CS3230 NP-Complete

Welcome to PlantUML!

You can start with a simple UML Diagram like:

```
Bob->Alice: Hello
```

Or

```
class Example
```

You will find more information about PlantUML syntax on https://plantuml.com

(If you use this software, you accept its license)
(Details by typing `license` keyword)

CS3230 NP-Complete

- Clique
  - Proof: clique is a NP-complete problem(Reduction From 3-SAT).
  - Reduce Independent Set to Clique (Independent Set $\leq_p$ Clique)
  - Reduce Vertex Cover to Clique
- Vertex cover
  - Proof: Vertex cover is NP-complete
  - Reduce 3-SAT to Vertex Cover
  - Reduce Indenpendent Set to Vertex Cover
- SET COVER
  - Definition 1
  - Reduce Vertex Cover to Set Cover
- INDEPENDENT-SET
  - Reduce 3-SAT to Independent Set
  - Reduce Clique to indenpendent set
- SUBSET-SUM
- KNAPSACK
  - Reduce Subset-Sum to Knapsack
- Ham-Cycle
  - Reduce Vertex Cover to Ham-Cycle

Clique

Given a graph G=(V,E), a clique of G is a subset $V' \subseteq V$ such that $\forall v, u \in V : uv \in E$. i.e. G'= (V', E) is a complete graph. |V'| is the size of a clique

- Optimization version: given a graph G, find the maximum size of clique of G
- Decisional version: given a graph G and a number k. decides whether there exists clique of G with size k

Proof: clique is a NP-complete problem(Reduction From 3-SAT).

1. Claim: clique is a NP problem

   1.1 Given instance G=(V,E), k of the clique problem, let subset $V' \subseteq V$ be the certificate and verify it with the following algorithm:

   1.1.1 Verify whether |V'| = k

   1.1.2 Verify all $u, v \in V', u \neq v$: $(u, v) \in E$

   1.2 The algorithm runs in O($|V|^2$), which is polynomia.

2. Claim: clique is a NP-hard problem:

   2.1 Polynomial reduction function from 3-SAT to clique in polynomial:

   2.1.1 Given an instance ϕ = $C_1 \wedge C_2 \wedge ... \wedge C_k$, where $C_i = (l_1^i \vee l_2^i \vee l_3^i)$

   2.1.2 Construct a graph G=(V,E) in the following way:

   - Each l_j^i as a vertex, where $j \in \{1, 2, 3\}, i \in \{1, \ldots, k\}$
   - For any two vertices u, v, there is an edge between them if:
     - They are not in the same clause (i.e. $l_j^i, l_{j'}^{i'} : i \neq i'$)
     - Their values are consistent (i.e. $l_j^i, l_{j'}^{i'} : l_j^i \neq \neg l_{j'}^{i'}$)

   2.1.3 G is an instance of vertex cover

   2.1.4 The reduction function runs in $O(3 * (k - 1)) = O(k)$

   2.2 3-SAT returns YES $\rightarrow$ Clique returns YES:

   2.2.1 Given that 3-SAT returns YES, there is a truth assignment to boolean variables such that ϕ returns True, which implies that each clause C_i will be true, and hence there must be at least one literal in each clause that evaluates to be true.

   2.2.2 Select one true literals in each clause and makes $V' \subseteq V$, and claims that $|V'| = k$ and V' is a clique of G, because $\forall u, v \in V'$, u,v are not from the same clause and value of u, v are both True, which is consistent, and therefore $(u, v) \in E$

   2.2.3 Hence Clique returns YES

   2.3 Clique returns YES $\rightarrow$ 3-SAT returns YES

   2.3.1 Given that Clique returns YES, there exists $V' \subseteq V$ and V' is Clique of G and $|V'| = k$

   2.3.2 Claim: $\forall u, v \in V'$: u,v are in distinct clause. Otherwise $(u, v) \notin E$, conflicts 2.3.1

   2.3.3 Claim: $\forall i \in \{1, \ldots, k\} : \exists l_j^i \in C_i : l_j^i \in V'$, otherwise conflicts that $|V|$

   2.3.4 By assigning each vertices in V' to True, each Clause will evaluate to True and ϕ will be evaluated to True eventually

   2.3.5 Hence 3-SAT returns YES

   2.4 Hence 3-SAT$\leq_p$ $Clique$ and Clique is in NP-hard

3. Therefore Clique is NP-Complete.

Reduce Independent Set to Clique (Independent Set $\leq_p$ Clique)

1. Given an instance G=(V,E), k of Independent Set Problem, where $|V| = n$, a polynomial reduction function can be constructee as follows:

   1.1 Let $\overline{G} = (V, \overline{E})$ be the complementary graph of G, k be the instance of Clique

   1.2 The reduction fucntion runs in constant time hence clearly in polynomial

2. Input to IndependentSet(G, k) is YES instance $\rightarrow$ Input to Clique($\overline{G}, k$) is YES instance:

   2.1 Given that Independent set returns YES, then there exists $V' \subseteq V$ such that $\forall u, v \in V', (u, v) \notin E$ and $|V'| = k$

   2.2. Then $\forall u, v \in V' : (u, v) \in \overline{E}$ by the definition of $\overline{E}$ and $|V'| = k$, hence Clique returns YES

3. Input to Clique(G, k) is YES instance$\rightarrow$ Input to IndenpendentSet($\overline{G}, k$) is YES instance

   3.1 Given that Clique returns YES then there exists $V' \subseteq V$ such that $\forall u, v \in V' : (u, v) \in \overline{E}$, and $|V'| = k$

