $E(T) \leq C'$
- recursive call with:
  $k' \leftarrow k' - |T|
  C' \leftarrow C - E(T)
  s \leftarrow$ left endpoint of the rightmost interval after $T$
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:
- we sort intervals according to their right endpoints
- parameters of the dynamic programming:
  $k' \leftarrow k$, $C' \leftarrow C^*$, $s \leftarrow$ left endpoint of the leftmost interval
- given the parameters, we construct all subsets $T$ s.t.
  (i) $T$ is connected
  (ii) $T$ starts after $s$ (i.e. to the right of $s$)
  (iii) $|T| \leq k'$ and $E(T) \leq C'$
- recursive call with:
  $\triangleright k' \leftarrow k' - |T|$
  $\triangleright C' \leftarrow C - E(T)$
  $\triangleright s \leftarrow$ left endpoint of the rightmost interval after $T$
- $\Rightarrow$ at most $k.C^*.n$ different inputs
- what about the running time of one call?
FPT Algorithm in Interval Graphs

Given a set $\mathcal{I}$ of intervals, $k \leq |\mathcal{I}|$ and a cost $C^*$

Idea of the algorithm:

- we sort intervals according to their right endpoints
- parameters of the dynamic programming:
  $k' \leftarrow k$, $C' \leftarrow C^*$, $s \leftarrow$ left endpoint of the leftmost interval
- given the parameters, we construct all subsets $T$ s.t.
  (i) $T$ is connected
  (ii) $T$ starts after $s$ (i.e. to the right of $s$)
  (iii) $|T| \leq k'$ and $E(T) \leq C'$
- recursive call with:
  - $k' \leftarrow k' - |T|$
  - $C' \leftarrow C - E(T)$
  - $s \leftarrow$ left endpoint of the rightmost interval after $T$
- $\Rightarrow$ at most $k.C^*.n$ different inputs

what about the running time of one call?

Let $\Omega_s(C')$ be the set of all subsets satisfying (i), (ii) and (iii)
FPT Algorithm in Interval Graphs

- given the parameters, we construct all subsets $T$ s.t.
  - (i) $T$ is connected
  - (ii) $T$ starts after $s$ (i.e. to the right of $s$)
  - (iii) $|T| \leq k'$ and $E(T) \leq C'$

Let $\Omega_s(C')$ be the set of all subsets satisfying (i), (ii) and (iii)
FPT Algorithm in Interval Graphs

- given the parameters, we construct all subsets \( T \) s.t.
  
  \[
  \begin{align*}
  (i) & \quad T \text{ is connected} \\
  (ii) & \quad T \text{ starts after } s \text{ (i.e. to the right of } s) \\
  (iii) & \quad |T| \leq k' \text{ and } E(T) \leq C'
  \end{align*}
  \]

Let \( \Omega_s(C') \) be the set of all subsets satisfying (i), (ii) and (iii).

### Lemma

\[
\begin{array}{c}
\Omega_s(C') \\
T_1 \\
T_2 \\
\vdots \\
T_l \\
\end{array} \quad \xrightarrow{\text{cost}} \quad \begin{array}{c}
\Gamma_s(C') \\
T'_1 \\
T'_2 \\
\vdots \\
T'_l \\
\end{array}
\]

- can be enumerated in FPT time
- \(< y_1, \ldots, y_i, \ldots, y_t >\)
FPT Algorithm in Interval Graphs

Restructuration of a connected component $T$. We process intervals of $T$ using a cursor $S_i$. 

FPT Algorithm in Interval Graphs

Restructuration of a connected component $T$. We process intervals of $T$ using a cursor $S_i$.

$l_{current}$: leftmost interval of $T$ crossing $S_i$
FPT Algorithm in Interval Graphs

Restructuration of a connected component $T$. We process intervals of $T$ using a cursor $S_i$.

$l_{current}$: leftmost interval of $T$ crossing $S_i$
FPT Algorithm in Interval Graphs

Restructuration of a connected component $T$. We process intervals of $T$ using a cursor $S_i$.
FPT Algorithm in Interval Graphs

Restructuration of a connected component $T$. We process intervals of $T$ using a cursor $S_i$. 

![Diagram](image-url)
Restructuration of a connected component $T$. We process intervals of $T$ using a cursor $S_i$.

$\Rightarrow$ we only have to "guess" the number $y_i$ of intervals overlapping $I_{\text{current}}$
FPT Algorithm in Interval Graphs

Any connected component $T \in \Gamma_s(C)$ can be encoded by a vector $< y_1, ..., y_i, ..., y_t >$

We now bound the size of $\Gamma_s(C)$:

- $y_i = B \Rightarrow$ there exists a clique of size $B$ in the solution
Any connected component $T \in \Gamma_s(C)$ can be encoded by a vector $<y_1, \ldots, y_i, \ldots, y_t>$

We now bound the size of $\Gamma_s(C)$:

- $y_i = B \Rightarrow$ there exists a clique of size $B$ in the solution
  $\Rightarrow y_i \leq \sqrt{2C} + 2$
Any connected component $T \in \Gamma_s(C)$ can be encoded by a vector $<y_1, \ldots, y_i, \ldots, y_t>$.

