# Introduction to NP-Completeness: Lecture 19


**Lecturer:** S. S. Ravi

**Affiliation:** Biocomplexity Institute and Initiative, UVA

**Email:** ssravi@virginia.edu


**Date:** Oct. 29, 2020

**Request:** If you send email to Ravi regarding this lecture, please be sure to cc the message to Professor Haifeng Xu (hx4ad@virginia.edu).

# Outline for Lecture 19

1. Steps to prove **NP**-completeness (brief review from Lecture 18)
2. Additional reductions to show **NP**-completeness
3. Coping with **NP**-complete problems

# Steps to Prove **NP**-completeness

**Goal:** To prove that Problem $Q$ is **NP**-complete.

- Show that $Q$ is in **NP**. (This step shows the **membership** in **NP**.)
- Identify a suitable problem $P$ which is known to be **NP**-complete.
- Show that $P \leq_p Q$. (This step shows the **NP-hardness** of $Q$.)

**Minimum Set Cover** (MSC):

Instance: A universal set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $S = \{S_1, S_2, \ldots, S_m\}$, where each $S_j$ is a subset of $U$ $(1 \leq j \leq m)$ and an integer $r \leq m$.

Question: Is there is a subcollection $S'$ of $S$ such that $|S'| \leq r$ and the union of the sets in $S'$ is equal to $U$?

**Example:** Let $U = \{u_1, u_2, u_3, u_4, u_5\}$ and $S = \{S_1, S_2, S_3, S_4\}$, where $S_1 = \{u_1, u_3\}$, $S_2 = \{u_2, u_4\}$, $S_3 = \{u_3, u_5\}$ and $S_4 = \{u_2, u_5\}$.

- With $r = 3$, we have a "YES" instance of MSC: choose $S'$ $=\{S_1, S_2, S_3\}$ as a solution.

**Minimum Set Cover** (MSC):

Instance: A universal set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $S = \{S_1, S_2, \ldots, S_m\}$, where each $S_j$ is a subset of $U$ $(1 \leq j \leq m)$ and an integer $r \leq m$.

Question: Is there is a subcollection $S'$ of $S$ such that $|S'| \leq r$ and the union of the sets in $S'$ is equal to $U$?

**Example:** Let $U = \{u_1, u_2, u_3, u_4, u_5\}$ and $S = \{S_1, S_2, S_3, S_4\}$, where $S_1 = \{u_1, u_3\}$, $S_2 = \{u_2, u_4\}$, $S_3 = \{u_3, u_5\}$ and $S_4 = \{u_2, u_5\}$.

- With $r = 3$, we have a "YES" instance of MSC: choose $S'$ $=\{S_1, S_2, S_3\}$ as a solution.
- With $r = 2$, we have a "NO" instance of MSC. **(Why?)**

**An Application of MSC in Software Testing:**

- Suppose the source code for some software has $N$ lines numbered 1 through $N$. ($U = \{1, 2, \ldots, N\}$.)

**An Application of MSC in Software Testing:**

- Suppose the source code for some software has $N$ lines numbered 1 through $N$. ($U = \{1, 2, \ldots, N\}$.)
- For a test input $t_i$, let $S_i \subseteq U$ be the subset of lines of the source code reached by $t_i$.

**An Application of MSC in Software Testing:**

- Suppose the source code for some software has $N$ lines numbered 1 through $N$. ($U = \{1, 2, \dots, N\}$.)
- For a test input $t_i$, let $S_i \subseteq U$ be the subset of lines of the source code reached by $t_i$.
- Let $T = \{t_1, t_2, \dots, t_m\}$ consist of $m$ tests such that running all the tests in $T$ will reach all the lines.

**An Application of MSC in Software Testing:**

- Suppose the source code for some software has $N$ lines numbered 1 through $N$. ($U = \{1, 2, \ldots, N\}$.)
- For a test input $t_i$, let $S_i \subseteq U$ be the subset of lines of the source code reached by $t_i$.
- Let $T = \{t_1, t_2, \ldots, t_m\}$ consist of $m$ tests such that running all the tests in $T$ will reach all the lines.
- There may be smaller test set $T' \subseteq T$ that may also be sufficient.

**An Application of MSC in Software Testing:**

- Suppose the source code for some software has $N$ lines numbered 1 through $N$. ($U = \{1, 2, \ldots, N\}$.)
- For a test input $t_i$, let $S_i \subseteq U$ be the subset of lines of the source code reached by $t_i$.
- Let $T = \{t_1, t_2, \ldots, t_m\}$ consist of $m$ tests such that running all the tests in $T$ will reach all the lines.
- There may be smaller test set $T' \subseteq T$ that may also be sufficient.
- Finding $T'$ is exactly the MSC problem where $U = \{1, 2, \ldots, N\}$, the set $S_i \subseteq U$ corresponds to test $t_i$ ($1 \leq i \leq n$).

**Theorem 1:** MSC is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $S'$ to MSC is a subcollection of $S$.)

**Theorem 1:** MSC is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $S'$ to MSC is a subcollection of $S$.)
- Proof of **NP**-hardness: Reduction from Minimum Vertex Cover (MVC).

**Theorem 1:** MSC is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $S'$ to MSC is a subcollection of $S$.)
- Proof of **NP**-hardness: Reduction from Minimum Vertex Cover (MVC).
- **Recall:** An MVC instance has graph $G(V, E)$ and integer $k$. The question is whether $G$ has a vertex cover of size $\leq k$.

**Theorem 1:** MSC is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $S'$ to MSC is a subcollection of $S$.)
- Proof of **NP**-hardness: Reduction from Minimum Vertex Cover (MVC).
- **Recall:** An MVC instance has graph $G(V, E)$ and integer $k$. The question is whether $G$ has a vertex cover of size $\leq k$.
- **Intuitive idea:**

**Theorem 1:** MSC is **NP**-complete.

**Proof:**

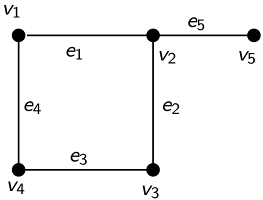
- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $S'$ to MSC is a subcollection of $S$.)
- Proof of **NP**-hardness: Reduction from Minimum Vertex Cover (MVC).
- **Recall:** An MVC instance has graph $G(V, E)$ and integer $k$. The question is whether $G$ has a vertex cover of size $\leq k$.
- **Intuitive idea:**
  - Edges in MVC become elements in MSC.

**Theorem 1:** MSC is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $S'$ to MSC is a subcollection of $S$.)
- Proof of **NP**-hardness: Reduction from Minimum Vertex Cover (MVC).
- **Recall:** An MVC instance has graph $G(V, E)$ and integer $k$. The question is whether $G$ has a vertex cover of size $\leq k$.
- **Intuitive idea:**
  - Edges in MVC become elements in MSC.
  - Nodes in MVC become sets in MSC.