   3.2 Then $\forall u, v \in V', (u, v) \notin E$, and $|V'| = k$

   3.3 hence Indenpendent Set returns YES

4. hence Independent Set $\leq_p$ Clique

Reduce Vertex Cover to Clique

1. Given an instance $G = (V, E)$, k of vertex cover, and $|V| = n$
2. Let $E' = \{(u, v) | u, v \in V \wedge (u, v) \notin E\}$
3. $G' = (V, E'), n - k$ be instance of Clique

- YES instance of Vertex Cover $\rightarrow$ YES instance of Clique
  1. If $\exists V_{sub} \subseteq V : \forall (u, v) \in E : u \in V_{sub} \vee v \in V_{sub}$ and $|V_{sub}| \leq k$
  2. Then $\forall u, v \in V : u, v \notin V_{sub} \rightarrow (u, v) \notin E \rightarrow (u, v) \in E'$
  3. Hence $\forall u, v \in V \backslash V_{sub} : (u, v) \in E'$ and $|V \backslash V_{sub}| \geq n - k$

Vertex cover

Given a graph G=(V, E), a subset $V' \subseteq V$ is said a vertex cover if $\forall e = (u, v) \in E :$ either $u \in V'$ or $v \in V'$

|V'| is said to be the size of the vertex cover

- Optimization version: find the maximum size of vertex cover.
- Decision version: decides whether there is a vertex cover with size k.

Proof: Vertex cover is NP-complete

1. Claim: Vertex cover is a NP problem

   1.1 Given instance G=(V,E), k of vertex cover problem, let $V' \subseteq V$ be the certificate and an algorithm can verify the certificate in the following way:

      1.1.1 Verify |V'|=k

      1.1.2 Go though all edges $e = (u, v) \in E$ and verify that either u or v in V'

   1.2 The algorithm runs in O(E), which is polynomial

   1.3 Hence VERTEX-COVER $\in$ NP

2. Claim: Vertex cover is a NP-hard problem

   2.1 Polynomial reduction function from Clique to Vertex cover

      2.1.1 Given an instance G=(V, E), k of Clique, where |V|=n

      2.1.2 Let $\overline{G} = (V, \overline{E})$ be complementary graph of G, then $\overline{G}$, n-k are instance of Vertex cover

      2.2.3 The reduction function is in O(E), which is polynomial

   2.2 Clique returns YES $\rightarrow$ Vertex cover returns YES:

      2.2.1 Given that Clique returns YES, then there exists $V' \subseteq V$, $|V'| = k$ and $\forall u, v \in V'$, $(u, v) \in E$

      2.2.2 Thereofore $\forall u, v \in V' : (u, v) \notin \overline{E}.$ (by definition of $\overline{E}$)

      2.2.3 Claim: $\forall e = (u', v') \in \overline{E}:$ either u or v in $V \backslash V'$, where $|V \backslash V'| = n - k$, otherwise, both u, v

         are in V', which conflicts 2.2.2

      2.2.4 Hence $\forall e = (u, v) \in \overline{E}$: either u or v in $V \backslash V'$ and $|V \backslash V'| = n - k$

   2.3 Vertex cover returns YES $\rightarrow$ Clique returns YES

      2.3.1 Given that the Vertex cover returns YES, then exists $V'' \subseteq V$ such that $\forall e = (u, v) \in \overline{E}$:

         either u or v in V'' and V''= n-k

      2.3.2 then let V' = $V \backslash V''$, then $|V'| = k$, and $\forall u', v' \in V' : (u', v') \notin \overline{E}$, otherwise $u', v' \in V''$

         conflicts

      2.3.3 Hence $\forall u', v' \in V' : (u', v') \in E$ (from 2.3.2 and definition of $\overline{E}$)

      2.3.4 Hence Vertex cover returns YES

   2.4 Therefore Clique$\leq_p$ Vertex cover, and Vertex cover is NP-hard

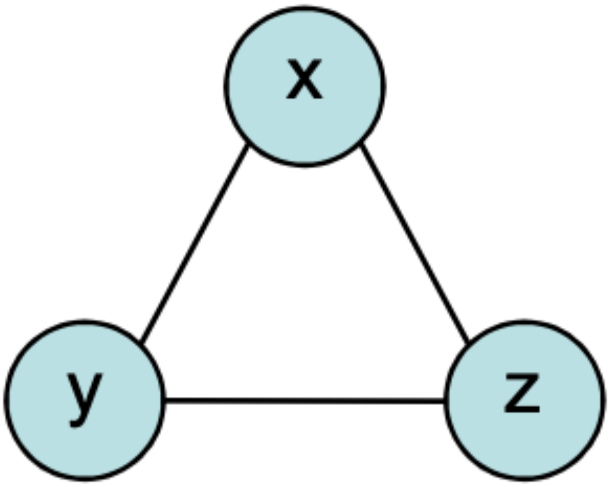
3. Therefore Vertex cover is NP-complete

Reduce 3-SAT to Vertex Cover

Given an instance $C_1 \wedge C_2 \wedge ... \wedge C_n$ of 3-SAT, with m variables and l clauses. We will try to reduce it to a Vertex Cover problem of size $m + 2 \cdot l$

