(16.4) Reductions from Label Cover

1805006 - Tanjeem Azwad Zaman

Overview

1. Problem Definition

2. Optimization Problems
   a. Minimization Version
   b. Maximization Version

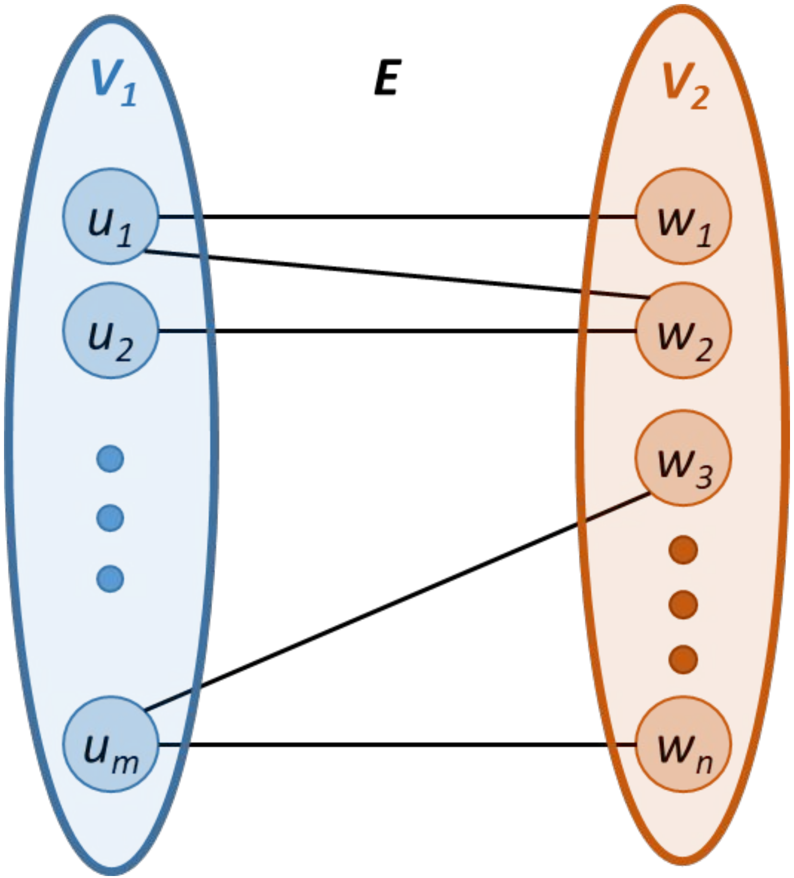
3. Hard to approximate the Maximization Problem (Reduction from MAX E3SAT)

4. Some Theorems

1. Problem Definition

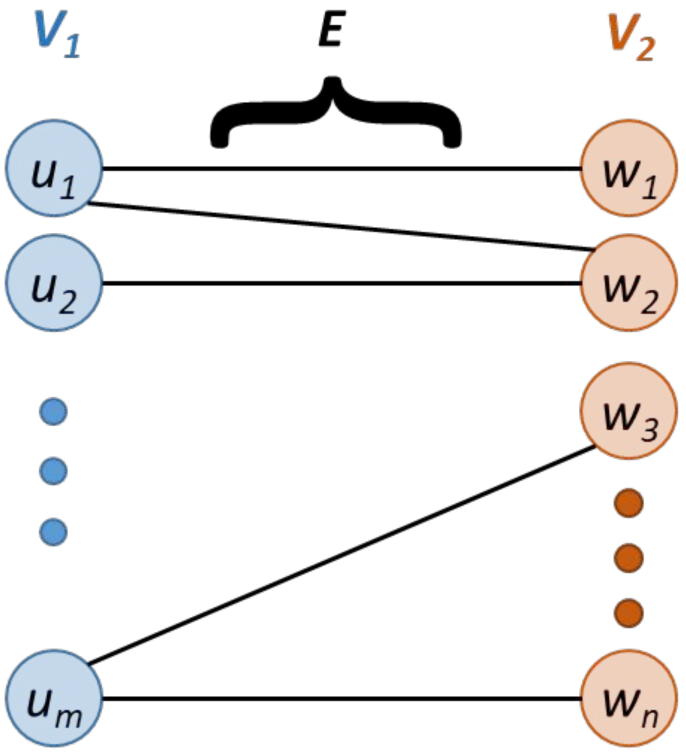
Label Cover

- Bipartite Graph (V_1, V_2, E)
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge (u,w) is ***"satisfied"*** if
  - u is assigned label l_1 and
  - **w** is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



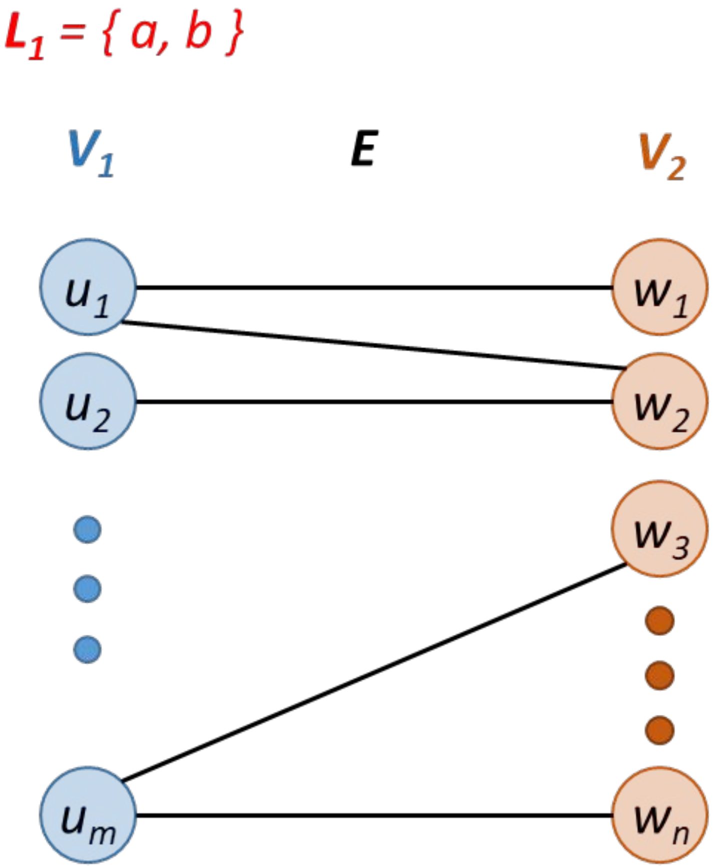
Label Cover

- Bipartite Graph ***(V_1, V_2, E)***
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge ***(u,w)*** is ***"satisfied"*** if
  - ***u*** is assigned label l_1 and
  - **w** is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



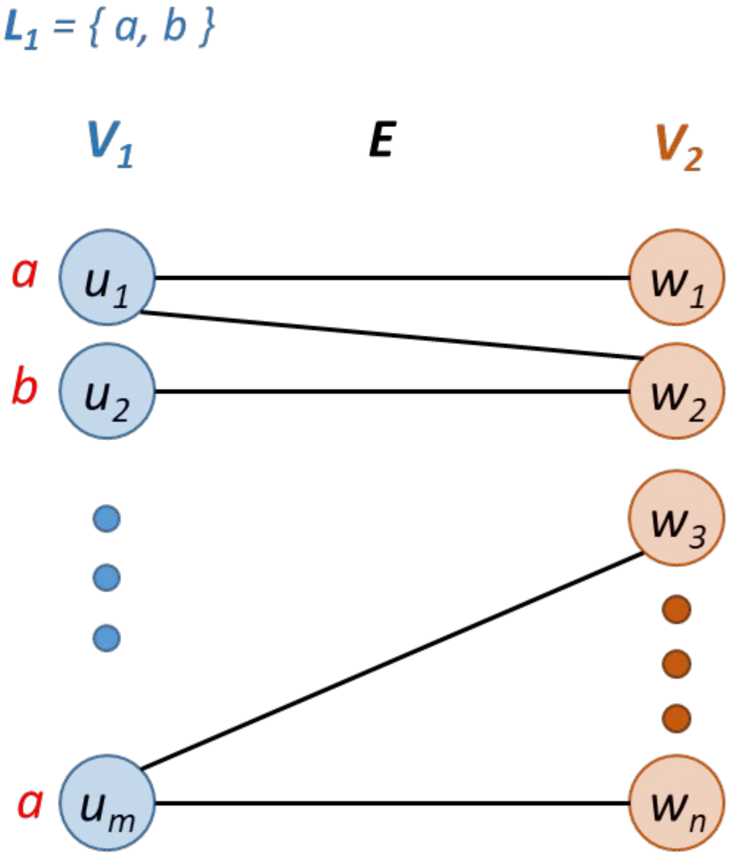
Label Cover

- Bipartite Graph ***(V_1, V_2, E)***
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge ***(u,w)*** is ***"satisfied"*** if
  - ***u*** is assigned label l_1 and
  - **w** is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