**An Example to Illustrate the Reduction:**




MVC Instance with $k = 2$

**Resulting MSC instance:**

- $U = \{u_1, u_2, u_3, u_4, u_5\}$
- $S = \{S_1, S_2, S_3, S_4, S_5\}$, where
  $S_1 = \{u_1, u_4\}$ $S_2 = \{u_1, u_2, u_5\}$
  $S_3 = \{u_2, u_3\}$ $S_4 = \{u_3, u_4\}$
  $S_5 = \{u_5\}$
- Bound $r$ on the number of sets $= 2$

**Steps of the Reduction from MVC:**

- MVC instance $I$ has graph $G(V, E)$ and integer $k$. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$.
  (Size of the MVC instance $= O(m + n)$.)

**Steps of the Reduction from MVC:**

- MVC instance $I$ has graph $G(V, E)$ and integer $k$. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. (Size of the MVC instance $= O(m + n)$.)
- MSC instance $I'$ has $U$, $S$ and integer $r$.

**Steps of the Reduction from MVC:**

- MVC instance $I$ has graph $G(V, E)$ and integer $k$. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. (Size of the MVC instance $= O(m + n)$.)
- MSC instance $I'$ has $U$, $S$ and integer $r$.
- Construct $U = \{u_1, u_2, \ldots, u_m\}$. (Thus, element $u_i$ corresponds to edge $e_i$, $1 \leq i \leq m$.) **[Time: $O(m)$]**

**Steps of the Reduction from MVC (continued):**

- Construct $S = \{S_1, S_2, \ldots, S_n\}$.

**Steps of the Reduction from MVC (continued):**

- Construct $S = \{S_1, S_2, \ldots, S_n\}$.
- Set $S_j$ corresponding to node $v_j$ is chosen as follows: for each edge $e_i$ that touches node $v_j$, add the corresponding element $u_i$ to $S_j$. **[Time: $O(m)$]**

**Steps of the Reduction from MVC (continued):**

- Construct $S = \{S_1, S_2, \ldots, S_n\}$.
- Set $S_j$ corresponding to node $v_j$ is chosen as follows: for each edge $e_i$ that touches node $v_j$, add the corresponding element $u_i$ to $S_j$. **[Time: $O(m)$]**
- Set $r$ (bound on the number of sets in MSC) $= k$ (bound on the number of nodes in MVC). **[Time: $O(1)$]**

**Steps of the Reduction from MVC (continued):**

- Construct $S = \{S_1, S_2, \ldots, S_n\}$.
- Set $S_j$ corresponding to node $v_j$ is chosen as follows: for each edge $e_i$ that touches node $v_j$, add the corresponding element $u_i$ to $S_j$. **[Time: $O(m)$]**
- Set $r$ (bound on the number of sets in MSC) $= k$ (bound on the number of nodes in MVC). **[Time: $O(1)$]**
- Total time for the reduction $= O(m + n)$. So, the reduction is efficient.

**Correctness:**

**Part 1:** Suppose MSC instance $I'$ has a solution. (We must show that the MVC instance $I$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ be a solution to MSC, where $\ell \leq r = k$.

**Correctness:**

**Part 1:** Suppose MSC instance $I'$ has a solution. (We must show that the MVC instance $I$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ be a solution to MSC, where $\ell \leq r = k$.
- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$.

**Correctness:**

**Part 1:** Suppose MSC instance $I'$ has a solution. (We must show that the MVC instance $I$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ be a solution to MSC, where $\ell \leq r = k$.
- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$.
- **Claim:** $V'$ is a solution to MVC.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**
  - $|V'| = \ell < r = k$. (So $V'$ has the right size.)

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**
  - $|V'| = \ell < r = k$. (So $V'$ has the right size.)
  - Consider any edge $e_x = \{v_i, v_j\}$ of $G$.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**
  - $|V'| = \ell < r = k$. (So $V'$ has the right size.)
  - Consider any edge $e_x = \{v_i, v_j\}$ of $G$.
  - The element $u_x$ corresponding to $e_x$ appears only in sets $S_i$ and $S_j$.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**
  - $|V'| = \ell < r = k$. (So $V'$ has the right size.)
  - Consider any edge $e_x = \{v_i, v_j\}$ of $G$.
  - The element $u_x$ corresponding to $e_x$ appears only in sets $S_i$ and $S_j$.
  - $S'$ must contain at least one of $S_i$ and $S_j$ (since $S'$ is a valid set cover).

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**
  - $|V'| = \ell < r = k$. (So $V'$ has the right size.)
  - Consider any edge $e_x = \{v_i, v_j\}$ of $G$.
  - The element $u_x$ corresponding to $e_x$ appears only in sets $S_i$ and $S_j$.
  - $S'$ must contain at least one of $S_i$ and $S_j$ (since $S'$ is a valid set cover).
  - So, $V'$ contains at least one of $v_i$ and $v_j$.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MVC.
- **Reason:**
  - $|V'| = \ell < r = k$. (So $V'$ has the right size.)
  - Consider any edge $e_x = \{v_i, v_j\}$ of $G$.
  - The element $u_x$ corresponding to $e_x$ appears only in sets $S_i$ and $S_j$.
  - $S'$ must contain at least one of $S_i$ and $S_j$ (since $S'$ is a valid set cover).
  - So, $V'$ contains at least one of $v_i$ and $v_j$.
  - Thus, $V'$ is a solution to the MVC instance $I'$.

**Part 2:** Suppose MVC instance $I$ has a solution. (We need to show that the MSC instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$ be a solution to MVC, where $\ell \leq k = r$.

**Part 2:** Suppose MVC instance $I$ has a solution. (We need to show that the MSC instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$ be a solution to MVC, where $\ell \leq k = r$.
- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$.

**Part 2:** Suppose MVC instance $I$ has a solution. (We need to show that the MSC instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$ be a solution to MVC, where $\ell \leq k = r$.
- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$.
- **Claim:** $S'$ is a solution to MSC.

**Part 2:** Suppose MVC instance $I$ has a solution. (We need to show that the MSC instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$ be a solution to MVC, where $\ell \leq k = r$.
- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$.
- **Claim:** $S'$ is a solution to MSC.
- Proof similar to that of Part 1. **(Reading exercise)**

**Part 2:** Suppose MVC instance $I$ has a solution. (We need to show that the MSC instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_\ell\}$ be a solution to MVC, where $\ell \leq k = r$.
- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$.
- **Claim:** $S'$ is a solution to MSC.
- Proof similar to that of Part 1. **(Reading exercise)**
- Thus, MSC is **NP**-complete.

