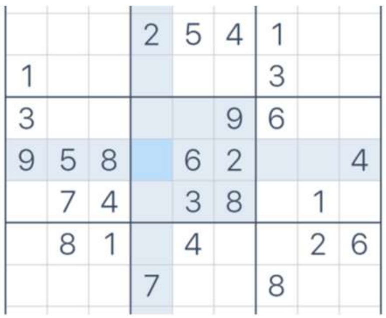
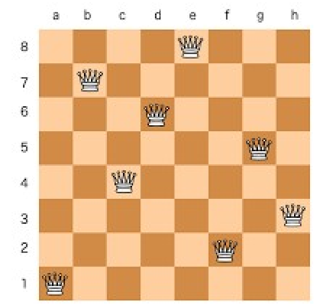
Parameterized Approximations and CSPs

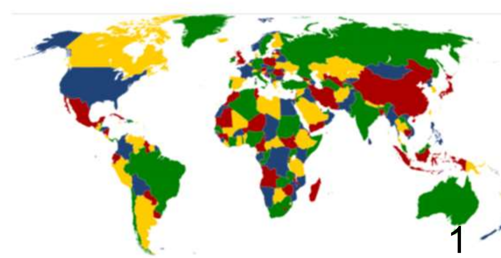




Hadas Shachnai

Parameterized Complexity of Computational Reasoning (PCCR)

July 2022



Technion
Israel Institute of Technology

Talk Outline

- Parameterization and Approximation
- FPT approximations for W-hard Problems
- FPT approximations for CSPs
- Summary and Open Problems

Parameterized Complexity

Main idea: Instead of expressing the running time as a function $T(n)$ of n, we express it as a function $T(n, k)$ of the input size, n, and some parameter k of the input.

We want to be efficient for inputs where k is small

Possible choices for the parameter k:

- The size k of the solution we are looking for.
- The maximum degree of the input graph.
- The diameter of the input graph.
- Maximum length of a clause in the input Boolean formula.
- …

Fixed-parameter tractability

A parameterized problem is **fixed-parameter tractable (FPT)** if for an input of size n and parameter value k there is an $f(k)n^c$ time algorithm for some constant c.

Example: Vertex Cover parameterized by the solution size k is FPT: can be solved in time $O(1.273^k + kn)$

W-hardness

Negative evidence similar to NP-completeness. If a problem is **W-hard,** then it is unlikely to be in FPT.

$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots$

Some W-hard problems:

- Finding a clique/independent set of size k.
- Finding a dominating set of size k.
- Finding k pairwise disjoint sets.
- ...

Talk Outline

- Parameterization and Approximation
- FPT approximations for W-hard Problems
- FPT approximations for CSPs
- Summary and Open Problems

Approximation using Problem Parameters

Idea: Instead of finding an approximation algorithm with running time $n^{O(1)}$, find an approximation algorithm with running time $f(k) \cdot n^{O(1)}$, where k is some parameter of the optimization problem instance.

Example: [Böckenhauer et al., 2007] Metric TSP with Deadlines is the standard metric TSP problem, extended with a set D of deadline nodes. The salesperson must reach $v \in D$ within time at most $d(v)$.

Let $|D|$ be the parameter.

Metric TSP with Deadlines

- Approximation: The problem has no constant factor approximation (unless P = NP).

- Parameterization: The problem parameterized by $|D|$ is not in FPT

- Approximation + parameterization: A 2.5-approximation can be found in time $O(n^3 + |D|! \cdot |D|)$

Approximation schemes

Polynomial-time approximation scheme (PTAS):

Input: Instance x, $\varepsilon > 0$

Output: $(1+\varepsilon)$-approximate solution

- **PTAS:** running time is $|x|^{f(1/\varepsilon)}$
- **EPTAS:** running time is $f(1/\varepsilon) \cdot |x|^{O(1)}$
- **FPTAS:** running time is $(1/\varepsilon)^{O(1)} \cdot |x|^{O(1)}$

Connections with parameterized complexity:

- Methodological similarities between EPTAS and FPT design.
- Lower bounds on the efficiency of approximation schemes.

FPT-approximation schemes

FPT-AS:

Input: Instance (x, k), $\varepsilon > 0$

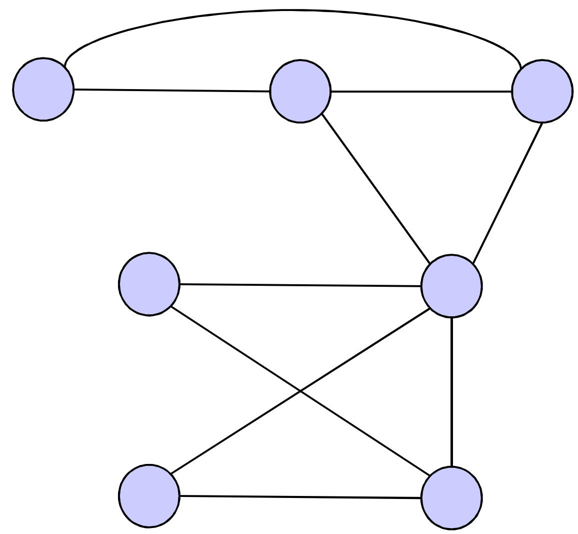
Output: $(1+ \varepsilon)$-approximate solution

Running time: $f(\varepsilon ,k) \cdot |x|^{O(1)}$, for some computable function f.