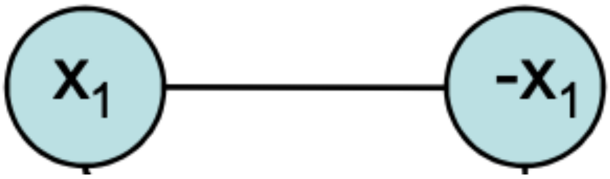
The vertices

1. Clause Widget



The above is the clause widget for $(x \vee y \vee z)$

2. Variable Widget



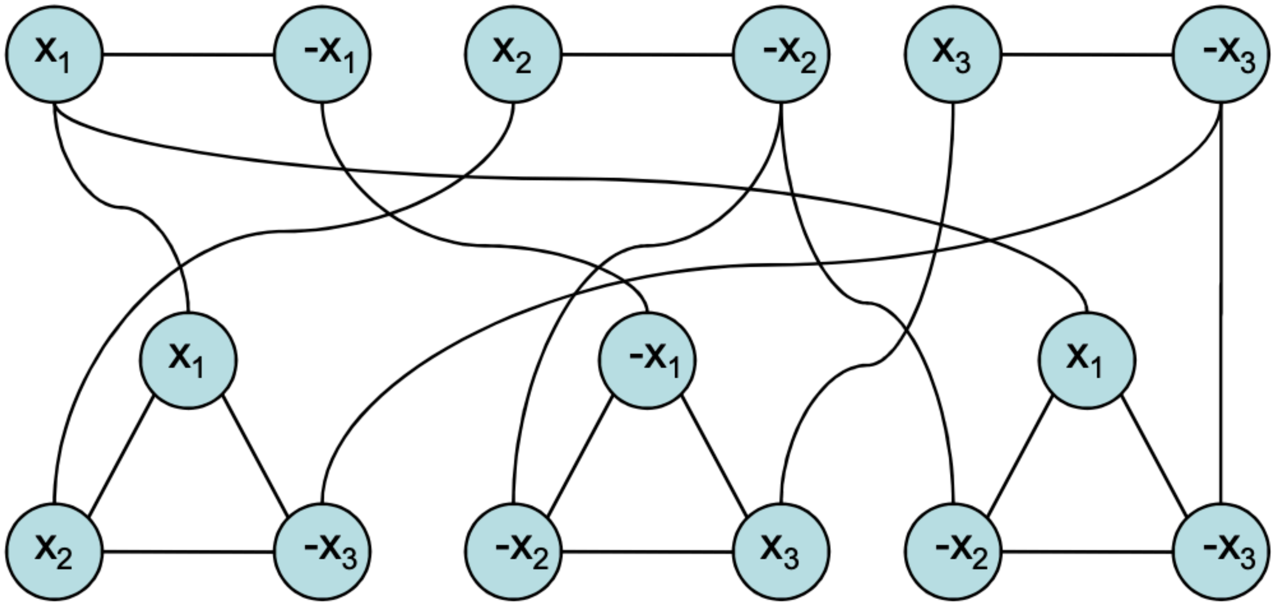
The above is the vairiable widget for variable x_1

Hence we will draw $2 \cdot m + 3 \cdot l$ vertices

The edge

1. We will draw edge connecting vertices in the same widget
2. We will draw edge connecting vertices of Clause Widget to corresponding Variable Widget

E.g. The graph $G = (V, E)$ correspondes to
$\phi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \vee (x_1 \vee \neg x_2 \vee \neg x_3)$ is as follows:



Proof of reduction

We now have to prove that this reduction is correct, meaning that a vertex cover of size $k = m + 2l$ exists if and only if the given ϕ is satisfiable. Suppose we have a satisfying assignment for ϕ. For each variable, add the node corresponding to the true version of the literal for that variable to the vertex cover. Then, for each clause, select one true literal in the clause and add the remaining two literals to the vertex cover. We have used 2 vertices in each clause gadget and 1 vertex in each variable gadget for this cover, which meets the size requirement for the cover. Each edge in a variable gadget is covered by the node selected from that gadget. All three edges in each clause are covered by the two nodes we selected in that clause gadget. One true literal in each clause gadget is left out of the cover, but because it is a true literal it is connected by an edge to the node corresponding to the true literal in a variable clause. Therefore these nodes cover all of the edges in the graph, and we have a valid vertex cover.

We must still prove the other direction of this implication. If we have a vertex cover of size k for this graph, that cover must contain one node in each variable gadget and two nodes in each clause gadget to cover the edges in those gadgets. This requires exactly $k = m+2l$ nodes. Suppose we take literal corresponding to a covered node in a variable gadget to be true. This assignment satisfies ϕ because, for each clause gadget, each of the three edges connecting the clause gadget to the variable gadgets is covered, and only two nodes in the clause gadget are in the cover. Therefore one of these clause-gadget edges must be covered by a node in a variable gadget, and so the assignment of that covered variable gadget literal to true in ϕ will satisfy the clause. This holds for every clause gadget and clause, so this assignment satisfies ϕ. Therefore a k-covering of this graph corresponds to a satisfying assignment for ϕ, while a satisfying assignment for ϕ corresponds to a k-covering of this graph, and this reduction is correct.