# Additional Proofs of **NP**-completeness

**Review:** 3SAT and Maximum Independent Set (MIS)

- **3SAT**:

  Instance: A set $X = \{x_1, x_2, \ldots, x_n\}$ of Boolean variables and a set $F = \{C_1, C_2, \ldots, C_m\}$ of $m$ clauses using the variables in $X$. Each clause has *exactly* 3 literals.

  Question: Is $F$ satisfiable?

**Review:** 3SAT and Maximum Independent Set (MIS)

- **3SAT**:

  Instance: A set $X = \{x_1, x_2, \ldots, x_n\}$ of Boolean variables and a set $F = \{C_1, C_2, \ldots, C_m\}$ of $m$ clauses using the variables in $X$. Each clause has *exactly* 3 literals.

  Question: Is $F$ satisfiable?

- **Maximum Independent Set (MIS)**:

  Instance: An undirected graph $G(V, E)$ and and an integer $\ell \leq |V|$.

  Question: Does $G$ have an **independent set** with at least $\ell$ vertices?

# Additional Proofs of **NP**-completeness

**Theorem 2:** MIS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MIS is a subset of nodes.)

# Additional Proofs of **NP**-completeness

**Theorem 2:** MIS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MIS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from 3SAT.

**Theorem 2:** MIS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MIS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from 3SAT.
- **Intuitive idea:**

**Theorem 2:** MIS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MIS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from 3SAT.
- **Intuitive idea:**
  - From each clause of 3SAT construct a subgraph (of the eventual graph of MIS).

**Theorem 2:** MIS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MIS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from 3SAT.
- **Intuitive idea:**
  - From each clause of 3SAT construct a subgraph (of the eventual graph of MIS).
  - 3SAT requires at least one literal to have the value 1 in each clause. This corresponds to choosing one vertex from each subgraph in the independent set.

# Additional Proofs of **NP**-completeness

**Theorem 2:** MIS is **NP**-complete.

**Proof:**

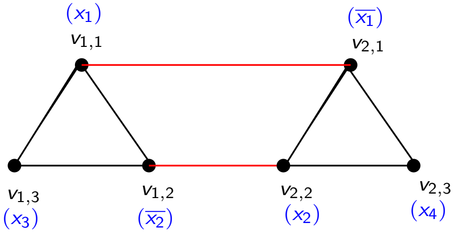
- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MIS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from 3SAT.
- **Intuitive idea:**
  - From each clause of 3SAT construct a subgraph (of the eventual graph of MIS).
  - 3SAT requires at least one literal to have the value 1 in each clause. This corresponds to choosing one vertex from each subgraph in the independent set.
  - Must avoid **conflicts** in assignments to 3SAT (i.e., two *complementary* literals should not both be set to 1); this is ensured by adding suitable edges in the MIS instance.

**Example to Illustrate the reduction from 3SAT to MIS:**

**3SAT Instance:**

$$X = \{x_1, x_2, x_3, x_4\}$$
$$F = \{C_1, C_2\}, \text{ where}$$
$$C_1 = (x_1 \vee \overline{x_2} \vee x_3)$$
$$C_2 = \{\overline{x_1} \vee x_2 \vee x_4)$$

**Resulting MIS Instance:**




Indep. set size $\ell = 2$

**Steps of the Reduction from 3SAT to MIS:**

- Size of the 3SAT instance $I = O(m)$.

**Steps of the Reduction from 3SAT to MIS:**

- Size of the 3SAT instance $I = O(m)$.
- For each clause $C_j$, construct a 3-node complete graph $G_j$. (Each node of $G_j$ corresponds to a literal in $C_j$.) **[Time: $O(m)$]**

# Additional Proofs of **NP**-completeness

**Steps of the Reduction from 3SAT to MIS:**

- Size of the 3SAT instance $I = O(m)$.
- For each clause $C_j$, construct a 3-node complete graph $G_j$. (Each node of $G_j$ corresponds to a literal in $C_j$.) **[Time: $O(m)$]**
- **Note:** From each $G_j$, only one node can be chosen in any independent set.

**Steps of the Reduction from 3SAT to MIS:**

- Size of the 3SAT instance $I = O(m)$.
- For each clause $C_j$, construct a 3-node complete graph $G_j$. (Each node of $G_j$ corresponds to a literal in $C_j$.) **[Time: $O(m)$]**
- **Note:** From each $G_j$, only one node can be chosen in any independent set.
- For any pair of subgraphs $G_p$ and $G_q$ ($p \neq q$), if a node $v$ in $G_p$ and a node $w$ in $G_q$ correspond to complementary literals, add the edge $\{v, w\}$. (These are **conflict** edges.) **[Time: $O(m^2)$]**

**Steps of the Reduction from 3SAT to MIS:**

- Size of the 3SAT instance $I = O(m)$.
- For each clause $C_j$, construct a 3-node complete graph $G_j$. (Each node of $G_j$ corresponds to a literal in $C_j$.) **[Time: $O(m)$]**
- **Note:** From each $G_j$, only one node can be chosen in any independent set.
- For any pair of subgraphs $G_p$ and $G_q$ ($p \neq q$), if a node $v$ in $G_p$ and a node $w$ in $G_q$ correspond to complementary literals, add the edge $\{v, w\}$. (These are **conflict** edges.) **[Time: $O(m^2)$]**
- The size $\ell$ of independent set $= m$ (number of clauses). **[Time: $O(1)$]**

# Additional Proofs of **NP**-completeness

**Steps of the Reduction from 3SAT to MIS:**

- Size of the 3SAT instance $I = O(m)$.
- For each clause $C_j$, construct a 3-node complete graph $G_j$. (Each node of $G_j$ corresponds to a literal in $C_j$.) **[Time: $O(m)$]**
- **Note:** From each $G_j$, only one node can be chosen in any independent set.
- For any pair of subgraphs $G_p$ and $G_q$ ($p \neq q$), if a node $v$ in $G_p$ and a node $w$ in $G_q$ correspond to complementary literals, add the edge $\{v, w\}$. (These are **conflict** edges.) **[Time: $O(m^2)$]**
- The size $\ell$ of independent set $= m$ (number of clauses). **[Time: $O(1)$]**
- Time used by the reduction $= O(m^2)$.

**Correctness:**

**Part 1:** Suppose 3SAT instance $I$ has a solution. (Must show that the MIS instance $I'$ has a solution.)

- Each clause $C_j$ of 3SAT has one or more literals with value 1. Choose one such literal $a_j$ from $c_j$ and $v_j$ be the node corresponding to $a_j$ in $G_j$, $1 \leq j \leq m$.

# Additional Proofs of **NP**-completeness

**Correctness:**

**Part 1:** Suppose 3SAT instance $I$ has a solution. (Must show that the MIS instance $I'$ has a solution.)

- Each clause $C_j$ of 3SAT has one or more literals with value 1. Choose one such literal $a_j$ from $c_j$ and $v_j$ be the node corresponding to $a_j$ in $G_j$, $1 \leq j \leq m$.
- Let $V' = \{v_1, v_2, \ldots, v_m\}$.

