8b. Iterative Compression

COMP6741: Parameterized and Exact Computation

Serge Gaspers

School of Computer Science and Engineering, UNSW Sydney, Australia

19T3

Outline

Outline

Iterative Compression

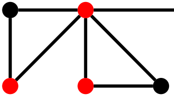
For a minimization problem:

- **Compression step:** Given a solution of size $k+1$, compress it to a solution of size k or prove that there is no solution of size k
- **Iteration step:** Incrementally build a solution to the given instance by deriving solutions for larger and larger subinstances

Example: Vertex Cover

A vertex cover in a graph $G = (V, E)$ is a subset of its vertices $S \subseteq V$ such that every edge of G has at least one endpoint in S.

Vertex Cover	
Input:	A graph $G = (V, E)$ and an integer k
Parameter:	k
Question:	Does G have a vertex cover of size k?



We will design a (slow) iterative compression algorithm for Vertex Cover to illustrate the technique.

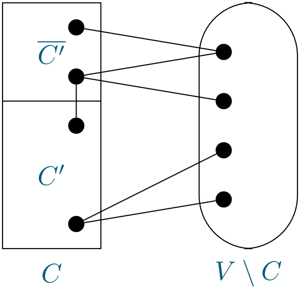
Vertex Cover: Compression Step

Comp-VC

Input:	graph $G = (V, E)$, integer k, vertex cover C of size $k+1$ of G
Output:	a vertex cover C^* of size $\leq k$ of G if one exists

Vertex Cover: Compression Step

Comp-VC
- Input: graph $G = (V, E)$, integer k, vertex cover C of size $k+1$ of G
- Output: a vertex cover C^* of size $\leq k$ of G if one exists



- Go over all partitions $(C', \overline{C'})$ of C
- $C^* = C' \cup N(\overline{C'})$
- If $\overline{C'}$ is an independent set and $|C^*| \leq k$ then return C^*

Vertex Cover: Iteration Step

Use algorithm for Comp-VC to solve Vertex Cover.

VERTEX COVER: Iteration Step

Use algorithm for COMP-VC to solve VERTEX COVER.

- Order vertices: $V = \{v_1, v_2, \ldots, v_n\}$
- Define $G_i = G[\{v_1, v_2, \ldots, v_i\}]$
- $C_0 = \emptyset$
- For $i = 1..n$, find a vertex cover C_i of size $\leq k$ of G_i using the algorithm for COMP-VC with input G_i and $C_{i-1} \cup \{v_i\}$. If G_i has no vertex cover of size $\leq k$, then G has no vertex cover of size $\leq k$.

Final running time: $O^*(2^k)$

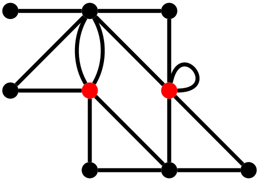
Outline

Feedback Vertex Set

A feedback vertex set of a multigraph $G = (V, E)$ is a set of vertices $S \subseteq V$ such that $G - S$ is acyclic.

FEEDBACK VERTEX SET (FVS)

Input:	Multigraph $G = (V, E)$, integer k
Parameter:	k
Question:	Does G have a feedback vertex set of size at most k?



Note: We already saw an $O^*((3k)^k)$ time algorithm (and a $O^*(4^k)$ time randomized algorithm) for FVS.
We will now aim for a $O^*(c^k)$ time deterministic algorithm, with $c \in O(1)$.

Compression Problem

Comp-FVS

Input: graph $G = (V, E)$, integer k, feedback vertex set S of size $k + 1$ of G

Output: a feedback vertex set S^* of size $\leq k$ of G if one exists

Iteration step

- Order vertices: $V = \{v_1, v_2, \dots, v_n\}$
- Define $G_i = G[\{v_1, v_2, \dots, v_i\}]$
- $S_0 = \emptyset$
- For $i = 1..n$, find a feedback vertex set S_i of size $\leq k$ of G_i using the algorithm for $\textsc{Comp-FVS}$ with input G_i and $S_{i-1} \cup \{v_i\}$. If G_i has no feedback vertex set of size $\leq k$, then G has no feedback vertex set of size $\leq k$.

