FPT Approximation using Treewidth: CVC, VDS and TSS

Huairui Chu and Bingkai Lin

Nanjing University, China

Introduction

$f(w)|G|^{O(1)}$-time (**FPT**).

Many hard problems have **efficient algorithm** when the input graph's **treewidth** is at most $\boldsymbol{w}$.

Treewidth: A measure of how "tree-like" the graph is. (Introduced by Robertson and Seymour.)

Introduction

Some problems are **hard** even when the input graph's **treewidth** is small.

This Work: $f(w)|G|^{O(1)}$-time approximation for **three** such problems.

CVC: Capacitated Vertex Cover
VDS: Vector Dominating Set
TSS: Target Set Selection

Overview

- Treewidth: An Introduction
- Target Set Selection and Vector Dominating Set (TSS & VDS)
- Capacitated Vertex Cover (CVC)
- Conclusion

Overview

- **Treewidth: An Introduction**
- Target Set Selection and Vector Dominating Set (TSS & VDS)
- Capacitated Vertex Cover (CVC)
- Conclusion

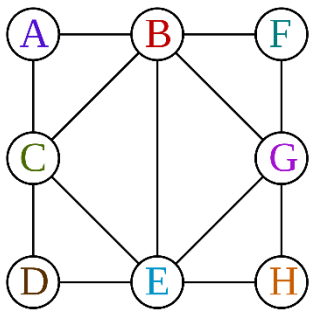
Treewidth-Definitions

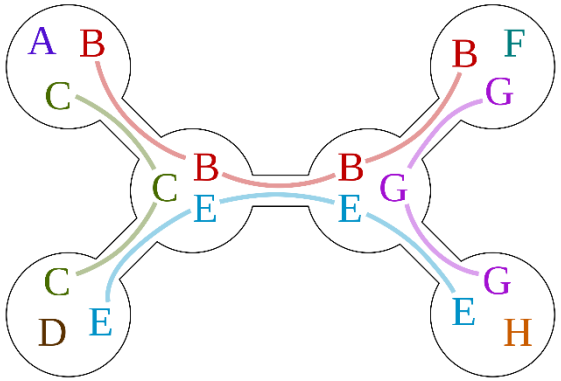
Definition (tree decomposition)

A tree decomposition of a graph G is a pair $(T, \mathcal{X})$ such that

- T is a rooted tree and $\mathcal{X} = \{X_\alpha : \alpha \in V(T), X_\alpha \subseteq V(G)\}$ is a collection of subsets of $V(G)$;
- $\bigcup_{\alpha \in V(T)} X_\alpha = V(G)$;
- For every edge $e \in E(G)$, there exists a $X_\alpha \in \mathcal{X}$ such that $e \subseteq X_\alpha$;
- For every vertex $v \in V(G)$, the set $\{\alpha \in V(T) : v \in X_\alpha\}$ induces a subtree of T.

$X_\alpha \in \mathcal{X}$: 'bag'