Proof of polynomial is easy and hence omitted

Reduce Indenpendent Set to Vertex Cover

1. Given an instance $G=(V,E)$, k of indenpendent set, then $G=(V,E), |V|-k$ is instance of Vertex Cover
   - The reducing function is obviously in polynomial
   - YES instance of Independent Set $\to$ YES instance of Vertex Cover
     - If $\exists\ V_{sub} \subseteq V : \forall u,v \in V_{sub} : (u,v) \notin E$ and $V_{sub} \geq k$
     - Then $\forall (u,v) \in E : u \notin V_{sub} \vee v \notin V_{sub}$
     - Hence $\forall (u,v) \in E : u \in V \backslash V_{sub} \vee v \in V \backslash V_{sub}$ and $V_{sub} \leq |V|-k$
     - Hence YES instance of Vertex Cover
   - The other direction is omitted .

SET COVER

Definition 1

Given a set $U=\{a_1,a_2,\ldots,a_n\}$, where U is also called universe and a collection of m subsets of U: $S=\{S_1,\ldots,S_m\}$, for any sub-collection of S say $S' \subseteq S$,if union of S' is U, then S' is set cover of U and $|S'|$ is the size of the set cover.

- Optimization version: find the subcollection $S' \subseteq S$ with the smallest size
- Decision version: decide whether there is a subcollection $S' \subseteq S$ With size less or equal to k.

Reduce Vertex Cover to Set Cover

1. Given instance of Vertex Cover $G=(V,E)$
2. Let $U=E$
3. Let $S=\{S_u | u \in V\}$, where S_u denotes the set of edges that incident on u

📔 **NOTE:**

There is no reduction from Set Cover to Vertex Cover, because there is case where each element of $\in U$ is contained by more than two $S_i \in S$, which would implies that one edge incident on more than 2 vertices, which does not make sense.

INDEPENDENT-SET

Given a graph G=(V,E), an subset $V' \subseteq V$ is said to be an independent set of G if $\forall u, v(u \neq v') \in V : (u,v) \notin E$. $|V'|$ is said to be the size of independent set.

- Optimization version: given a graph G=(V,E), find the independent set with maxmum size.
- Decision version: given a graph G(V,E), decides whether there is an independent set with size k.

Reduce 3-SAT to Independent Set

Given an instance $C_1 \wedge C_2 \wedge ... \wedge C_k$ of 3-SAT, where $C_i = l_i^1 \vee l_i^2 \vee l_i^3$ for $i \in \{1, 2, \ldots, n\}$

- Each clause contains three literals, we draw one vertex for each literal
- We connet the vertex in the same Clause, and build Clause Literal
- We connect literals with their complementary

In this way we build widget $G = (V, E)$ and G, k is the instance for independent set.

The intuition is that we have to find one true iteral each clause, which does not conflicts with the assignment of other clause.

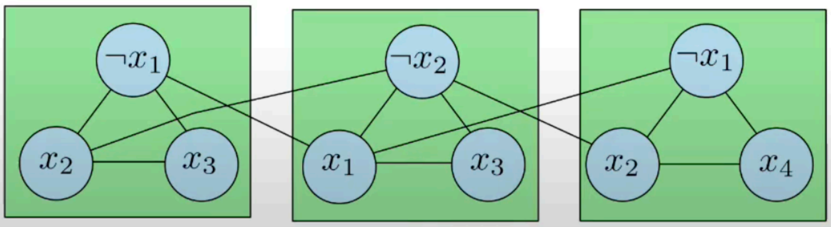


Figure: Graph for $\varphi = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$

Reduce Clique to indenpendent set

1. Given an instance $G = (V, E)$, k of clique, let $\overline{G} = (V, \overline{E})$, k be instance of independent set.

SUBSET-SUM

Given a set of non-negative numbers $S = \{v_1, v_2, \ldots, v_n\}$ and a number V, check whether there is a subset $I \subset \{1, \ldots, n\}$ such that $\sum_{i \in I} v_i = V$

KNAPSACK

Given a set of non-negative weight $S_w = \{w_1, \ldots, w_n\}$ and a set of non-nagative value $S_v = \{v_1, \ldots, v_n\}$, a maximum weight W, knapsack is any subset of $I \in \{1, \ldots, n\}$ such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i$ is called value of the knapsack

- Optimization version: Find the knapsack with maximum value
- Decision version: instance involing an additional integer k, decide whether there is a kanpsack with value equal or more than k