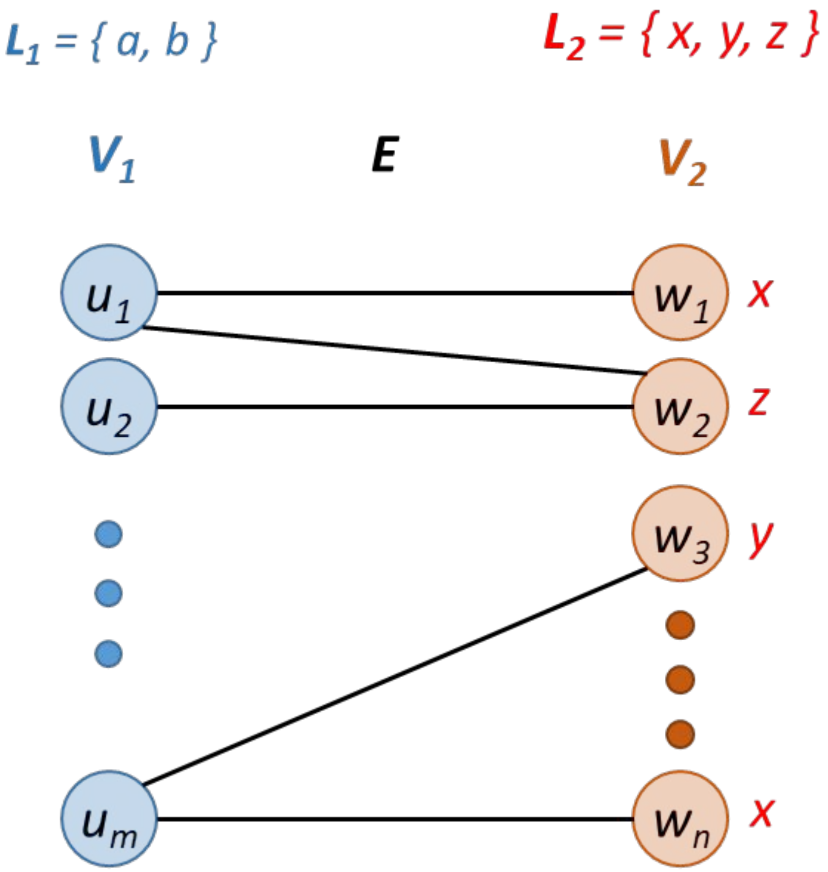
Label Cover

- Bipartite Graph ***(V₁, V₂, E)***
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge ***(u,w)*** is ***"satisfied"*** if
  - u is assigned label l_1 and
  - w is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



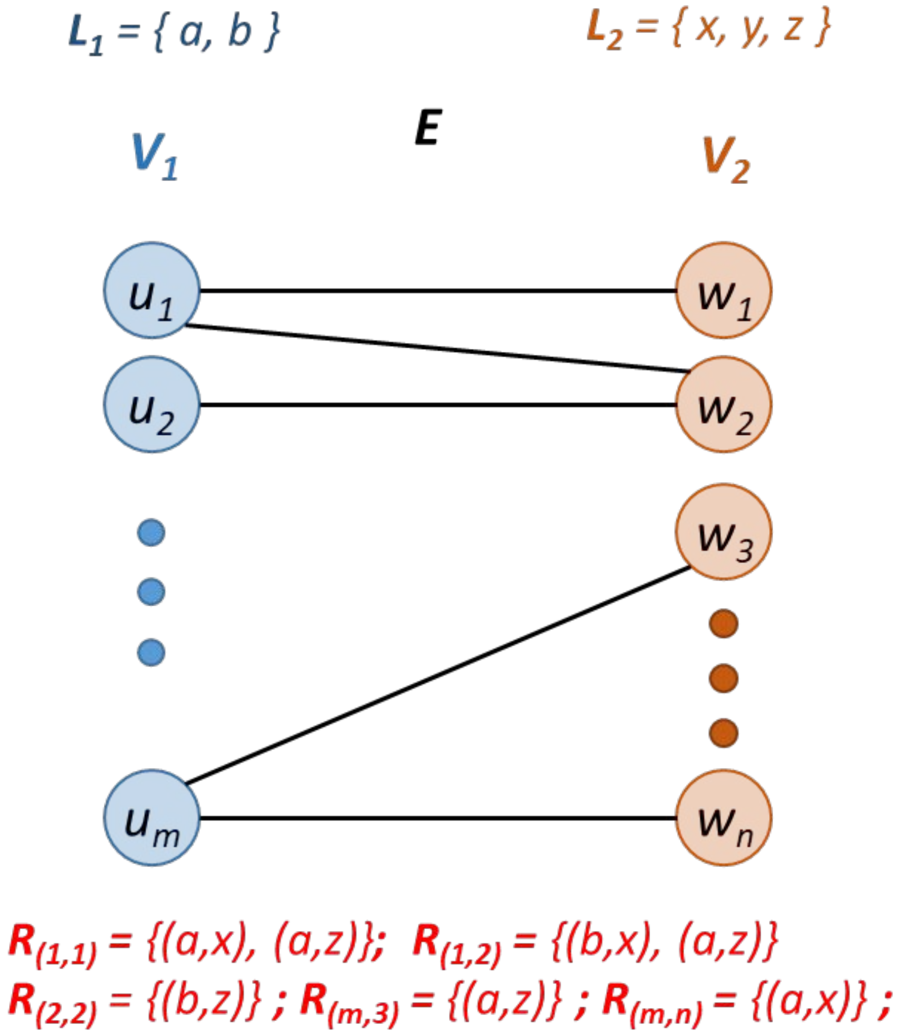
Label Cover

- Bipartite Graph (V_1, V_2, E)
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
    - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge (u,w) is ***"satisfied"*** if
    - u is assigned label l_1 and
    - w is assigned label l_2 s.t.
    - $l_1 \in L_1$; $l_2 \in L_2$ and
    - $(l_1, l_2) \in R_{(u,w)}$



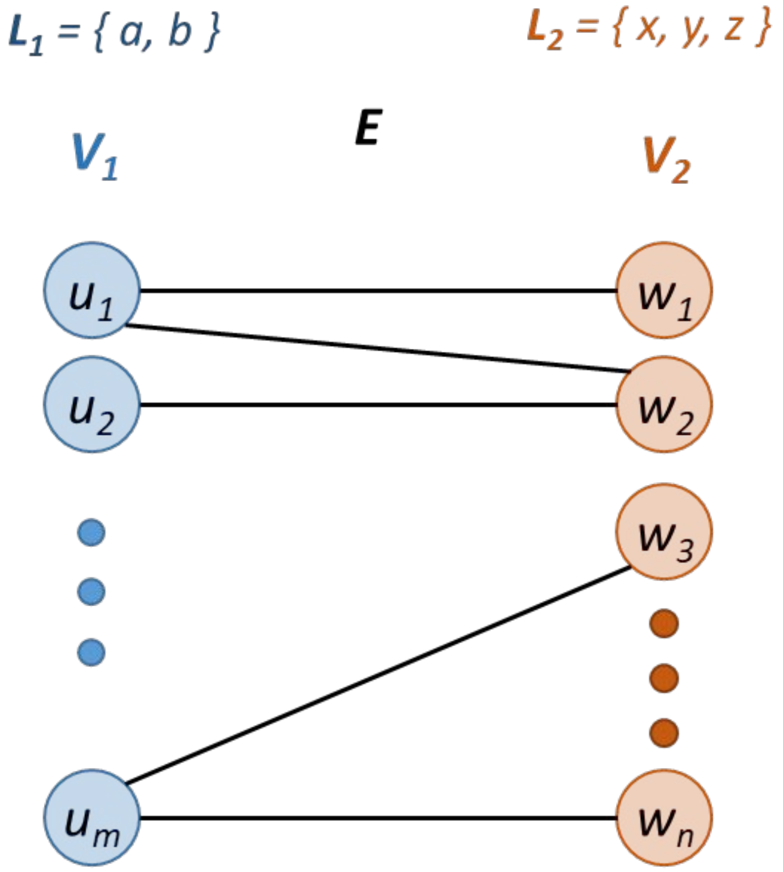
Label Cover

- Bipartite Graph (V_1, V_2, E)
- L_1: Set of possible labels for $u \in V_1$
- L_2: Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge (u,w) is ***"satisfied"*** if
  - u is assigned label l_1 and
  - w is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



Label Cover

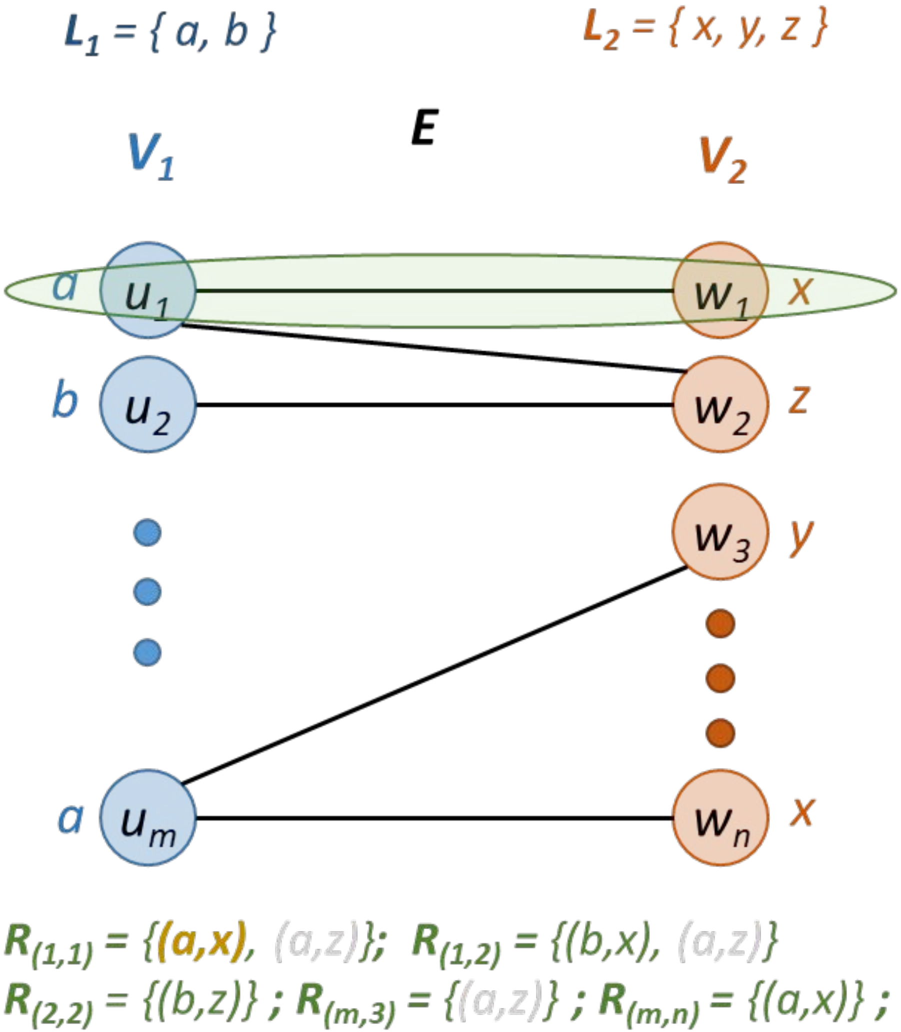
- Bipartite Graph ***(V₁, V₂, E)***
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge (u,w) is ***"satisfied"*** if
  - u is assigned label l_1 and
  - **w** is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