Suppose $\textsc{Comp-FVS}$ can be solved in $O^*(c^k)$ time.
Then, using this iteration, FVS can be solved in $O^*(c^k)$ time.

Compression step

To solve $\textsc{Comp-FVS}$: for each partitions $(S', \overline{S'})$ of S, find a feedback vertex set S^* of G with $|S^*| < |S|$ and $S' \subseteq S^* \subseteq V \setminus \overline{S'}$ if one exists.

Compression step

To solve $\textsc{Comp-FVS}$: for each partitions $(S', \overline{S'})$ of S, find a feedback vertex set S^* of G with $|S^*| < |S|$ and $S' \subseteq S^* \subseteq V \setminus \overline{S'}$ if one exists.
Equivalently, find a feedback vertex set S'' of $G - S'$ with $|S''| < |\overline{S'}|$ and $S'' \cap \overline{S'} = \emptyset$.

Compression step

To solve COMP-FVS: for each partitions $(S', \overline{S'})$ of S, find a feedback vertex set S^* of G with $|S^*| < |S|$ and $S' \subseteq S^* \subseteq V \setminus \overline{S'}$ if one exists.
Equivalently, find a feedback vertex set S'' of $G - S'$ with $|S''| < |\overline{S'}|$ and $S'' \cap \overline{S'} = \emptyset$.
We arrive at the following problem:

DISJOINT-FVS

Input:	graph $G = (V, E)$, integer k, feedback vertex set S of size $k + 1$ of G
Output:	a feedback vertex set S^* of G with $|S^*| \leq k$ and $S^* \cap S = \emptyset$, if one exists

Compression step

To solve COMP-FVS: for each partitions $(S', \overline{S'})$ of S, find a feedback vertex set S^* of G with $|S^*| < |S|$ and $S' \subseteq S^* \subseteq V \setminus \overline{S'}$ if one exists.
Equivalently, find a feedback vertex set S'' of $G - S'$ with $|S''| < |\overline{S'}|$ and $S'' \cap \overline{S'} = \emptyset$.
We arrive at the following problem:

DISJOINT-FVS

Input:	graph $G = (V, E)$, integer k, feedback vertex set S of size $k+1$ of G
Output:	a feedback vertex set S^* of G with $|S^*| \le k$ and $S^* \cap S = \emptyset$, if one exists

If DISJOINT-FVS can be solved in $O^*(d^k)$ time, then COMP-FVS can be solved in

$$O^*\left(\sum_{i=0}^{k+1} \binom{k+1}{i} d^i\right) \subseteq O^*((d+1)^k) \text{ time}$$

by the Binomial Theorem: $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$.

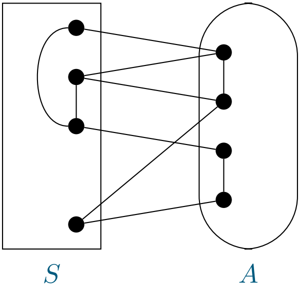
Algorithm for Disjoint-FVS

Disjoint-FVS

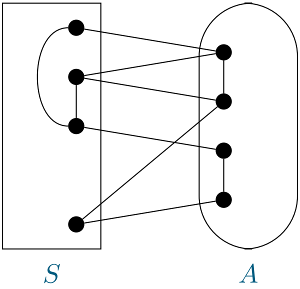
Input: graph $G = (V, E)$, integer k, feedback vertex set S of size $k + 1$ of G

Output: a feedback vertex set S^* of G with $|S^*| \leq k$ and $S^* \cap S = \emptyset$, if one exists

Denote $A := V \setminus S$.



Simplification rules for DISJOINT-FVS



Start with $S^* = \emptyset$.