We now bound the size of $\Gamma_s(C)$:

- $y_i = B \Rightarrow$ there exists a clique of size $B$ in the solution
  \[ \Rightarrow y_i \leq \sqrt{2C} + 2 \]
- each $T \in \Gamma_s(C)$ is a connected component, and each $S_i$ crosses a different interval of $T$
FPT Algorithm in Interval Graphs

Any connected component $T \in \Gamma_s(C)$ can be encoded by a vector $<y_1, ..., y_i, ..., y_t>$

We now bound the size of $\Gamma_s(C)$:

- $y_i = B \Rightarrow$ there exists a clique of size $B$ in the solution
  $\Rightarrow y_i \leq \sqrt{2C} + 2$

- each $T \in \Gamma_s(C)$ is a connected component, and each $S_i$ crosses a different interval of $T$
  $\Rightarrow t \leq C + 1$
Any connected component $T \in \Gamma_s(C)$ can be encoded by a vector $< y_1, \ldots, y_i, \ldots, y_t >$

We now bound the size of $\Gamma_s(C)$:

- $y_i = B \Rightarrow$ there exists a clique of size $B$ in the solution
  $\Rightarrow y_i \leq \sqrt{2C} + 2$
- each $T \in \Gamma_s(C)$ is a connected component, and each $S_i$ crosses a different interval of $T$
  $\Rightarrow t \leq C + 1$

Thus:

$$|\Gamma_s(C)| \leq (\sqrt{2C} + 2)^{C+1}$$

and each step of the dynamic programming runs in $FPT$ time.
FPT Algorithm in Interval Graphs

Any connected component \( T \in \Gamma_s(C) \) can be encoded by a vector \(< y_1, \ldots, y_i, \ldots, y_t >\).

We now bound the size of \( \Gamma_s(C) \):

- \( y_i = B \Rightarrow \) there exists a clique of size \( B \) in the solution
  \( \Rightarrow y_i \leq \sqrt{2C} + 2 \)
- each \( T \in \Gamma_s(C) \) is a connected component, and each \( S_i \) crosses a different interval of \( T \)
  \( \Rightarrow t \leq C + 1 \)

Thus:

\[
|\Gamma_s(C)| \leq (\sqrt{2C} + 2)^{C+1}
\]

and each step of the dynamic programming runs in FPT time.

**Theorem**

Sparsest \( k \)-Subgraph in Interval Graphs is FPT parameterized by the cost of the solution.
Contents

1 Introduction

2 PTAS in Proper Interval Graphs

3 FPT Algorithm in Interval Graphs

4 Open Problems and Future Work
Open problems and Future Work

A diagram showing a hierarchy of graph classes, starting with Perfect graphs at the top, followed by Bipartite, Chordal, Tree, Interval, Split, and Proper Int. graphs at the bottom.
Open problems and Future Work

Perfect
NP-hard

Bipartite

Chordal

Tree

Interval

Split

Proper Int.
Open problems and Future Work

- **Perfect NP-hard**
- **Bipartite**
- **Chordal Tree Poly**
- **Interval Proper Int.**
- **Split Poly**
Open problems and Future Work

- Perfect
  - NP-hard
- Bipartite
  - NP-hard
- Chordal
- Tree
  - Poly
- Interval
- Split
  - Poly
- Proper
  - Int.
Open problems and Future Work

- Perfect: NP-hard
- Bipartite: NP-hard
- Chordal: NP-hard
- Tree: Poly
- Interval
- Split: Poly
- Proper Int.
Open problems and Future Work

- Perfect
  - NP-hard

- Bipartite
  - NP-hard

- Chordal
  - NP-hard

- Tree
  - Poly

- Interval
  - FPT
  - 2-apx

- Split
  - Poly

- Proper
  - Int.
Open problems and Future Work

![Diagram showing graph classes and their complexity]

- **Perfect**
  - NP-hard
- **Bipartite**
  - NP-hard
- **Chordal**
  - NP-hard
- **Tree**
  - Poly
- **Interval**
  - FPT
  - 2-apx
- **Split**
  - Poly
- **Proper Int.**
Open problems and Future Work

- $NP$-h/$Poly$ in (Proper) Interval graphs?
- Extend FPT and/or approximation results to Chordal graphs?
- Polynomial kernel in Interval graphs?
Thank you for your attention!