**Correctness:**

**Part 1:** Suppose 3SAT instance $I$ has a solution. (Must show that the MIS instance $I'$ has a solution.)

- Each clause $C_j$ of 3SAT has one or more literals with value 1. Choose one such literal $a_j$ from $c_j$ and $v_j$ be the node corresponding to $a_j$ in $G_j$, $1 \leq j \leq m$.
- Let $V' = \{v_1, v_2, \ldots, v_m\}$.
- **Claim:** $V'$ is a solution to MIS.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.
- **Reason:**

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.
- **Reason:**
  - $|V'| = m$. (So, $V'$ has the right size.)

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.
- **Reason:**
  - $|V'| = m$. (So, $V'$ has the right size.)
  - Suppose $V'$ is not an independent set. Then there is an edge $e = \{v_i, v_j\}$ in the MIS graph $G$.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.
- **Reason:**
  - $|V'| = m$. (So, $V'$ has the right size.)
  - Suppose $V'$ is not an independent set. Then there is an edge $e = \{v_i, v_j\}$ in the MIS graph $G$.
  - $v_i$ and $v_j$ are from different subgraphs; that is, $e$ is a **conflict** edge.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.
- **Reason:**
  - $|V'| = m$. (So, $V'$ has the right size.)
  - Suppose $V'$ is not an independent set. Then there is an edge $e = \{v_i, v_j\}$ in the MIS graph $G$.
  - $v_i$ and $v_j$ are from different subgraphs; that is, $e$ is a **conflict** edge.
  - So, the literals corresponding to $v_i$ and $v_j$ are *complements* of each other.

**Correctness Part 1 (continued):**

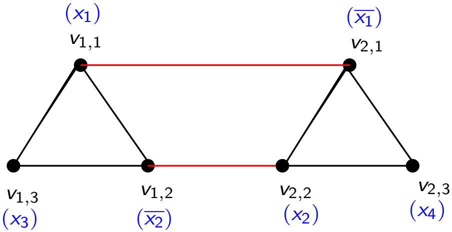
- **Claim:** $V'$ is a solution to MIS.
- **Reason:**
  - $|V'| = m$. (So, $V'$ has the right size.)
  - Suppose $V'$ is not an independent set. Then there is an edge $e = \{v_i, v_j\}$ in the MIS graph $G$.
  - $v_i$ and $v_j$ are from different subgraphs; that is, $e$ is a **conflict** edge.
  - So, the literals corresponding to $v_i$ and $v_j$ are *complements* of each other.
  - Hence, the given solution to 3SAT sets both a variable and its complement to 1, a contradiction.

**Correctness Part 1 (continued):**

- **Claim:** $V'$ is a solution to MIS.
- **Reason:**
  - $|V'| = m$. (So, $V'$ has the right size.)
  - Suppose $V'$ is not an independent set. Then there is an edge $e = \{v_i, v_j\}$ in the MIS graph $G$.
  - $v_i$ and $v_j$ are from different subgraphs; that is, $e$ is a **conflict** edge.
  - So, the literals corresponding to $v_i$ and $v_j$ are *complements* of each other.
  - Hence, the given solution to 3SAT sets both a variable and its complement to 1, a contradiction.
  - Thus, $V'$ is a solution to the MIS instance $I'$.

**Part 2:** Suppose MIS instance $I'$ has a solution. (Must show that the 3SAT instance $I'$ has a solution.)

**Example:**




- Suppose MIS solution $V' = \{v_{1,2}, v_{2,3}\}$.
- The corresponding literals: $\overline{x_2}$, $x_4$.
- Set $x_2 = 0$ and $x_4 = 1$.
- Set the remaining variables $x_1$ and $x_3$ to 0.

# Additional Proofs of **NP**-completeness

**Part 2:** Suppose MIS instance $I'$ has a solution. (Must show that the 3SAT instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_m\}$ be a solution to MIS.

# Additional Proofs of **NP**-completeness

**Part 2:** Suppose MIS instance $I'$ has a solution. (Must show that the 3SAT instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_m\}$ be a solution to MIS.
- Construct a solution to 3SAT as follows:

# Additional Proofs of **NP**-completeness

**Part 2:** Suppose MIS instance $I'$ has a solution. (Must show that the 3SAT instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_m\}$ be a solution to MIS.
- Construct a solution to 3SAT as follows:
  - For each $v_i \in V'$, let $a_i$ be the corresponding literal in the 3SAT instance. Set $a_i$ to 1 (and thus $\overline{a_i}$ to 0).

# Additional Proofs of **NP**-completeness

**Part 2:** Suppose MIS instance $I'$ has a solution. (Must show that the 3SAT instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_m\}$ be a solution to MIS.
- Construct a solution to 3SAT as follows:
  - For each $v_i \in V'$, let $a_i$ be the corresponding literal in the 3SAT instance. Set $a_i$ to 1 (and thus $\overline{a_i}$ to 0).
  - If there is a variable $x_b$ that has not been assigned a value, set $x_b$ to 0.

**Part 2:** Suppose MIS instance $I'$ has a solution. (Must show that the 3SAT instance $I'$ has a solution.)

- Let $V' = \{v_1, v_2, \ldots, v_m\}$ be a solution to MIS.
- Construct a solution to 3SAT as follows:
  - For each $v_i \in V'$, let $a_i$ be the corresponding literal in the 3SAT instance. Set $a_i$ to 1 (and thus $\overline{a_i}$ to 0).
  - If there is a variable $x_b$ that has not been assigned a value, set $x_b$ to 0.
- **Claim:** We have a solution to the 3SAT instance $I$.

- **Claim:** We have a solution to the 3SAT instance $I$.

# Additional Proofs of **NP**-completeness

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**
  - Each variable has been assigned a value.

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**
  - Each variable has been assigned a value.
  - Since $V'$ is an independent set, the chosen assignment does not have any conflicts.

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**
  - Each variable has been assigned a value.
  - Since $V'$ is an independent set, the chosen assignment does not have any conflicts.
  - Since $|V'| = m$, for each subgraph $G_j$, $V'$ has node, say $v_j$ from $G_j$.

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**
  - Each variable has been assigned a value.
  - Since $V'$ is an independent set, the chosen assignment does not have any conflicts.
  - Since $|V'| = m$, for each subgraph $G_j$, $V'$ has node, say $v_j$ from $G_j$.
  - consider any clause $C_j$ and let $v_j$ be the node in $V'$ from subgraph $G_j$.

