On Finding a Sparse Subgraph in Subclasses of Perfect Graphs

Rémi Watrigant, Marin Bougeret and Rodolphe Giroudeau

LIRMM, Montpellier, France

1ères journées du GT CoA - Complexité et Algorithmes
Contents

1 Introduction

2 PTAS in Proper Interval Graphs

3 Open Problems and Future Work
## $k$-Sparsest Subgraph Problem ($k$-SS)

**Input:** a 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 $S$)

### Watrigant, Bougeret, Giroudeau

On Finding a Sparse Subgraph in Subclasses of Perfect Graphs
**$k$-Sparsest Subgraph Problem ($k$-SS)**

**Input:** a 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
  - $k$-SS $NP$-hard in general graphs (+ no FPT, approximation algorithm)
**$k$-Sparsest Subgraph Problem ($k$-SS)**

**Input:** a 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
  - $k$-SS NP-hard in general graphs (+ no FPT, approximation algorithm)
- But INDEPENDENT SET is polynomial in perfect graphs!
- what about $k$-SS ?
**$k$-Sparsest Subgraph Problem ($k$-SS)**

**Input:** a 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**
  - $k$-SS **NP**-hard in general graphs (+ no FPT, approximation algorithm)
  - But **INDEPENDENT SET** is polynomial in perfect graphs!
  - what about $k$-SS?

- maximization version: $k$-Densest Subgraph:
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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
  - $k$-SS NP-hard in general graphs (+ no FPT, approximation algorithm)
  - But INDEPENDENT SET is polynomial in perfect graphs!
    - what about $k$-SS?

- maximization version: $k$-Densest Subgraph:
  - NP-hard in chordal graphs [Corneil and Perl, 1984]
  - unknown in (proper) interval graphs (longstanding open problem) [CP84]
  - PTAS in interval graphs [Nonner, 2011]
  - constant approximation algorithm in chordal graphs [Liazi et al., 2008]
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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**  
  $\Rightarrow$ $k$-SS $NP$-hard in general graphs (+ no FPT, approximation algorithm)  
  But **INDEPENDENT SET** is polynomial in perfect graphs!  
  what about $k$-SS ?

- maximization version: $k$-Densest Subgraph:  
  - $NP$-hard in chordal graphs [Corneil and Perl, 1984]  
  - unknown in (proper) interval graphs (longstanding open problem) [CP84]  
  - PTAS in interval graphs [Nonner, 2011]  
  - constant approximation algorithm in chordal graphs [Liazi et al., 2008]

- $k$-SS polynomial in:  
  - split graphs (obvious)  
  - bounded cliquewidth ($\Rightarrow$ trees, cographs, ...) [Boersma et al., 2012]

Recall that $\text{proper interval} \subset \text{interval} \subset \text{chordal} \subset \text{perfect}$

$\text{split} \subset \text{chordal} \subset \text{perfect}$
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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 $S$)

Our results [W., Bougeret, Giroudeau, 2012]: 
**$k$-Sparsest Subgraph Problem ($k$-SS)**

**Input:** a 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 $S$)

Our results [W., Bougeret, Giroudeau, 2012]:
- FPT algorithm in interval graphs (parameterized by the cost of the solution)
- PTAS in proper interval graphs
- $NP$-hardness in chordal graphs
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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 \( S \))

Our results [W., Bougeret, Giroudeau, 2012]:
- FPT algorithm in interval graphs (parameterized by the cost of the solution)
- PTAS in proper interval graphs \( \leftarrow \) This talk
- \( NP \)-hardness in chordal graphs
**k-Sparkest Subgraph Problem (k-SS)**

**Input:** a 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 $S$)

Our results [W., Bougeret, Giroudeau, 2012]:
- FPT algorithm in interval graphs (parameterized by the cost of the solution)
- PTAS in proper interval graphs ← This talk
- NP-hardness in chordal graphs

**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 $< (1 + \epsilon)OPT$
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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 $S$)

Our results [W., Bougeret, Giroudeau, 2012]:
- FPT algorithm in interval graphs (parameterized by the cost of the solution)
- PTAS in proper interval graphs ← This talk
- $NP$-hardness in chordal graphs

Interval graph = intersection graph of intervals in the real line.
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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 $S$)

---

Our results [W., Bougeret, Giroudeau, 2012]:
- FPT algorithm in interval graphs (parameterized by the cost of the solution)
- PTAS in proper interval graphs ← This talk
- $NP$-hardness in chordal graphs

Interval graph = intersection graph of intervals in the real line.
**k-Sparsest Subgraph Problem (k-SS)**

**Input:** a 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 $S$)

Our results [W., Bougeret, Giroudeau, 2012]:
- FPT algorithm in interval graphs (parameterized by the cost of the solution)
- PTAS in proper interval graphs ← This talk
- $NP$-hardness in chordal graphs

Interval graph = intersection graph of intervals in the real line.