This is a parameterized version of EPTAS.

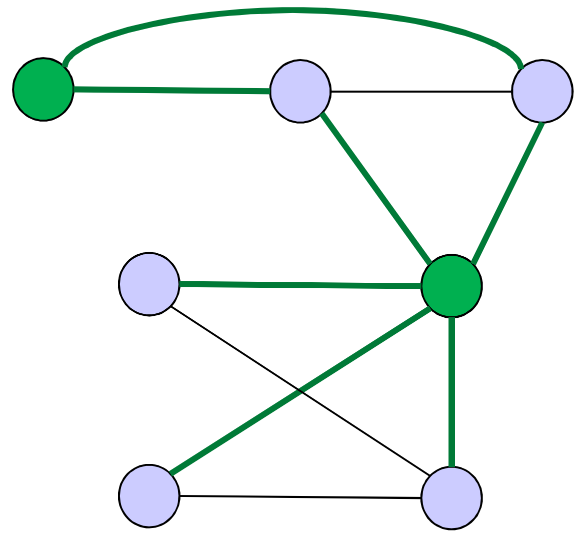
Partial Vertex Cover

Select k vertices, maximizing the number of covered edges.



Partial Vertex Cover

Select k vertices, maximizing the number of edges covered.



Partial Vertex Cover

- Approximation: The problem has a constant factor approximation, but has no PTAS (unless P = NP) [Hochbaum and Pathria 1998, Petrank 1994]

- Parameterization: The problem parameterized by k is W[1]-hard [Guo, Niedermeier and Wernicke 2005]

- Approximation + parameterization: A $(1 + \varepsilon)$-approximation can be found in time $f(k, \varepsilon) \cdot n^{O(1)}$ [Marx 2008]

Partial Vertex Cover: FPT-AS

Theorem: Partial Vertex Cover admits an FPT-AS with parameter k, the number of vertices in the solution.

Proof: Sort the vertices in non-increasing order by degrees, and let $D = 2\binom{k}{2}/\varepsilon$.

- Case 1: $d(v_1) > D$.

The algorithm outputs $S=\{v_1, \ldots, v_k\}$.

Partial Vertex Cover: FPT-AS

- Case 1: $d(v_1) > D$.

- The algorithm outputs $S=\{v_1,\dots v_k\}$. These k vertices cover at least $\sum_{i=1}^{k} d(v_i) - \binom{k}{2}$ edges.

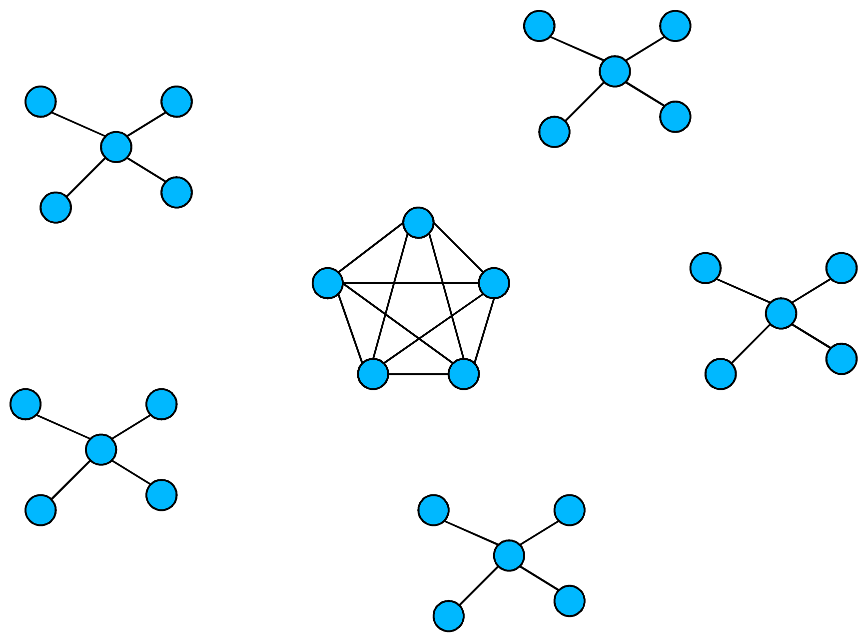
- An optimal solution covers at most $\sum_{i=1}^{k} d(v_i)$ edges.

Partial Vertex Cover: FPT-AS

- Case 1: $d(v_1) \geq D$.

The algorithm covers at least $\sum_{i=1}^{k} d(v_i) - \binom{k}{2}$ edges. An optimal solution covers at most $\sum_{i=1}^{k} d(v_i)$ edges.

Example:



Assume k=5.