$R_{(1,1)} = \{(a,x), (a,z)\}$; $R_{(1,2)} = \{(b,x), (a,z)\}$
$R_{(2,2)} = \{(b,z)\}$; $R_{(m,3)} = \{(a,z)\}$; $R_{(m,n)} = \{(a,x)\}$;

Label Cover

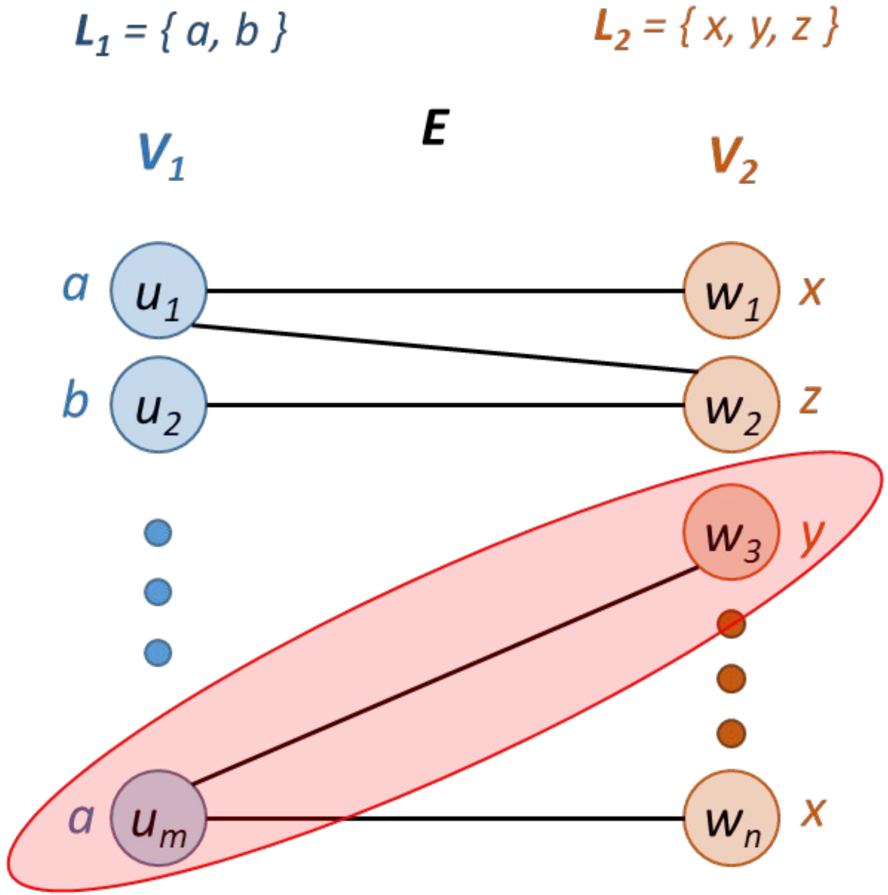
- Bipartite Graph (V_1, V_2, E)
- L_1 : Set of possible labels for $u \in V_1$
- L_2 : Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$
- An edge (u,w) is ***"satisfied"*** if
  - u is assigned label l_1 and
  - **w** is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$



Label Cover

- Bipartite Graph (V_1, V_2, E)
- L_1: Set of possible labels for $u \in V_1$
- L_2: Set of possible labels for $w \in V_2$
- $R_{(u,w)} \subseteq L_1 \times L_2$
  - A set of acceptable assignments for each edge $(u,w) \in E$

- An edge (u,w) is ***"satisfied"*** if
  - u is assigned label l_1 and
  - w is assigned label l_2 s.t.
  - $l_1 \in L_1$; $l_2 \in L_2$ and
  - $(l_1, l_2) \in R_{(u,w)}$

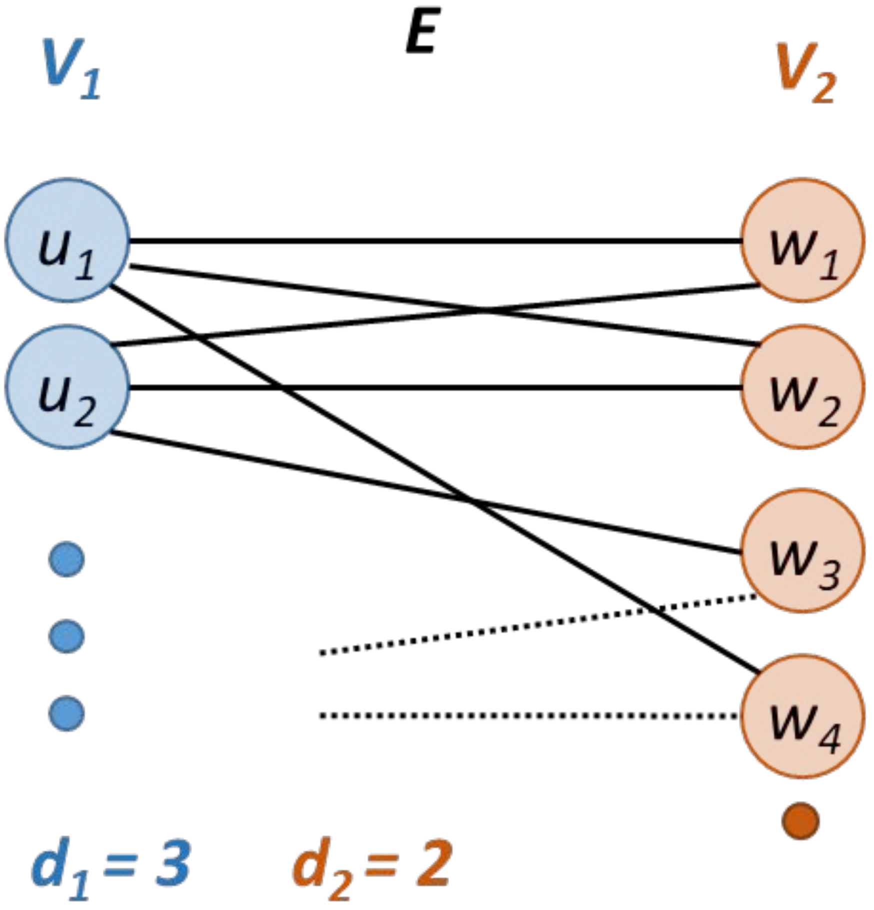


$R_{(1,1)} = \{(a,x), (a,z)\}$; $R_{(1,2)} = \{(b,x), (a,z)\}$

$R_{(2,2)} = \{(b,z)\}$; $R_{(m,3)} = \{(a,z)\}$; $R_{(m,n)} = \{(a,x)\}$;

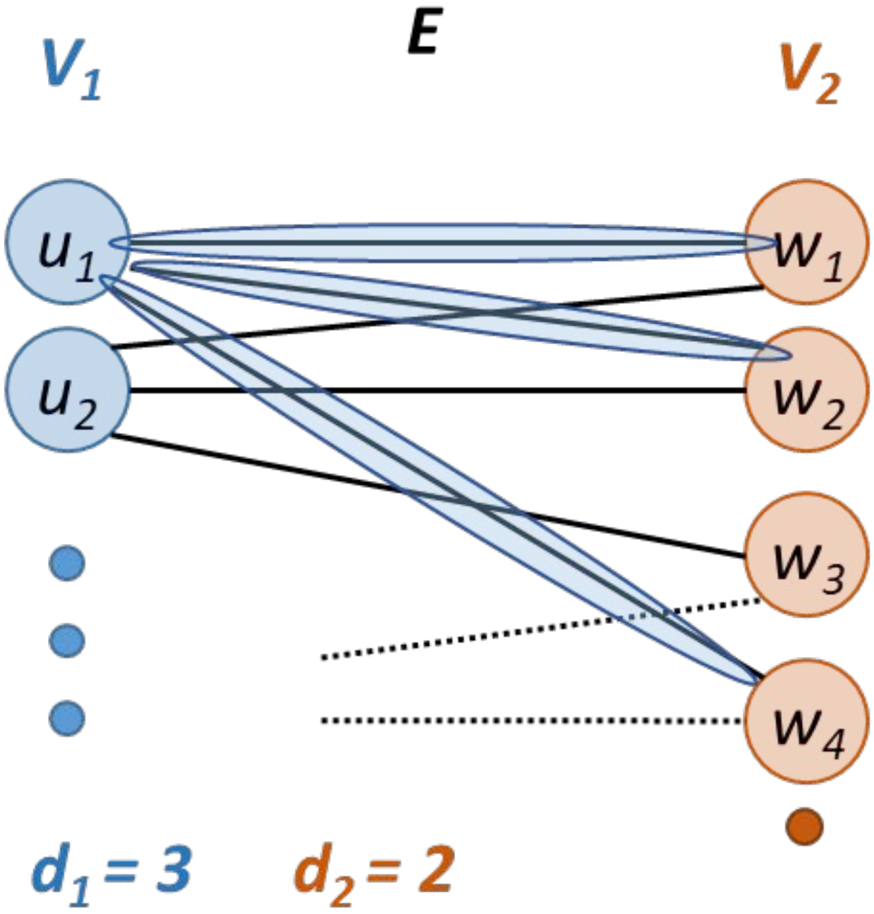
(d_1, d_2) - Regular instances

- Each vertex $\boldsymbol{u \in V_1}$ has degree $\boldsymbol{d_1}$
- Each vertex $\boldsymbol{w \in V_2}$ has degree $\boldsymbol{d_2}$