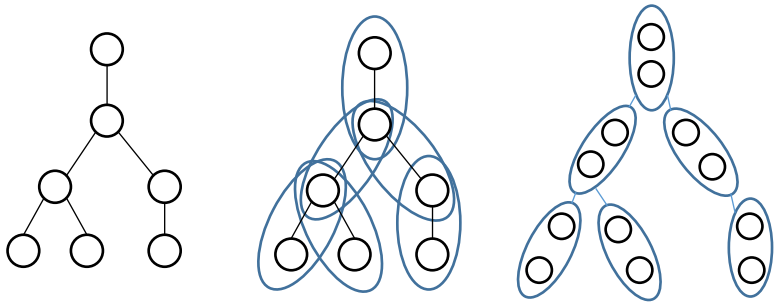


Tree decomposition width: $\max_{\alpha \in V(T)} |X_\alpha| - 1$

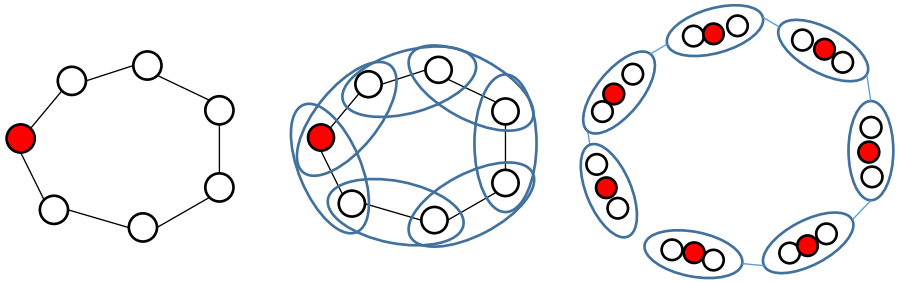
Treewidth of G: minimum width over all tree decompositions of G

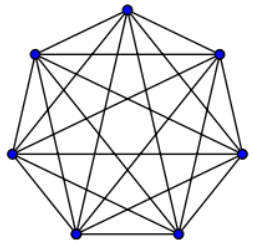
Treewidth-Examples

Tree has treewidth 1

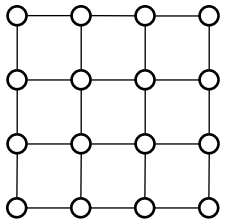


Cycle has treewidth 2





k-Clique has treewidth $k-1$

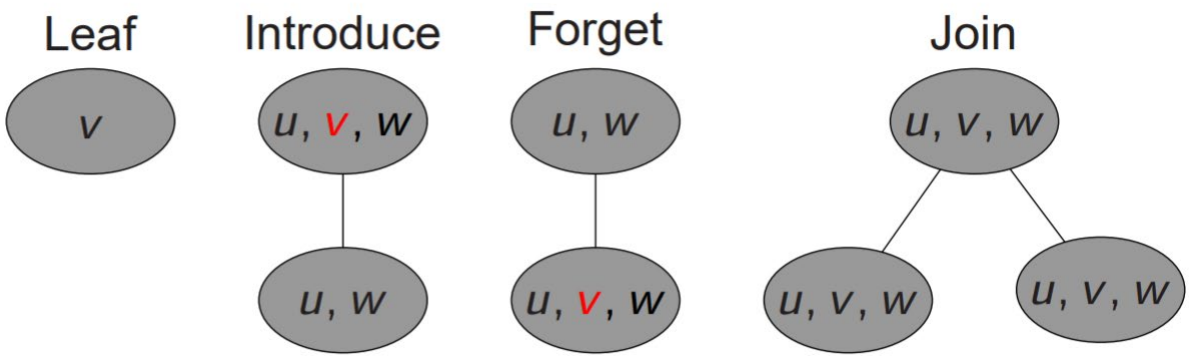


k by k grid has treewidth k

Control flow graphs have small treewidth

Treewidth-Nice Tree Decomposition

A tree decomposition T is **nice** if any node $\alpha \in V(T)$ is one of the following:



Any tree decomposition can be transformed into a nice tree decomposition in polynomial time, with width w kept.

Treewidth-Applications

Treewidth can be computed in FPT time.

- **[Bodlaender, 1996]:** exact, $2^{O(w^3)}n^{O(1)}$
- **[Korhonen, 2021]:** $2 + o(1)$-approximation, $2^{O(w)}n^{O(1)}$

Theorem [Courcelle, 1990]

Every graph property definable in **monadic second-order logic** can be decided in FPT time.