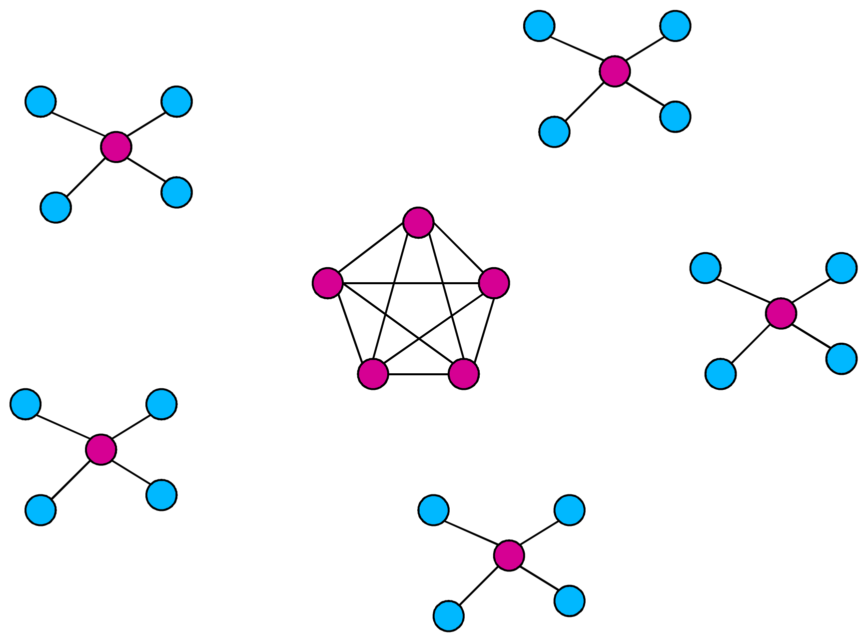
There are 10 vertices with degree 4

Partial Vertex Cover: FPT-AS

- Case 1: $d(v_1) \geq D$.

The algorithm covers at least $\sum_{i=1}^{k} d(v_i) - \binom{k}{2}$ edges. An optimal solution covers at most $\sum_{i=1}^{k} d(v_i)$ edges.

Example:



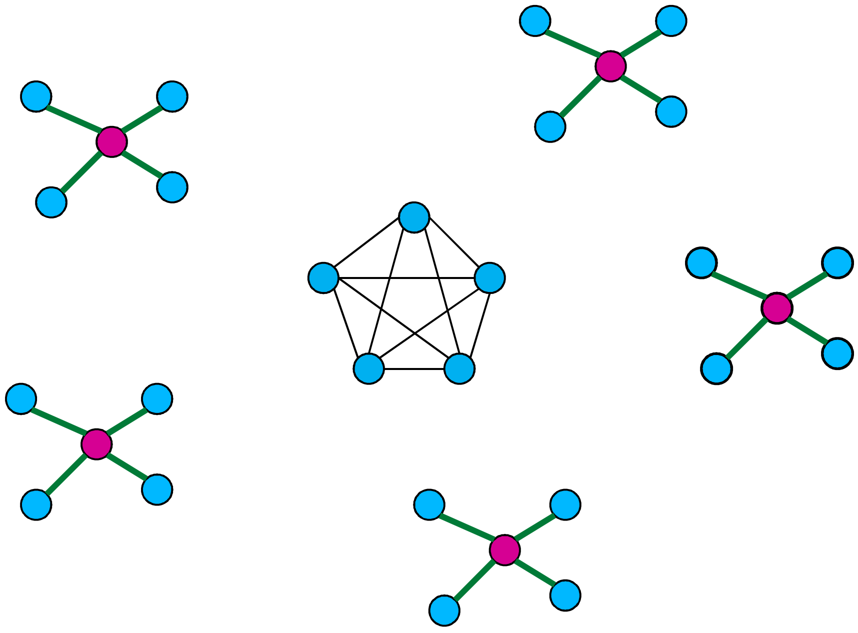
Assume $k=5$.
There are 10 vertices with degree 4

Partial Vertex Cover: FPT-AS

- Case 1: $d(v_1) \geq D$.

The algorithm covers at least $\sum_{i=1}^{k} d(v_i) - \binom{k}{2}$ edges. An optimal solution covers at most $\sum_{i=1}^{k} d(v_i)$ edges.

Example:



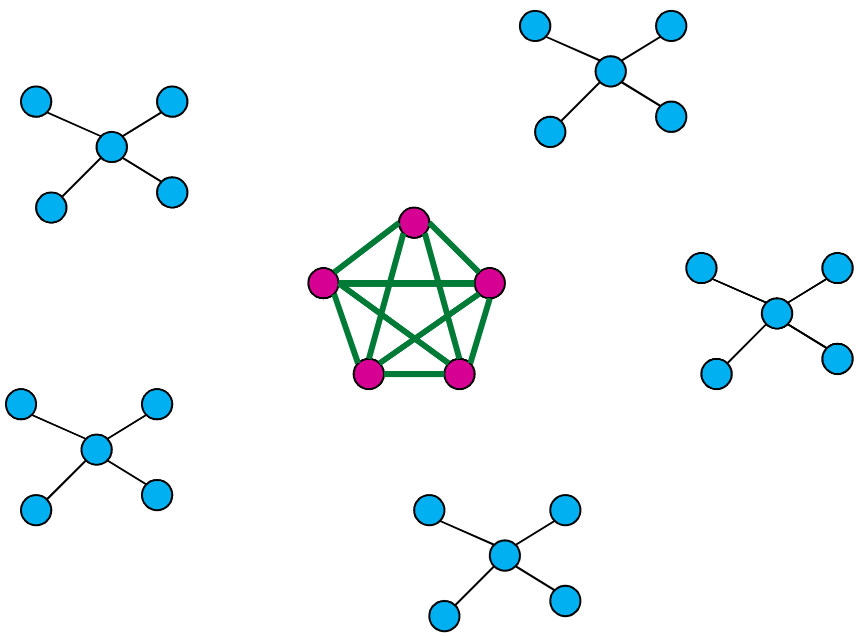
If the algorithm selects the 5 star centers: 20 edges are covered.