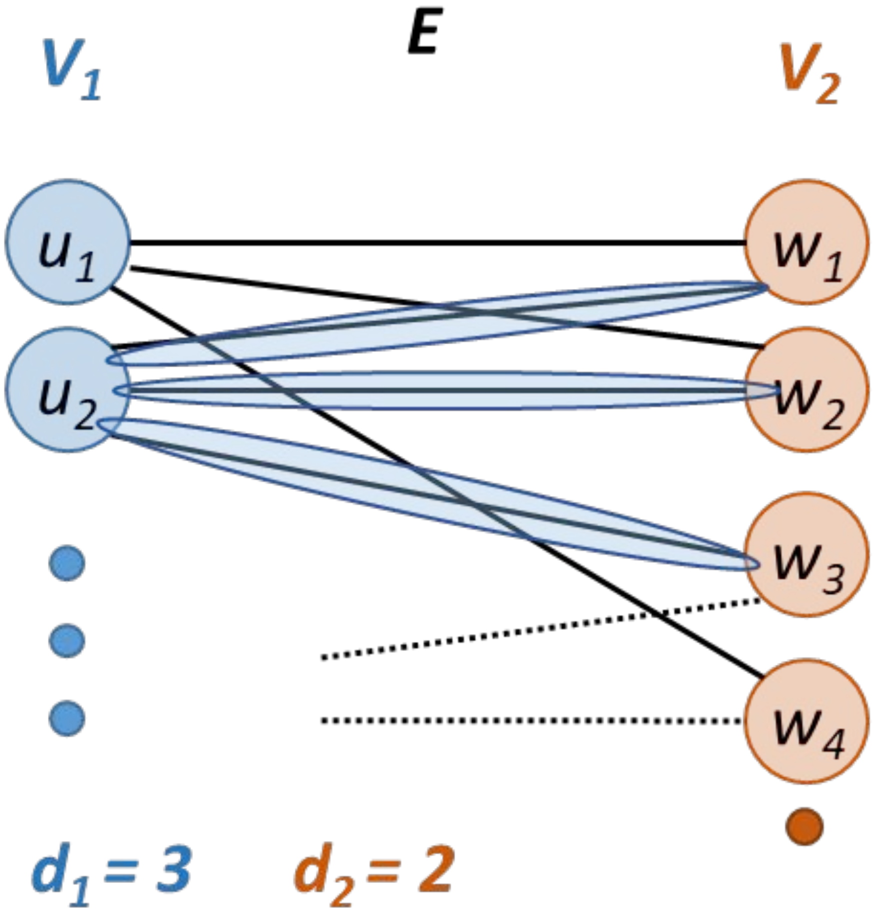
Regular instances (d_1, d_2)

- Each vertex $u \in V_1$ has degree d_1
- Each vertex $w \in V_2$ has degree d_2



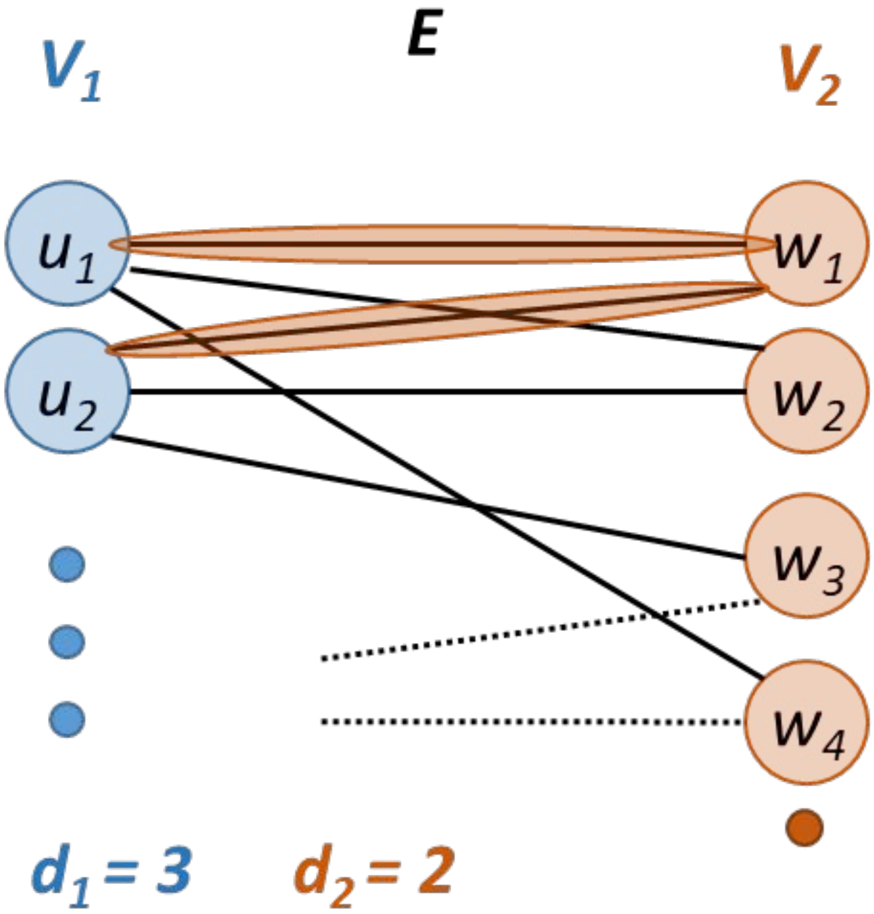
Regular instances (d_1, d_2)

- Each vertex $u \in V_1$ has degree d_1
- Each vertex $w \in V_2$ has degree d_2



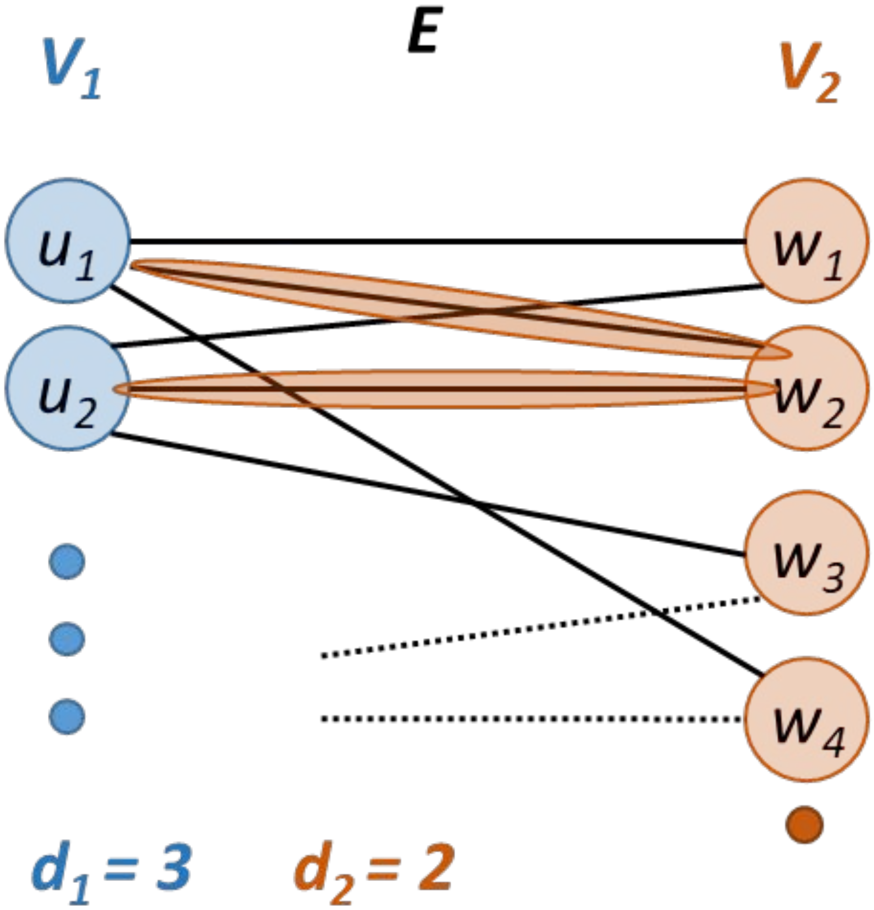
Regular instances (d_1, d_2)

- Each vertex $u \in V_1$ has degree d_1
- Each vertex $w \in V_2$ has degree d_2



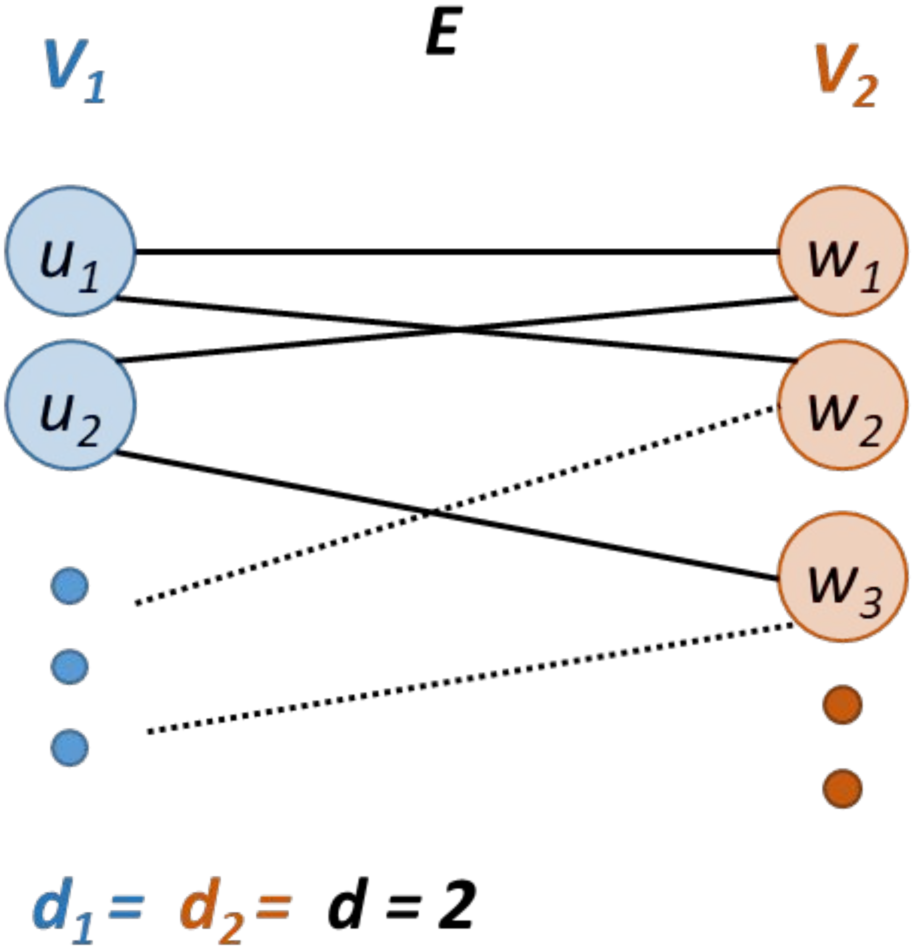
Regular instances (d_1, d_2)