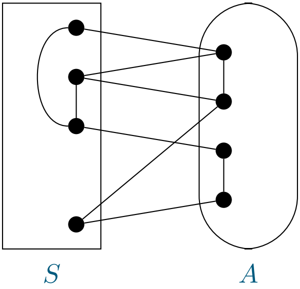
(cycle-in-S)

If $G[S]$ is not acyclic, then return NO.

(budget-exceeded)

If $k < 0$, then return NO.

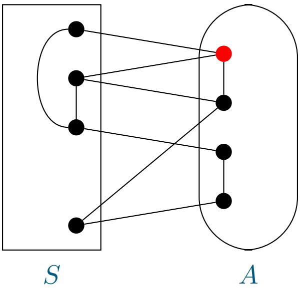
Simplification rules for DISJOINT-FVS



(finished)

If $G - S^*$ is acyclic, then return S^*.

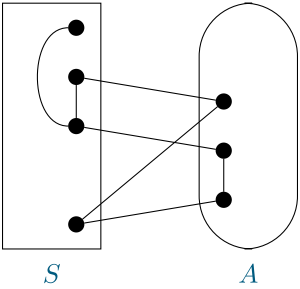
Simplification rules for DISJOINT-FVS



(creates-cycle)

If $\exists v \in A$ such that $G[S \cup \{v\}]$ is not acyclic, then add v to S^* and remove v from G.

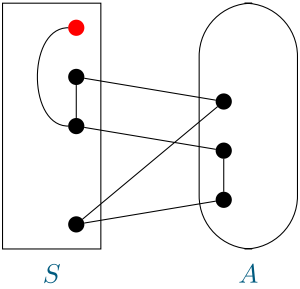
Simplification rules for Disjoint-FVS



(creates-cycle)

If $\exists v \in A$ such that $G[S \cup \{v\}]$ is not acyclic, then add v to S^* and remove v from G.

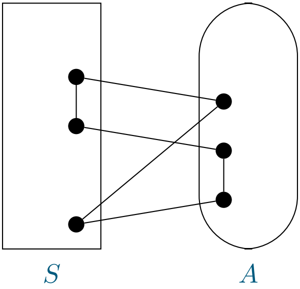
Simplification rules for Disjoint-FVS



(Degree-(≤ 1))

If $\exists v \in V$ with $d_G(v) \leq 1$, then remove v from G.

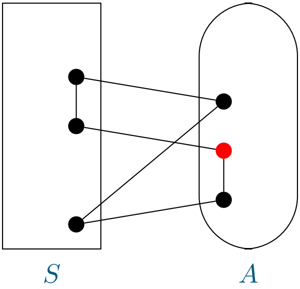
Simplification rules for Disjoint-FVS



(Degree-(≤ 1))

If $\exists v \in V$ with $d_G(v) \leq 1$, then remove v from G.

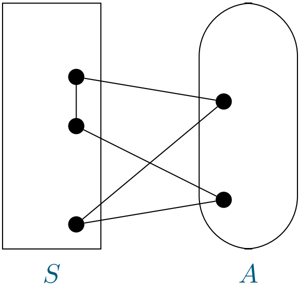
Simplification rules for DISJOINT-FVS



(Degree-2)

If $\exists v \in V$ with $d_G(v) = 2$ and at least one neighbor of v is in A, then add an edge between the neighbors of v (even if there was already an edge) and remove v from G.

Simplification rules for DISJOINT-FVS



(Degree-2)

If $\exists v \in V$ with $d_G(v) = 2$ and at least one neighbor of v is in A, then add an edge between the neighbors of v (even if there was already an edge) and remove v from G.

Branching rule for Disjoint-FVS

Select a vertex $v \in A$ with at least 2 neighbors in S.
Such a vertex exists if no simplification rule applies (for example, we can take a leaf in $G[A]$).
Branch into two subproblems:

- $v \in S^*$: add v to S^*, remove v from G, and decrease k by 1
- $v \notin S^*$: add v to S

Exercise: Running time

- Prove that this algorithm has running time $O^*(4^k)$.