Partial Vertex Cover: FPT-AS

- Case 1: $d(v_1) \geq D$.

The algorithm covers at least $\sum_{i=1}^{k} d(v_i) - \binom{k}{2}$ edges. An optimal solution covers at most $\sum_{i=1}^{k} d(v_i)$ edges.

Example:



If the algorithm selects the clique vertices: 10 edges are covered.

Partial Vertex Cover: FPT-AS

- Case 1: $d(v_1) > D$.

The value of the solution is at least

$$\frac{\sum_{i=1}^{k} d(v_i) - \binom{k}{2}}{\sum_{i=1}^{k} d(v_i)} \geq 1 - \frac{\binom{k}{2}}{D} \geq \frac{1}{1+\varepsilon}$$

times the optimum.

Partial Vertex Cover: FPT-AS

- Case 2: $d(v_1) \leq D$.

Then the optimum value is at most kD.

Use Color Coding to find OPT:

for all $1 \leq \ell \leq kD$, check if it is possible to cover at least ℓ edges with k vertices.

Running time dominated by the Color Coding subroutine.
Overall, $f(k, \varepsilon) \cdot poly(n)$.

FPT approximations for W-hard Problems: Summary

- A straightforward combination of approximation and FPT.

$f(k) \cdot n^{O(1)}$ or $f(k, \varepsilon) \cdot n^{O(1)}$ time approximation algorithms, where k is some parameter of the optimization problem.

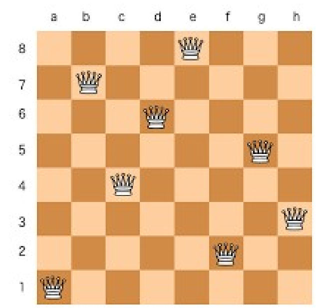
- We can obtain constant factor approximation or approximation schemes for problems where polynomial-time algorithms cannot.

Talk Outline

- Parameterization and Approximation
- FPT approximations for W-hard Problems
- FPT approximations for CSPs
- Summary and Open Problems

Constraint Satisfaction Problems

- An instance of constraint satisfaction problem (CSP) is a set of constraints over a set of variables that can take values in certain domain.



- Question: Can we assign values to the variables such that all constraints are satisfied?
- Boolean CSP: Ψ is a set $\{C_1, \ldots, C_m\}$ of m constraints over n variables $X(\Psi) = \{x_1, \ldots, x_n\}$ and their negations (literals).
- Each constraint C_j maps a set of literals to $\{0,1\}$; a literal can take the values '0' or '1'.

Constraint Satisfaction Problems (Cont'd)

- Various types of constraints (e.g., OR, AND, MAJORITY,..).

Example: $(x_1 \vee \neg x_2 \vee \neg x_4 \vee x_5) \wedge (x_2 \vee \neg x_3) \wedge x_4$ (CNF-SAT, or SAT)

- In the optimization version of CSPs (Max-CSPs) we seek an assignment to the variables which satisfies a maximal number of constraints

- The vast majority of interesting CSPs are NP-hard.

- Two common approaches for solving CSPs (and Max-CSPs): parameterization and approximation.

Solving CSPs

Approximations

- There are polynomial-time approximations for Max-CSPs [Alon, Fernandez de la Vega, Kannan and Karpinski 2002] [Trevisan 2004] [Austrin and Khot 2013] [Khot and Saket 2015]
- Many Max-CSPs are APX-hard [Creignou 1995, Elbassioni] [Raman, Ray and Sitters 2009] even when the decision version is in P (e.g., 2-SAT)

Parameterized CSPs

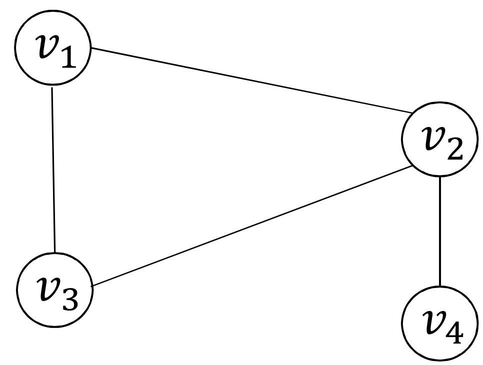
- Wide literature on parameterized versions of CSPs [Grohe 2006] [Samer and Szeider 2010] [Szeider 2011] [Gaspers and Szeider 2011] [Pichler, Rümmele, Szeider and Woltran 2014] [Gaspers and Szeider 2014] [Ganian, Ramanujan and Szeider 2017] [Bannach, Fleischmann and Skambath 2022]
- The parameters often relate to the structure of the input instance