- Each vertex $u \in V_1$ has degree d_1
- Each vertex $w \in V_2$ has degree d_2



d - Regular instances

- All vertices have degree ***d***
- $d_1 = d_2 = d$



2.a Optimization Problems: Minimization Version

Minimization problem

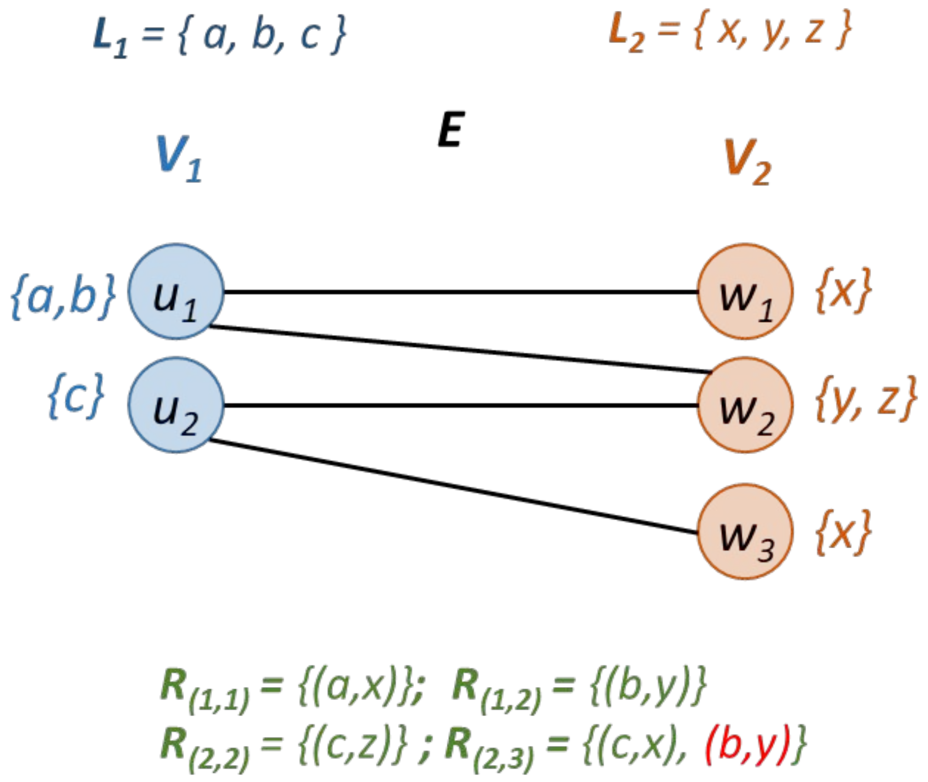
- Assign each vertex a ***subset of labels***.
  - All $\boldsymbol{u}$ in $\boldsymbol{V_1}$ is assigned its own $\boldsymbol{L_u \subseteq L_1}$
  - All $\boldsymbol{w}$ in $\boldsymbol{V_2}$ is assigned its own $\boldsymbol{L_w \subseteq L_2}$

- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $R_{(u,w)}$

- ***Minimize*** total number of labels used
  - $\sum_{u\,\in} |L_u| + \sum_{w\,\in} |L_w|$

- Example: Cost = 7

- Alternate: Cost = 8



Minimization problem

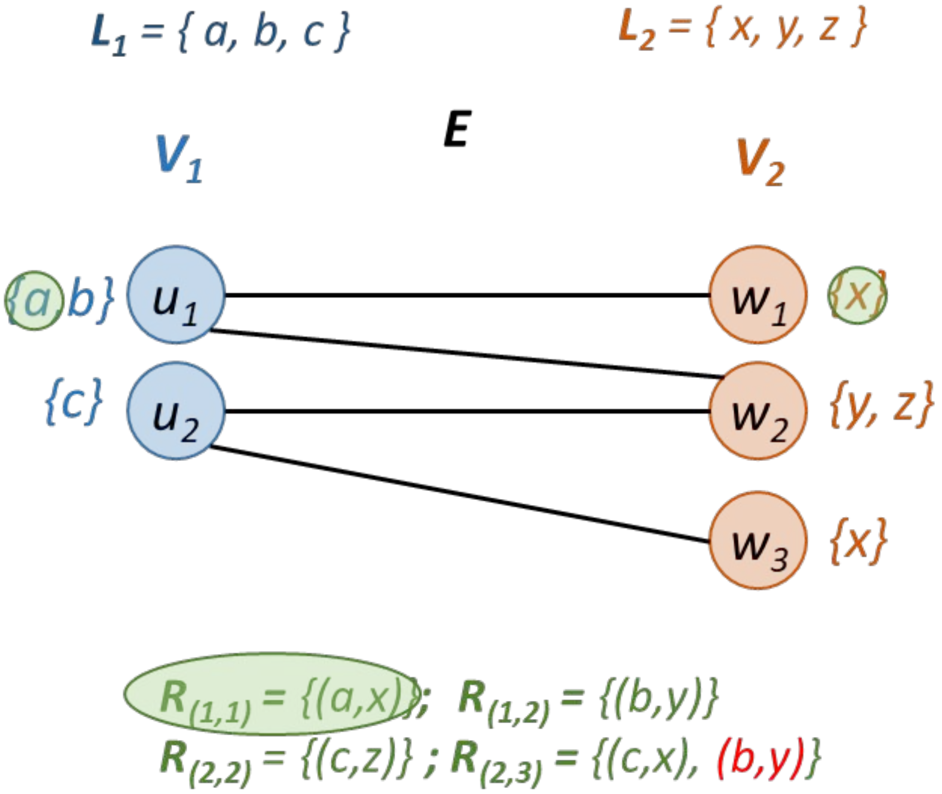
- Assign each vertex a ***<u>subset of labels.</u>***
  - All $\boldsymbol{u}$ in $\boldsymbol{V_1}$ is assigned its own $\boldsymbol{L_u \subseteq L_1}$
  - All $\boldsymbol{w}$ in $\boldsymbol{V_2}$ is assigned its own $\boldsymbol{L_w \subseteq L_2}$

- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $\boldsymbol{R_{(u,w)}}$

- ***Minimize*** total number of labels used
  - $\sum_{u \in} |L_u| + \sum_{w \in} |L_w|$

- Example: Cost = 7

- Alternate: Cost = 8



Minimization problem

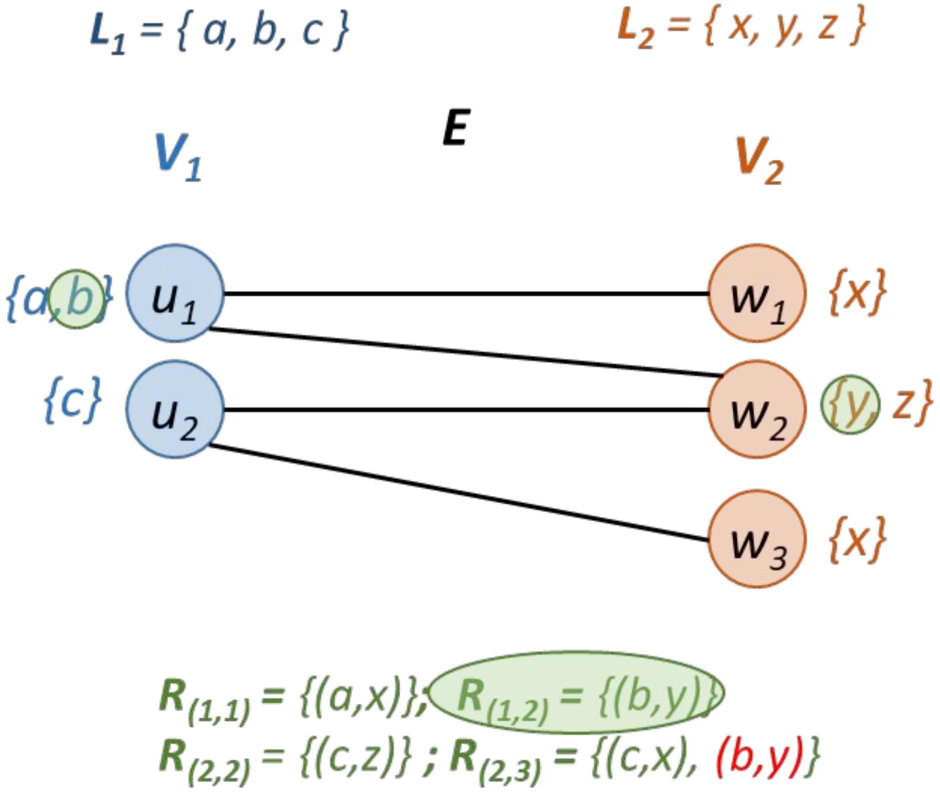
- Assign each vertex a ***subset of labels.***
  - All $\boldsymbol{u}$ in $\boldsymbol{V_1}$ is assigned its own $\boldsymbol{L_u \subseteq L_1}$
  - All $\boldsymbol{w}$ in $\boldsymbol{V_2}$ is assigned its own $\boldsymbol{L_w \subseteq L_2}$

- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $\boldsymbol{R_{(u,w)}}$

- ***Minimize*** total number of labels used
  - $\sum_{u \in} |L_u| + \sum_{w \in} |L_w|$