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**
  - Each variable has been assigned a value.
  - Since $V'$ is an independent set, the chosen assignment does not have any conflicts.
  - Since $|V'| = m$, for each subgraph $G_j$, $V'$ has node, say $v_j$ from $G_j$.
  - consider any clause $C_j$ and let $v_j$ be the node in $V'$ from subgraph $G_j$.
  - The literal $y$ corresponding to $v_j$ is in clause $C_j$ and $y$ has been set to 1, $C_j$ is satisfied.

# Additional Proofs of **NP**-completeness

- **Claim:** We have a solution to the 3SAT instance $I$.
- **Reason:**
  - Each variable has been assigned a value.
  - Since $V'$ is an independent set, the chosen assignment does not have any conflicts.
  - Since $|V'| = m$, for each subgraph $G_j$, $V'$ has node, say $v_j$ from $G_j$.
  - consider any clause $C_j$ and let $v_j$ be the node in $V'$ from subgraph $G_j$.
  - The literal $y$ corresponding to $v_j$ is in clause $C_j$ and $y$ has been set to 1, $C_j$ is satisfied.
- Thus, MIS is **NP**-complete.

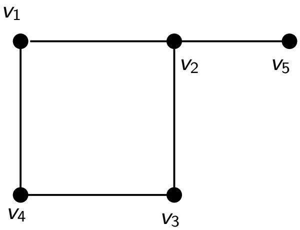
**Minimum Dominating Set (MDS)**

Instance: An undirected graph $G(V, E)$ and an integer $r \leq |V|$.

Question: Does $G$ have a **dominating set** of size at most $r$, that is, is there a subset $V' \subseteq V$ such that $|V'| \leq r$ and for each node $v_i \in V - V'$, there is some node $v_j \in V'$ such that $\{v_i, v_j\} \in E$?

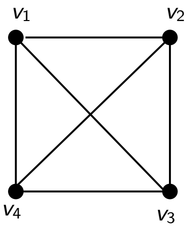
**Note:** Suppose $V'$ is a dominating set. We say that each node in $V - V'$ is **dominated** by a node in $V'$.

**Example:**



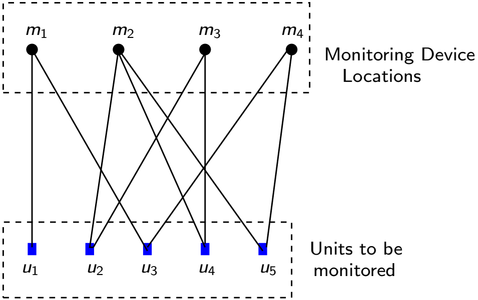

- Here, $V_1 = \{v_4, v_5\}$ is a dominating set; $v_4$ dominates $v_1$ and $v_3$ while $v_5$ dominates $v_2$. (It is also a minimum dominating set.)
- $V_2 = \{v_1, v_3\}$ is not a dominating set since $v_5$ is not dominated by any vertex in $V_2$.

**Vertex Cover and Dominating Set:**




- For the above graph, any vertex cover must have at least 3 vertices.
- However, a dominating set needs only one vertex. (That vertex dominates the other three.)

**An Application of Dominating Sets:**




Minimum set of monitoring devices needed $= \{m_1, m_2\}$

**Theorem 3:** MDS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MDS is a subset of nodes.)

**Theorem 3:** MDS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MDS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from MSC.

**Theorem 3:** MDS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MDS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from MSC.
- **Intuitive idea:**

**Theorem 3:** MDS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MDS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from MSC.
- **Intuitive idea:**
  - The graph for MDS has two classes of nodes (one to represent sets and the other to represent elements of MSC).

# Additional Proofs of **NP**-completeness

**Theorem 3:** MDS is **NP**-complete.

**Proof:**

- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MDS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from MSC.
- **Intuitive idea:**
  - The graph for MDS has two classes of nodes (one to represent sets and the other to represent elements of MSC).
  - "Set covering an element" corresponds to "set node dominating an element node".

**Theorem 3:** MDS is **NP**-complete.

**Proof:**

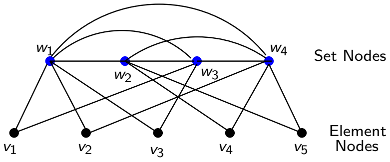
- Membership in **NP**: **Exercise**. (**Hint:** Proposed solution $V'$ for MDS is a subset of nodes.)
- Proof of **NP**-hardness: Reduction from MSC.
- **Intuitive idea:**
  - The graph for MDS has two classes of nodes (one to represent sets and the other to represent elements of MSC).
  - "Set covering an element" corresponds to "set node dominating an element node".
  - Make sure that there is a solution to MDS consisting only of nodes representing sets.

**Example to Illustrate the reduction from MSC to MDS:**

**MSC Instance:**

- $U = \{u_1, u_2, u_3, u_4, u_5\}$
- $S = \{S_1, S_2, S_3, S_4\}$ where
$$\begin{array}{rcl} S_1 & = & \{u_1, u_2, u_3\} \\ S_2 & = & \{u_4, u_5\} \\ S_3 & = & \{u_1, u_3\} \\ S_4 & = & \{u_2, u_4, u_5\} \end{array}$$
- $k = 2$

**MDS Instance:**




$r = 2$

**Note:** Set nodes are in **blue**.

**Steps of the Reduction from MSC to MDS:**

- In the given MSC instance, $U = \{u_1, u_2, \ldots, u_n\}$ and $S = \{S_1, S_2, \ldots, S_m\}$. (Size of MSC instance $I = O(mm)$.)

**Steps of the Reduction from MSC to MDS:**

- In the given MSC instance, $U = \{u_1, u_2, \ldots, u_n\}$ and $S = \{S_1, S_2, \ldots, S_m\}$. (Size of MSC instance $I = O(mm)$.)
- For the MDS instance $I'$, the node set $V = V_1 \cup V_2$, where $V_1 = \{v_1, v_2, \ldots, v_n\}$ corresponds to elements and $V_2 = \{w_1, w_2, \ldots, w_n\}$ corresponds to sets. (Thus, $V_1$ has **element nodes** and $V_2$ has **set nodes**.) **[Time: $O(m + n)$]**

## Steps of the Reduction from MSC to MDS:

- In the given MSC instance, $U = \{u_1, u_2, \ldots, u_n\}$ and $S = \{S_1, S_2, \ldots, S_m\}$. (Size of MSC instance $I = O(mm)$.)
- For the MDS instance $I'$, the node set $V = V_1 \cup V_2$, where $V_1 = \{v_1, v_2, \ldots, v_n\}$ corresponds to elements and $V_2 = \{w_1, w_2, \ldots, w_n\}$ corresponds to sets. (Thus, $V_1$ has **element nodes** and $V_2$ has **set nodes**.) **[Time: $O(m+n)$]**
- The edge set for MDS $E = E_1 \cup E_2$, where

$$E_1 = \{\{v_i, w_j\} \ : \ u_i \in S_j\} \ \text{ and } \ E_2 = \{\{w_i, w_j\} \ : \ _i \neq j\}$$