CSP Graphs and Structural Parameters

Example: Max-SAT Ψ

$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_4)$

The primal graph G_Ψ^*:

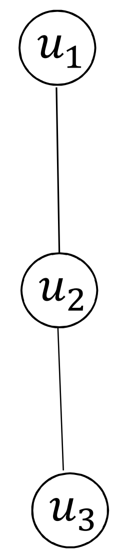


CSP Graphs and Structural Parameters

Example: Max-SAT Ψ

$(x_1 \lor \neg x_2 \lor x_3) \land (x_2 \lor \neg x_4) \land (x_4 \lor x_6)$

The dual graph G^*_Ψ:

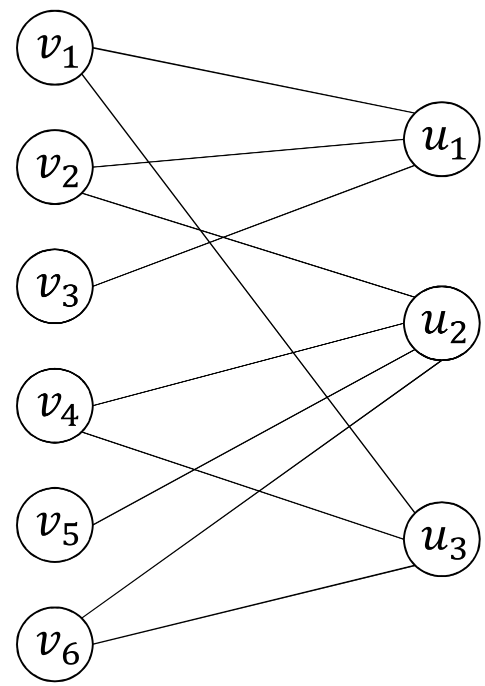


CSP Graphs and Structural Parameters

Example: Max-SAT Ψ

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_4 \vee \neg\, x_5 \vee x_6) \wedge (\neg x_1 \vee x_4 \ \vee \neg x_6)$$

The incidence graph G^*_Ψ:



There are other types of graphs for a given CSP instance (e.g., constraint hypergraph).

Common Basic Parameters

- Number of variables
- Number of constraints
- Largest arity (size of a constraint scope)
- Largest overlap between two constraints scopes
- Largest difference between constraints scopes
- ….

Some Common Structural Parameters

- Maximal degree of the instance graph (primal/dual/incidence graph, constraint hypergraph)
- Treewidth: The notion of treewidth measures how close is a graph to being a tree.

A tree decomposition of $G = (V, E)$ is a tree T consisting of nodes $X_1, \ldots, X_n$, where X_i is a subset of vertices. The tree satisfies the following properties:

i. $\cup_i X_i = V$

ii. If v is contained in both X_i and X_j then v is contained in any node on the single path in T between X_i and X_j

iii. For any edge (u, v) there is a node that contains both u and v.

Structural Parameters (Cont'd)

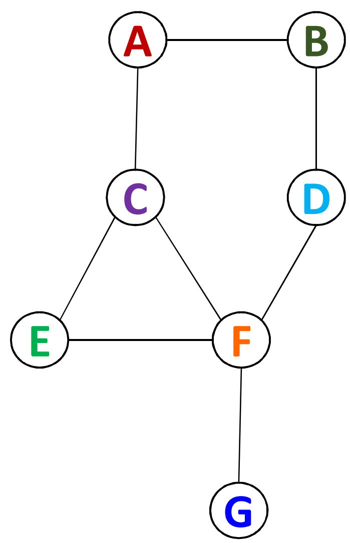
- Treewidth:

The width of a tree decomposition is the the cardinality of the largest node in T minus 1.

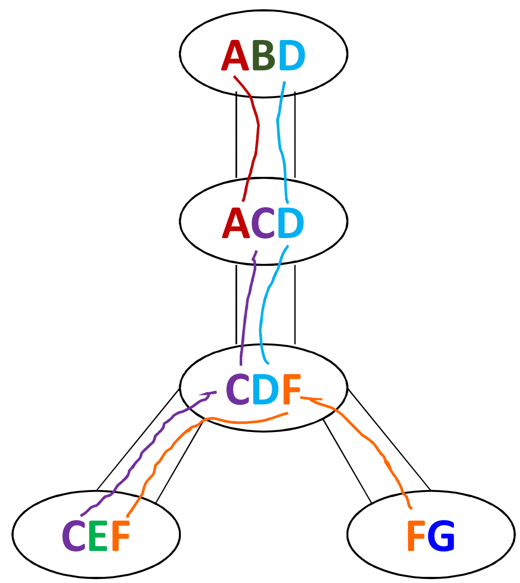
The treewidth of the graph G, $tw(G)$, is the minimum width of any tree decomposition of G.