- Example: Cost = 7

- Alternate: Cost = 8



Minimization problem

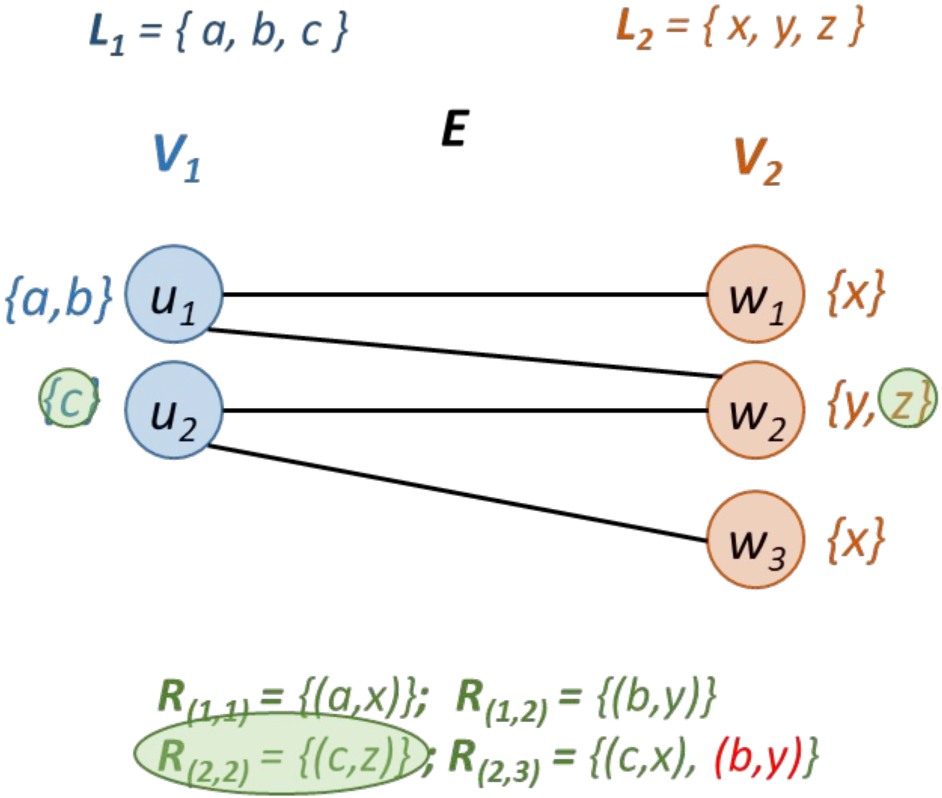
- Assign each vertex a ***subset of labels.***
  - All u in V_1 is assigned its own $L_u \subseteq L_1$
  - All w in V_2 is assigned its own $L_w \subseteq L_2$

- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $R_{(u,w)}$

- ***Minimize*** total number of labels used
  - $\sum_{u \in} |L_u| + \sum_{w \in} |L_w|$

- Example: Cost = 7

- Alternate: Cost = 8



Minimization problem

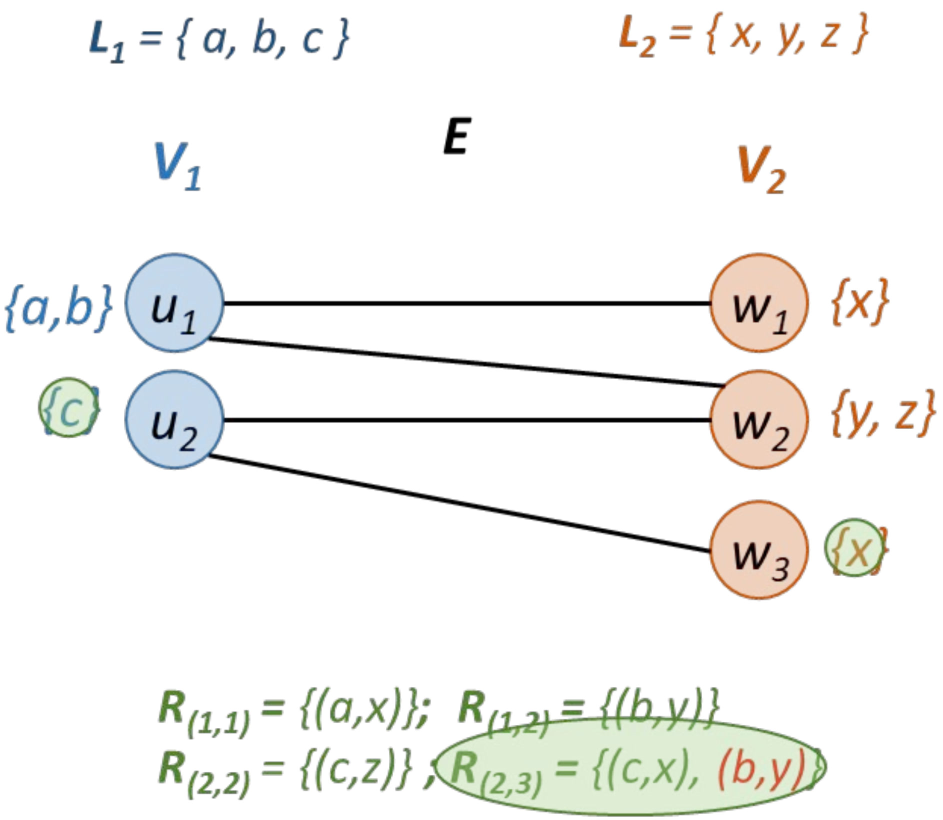
- Assign each vertex a ***<u>subset of labels.</u>***
  - All $\boldsymbol{u}$ in $\boldsymbol{V_1}$ is assigned its own $\boldsymbol{L_u} \subseteq \boldsymbol{L_1}$
  - All $\boldsymbol{w}$ in $\boldsymbol{V_2}$ is assigned its own $\boldsymbol{L_w} \subseteq \boldsymbol{L_2}$

- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $\boldsymbol{R_{(u,w)}}$

- ***Minimize*** total number of labels used
  - $\sum_{u \in} |L_u| + \sum_{w \in} |L_w|$

- Example: Cost = 7

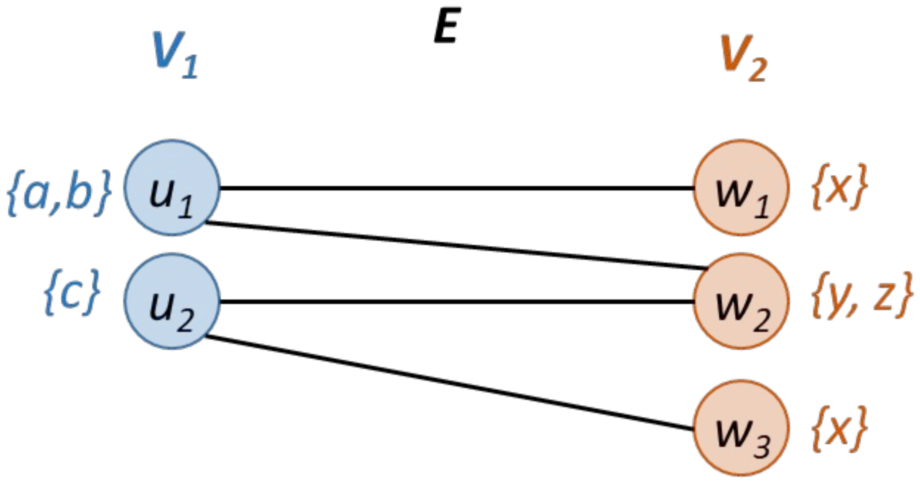
- Alternate: Cost = 8



Minimization problem

- Assign each vertex a ***subset of labels.***
  - All u in V_1 is assigned its own $L_u \subseteq L_1$
  - All w in V_2 is assigned its own $L_w \subseteq L_2$
- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $R_{(u,w)}$
- ***Minimize*** total number of labels used
  - $\sum_{u \in V_1} |L_u| + \sum_{w \in V_2} |L_w|$
- Example: Cost = 7
- Alternate: Cost = 8

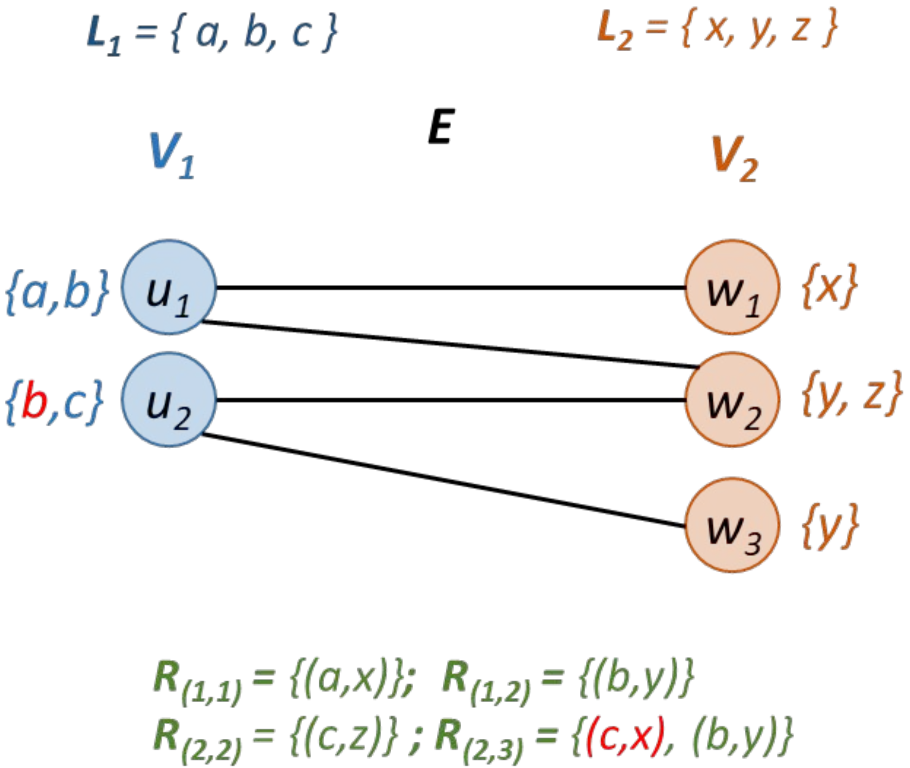
$L_1 = \{a, b, c\}$ $L_2 = \{x, y, z\}$



$R_{(1,1)} = \{(a,x)\}$; $R_{(1,2)} = \{(b,y)\}$
$R_{(2,2)} = \{(c,z)\}$; $R_{(2,3)} = \{(c,x), (b,y)\}$