Reduce Subset-Sum to Knapsack

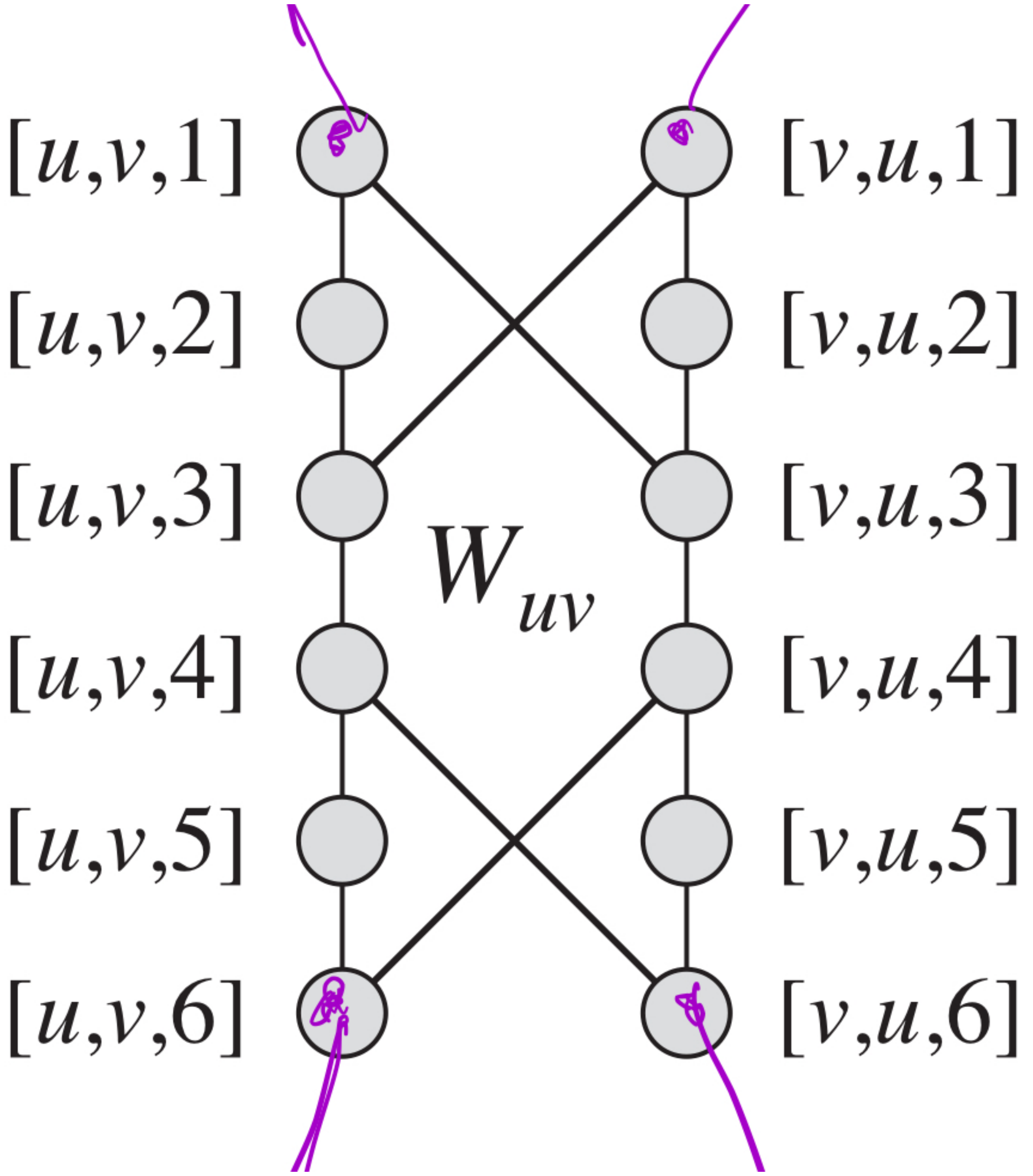
Given instance $\{a_1, a_2, \ldots, a_n\}$ and V of knapsack problem. Let $v_i = a_i$ and $w_i = a_i$ for $i \in \{1, \ldots, n\}$, $W = V$, then $(w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n)$, W, V will be instance of KnapSack

Ham-Cycle

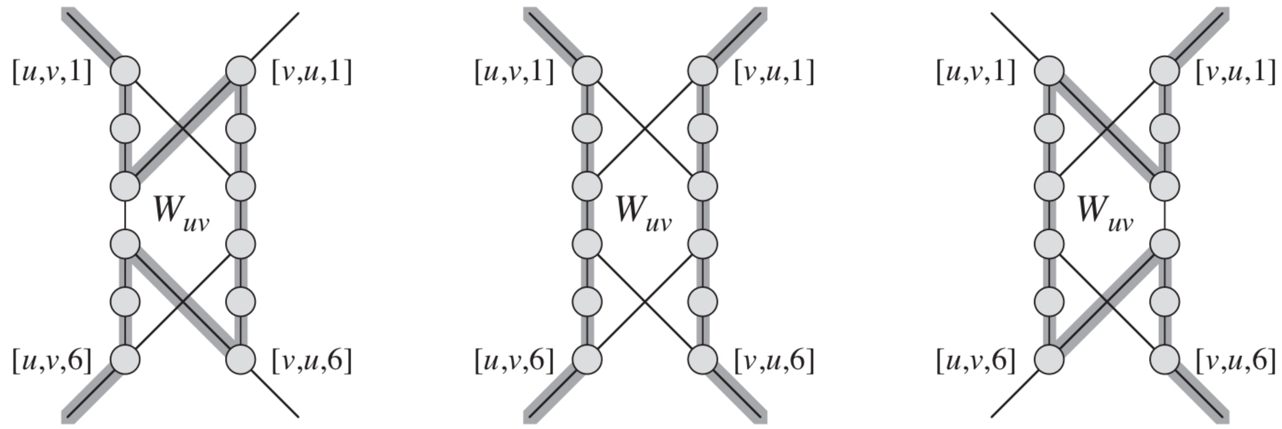
Given. a graph G=(V,E), find a simple path that passes every $v \in V$