Structural Parameters: Treewidth

Graph G



Tree decomposition of G



tw(G) =2

Structural Parameters (Cont'd)

- Clique-width is the minimal number of labels required to construct a graph G using in each step one of the operations:

i. Creation of a new vertex with a label i

ii. Disjoint union of two labeled graphs

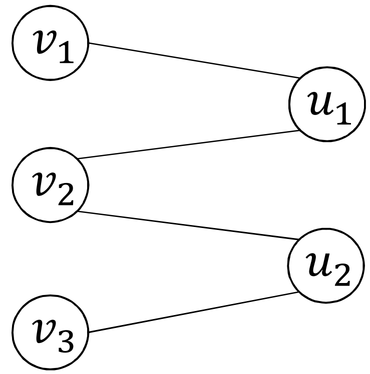
iii. Joining by an edge each vertex labeled with i with each vertex labeled with j, for some $i \neq j$

iv. Renaming a vertex with label i to label j

Structural Parameters: Clique-width

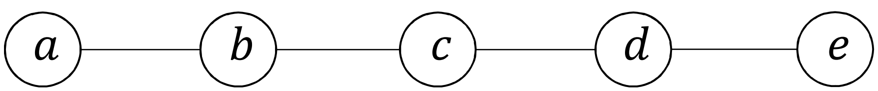
Max-SAT instance: $(x_1 \lor \neg x_2) \land (x_2 \lor \neg x_3)$

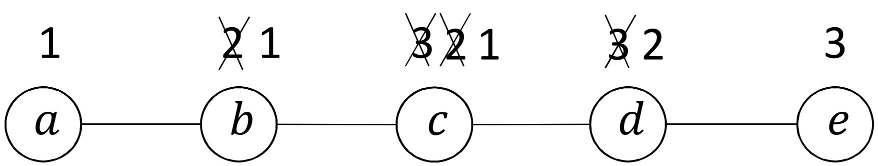
The incidence graph G^*_Ψ:



$\text{cw}(G^*_\Psi)=3$

Finding the clique-width of a path graph G:





Solving Hard Instances of Max-SAT

- Approximation: The problem has constant factor approximations, but does not admit a polynomial-time approximation scheme (unless P = NP) [Krentel 1986] [Cohen, Cooper and Jeavons 2004]

- Parameterization by clique-width: The problem is W[1]-hard [Ordyniak, Paulusma and Szeider 2013]

- Approximation + parameterization: A $(1 + \varepsilon)$-approximation can be found in time $f(k, \varepsilon) \cdot n^{O(1)}$ for Max-SAT parameterized by clique-width [Dell, Kim, Lampis, Mitsou and Mömke 2017]

Example: FPT-AS for Max-SAT

Use as parameter the clique-width

Main ideas:

- Distinguish between small, medium and large clauses

$$\overbrace{\underbrace{C_1, C_2, \dots, C_{|S|}}_{|C_i| \le d}}^{\text{small clauses}} \quad \overbrace{\underbrace{C_{|S|+1}, C_{|S|+2}, \dots, C_{|S|+|M|}}_{d < |C_i| \le D}}^{\text{medium clauses}} \quad \overbrace{\underbrace{C_{m-|L|+1}, \dots, C_m}_{|C_i| > D}}^{\text{large clauses}}$$

FPT-AS for Max-SAT (Cont'd)

- Given Ψ and $\varepsilon > 0$, we can choose $d \geq 1$ and $D = \varepsilon^{-4}\mathrm{d}$ such that $|M| \leq \varepsilon m$, where $m = |\Psi|$.
- Omit the medium clauses to obtain a well-separated instance

$$\overbrace{\underbrace{C_1, C_2, \dots, C_{|S|}}_{|C_i| \leq d}}^{\text{small clauses}} \quad \cancel{\overbrace{\underbrace{C_{|S|+1}, C_{|S|+2}, \dots, C_{|S|+|M|}}_{d < |C_i| \leq D}}^{\text{medium clauses}}} \quad \overbrace{\underbrace{C_{m-|L|+1}, \dots, C_m}_{|C_i| > D}}^{\text{large clauses}}$$

- Consider several cases for the remaining instance

FPT-AS for Max-SAT: Case 1

$|S| \leq \frac{\varepsilon m}{4}$: then ignore S. Return a random assignment SOL for the clauses in L.

~~small clauses~~

$$\cancel{\underbrace{C_1, C_2, \ldots, C_{|S|}}_{|C_i| \leq d}}$$

~~medium clauses~~

$$\cancel{\underbrace{C_{|S|+1}, C_{|S|+2}, \ldots, C_{|S|+|M|}}_{d < |C_i| \leq D}}$$