Minimization problem

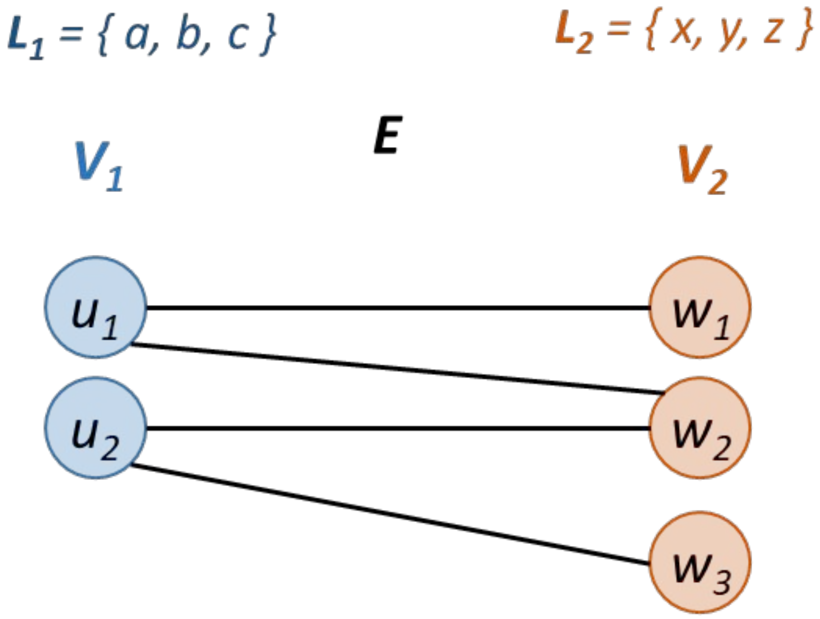
- Assign each vertex a ***<u>subset of labels.</u>***
  - All $\boldsymbol{u}$ in $\boldsymbol{V_1}$ is assigned its own $\boldsymbol{L_u \subseteq L_1}$
  - All $\boldsymbol{w}$ in $\boldsymbol{V_2}$ is assigned its own $\boldsymbol{L_w \subseteq L_2}$
- Such that, for all edges, **at least one pair** of its **vertex labels** is in its $\boldsymbol{R_{(u,w)}}$
- ***Minimize*** total number of labels used
  - $\sum_{u \in V_1} |L_u| + \sum_{w \in V_2} |L_w|$
- Example: Cost = 7
- Alternate: Cost = 8



2.b Optimization Problems: Maximization Version

Maximization Problem

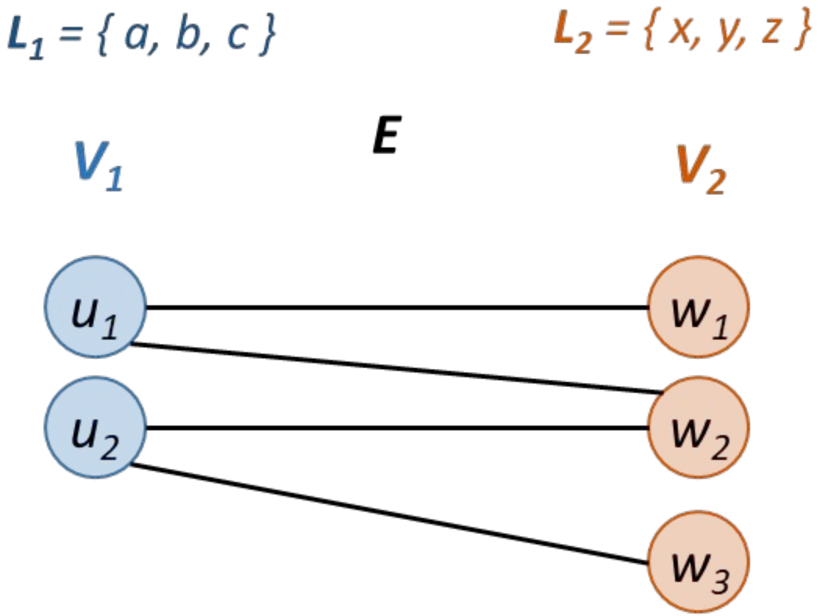
- Assign each vertex ***exactly one label***
- Try to satisfy ***as many edges as possible***



$\boldsymbol{R_{(1,1)}} = \{(a,x)\}$; $\boldsymbol{R_{(1,2)}} = \{(b,y), (a,z)\}$
$\boldsymbol{R_{(2,2)}} = \{(c,z)\}$; $\boldsymbol{R_{(2,3)}} = \{(a,x), (b,y)\}$

Maximization Problem

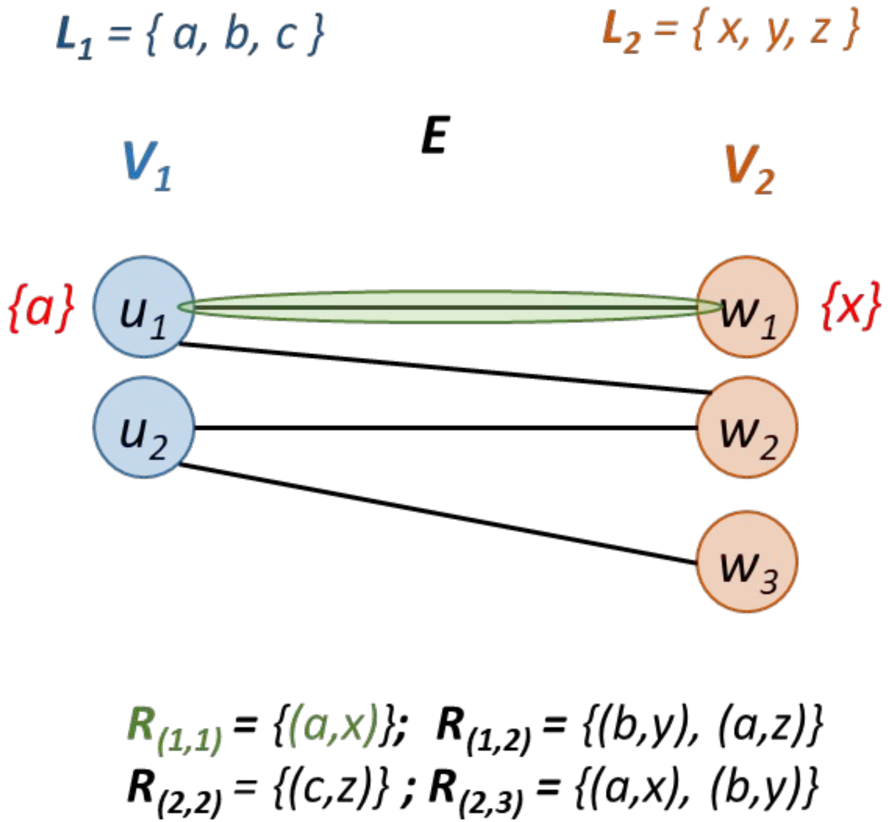
- Assign each vertex ***exactly one label***
- Try to satisfy ***as many edges as possible***



$\mathbf{R}_{(1,1)} = \{(a,x)\}$; $\mathbf{R}_{(1,2)} = \{(b,y), (a,z)\}$
$\mathbf{R}_{(2,2)} = \{(c,z)\}$; $\mathbf{R}_{(2,3)} = \{(a,x), (b,y)\}$

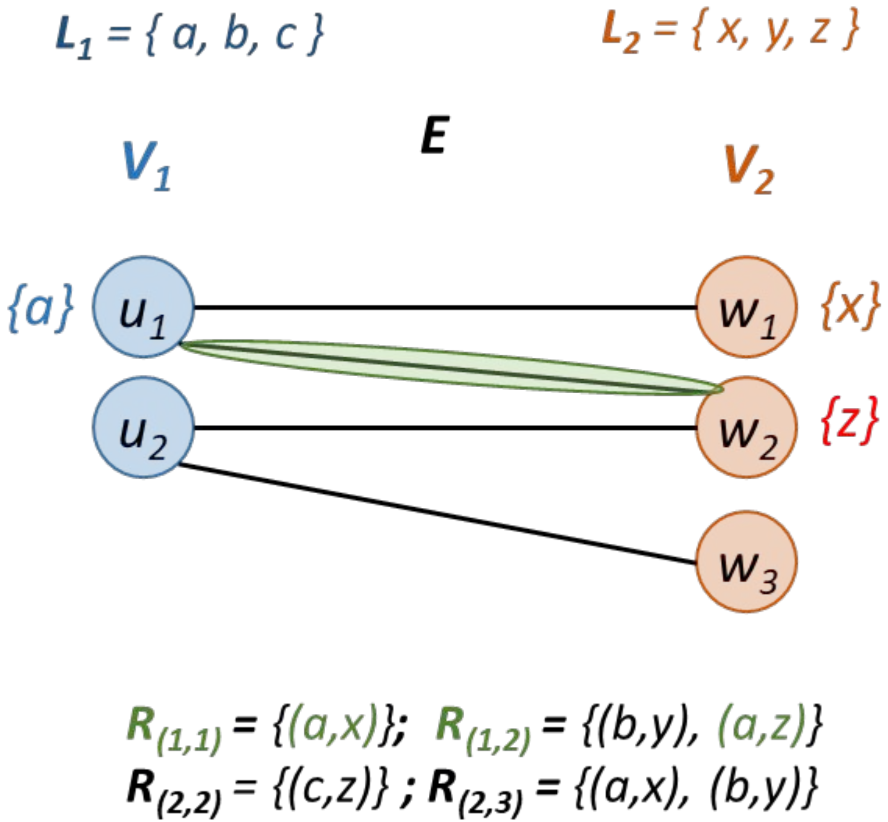
Maximization Problem

- Assign each vertex ***exactly one label***
- Try to satisfy ***as many edges as possible***