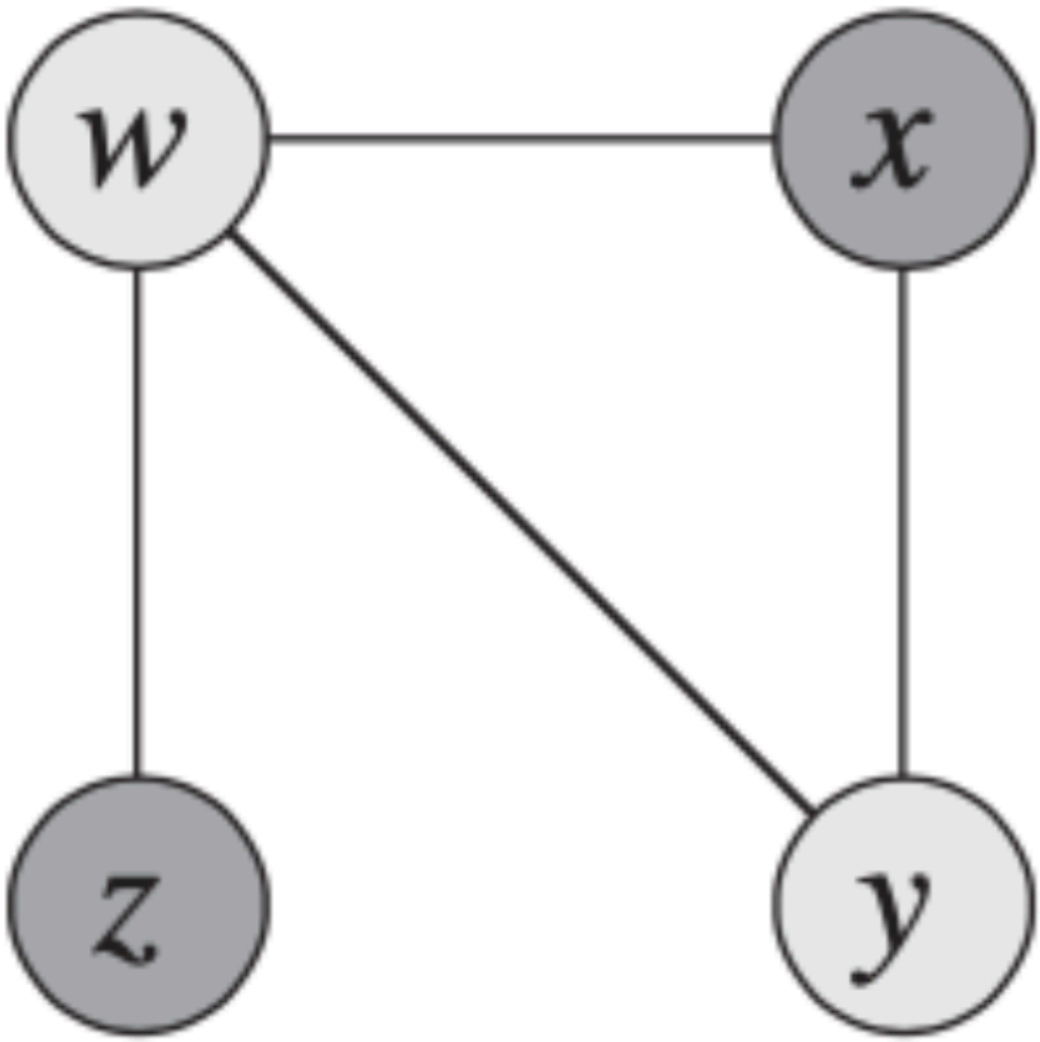
Reduce Vertex Cover to Ham-Cycle

Given instance $G = (V, E)$ and k of Vertex Cover



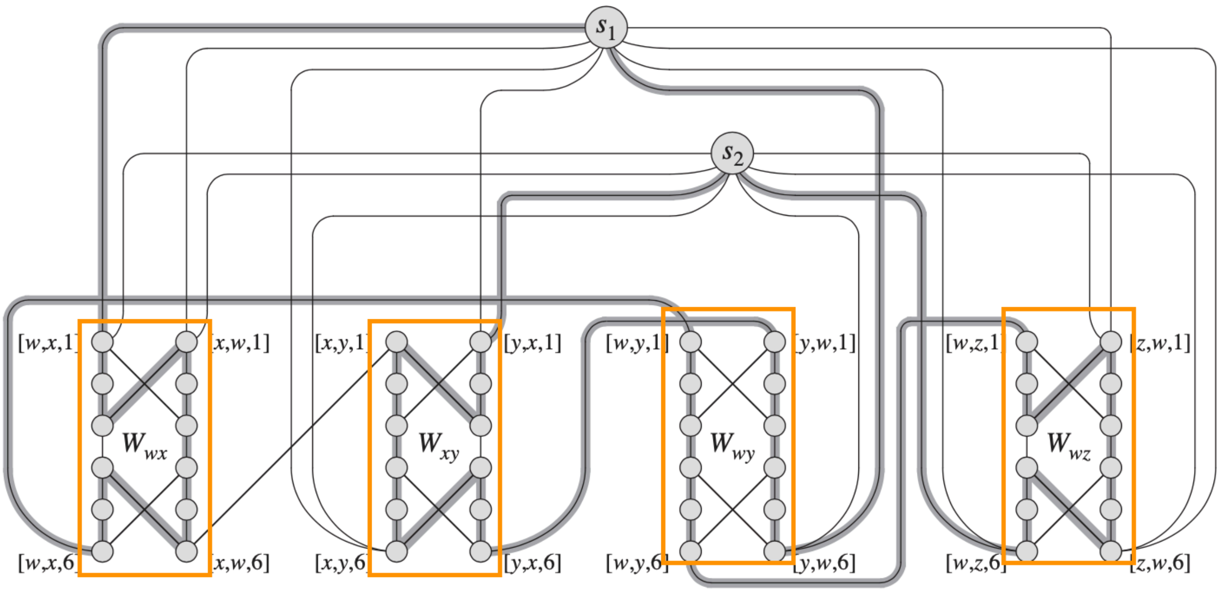
- Each widge W_{uv} represents an edge $(u, v) \in E$, with only $[u, v, 1]$, $[u, v, 6]$, $[v, u, 1]$, $[v, u, 6]$ can connect with other components
- The widget has 14 edges such that for any path enter teh W_{uv} from $[u, v, 1]$, $[u, v, 6]$, $[v, u, 1]$, $[v, u, 6]$, there are only the following three ways to go out