Example:

3-COLORING

$\exists C_1, C_2, C_3 \subseteq V$

$\big(\forall v \in V(v \in C_1 \vee v \in C_2 \vee v \in C_3)\big)$

$\wedge \Big(\forall u, v \in V\ adj(u, v) \rightarrow \big(\neg(u \in C_1 \wedge v \in C_1) \wedge \neg(u \in C_2 \wedge v \in C_2) \wedge \neg(u \in C_3 \wedge v \in C_3)\big)\Big)$

Hard Problems on Bounded Treewidth Graphs

Unlikely to have $f(w)|G|^{O(1)}$-time Algorithm

$\boldsymbol{W[1]}$**-hard problems** parameterized by treewidth w:

- **Cover & domination**: **Capacitated Vertex Cover**, Capacitated Dominating Set, **Vector Dominating Set**, etc.
- **Coloring**: Equitable Coloring, List Coloring, etc.
- **Others**: **Target Set Selection**, Constraint Satisfaction Problem, etc.

Our approach: Combine **Approximation** and **Parameterization.**

Our Results

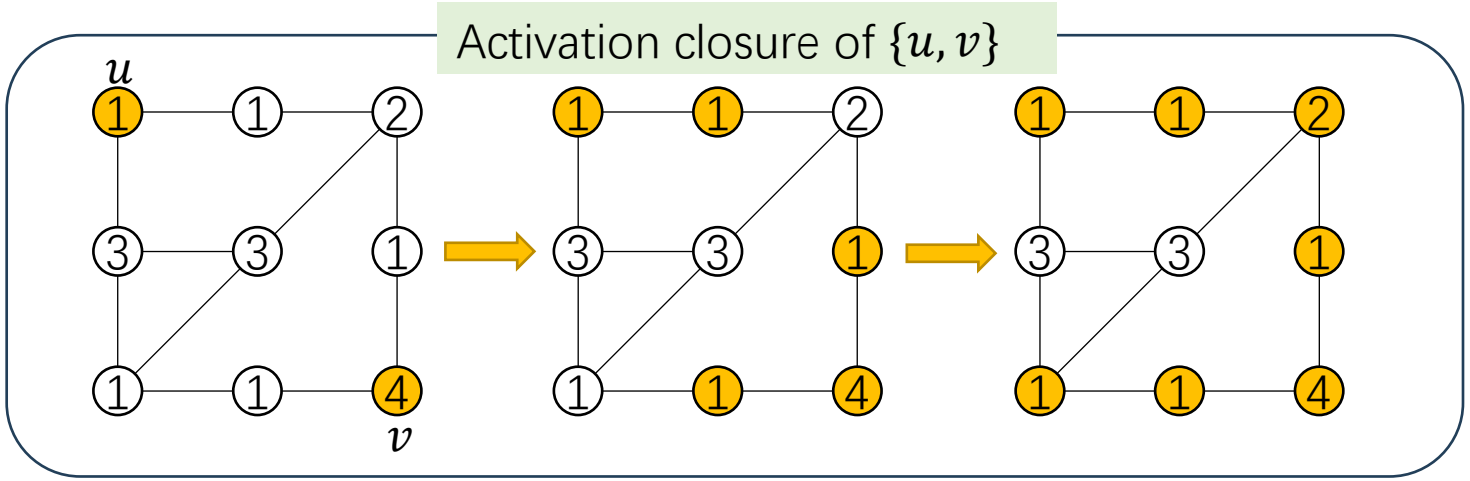
Problem	Running time	Approximation ratio
CVC	FPT	$1+\left(\frac{1}{w\log n}\right)^{O(1)}$
TSS	$n^{C+O(1)}$	$1+\frac{w+1}{C+1}$
VDS	FPT	$1+\left(\frac{1}{w\log\log n}\right)^{O(1)}$

Overview

- Treewidth: An Introduction
- **Target Set Selection and Vector Dominating Set (TSS & VDS)**
- Capacitated Vertex Cover (CVC)
- Conclusion

TSS-Introduction

Target Set Selection
Input: Given graph $G = (V, E)$ and a threshold function $t: V \to \mathbb{N}$.
Goal: Find the smallest set $S \subseteq V$ that can **activate** V.