large clauses

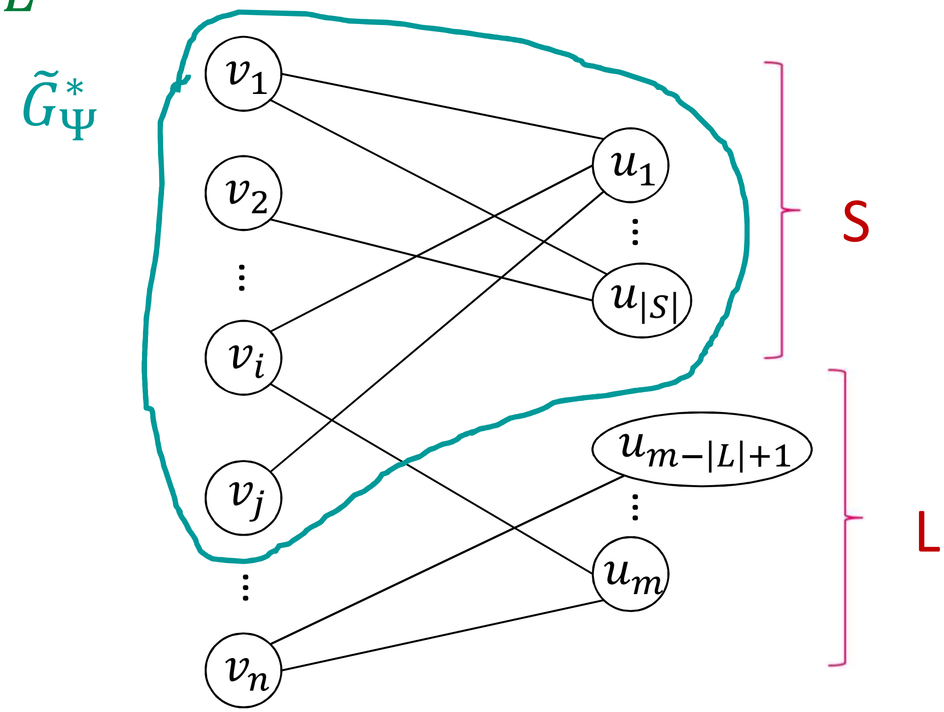
$$\underbrace{C_{m-|L|+1}, \ldots, C_m}_{|C_i| > D}$$

Almost all clauses in Ψ are in L

FPT-AS for Max-SAT: Case 2

- $|L| \leq \frac{\varepsilon m}{4}$: Then ignore L

Solve for S using an FPT algorithm for Max-SAT with bounded treewidth



$\tilde{G}_{\Psi}^{*}$ has no large bicliques

FPT-AS for Max-SAT: Case 3

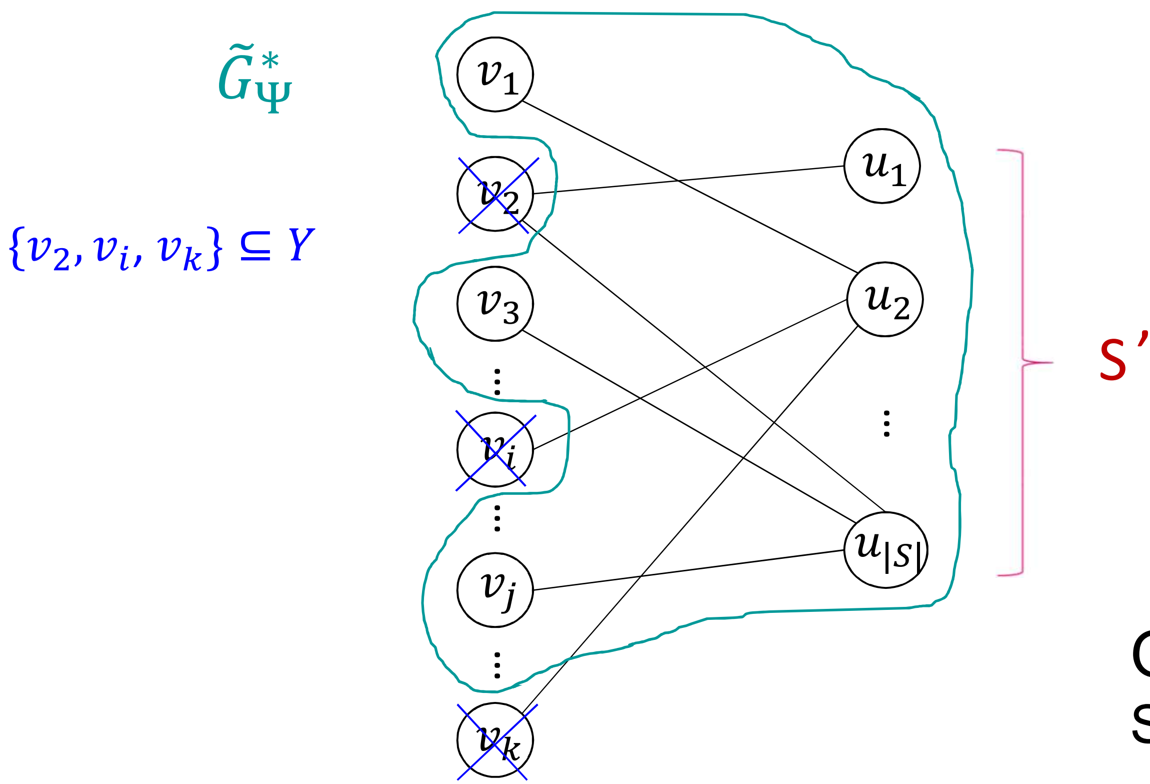
- If $|L| > \frac{\varepsilon m}{4}$ and $|S| > \frac{\varepsilon m}{4}$, find a good set of variables Y, i.e.,

  (i) There are at most $\frac{\varepsilon m}{4}$ small clauses which contain variables in Y.