Result for FEEDBACK VERTEX SET

Theorem 1

FEEDBACK VERTEX SET *can be solved in* $O^*(5^k)$ *time.*

Outline

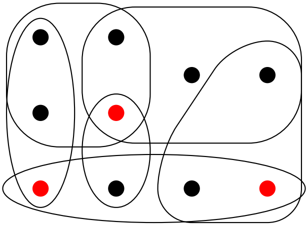
r-Hitting Set

A set system $\mathcal{S}$ is a pair (V, H), where V is a finite set of elements and H is a collection of subsets of V. The rank of $\mathcal{S}$ is the maximum size of a set in H, i.e., $\max_{Y \in H} |Y|$.

A hitting set of a set system $\mathcal{S} = (V, H)$ is a subset X of V such that X contains at least one element of each set in H, i.e., $X \cap Y \neq \emptyset$ for each $Y \in H$.

r-HITTING SET (r-HS)

Input:	A rank r set system $\mathcal{S} = (V, H)$, an integer k
Parameter:	k
Question:	Does $\mathcal{S}$ have a hitting set of size at most k



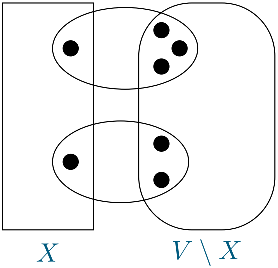
Note: There is an easy $O^*(r^k)$ branching algorithm.

Compression Step

COMP-r-HS

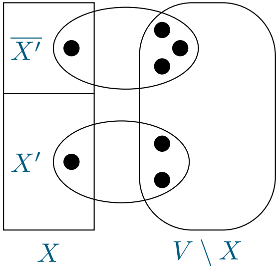
Input: set system $\mathcal{S} = (V, H)$, integer k, hitting set X of size $k+1$ of $\mathcal{S}$

Output: a hitting set X^* of size $\leq k$ of $\mathcal{S}$ if one exists



Compression Step

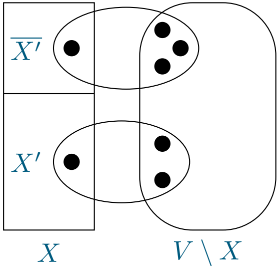
Comp-r-HS
Input: set system $\mathcal{S} = (V, H)$, integer k, hitting set X of size $k+1$ of $\mathcal{S}$
Output: a hitting set X^* of size $\leq k$ of $\mathcal{S}$ if one exists



Go over all partitions $(X', \overline{X'})$ of X

Compression Step

Comp-r-HS
Input: set system $\mathcal{S} = (V, H)$, integer k, hitting set X of size $k+1$ of $\mathcal{S}$
Output: a hitting set X^* of size $\leq k$ of $\mathcal{S}$ if one exists

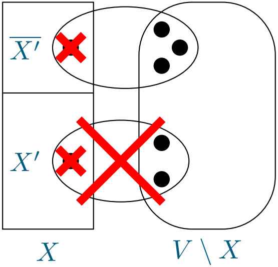


Reject a partition if there is a $Y \in H$ such that $Y \subseteq \overline{X'}$.

Compression Step

Comp-r-HS

Input:	set system $\mathcal{S} = (V, H)$, integer k, hitting set X of size $k+1$ of $\mathcal{S}$
Output:	a hitting set X^* of size $\leq k$ of $\mathcal{S}$ if one exists



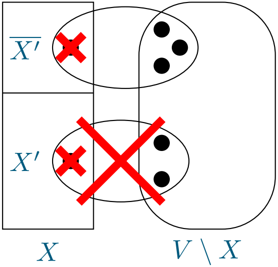
Compute a hitting set X'' of size $\leq k - |X'|$ for (V', H'), where $V' = V \setminus X$ and $H' = \{Y \cap V' \ : \ Y \in H \wedge Y \cap X' = \emptyset\}$, if one exists.