Application:
Spread information in a social network

TSS-Introduction

Hardness of TSS

Parameter	Hardness	Approximation ratio	Hypothesis	Reference
k: solution size	$W[P]$-hard	1		[Abrahamson, 1995]
	$n^{\omega(1)}$	$2^{\log^{1-\epsilon} n}$	$P \neq NP$	[Chen, 2009] [Charikar, 2016]
w: treewidth	$n^{\Omega(\sqrt{w})}$	1	ETH	[Ben-Zwi, 2011]

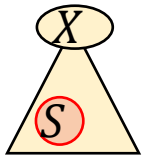
Our result: For any $C \in \mathbb{N}$, an algorithm for TSS:

- running time $n^{C+O(1)}$
- approximation ratio $1 + \frac{w+1}{C+1}$.

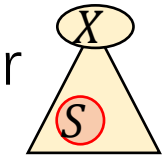
TSS-Observation

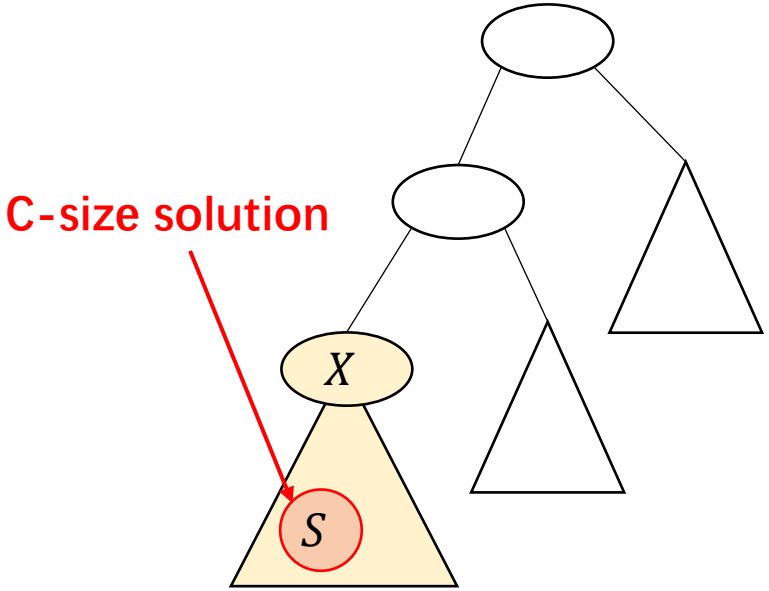
Lemma
Every bag $X \in \mathcal{X}$ in a tree decomposition $(T, \mathcal{X})$ is a separator of the original graph.

If the **opt solution** has C vertices in:



Then we can pick **X** and the C-size solution **S** as a solution for





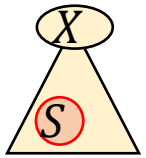
Approx. Ratio: $\frac{|X|+C}{C}$

TSS-Observation

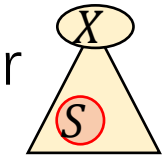
Lemma
Every bag $X \in \mathcal{X}$ in a tree decomposition $(T, \mathcal{X})$ is a separator of the original graph.

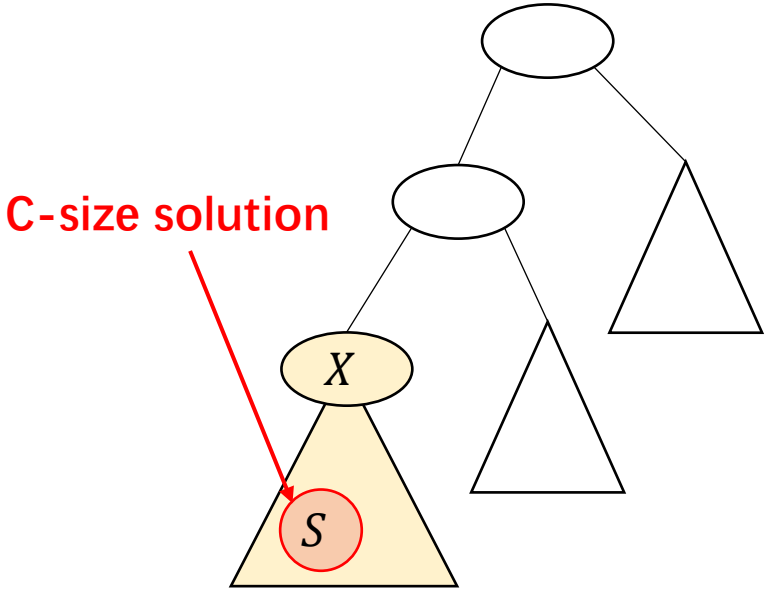
Can be tested in n^{C+1}-time

If the **opt solution** has C vertices in:



Then we can pick **X** and the C-size solution **S** as a solution for





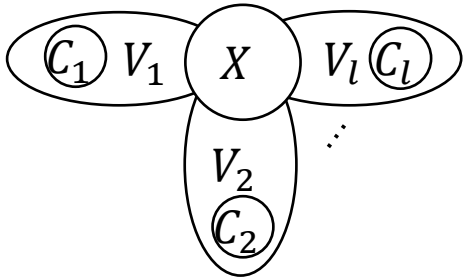
Approx. Ratio: $\frac{|X|+C}{C}$

TSS-Generalization

Observations:

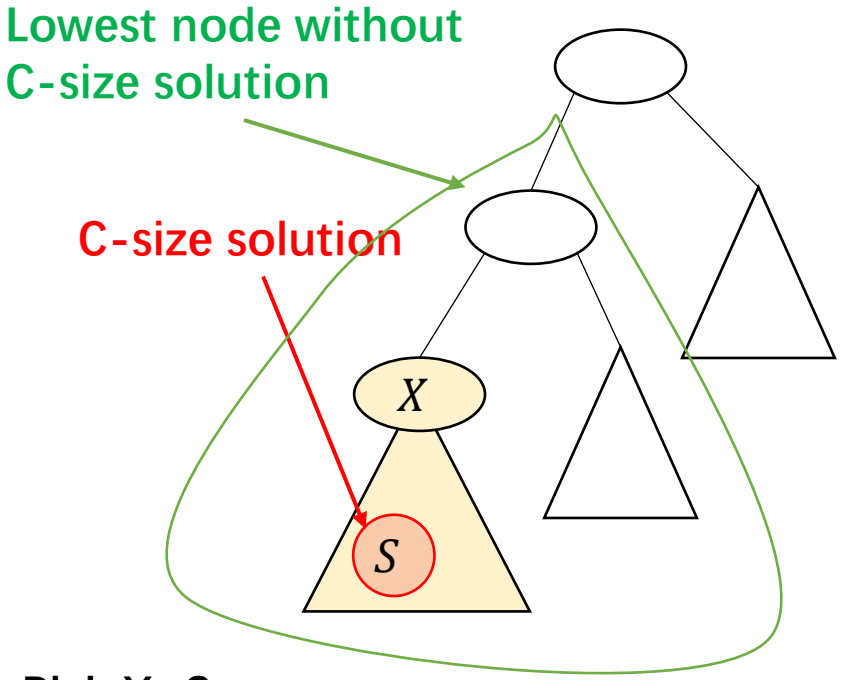
- TSS is **monotone**
- TSS is **splittable**

splittable



If X is a **separator** of G and each C_i is a **solution** for $G[V_i]$, then

- $X \cup C_1 \cup C_2 \cdots C_l$ is a solution for G



Pick X∪S

Approx. Ratio$=\frac{|X \cup S|}{c} \leq (1+\frac{w+1}{c})$

A General Approximation Algorithm

Main Theorem 1 (informal):

- **Monotone**
- **Splittable**
- **Algorithm $\boldsymbol{A}$**: 'finding C-sized solution', running time $f(C, w, n)$

$f(C, w, n)n^{O(1)}$-time, $1 + \frac{w+1}{C+1}$-ratio algorithm

TSS: $f(C, w, n) = n^{C+O(1)}$ $\xrightarrow{\text{Set } C \text{ a constant}}$ $n^{C+O(1)}$ time, $1 + \frac{w+1}{C+1}$-ratio

Application to VDS

Vector Dominating Set
Input: $G = (V, E)$, a threshold function $t: V \to \mathbb{N}$.
Goal: Find the smallest set S that $\forall v \in V \setminus S, |N(v) \cap S| \geq t(v)$.