Thus, $E_1$ has the membership edges and $E_2$ connects the nodes in $V_2$ as a complete graph (or clique). **[Time: $O(mn + m^2)$]**

## Steps of the Reduction from MSC to MDS:

- In the given MSC instance, $U = \{u_1, u_2, \ldots, u_n\}$ and $S = \{S_1, S_2, \ldots, S_m\}$. (Size of MSC instance $I = O(mm)$.)
- For the MDS instance $I'$, the node set $V = V_1 \cup V_2$, where $V_1 = \{v_1, v_2, \ldots, v_n\}$ corresponds to elements and $V_2 = \{w_1, w_2, \ldots, w_n\}$ corresponds to sets. (Thus, $V_1$ has **element nodes** and $V_2$ has **set nodes**.) **[Time: $O(m+n)$]**
- The edge set for MDS $E = E_1 \cup E_2$, where

$$E_1 = \{\{v_i, w_j\} \ : \ u_i \in S_j\} \ \text{ and } \ E_2 = \{\{w_i, w_j\} \ : \ {}_i \neq j\}$$

  Thus, $E_1$ has the membership edges and $E_2$ connects the nodes in $V_2$ as a complete graph (or clique). **[Time: $O(mn + m^2)$]**
- The size $r$ of dominating set $= k$ (the size of set cover). **[Time: $O(1)$]**

## Steps of the Reduction from MSC to MDS:

- In the given MSC instance, $U = \{u_1, u_2, \ldots, u_n\}$ and $S = \{S_1, S_2, \ldots, S_m\}$. (Size of MSC instance $I = O(mm)$.)
- For the MDS instance $I'$, the node set $V = V_1 \cup V_2$, where $V_1 = \{v_1, v_2, \ldots, v_n\}$ corresponds to elements and $V_2 = \{w_1, w_2, \ldots, w_n\}$ corresponds to sets. (Thus, $V_1$ has **element nodes** and $V_2$ has **set nodes**.) **[Time: $O(m+n)$]**
- The edge set for MDS $E = E_1 \cup E_2$, where

$$E_1 = \{\{v_i, w_j\} \ : \ u_i \in S_j\} \ \text{ and } \ E_2 = \{\{w_i, w_j\} \ : \ _i \neq j\}$$

  Thus, $E_1$ has the membership edges and $E_2$ connects the nodes in $V_2$ as a complete graph (or clique). **[Time: $O(mn + m^2)$]**
- The size $r$ of dominating set $= k$ (the size of set cover). **[Time: $O(1)$]**
- Time used by the reduction $= O(mn + m^2)$.

**Correctness:**

**Part 1:** Suppose MSC instance $I$ has a solution. (Must show that the MDS instance $I'$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ for some $\ell \leq k = r$ be a solution to the MSC instance.

**Correctness:**

**Part 1:** Suppose MSC instance $I$ has a solution. (Must show that the MDS instance $I'$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ for some $\ell \leq k = r$ be a solution to the MSC instance.
- Let $V' = \{w_1, w_2, \ldots, w_\ell\}$ be the set nodes corresponding to the sets in $S'$.

**Correctness:**

**Part 1:** Suppose MSC instance $I$ has a solution. (Must show that the MDS instance $I'$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ for some $\ell \leq k = r$ be a solution to the MSC instance.
- Let $V' = \{w_1, w_2, \ldots, w_\ell\}$ be the set nodes corresponding to the sets in $S'$.
- **Claim:** $V'$ is a solution to the MDS instance $I'$.

**Correctness:**

**Part 1:** Suppose MSC instance $I$ has a solution. (Must show that the MDS instance $I'$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ for some $\ell \leq k = r$ be a solution to the MSC instance.
- Let $V' = \{w_1, w_2, \ldots, w_\ell\}$ be the set nodes corresponding to the sets in $S'$.
- **Claim:** $V'$ is a solution to the MDS instance $I'$.
- $|V'| \leq r$. So, $V'$ is of the right size for MDS.

**Correctness:**

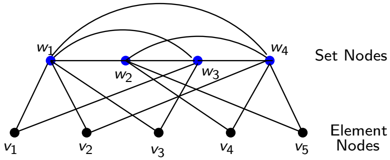
**Part 1:** Suppose MSC instance $I$ has a solution. (Must show that the MDS instance $I'$ has a solution.)

- Let $S' = \{S_1, S_2, \ldots, S_\ell\}$ for some $\ell \leq k = r$ be a solution to the MSC instance.
- Let $V' = \{w_1, w_2, \ldots, w_\ell\}$ be the set nodes corresponding to the sets in $S'$.
- **Claim:** $V'$ is a solution to the MDS instance $I'$.
- $|V'| \leq r$. So, $V'$ is of the right size for MDS.
- Show that $V'$ is a dominating set. **(Reading exercise)**

**Correctness:**

**Part 2:** Suppose MDS instance $I'$ has a solution. (Must show that the MSC instance $I$ has a solution.)

**Why is this proof different from Part 1?**




- Suppose $r = 3$.
- One possible solution to MDS is $D' = \{w_1, v_4, v_5\}$. (This solution has element nodes as well.)
- We can "convert" this solution to $D = \{w_1, w_2, w_4\}$ which contains only set nodes and has size 3.

**Correctness:**

**Part 2:** Suppose MDS instance $I'$ has a solution. (Must show that the MSC instance $I$ has a solution.)

- Let $D'$ be a solution to the MDS instance $I'$. (Thus, $|D'| \leq k$.)

**Correctness:**

**Part 2:** Suppose MDS instance $I'$ has a solution. (Must show that the MSC instance $I$ has a solution.)

- Let $D'$ be a solution to the MDS instance $I'$. (Thus, $|D'| \leq k$.)
- Partition $D'$ into $D_1$ and $D_2$ so that $D_1 \subseteq V_1$ (element nodes) and $D_2 \subseteq V_2$ (set nodes).

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)
  - Delete $v$ from $D_1$ and add $w$ to $D_2$ (if $w$ is not already in $D_2$).

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)
  - Delete $v$ from $D_1$ and add $w$ to $D_2$ (if $w$ is not already in $D_2$).
- Now, $D_2$ has only set nodes and $|D_2| \leq k$.

# Additional Proofs of **NP**-completeness

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)
  - Delete $v$ from $D_1$ and add $w$ to $D_2$ (if $w$ is not already in $D_2$).
- Now, $D_2$ has only set nodes and $|D_2| \leq k$.
- Let $D_2 = \{w_1, w_2, \ldots, w_\ell\}$ for some $\ell \leq k$.