  (ii) There are at most ε^2m large clauses that contain less than $1/\varepsilon$ variables in Y.

- Ignore the variables of Y in the small clauses.
- Compute a random assignment SOL1 for the variables in Y

FPT-AS for Max-SAT: Case 3



Output
SOL= SOL1 ∪ SOL2

$\tilde{G}^*_\Psi$ has no large bicliques

- Solve for S' using an FPT algorithm for Max-SAT with bounded treewidth and return SOL2

Analysis

- Assume w.l.o.g that $\varepsilon < \frac{1}{8}$, and show that at least

$(1-\varepsilon)OPT$ clauses are satisfied in each case.

Case 1: Ignore S and return a random assignment SOL for the clauses in L

- At most $\varepsilon^2 m$ small clauses are unsatisfied

- At most $\left(\frac{1}{2}\right)^D < \frac{\varepsilon}{4}$ of the large clauses are unsatisfied (in expectation)

- Therefore, at most

$$\varepsilon^2 m + \frac{\varepsilon m}{4} + \frac{\varepsilon m}{4} \le \varepsilon m$$

clauses are unsatisfied in expectation.

Analysis (Cont'd)

Case 2: Ignore L and solve for S using an FPT algorithm for Max-SAT with bounded treewidth

Lemma [Gurski and Wanke, 2000]: Let G be a graph of clique-width k, such that G has no subgraph $K_{n,n}$. Then G has treewidth at most $3k(n-1)-1$.

- The number of unsatisfied large or medium clauses is at most

$$\varepsilon^2 m + \frac{\varepsilon m}{4} \leq \frac{\varepsilon m}{2}$$

- Let OPT_S be the number of satisfied clauses in an optimal solution for S, then

$$OPT \ - \ OPT_S \leq \frac{\varepsilon m}{2} \leq \varepsilon \, OPT$$

Analysis (Cont'd)

Case 3: Find a good set of variables Y.

- Compute a random assignment SOL1 for the variables in Y.
- Compute SOL2, a solution for S without variables in Y, using an FPT algorithm for Max-SAT with bounded treewidth. Output SOL= SOL1U SOL2

- It can be shown that a good set Y exists.

Intuition:

- If L has more than $\frac{\varepsilon m}{4}$ clauses, there must be a variable that appears in 'many' large clauses and only in 'few' small clauses
- Use this property to construct Y iteratively

Analysis (Cont'd)

Case 3: Find a good set of variables Y.

- Compute a random assignment SOL1 for the variables in Y.
- Compute SOL2, a solution for S without variables in Y, using an FPT algorithm for Max-SAT with bounded treewidth. Output SOL= SOL1U SOL2

- The incidence graph for S' (S after the variables in Y are omitted) has bounded treewidth.

- By similar arguments,

$$OPT - OPT_{S'} \leq \frac{\varepsilon m}{2} \leq \varepsilon\, OPT$$

in expectation.

FPT-AS for Max-SAT: Summary

Theorem: Given $\varepsilon > 0$ and a Max-SAT instance Ψ with m clauses and n variables, and clique-width cw, there is a randomized algorithm that outputs a truth assignment that satisfies at least $(1-\varepsilon)OPT$ clauses in time

$f(\varepsilon, cw) poly(n+m)$.

More Results for Parameterized Approximations of CSPs

- Max-DNF parameterized by clique-width is W[1]-hard, and there is no FPT-AS, unless FPT = W[1] [Dell, Kim, Lampis, Mitsou and Mömke 2017]

- Max-CSP with MAJORITY constraints (i.e., a constraint is satisfied only if at least half of its literals are true) is W[1]-hard, parameterized by the Feedback Vertex Set. The problem admits and FPT-AS [Dell, Kim, Lampis, Mitsou and Mömke 2017]

Talk Outline

- Parameterization and Approximation
- FPT approximations for W-hard Problems
- FPT approximations for CSPs
- Summary and Open Problems

Summary

- Some parameterized CSPs are known to be W-hard. For such problems, it is natural to explore the existence of parameterized approximations.

- For some CSPs W-hardness can (almost) be circumvented using parameterized approximation, while others are inapproximable.

- Solving Max-SAT exactly is W-hard even for highly restricted dense graph parameters; however, Max-SAT admits an FPT-AS when parameterized by clique-width.

- In contrast, Max-DNF parameterized by clique- width is W-hard and admits no FPT-AS unless FPT=W[1].

Open Problems

- Parameterized approximations for other Max-CSPs that are W-hard? For example, Max-MAJORITY-CSP is W[1]-hard when parameterized by treewidth. Does it admit an FPT-AS?

- How the techniques used for design of FPT algorithms can be used to obtain parameterized approximations for CSPs?

- Choice of parameters/instance graph? So far, the parameters used for parameterized approximations relate to the incidence graph of the input instance.