Hardness:

- Generalizes Dominating Set.
- $W[1]$-hard parameterized by treewidth.

VDS:

$f(C, w, n) = 2^{O(wC^2 \log C)} n^{O(1)}$

[Raman, 2008]

Set $C = w^2 \left(\frac{\log \log n}{\log \log \log n}\right)^{0.5}$

FPT time,

$1 + \frac{1}{(w \log \log n)^{\Omega(1)}}$-ratio

Limitation

Some problems are not **splittable**!

- Capacitated Vertex Cover
- Capacitated Dominating Set

Overview

- Treewidth: An Introduction
- Target Set Selection and Vector Dominating Set (TSS & VDS)
- **Capacitated Vertex Cover (CVC)**
- Conclusion

Capacitated Vertex Cover-Introduction

Capacitated Vertex Cover(CVC)
Input: Given a graph $G = (V, E)$, a capacity function $c: V \to \mathbb{N}$.
Goal: Find a min vertex set S and a mapping $M: E \to S$ such that

- $\forall v \in S, \left|M^{-1}(v)\right| \leq c(v)$
- for each $e \in E, M(e) \in e$.

mapped to
=covered by

- **CVC** is $W[1]$-hard with parameter w.
- **[Lampis 2014]:** find a 'solution' of **OPT** size, but relax capacity constraint to $\left|M^{-1}(v)\right| \leq$ **(1+ϵ)** $\cdot\, c(v)$.

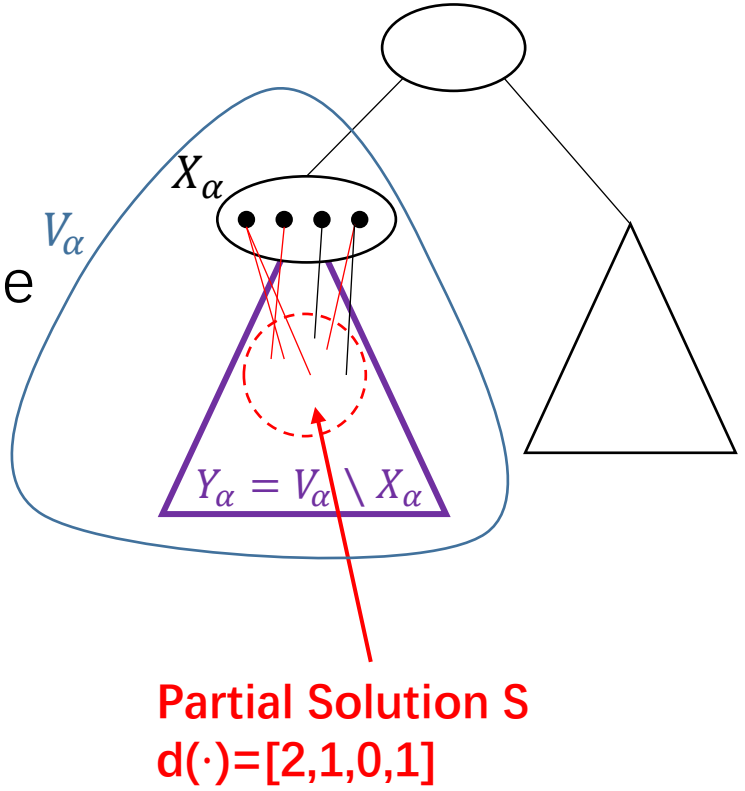
Question:
Can we find a solution of (1+ϵ)**OPT** size and satisfies all constraints?