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)
  - Delete $v$ from $D_1$ and add $w$ to $D_2$ (if $w$ is not already in $D_2$).
- Now, $D_2$ has only set nodes and $|D_2| \leq k$.
- Let $D_2 = \{w_1, w_2, \ldots, w_\ell\}$ for some $\ell \leq k$.
- Construct $S' = \{S_1, S_2, \ldots, S_\ell\}$. (Note that $|S| \leq k$).

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)
  - Delete $v$ from $D_1$ and add $w$ to $D_2$ (if $w$ is not already in $D_2$).
- Now, $D_2$ has only set nodes and $|D_2| \leq k$.
- Let $D_2 = \{w_1, w_2, \ldots, w_\ell\}$ for some $\ell \leq k$.
- Construct $S' = \{S_1, S_2, \ldots, S_\ell\}$. (Note that $|S| \leq k$).
- Showing that $S'$ is a solution to MSC is similar to Part 1.

**Correctness (continued):**

- If $D_1$ is nonempty, modify $D'$ repeatedly as follows until $D_1$ becomes empty:
  - Take any node $v \in D_1$.
  - Find a node $w \in V_2$ such that $\{v, w\}$ is an edge. (Such a node must exist since each element occurs in some set.)
  - Delete $v$ from $D_1$ and add $w$ to $D_2$ (if $w$ is not already in $D_2$).
- Now, $D_2$ has only set nodes and $|D_2| \leq k$.
- Let $D_2 = \{w_1, w_2, \ldots, w_\ell\}$ for some $\ell \leq k$.
- Construct $S' = \{S_1, S_2, \ldots, S_\ell\}$. (Note that $|S| \leq k$).
- Showing that $S'$ is a solution to MSC is similar to Part 1.
- Thus, MDS is **NP**-complete.

# Coping with **NP**-completeness in Practice

- Many optimization problems arising in practice are **NP**-hard.
- Two methods for coping with **NP**-hardness:
  1. Express the problem under the mathematical programming framework and use available software (e.g., Gurobi, CPLEX) for solving such problems.

# Coping with **NP**-completeness in Practice

- Many optimization problems arising in practice are **NP**-hard.
- Two methods for coping with **NP**-hardness:
  1. Express the problem under the mathematical programming framework and use available software (e.g., Gurobi, CPLEX) for solving such problems.
  2. Develop efficient approximation algorithms.

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Consider the optimization version of MVC (i.e., given $G(V, E)$, find a minimum vertex cover for $G$).

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Consider the optimization version of MVC (i.e., given $G(V, E)$, find a minimum vertex cover for $G$).
- **Integer Linear Program** (ILP): A formulation where the objective function and constraints are *linear* and variables are required to take on *integer* values.

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Consider the optimization version of MVC (i.e., given $G(V, E)$, find a minimum vertex cover for $G$).
- **Integer Linear Program** (ILP): A formulation where the objective function and constraints are *linear* and variables are required to take on *integer* values.
- **{0,1}-ILP**: Same as ILP except that each variable must take on a value from $\{0, 1\}$.

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Given $G(V, E)$, let $V = \{v_1, v_2, \ldots, v_n\}$.

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Given $G(V, E)$, let $V = \{v_1, v_2, \ldots, v_n\}$.
- For each node $v_i$, introduce a $\{0,1\}$-valued variable $x_i$, $1 \leq i \leq n$, with the following significance: $x_i = 1$ if $v_i$ is chosen in the (optimal) solution and 0 otherwise.

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Given $G(V, E)$, let $V = \{v_1, v_2, \ldots, v_n\}$.
- For each node $v_i$, introduce a $\{0,1\}$-valued variable $x_i$, $1 \le i \le n$, with the following significance: $x_i = 1$ if $v_i$ is chosen in the (optimal) solution and 0 otherwise.
- Now, $\sum_{i=1}^{n} x_i$ is the number of nodes chosen in the solution. Therefore, the objective of the ILP is:

$$\text{Minimize} \;\; \sum_{i=1}^{n} x_i$$

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Given $G(V, E)$, let $V = \{v_1, v_2, \ldots, v_n\}$.
- For each node $v_i$, introduce a $\{0,1\}$-valued variable $x_i$, $1 \leq i \leq n$, with the following significance: $x_i = 1$ if $v_i$ is chosen in the (optimal) solution and 0 otherwise.
- Now, $\sum_{i=1}^{n} x_i$ is the number of nodes chosen in the solution. Therefore, the objective of the ILP is:

$$\text{Minimize } \sum_{i=1}^{n} x_i$$

- **Constraints:** For each edge $e = \{v_i, v_j\}$, at least one of $v_i$ and $v_j$ must be in the solution; i.e., at least one of $x_i$ and $x_j$ must be set to 1. So:

$$x_i + x_j \geq 1 \quad \text{for each edge } \{v_i, v_j\}$$

**Illustration – An Integer Linear Program (ILP) for MVC:**

- Given $G(V, E)$, let $V = \{v_1, v_2, \ldots, v_n\}$.
- For each node $v_i$, introduce a $\{0,1\}$-valued variable $x_i$, $1 \leq i \leq n$, with the following significance: $x_i = 1$ if $v_i$ is chosen in the (optimal) solution and 0 otherwise.
- Now, $\sum_{i=1}^{n} x_i$ is the number of nodes chosen in the solution. Therefore, the objective of the ILP is:

$$\text{Minimize } \sum_{i=1}^{n} x_i$$

- **Constraints:** For each edge $e = \{v_i, v_j\}$, at least one of $v_i$ and $v_j$ must be in the solution; i.e., at least one of $x_i$ and $x_j$ must be set to 1. So:

$$x_i + x_j \geq 1 \quad \text{for each edge } \{v_i, v_j\}$$

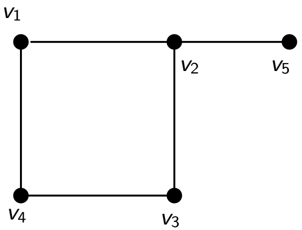
- So, the complete $\{0,1\}$-ILP for the MVC problem is:

$$\text{Minimize } \sum_{i=1}^{n} x_i \quad \text{subject to the following constraints:}$$

$$x_i + x_j \geq 1, \quad \text{for each edge } \{v_i, v_j\}$$

$$x_i \in \{0, 1\}, \quad 1 \leq i \leq n.$$

**Example:**




Minimize $x_1 + x_2 + x_3 + x_4 + x_5$

subject to the following constraints:

$$
\begin{aligned}
x_1 + x_2 &\geq 1 \\
x_1 + x_4 &\geq 1 \\
x_2 + x_3 &\geq 1 \\
x_2 + x_5 &\geq 1 \\
x_3 + x_4 &\geq 1 \\
x_i \in \{0, 1\},& \ \ 1 \leq i \leq 5
\end{aligned}
$$

**Note:** A solution to the above ILP is to set $x_2 = x_4 = 1$ and the other variables to 0. The corresponding minimum vertex cover is $\{v_2, v_4\}$.