- 



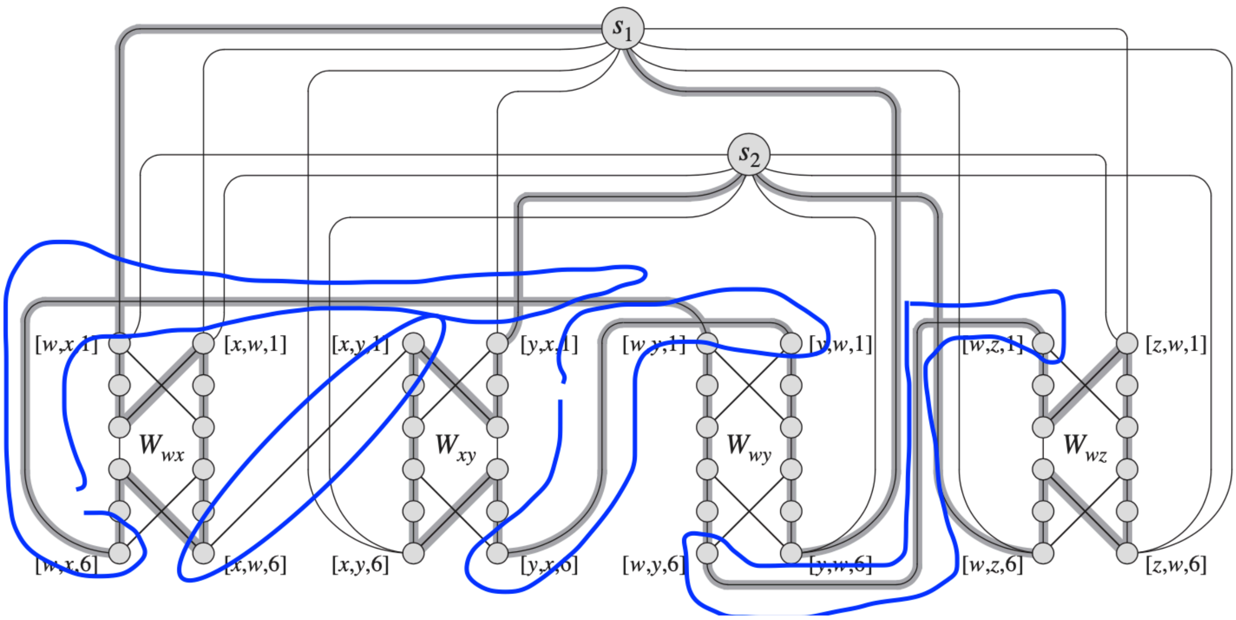
For the above graph of vertex cover, construct the graph for Ham-cycle as follows:

1. Represent each edge as widget and create an initial G'



Can see that the four widgets represent $(w, x), (x, y), (w, y), (w, z)$ respectively

2. For each vertex $u \in V$, we create create an edge between two widgets that incident on it in the following way:
    - Adding an edge between $[u, u^i, 6]$ and $[u, u^{i+1}, 1]$ for $1 \leq i \leq degree(u) - 1$, and gets the following edges cycled with blue mark



Here:

- $[w, x, 6] - [w, y, 1]$, $[w, y, 6] - [w, z, 1]$
- $[x, w, 6] - [x, y, 1]$
- $[y, x, 6] - [y, w, 1]$

The intuition behind these edges is that if we choose a vertex $u \in V$ in the vertex cover of G, we can construct a path from $[u, u^1, 1]$ to $[u, u^{degree(u)}, 6]$ that "covers" all widgets corresponding to edges incident on u

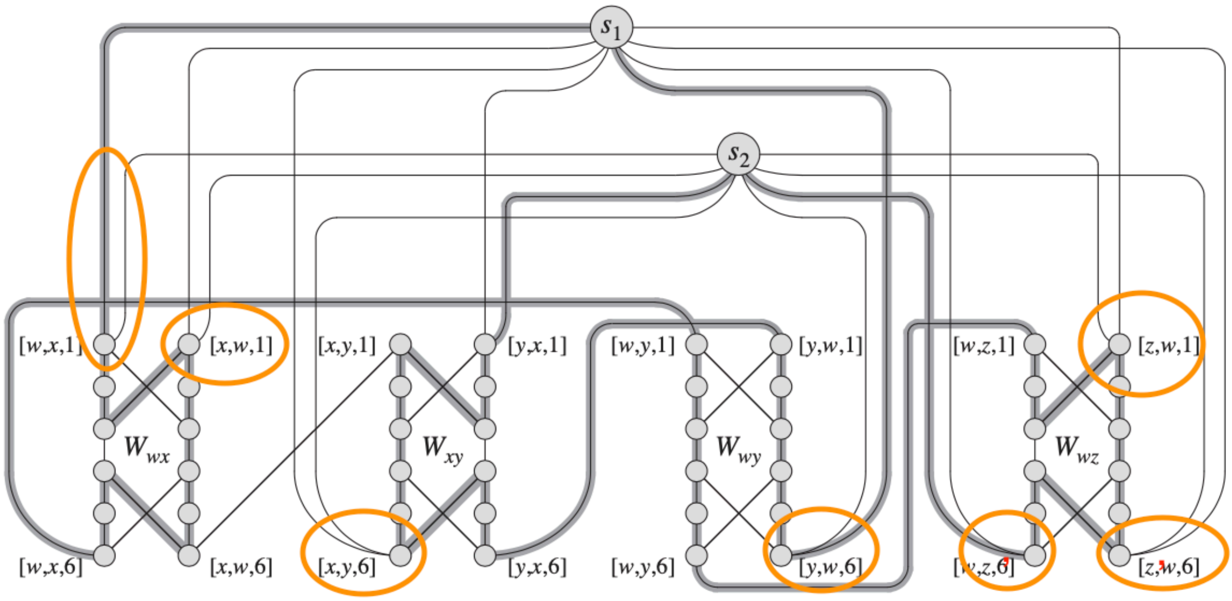
3. Selector vertices are tool used to decide the k vertices of cover in G of vertex cover