Naïve DP for CVC

Naïve DP:

- Let $R_\alpha := \{(d,k): d: X_\alpha \to \mathbb{N}, k \in \mathbb{N}\}$.
- (d,k) records a partial solution on graph $G[Y_\alpha]$
- $k \in \mathbb{N}$ is the solution size.
- For each $v \in X_\alpha$, $\exists d(v)$ of its incident edges can be covered by the solution. $\Rightarrow d(v) \leq k$
- Compute R_α by DP.

Problem: R_α has $n^{\Theta(w)}$ records ☹



Lampis' Idea

Lampis' Idea: compressing DP table

$$n^{w+O(1)} \rightarrow \left(\frac{\log n}{\epsilon}\right)^{w+O(1)}$$

- Pick $\epsilon \in (0,1)$
- Replace every $x \in [n]$ by x' with $(1+\epsilon)^{x'} \leq x < (1+\epsilon)^{x'}$
- Reduce $[n]$ to $[\log_{1+\epsilon} n]$

Error increase

use $a \sim_\gamma b$ to denote that $b/(1+\gamma) \leq a \leq (1+\gamma)b$.

▶ **Lemma 9.** *Let $a, b, a', b' \in \mathbb{R}, h \in \mathbb{N}^+$, $\epsilon_h \in (0, 0.01)$, $a' \sim_{\epsilon_h} a$ and $b' \sim_{\epsilon_h} b$. Then we have $[a'+b']_\epsilon \sim_{\epsilon_{h+1}} (a+b)$.*

- Use $O(\log n)$ depth Tree-decomposition
- Set $\epsilon = 1/\mathrm{poly}\log n$

Lampis' Idea

Lampis uses this framework to get following results:

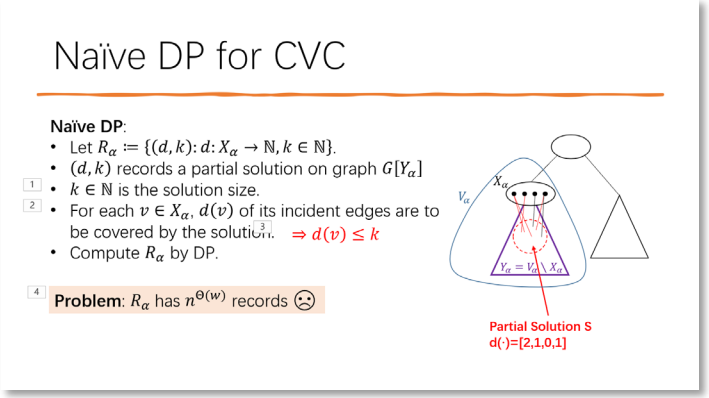
- find a 'solution' of **OPT** size, but relax capacity constraint to $\left|M^{-1}(v)\right| \leq \mathbf{(1+\epsilon)} \cdot c(v)$.

Question:
Can we find a solution of $(1+\epsilon)$**OPT** size and satisfies all constraints?

We say yes for CVC.

Approximation DP Algorithm-Intuition

Recall:



Approx. DP:

- Let $\widehat{R_\alpha}$ be an approximate sketch of R_α.
- Compute $\widehat{R_\alpha}$ by DP

Problem

Record $(\hat{d}, \hat{k}) \in \widehat{R_\alpha}$ may violate constraint, how to modify it to obtain a good one?

Modify Partial Solution

A simple case:

- $d \setminus v = d_m \setminus v, d_m(v) = d(v) + p$ for a fixed v
- we want to modify a partial solution (d, k) to $(d_m, k + p)$

Lemma
if we want to modify a partial solution (d, k) to $(d_m, k + p)$ where $d_m \setminus v = d \setminus v, d_m(v) = d(v) + p$ for a fixed v, we only need to test if (d_m, ∞) is feasible.

Lemma
We can test in polynomial time test if (d_m, ∞) is feasible for any d.