# Coping with **NP**-completeness

**Some Advantages of the ILP Formulation:**

- **Handling costs:** Suppose there is a cost $c_i$ for each node $v_i$ and we want a vertex cover of minimum cost. The objective can be modified to

  $$\text{Minimize} \sum_{i=1}^{n} c_i\, x_i.$$

  The objective remains *linear*.

**Some Advantages of the ILP Formulation:**

- **Handling costs:** Suppose there is a cost $c_i$ for each node $v_i$ and we want a vertex cover of minimum cost. The objective can be modified to

  $$\text{Minimize} \sum_{i=1}^{n} c_i\, x_i.$$

  The objective remains *linear*.

- **Additional constraints:** Suppose node $v_i$ must be in the solution and node $x_j$ should not be in the solution. Add the constraints "$x_i = 1$" and "$x_j = 0$" to the formulation.

# Coping with **NP**-completeness

**Approximation Algorithms:**

- In practice, optimal solutions may not be needed; near-optimal solutions may be sufficient.

# Coping with **NP**-completeness

**Approximation Algorithms:**

- In practice, optimal solutions may not be needed; near-optimal solutions may be sufficient.
- To obtain a near-optimal solution for a problem, it may be possible to devise an efficient algorithm for the problem.

# Coping with **NP**-completeness

**Approximation Algorithms:**

- In practice, optimal solutions may not be needed; near-optimal solutions may be sufficient.
- To obtain a near-optimal solution for a problem, it may be possible to devise an efficient algorithm for the problem.
- Such an algorithm is an **approximation algorithm** (or a **heuristic**).

# Coping with **NP**-completeness

**Approximation Algorithms:**

- In practice, optimal solutions may not be needed; near-optimal solutions may be sufficient.
- To obtain a near-optimal solution for a problem, it may be possible to devise an efficient algorithm for the problem.
- Such an algorithm is an **approximation algorithm** (or a **heuristic**).
- For some approximation algorithms, one can rigorously establish a **performance guarantee**:

  "For every instance, the solution produced by the algorithm is within a factor of 2 of the optimal solution."

# Coping with **NP**-completeness

**Approximation Algorithms:**

- In practice, optimal solutions may not be needed; near-optimal solutions may be sufficient.
- To obtain a near-optimal solution for a problem, it may be possible to devise an efficient algorithm for the problem.
- Such an algorithm is an **approximation algorithm** (or a **heuristic**).
- For some approximation algorithms, one can rigorously establish a **performance guarantee**:

  "For every instance, the solution produced by the algorithm is within a factor of 2 of the optimal solution."

- In other cases, researchers evaluate the performance of a heuristic experimentally.

**An Approximation Algorithm for MVC:**

**Definitions:**

- Given an undirected graph $G(V, E)$, a **matching** $M$ in $G$ is a subset of edges such that no two edges in $M$ share an end point.

**An Approximation Algorithm for MVC:**

**Definitions:**

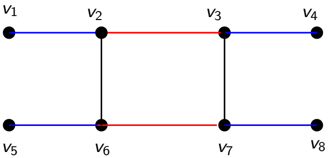
- Given an undirected graph $G(V, E)$, a **matching** $M$ in $G$ is a subset of edges such that no two edges in $M$ share an end point.
- A matching $M$ is **maximal** if no edge can be added to it without violating the matching property.

**An Approximation Algorithm for MVC:**

**Definitions:**

- Given an undirected graph $G(V, E)$, a **matching** $M$ in $G$ is a subset of edges such that no two edges in $M$ share an end point.
- A matching $M$ is **maximal** if no edge can be added to it without violating the matching property.
- A matching $M^*$ is a **maximum matching** if it has the *largest* number of edges among all the matchings of $G$.

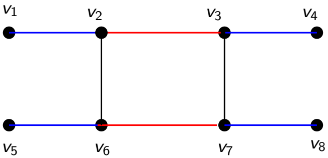
**Example:**




- For the graph $G$ (on the left), the set $M$ consisting of the edges $\{v_1, v_2\}$ and $\{v_2, v_6\}$ is not a matching; the edges share $v_2$.
- The set $M_1$ consisting of the **blue** edges is a **maximum** matching with 4 edges.
- The set $M_2$ consisting of the two **red** edges is a **maximal** matching.

- For any graph, finding a maximum or maximal matching can be done efficiently.

**Example:**



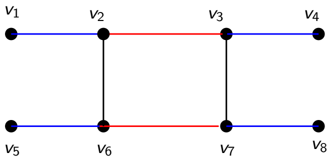

- For the graph $G$ (on the left), the set $M$ consisting of the edges $\{v_1, v_2\}$ and $\{v_2, v_6\}$ is not a matching; the edges share $v_2$.
- The set $M_1$ consisting of the **blue** edges is a **maximum** matching with 4 edges.
- The set $M_2$ consisting of the two **red** edges is a **maximal** matching.

- For any graph, finding a maximum or maximal matching can be done efficiently.
- There is a very simple algorithm for finding a *maximal* matching. (That algorithm is useful for approximating MVC.)

**Finding a Maximal Matching:**

1. Let $M = \emptyset$. ($M$ will contain a maximal matching at the end.)
2. **while** $E \neq \emptyset$ **do**
   - Choose any edge $\{x, y\}$ from $E$ and add it to $M$.
   - Delete the edges in $E$ that have $x$ or $y$ as an end point.
3. Output $M$.

**Example:**




**Note:** For the above graph $G$, the algorithm may output all the red edges or all the blue edges.

**Approximation Algorithm for MVC:**

- Find a maximal matching $M$ for $G$.
- Let $V'$ contain both the end points of each edge in $M$. Output $V'$.

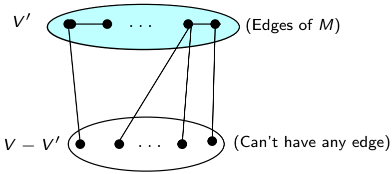
**Theorem 4:** Let $V^*$ denote a minimum vertex cover for $G$ and let $V'$ be the solution produced by the above approximation algorithm.

1. $V'$ is a vertex cover for $G$.
2. $|V'| \le 2|V^*|$.

**Note:** No algorithm with a better performance guarantee is currently known.

**Proof of Theorem 4:**

**Part 1:** $V'$ a vertex cover for $G$.




**Part 2:**

- Suppose $|M| = q$. Then $|V^*| \geq q$ since $V^*$ must contain at least one end point of each edge in $M$.
- $V'$ contains $2q$ nodes. Hence, $|V'| \leq 2\,|V^*|$.