Proper interval graph = no interval contains properly another one = unit interval graphs
Contents

1 Introduction

2 PTAS in Proper Interval Graphs

3 Open Problems and Future Work
PTAS in Proper Interval Graphs

Idea of the algorithm:
PTAS in Proper Interval Graphs

Idea of the algorithm:
- sort intervals according to their right (or left) endpoints
PTAS in Proper Interval Graphs

Idea of the algorithm:
- sort intervals according to their right (or left) endpoints
- greedy decomposition of the graph into a path of separators/cliques
PTAS in Proper Interval Graphs

Idea of the algorithm:

- sort intervals according to their right (or left) endpoints
- greedy decomposition of the graph into a path of separators/cliques
- re-structuration of an optimal solution into a near optimal solution such that all near optimal solutions can be enumerated in polynomial time
Idea of the algorithm:

- sort intervals according to their right (or left) endpoints
- greedy decomposition of the graph into a path of separators/cliques
- re-structuring 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:

\[ R_1 \]

\[ I_{m_1} \]

\[ B_1 \]
PTAS in Proper Interval Graphs

The decomposition:

\[ R_1 \]

\[ I_{m_1} \]

\[ B_1 \]
PTAS in Proper Interval Graphs

The decomposition:
PTAS in Proper Interval Graphs

The decomposition:
PTAS in Proper Interval Graphs

Restructuration of a solution: compaction $S \mapsto \text{comp}(S)$
PTAS in Proper Interval Graphs

Restructuration of a solution: compaction \( S \rightarrow \text{comp}(S) \)

Remark

If for each block, the compaction produces a \( \rho \)-approximated solution, then it is a \( \rho \)-approximated solution for the whole graph.
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 us build 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 \(q_L \rightarrow X_{1}^{L}, ..., X_{P}^{L}\)
- we divide \(X_R\) into \(P\) consecutive subsets of same size \(q_R \rightarrow X_{1}^{R}, ..., X_{P}^{R}\)

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

Let us build 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 \(q_L \rightarrow X^L_1, \ldots, X^L_P\)
- we divide \(X_R\) into \(P\) consecutive subsets of same size \(q_R \rightarrow X^R_1, \ldots, X^R_P\)

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

- \(Y^L_t\) are the \(q_L\) rightmost intervals of the \(t^{th}\) left block.
- \(Y^R_t\) are the \(q_R\) leftmost intervals of the \(t^{th}\) right block.
PTAS in Proper Interval Graphs
Re-structuration of optimal solutions

What do we need to construct such a solution?
What do we need to construct such a solution?

- the leftmost interval of the $t^{th}$ left block for $t \in \{1, \ldots, P\}$
- the rightmost interval of the $t^{th}$ right block for $t \in \{1, \ldots, P\}$
- $x_R, x_L$ (plus remainders of divisions by $P$...)

$\Rightarrow 2P + O(1)$ variables ranging in $\{0, \ldots, n\}$
Sketch of proof of the \((1 + \frac{4}{P})\) approximation ratio:
Sketch of proof of the \((1 + \frac{4}{P})\) approximation ratio:

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

- **SOL** = \((\frac{x_L}{2}) + (\frac{x_R}{2}) + \sum_{t=1}^{a} \sum_{u=1}^{a} E(Y_t^L, Y_u^R)\)
- **OPT** = \((\frac{x_L}{2}) + (\frac{x_R}{2}) + \sum_{t=1}^{a} \sum_{u=1}^{a} E(X_t^L, X_u^R)\)
PTAS in Proper Interval Graphs

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

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

But:
PTAS in Proper Interval Graphs

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

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

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 \(\bigcup_{i=1}^{u-1} X^R_i\)
PTAS in Proper Interval Graphs

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

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

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 \(\frac{SOL}{OPT} \leq 1 + \frac{4}{P}\)
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 $O(n^{O(P)})$.
Contents

1 Introduction

2 PTAS in Proper Interval Graphs

3 Open Problems and Future Work
Open problems and Future Work

Complexity of \( k \)-Sparsest Subgraph:

- Chordal
- Bipartite
- Tree
- Interval
- Split
- Proper Int.

Perfect
Open problems and Future Work

Complexity of $k$-Sparsest Subgraph:

- Perfect
  - NP-hard

- Bipartite

- Chordal

- Tree

- Interval

- Split

- Proper Int.
Open problems and Future Work

Complexity of $k$-Sparsest Subgraph:
Open problems and Future Work

Complexity of $k$-Sparsest Subgraph:

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

Complexity of $k$-Sparsest Subgraph:

- **Perfect**
  - NP-hard

- **Bipartite**
  - NP-hard?

- **Chordal**
  - Tree
    - Poly

- **Interval**
  - Split
    - Poly
  - Proper
    - Int.
    - PTAS
Open problems and Future Work

Complexity of $k$-Sparsest Subgraph:

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

Complexity of $k$-Sparsest Subgraph:

- Chordal: NP-hard
- Split: Poly
- Interval: FPT
- Proper Int.: PTAS
- Tree: Poly
- Bipartite: NP-hard?

Perfect: NP-hard
Future work/open questions:

- **k-sparsest subgraph:**
  - extend FPT and/or approximation results to Chordal graphs
  - NP-h/Poly on Interval, Proper interval?

- **k-densest subgraph:**
  - (NP-h/Poly on Interval, Proper interval)
  - FPT/W[1]-hardness on Chordal graphs?
Merci de votre attention !