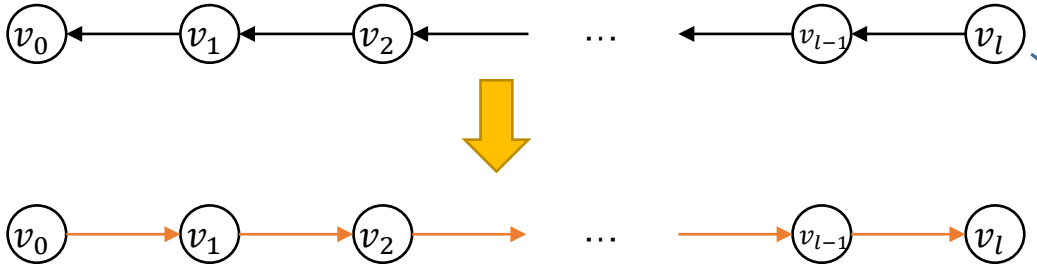
Proof:
Use network flow.

Modify Partial Solution-Augmenting

Lemma
if we want to modify a partial solution (d, k) to $(d_m, k+p)$ where $d_m \setminus v = d \setminus v, d_m(v) = d(v) + p$ for a fixed v, we only need to test if (d_m, ∞) is feasible.

Suppose we want d(v_0)→d(v_0)+1

An augmentable path



Possible case:
$v_l \notin$ solution

#edge covered by v_0 -= 1
#edge covered by v_l += 1

Solution size += 1

Modify Partial Solution-Barrier and Solution

Problem:
How to show that good approx. of $d(v)$ can lead to good approx. of k ?

A trick:
$d(v)$: the number of edges NOT covered$\Rightarrow d(v) \leq k$.

Error accumulation: $\epsilon_h d(v) + k \leq (1 + \epsilon_h)k$

Only ≤ 1 vertex needs to be modified! (nice tree decomposition)

Overview

- Treewidth: An Introduction
- Target Set Selection and Vector Dominating Set (TSS & VDS)
- Capacitated Vertex Cover (CVC)
- **Conclusion**

Conclusion

Main message:

- Combining **Approximation** and **DP on Tree Decomposition** is a promising research direction.

Two methods to handle **approx. error** during **DP.**

- Estimate opt solution in subtree + remove the bag
- Compress DP-table + modify solution

Open problem

- Constant FPT (In)approximability for TSS parameterized by tree width.

References

[Bodlaender, 1996] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM. J. Comput. 25(6), 1305-1317 (1996)
[Bodlaender, 1998] Hans L. Bodlaender and Torben Hagerup. Parallel algorithms with optimal speedup for bounded treewidth. SIAM J. Comput., 27(6):1725–1746, 1998.
[Korhonen, 2021] Tuukka Korhonen. A Single-Exponential Time 2-Approximation Algorithm for Treewidth SIAM. J. Comput. 25(6), 1305-1317 (1996)
[Courcelle, 1990] Bruno Courcelle. Graph rewriting: An algebraic and logic approach. Handbook of Theoretical Computer Science Volume B: Formal Models and Sematics (B)1990: 193-242.
[Lampis 2014] Michael Lampis. Parameterized Approximation Schemes using Graph Widths. ICALP 2014: 775-786
[Abrahamson, 1995] Karl R. Abrahamson, Rodney G. Downey and Michael R. Fellows. Fixed-Parameter Tractability and Completeness IV: On Completeness for W[P] and PSPACE Analogues. Ann. Pure Appl. Log. 73(3): 235-276 (1995)
[Chen, 2009] Ning Chen. On the Approximability of Influence in Social Networks. SIAM J. Discret. Math. 23(3): 1400-1415 (2009)
[Charikar, 2016] Moses Charikar, Yonatan Naamad, and Anthony Wirth. On approximating target set selection. APPROX/RANDOM 2016.
[Ben-Zwi, 2011] Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. Discret. Optim. 8(1): 87-96 (2011)
[Raman, 2008] Venkatesh Raman, Saket Saurabh, and Sriganesh Srihari. Parameterized algorithms for generalized domination. COCOA 2008: 116-126