   (If it takes at least k vertices in the vertex cover problem, then we need at least k selector vertices to make the ham-cycle, this will be proved latter)

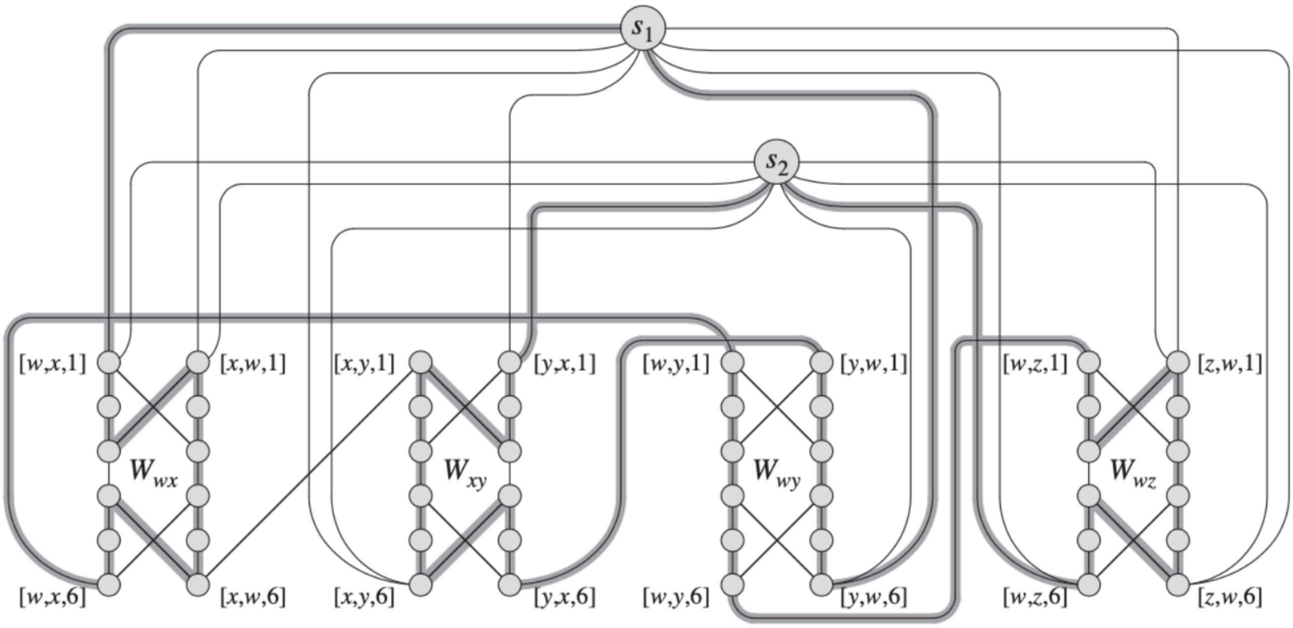
The selector vertices will connect $[u, u^1, 1]$ and $[u, u^{degree(u)}, 6]$ for all $u \in V$

Here each selector vertex should connect to $[w, x, 1]$, $[w, z, 6]$, $[x, w, 1]$, $[x, y, 6]$, $[y, w, 6]$, $[z, w, 1]$, $[z, w, 6]$ as

shown bellow

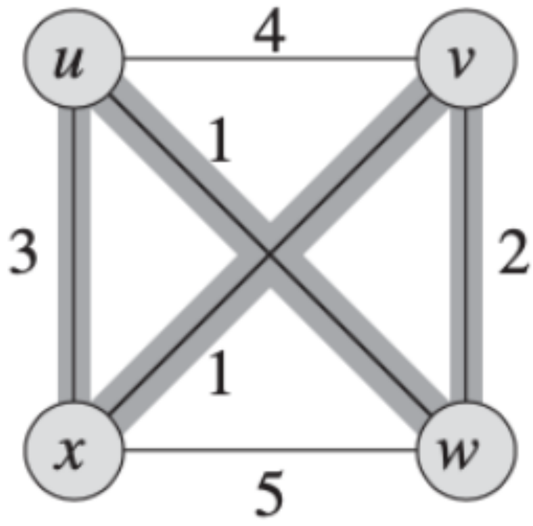


4. Then we finally construct G' :



Should first prove that G' is polynomial in size of G (See text book p1094)

5. Then we show that this is a reduction


u
v
x
w
4
1
3
2
1
5