Outline

1. Introduction
2. Feedback Vertex Set
3. Min r-Hitting Set
4. Further Reading

S. Gaspers (UNSW)
Outline

1. Introduction
2. Feedback Vertex Set
3. Min r-Hitting Set
4. Further Reading
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
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 $C^*$
Use algorithm for **Comp-VC** to solve **Vertex Cover**.
Use algorithm for \textsc{Comp-VC} to solve \textsc{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 \textsc{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

1. Introduction
2. Feedback Vertex Set
3. Min r-Hitting Set
4. Further Reading
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)**

<table>
<thead>
<tr>
<th>Input:</th>
<th>Multigraph $G = (V, E)$, integer $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter:</td>
<td>$k$</td>
</tr>
<tr>
<td>Question:</td>
<td>Does $G$ have a feedback vertex set of size at most $k$?</td>
</tr>
</tbody>
</table>

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: \( V = \{v_1, v_2, \ldots, v_n\} \)
- Define \( G_i = G[\{v_1, v_2, \ldots, v_i\}] \)
- \( S_0 = \emptyset \)
- For \( i = 1..n \), find a feedback vertex set \( S_i \) of size \( \leq k \) of \( G_i \) using the algorithm for 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 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 $\text{COMP-FVS}$: for each partitions $(S', \overline{S'})$ of $S$, find a feedback vertex set $S^*$ of $G$ with $|S^*| < |S|$ and $S' \subseteq S^* \subseteq V \setminus S'$ if one exists.
Compression step

To solve $\text{COMP-FVS}$: 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 $S'' \cap \overline{S'} = \emptyset$. 

If $\text{Disjoint-FVS}$ can be solved in $O^*((d^k))$ time, then $\text{Comp-FVS}$ can be solved in $O^*((k+1 \sum_{i=0}^{k+1} (k+1)^i d^i)) \subseteq O^*((d+1)^k)$ time by the Binomial Theorem: $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$. 

S. Gaspers (UNSW)
Compression step

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

Equivalently, find a feedback vertex set \(S''\) of \(G - S'\) with \(|S''| < |S'|\) and \(S'' \cap S' = \emptyset\).

We arrive at the following problem:

\[
\text{DISJOINT-FVS}
\]

| 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 |

If \( \text{DISJOINT-FVS} \) can be solved in \(O^*(d^k)\) time, then \( \text{COMP-FVS} \) 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 $A := V \setminus S$. 

---

**Diagram:**

- **$S$**
- **$A$**

Edges connecting $S$ and $A$.
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**

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

\text{(creates-cycle)}
Simplification rules for \textsc{Disjoint-FVS}

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

(creates-cycle)
Simplification rules for **DISJOINT-FVS**

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

**Degree-\((\leq 1)\)**
Simplification rules for \textsc{Disjoint-FVS}

\begin{itemize}
  \item \textbf{(Degree-\((\leq 1))\)}
    \begin{itemize}
      \item If \(\exists v \in V\) with \(d_G(v) \leq 1\), then remove \(v\) from \(G\).
    \end{itemize}
\end{itemize}
Simplification rules for **Disjoint-FVS**

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 \( G \).
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.
r-Hitting Set

A set system $S$ 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 $S$ is the maximum size of a set in $H$, i.e., $\max_{Y \in H} |Y|$. A hitting set of a set system $S = (V, H)$ 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 $S = (V, H)$, an integer $k$
Parameter: $k$
Question: Does $S$ 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 $S = (V, H)$, integer $k$, hitting set $X$ of size $k + 1$ of $S$
- **Output:** a hitting set $X^*$ of size $\leq k$ of $S$ if one exists

Diagram:

- $X$
- $V \setminus X$
**Compression Step**

**COMP-\(r\)-HS**

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

*Output:* a hitting set \( X^* \) of size \( \leq k \) of \( S \) if one exists

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

**Comp-\(r\)-HS**

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

**Output:** a hitting set \(X^*\) of size \(\leq k\) of \(S\) 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 \(S = (V, H)\), integer \(k\), hitting set \(X\) of size \(k + 1\) of \(S\)
- **Output:** a hitting set \(X^*\) of size \(\leq k\) of \(S\) if one exists

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

**COMP-\(r\)-HS**

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

Output: a hitting set \( X^* \) of size \( \leq k \) of \( S \) 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 
  \(H' = \{Y \cap V : Y \in H \land Y \cap X' = \emptyset\}\) are instances of \((r - 1)\)-HS.
The subinstances \((V', H')\) where \(V' = V \setminus X\) and \(H' = \{Y \cap V : Y \in H \land Y \cap X' = \emptyset\}\) are instances of \((r - 1)\)-HS.

Suppose \((r - 1)\)-HS can be solved in \(O^*((\alpha_{r-1})^k)\) time. Then, \(\text{COMP-} r\)-HS can be solved in

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

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

Suppose \((r - 1)\)-HS can be solved in \(O^*\left((\alpha_{r-1})^k\right)\) time. Then, \(\text{COMP-}r\)-HS can be solved in 
\[
O^* \left( \sum_{s=0}^{k} \binom{k}{s} (\alpha_{r-1})^{k-s} \right) \subseteq O^* \left( (\alpha_{r-1} + 1)^k \right)
\]
time.

**Note:** 2-HS is equivalent to \textsc{Vertex Cover} and can be solved in \(O^*\left(1.2738^k\right)\) time [CKX10].

**Note 2:** 3-HS can be solved in \(O^*\left(2.0755^k\right)\) 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 \(n\)
- \(H_i = \{Y \in H : Y \subseteq V_i\}\)
(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 \((V_i, H_i)\)
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.
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