Compression Step

Comp-r-HS

Input: set system $\mathcal{S} = (V, H)$, integer k, hitting set X of size $k+1$ of $\mathcal{S}$

Output: a hitting set X^* of size $\leq k$ of $\mathcal{S}$ if one exists



If one exists, then return $X^* = X' \cup X''$.

Compression Step II

- The subinstances (V', H') where $V' = V \setminus X$ and $H' = \{Y \cap V \ : \ Y \in H \wedge Y \cap X' = \emptyset\}$ are instances of $(r-1)$-HS.

Compression Step II

- The subinstances (V', H') where $V' = V \setminus X$ and $H' = \{Y \cap V \ : \ Y \in H \wedge Y \cap X' = \emptyset\}$ are instances of $(r-1)$-HS.
- Suppose $(r-1)$-HS can be solved in $O^*((\alpha_{r-1})^k)$ time. Then, Comp-r-HS can be solved in

$$O^* \left(\sum_{s=0}^{k} \binom{k+1}{s} (\alpha_{r-1})^{k-s} \right) \subseteq O^* \left((\alpha_{r-1} + 1)^k \right)$$

time.

Compression Step II

- The subinstances (V', H') where $V' = V \setminus X$ and $H' = \{Y \cap V \ : \ Y \in H \wedge Y \cap X' = \emptyset\}$ are instances of $(r-1)$-HS.
- Suppose $(r-1)$-HS can be solved in $O^*((\alpha_{r-1})^k)$ time. Then, Comp-r-HS can be solved in

$$O^*\left(\sum_{s=0}^{k} \binom{k+1}{s} (\alpha_{r-1})^{k-s}\right) \subseteq O^*\left((\alpha_{r-1}+1)^k\right)$$

  time.
- **Note:** 2-HS is equivalent to Vertex Cover and can be solved in $O^*(1.2738^k)$ time [CKX10].
- **Note 2:** 3-HS can be solved in $O^*(2.0755^k)$ time [Wah07].

Iteration Step

- (V, H) instance of r-HS with $V = \{v_1, v_2, \ldots, v_n\}$
- $V_i = \{v_1, v_2, \ldots, v_i\}$ for $i = 1$ to n
- $H_i = \{Y \in H \ : \ Y \subseteq V_i\}$

Iteration Step

- (V, H) instance of r-HS with $V = \{v_1, v_2, \ldots, v_n\}$
- $V_i = \{v_1, v_2, \ldots, v_i\}$ for $i = 1$ to n
- $H_i = \{Y \in H \ : \ Y \subseteq V_i\}$
- Note that $|X_{i-1}| \leq |X_i| \leq |X_{i-1}| + 1$ where X_j is a minimum hitting set of the instance (V_i, H_i)

r-HS running time

Theorem 2

For $r \geq 3$, r-HS can be solved in $O((r - 0.9245)^k)$ time.

By Monotone Local Search:

Theorem 3

For $r \geq 3$, r-HS can be solved in $O\left(\left(2 - \frac{1}{r-0.9245}\right)^n\right)$ time.

Outline

Further Reading

- Chapter 4, *Iterative Compression* in [Cyg+15]
- Section 11.3, *Iterative Compression* in [Nie06]
- Section 6.1, *Iterative Compression: The Basic Technique* in [DF13]
- Section 6.2, *Edge Bipartization* in [DF13]

References I

- [CKX10] Jianer Chen, Iyad A. Kanj, and Ge Xia. "Improved upper bounds for vertex cover". In: *Theoretical Computer Science* 411.40-42 (2010), pp. 3736–3756.
- [Cyg+15] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [DF13] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
- [Nie06] Rolf Niedermeier. *Invitation to Fixed Parameter Algorithms*. Oxford University Press, 2006.
- [Wah07] Magnus Wahlström. "Algorithms, measures and upper bounds for satisfiability and related problems". PhD thesis. Linköping University, Sweden, 2007.