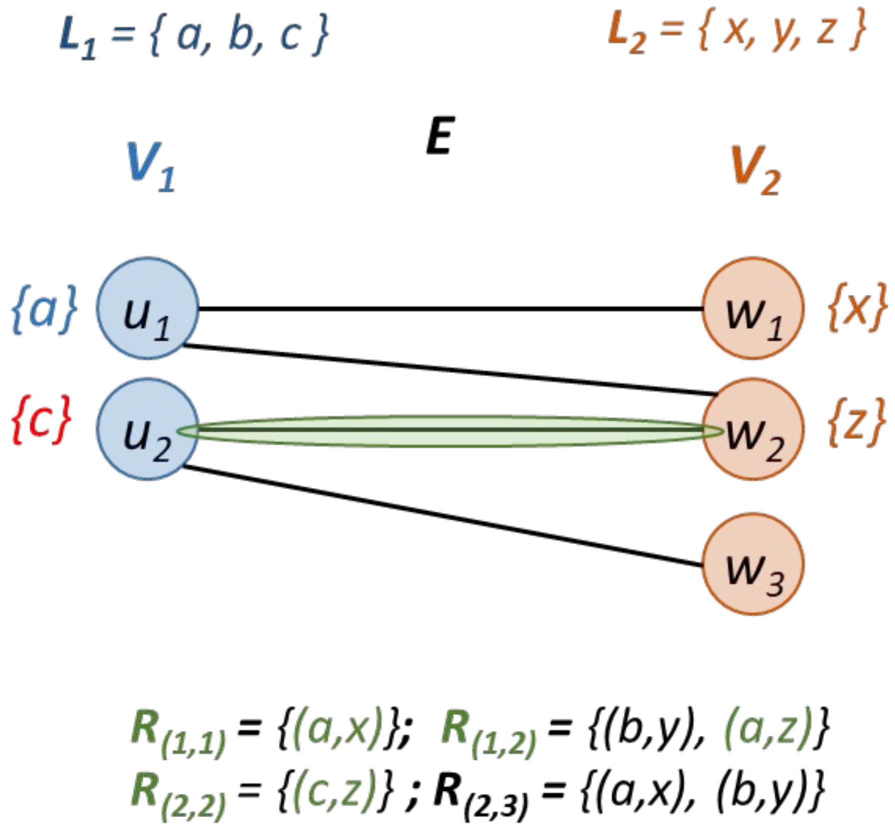
Maximization Problem

- Assign each vertex ***exactly one label***
- Try to satisfy ***as many edges as possible***



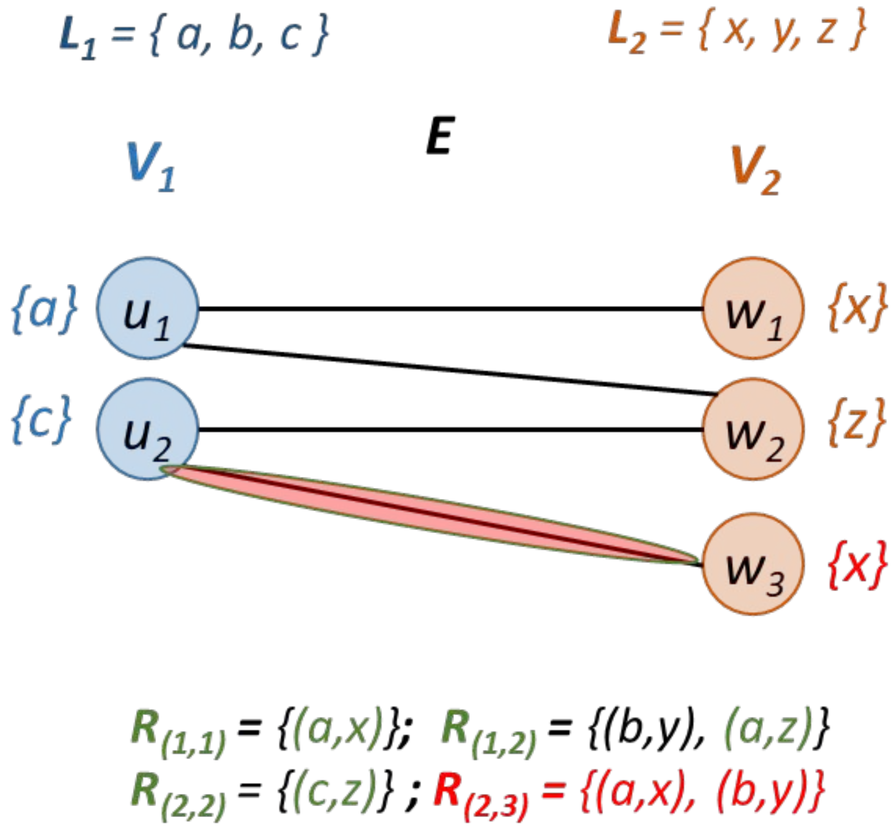
Maximization Problem

- Assign each vertex ***exactly one label***
- Try to satisfy ***as many edges as possible***



Maximization Problem

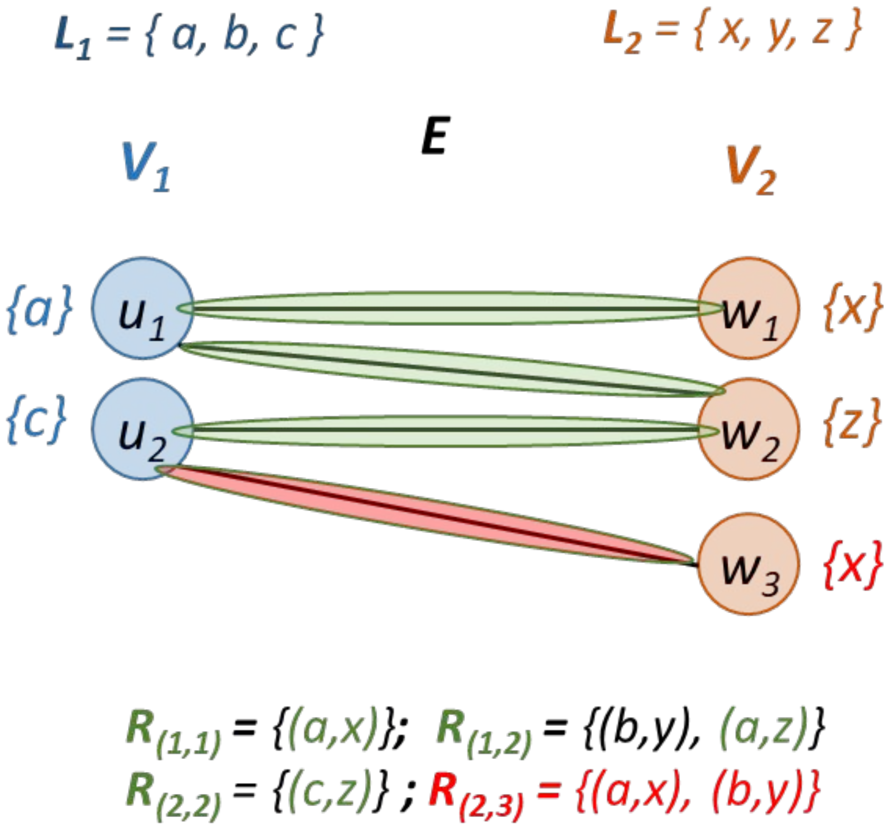
- Assign each vertex ***exactly one label***
- Try to satisfy ***as many edges as possible***



Maximization Problem

- Assign each vertex ***<u>exactly one label</u>***
- Try to satisfy ***as many edges as possible***

- ***SCORE = 3***



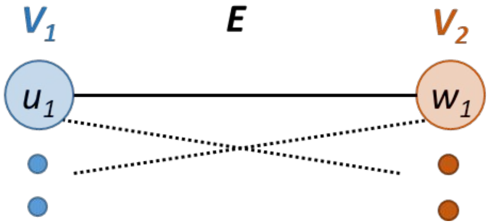
Example: Unique Games Problem

In short:

- A ***binary constraint satisfaction problem***, where
- If there is a constraint on $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, then
    - for one assignment of $\boldsymbol{x_i}$ there will be exactly one assignment of $\boldsymbol{x_j}$ that satisfies the constraint, and vice versa

Unique Games as a Label Cover (Maximization)

- $L_1 = L_2 = L$
- Each $R_{(u,w)}$ will be $L \rightarrow \pi(L)$, ie
  - a function from L to a permutation of L
- So that one label assignment to a vertex u allows exactly one assignment to w where (u,w) is an edge.

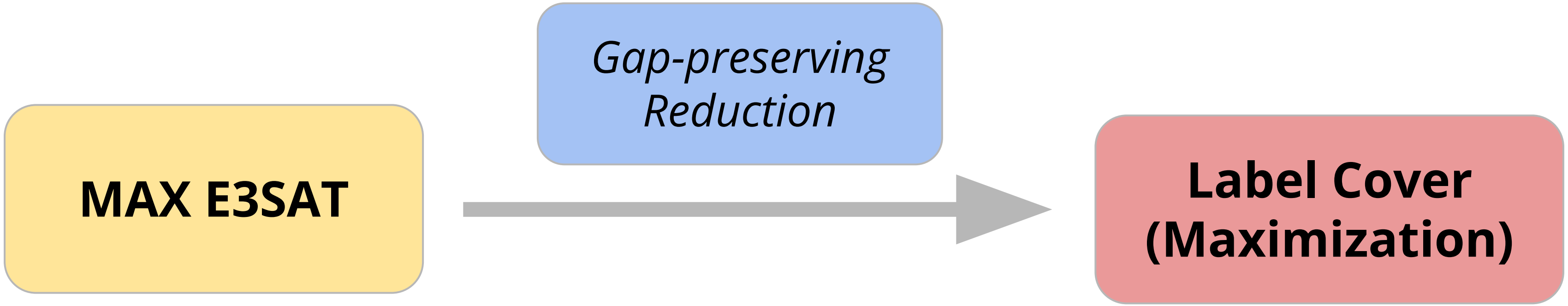


$L = \{a,b,c\}$

$R_{(1,1)} = \{(a,c), (b,a), (c,b)\}$

3. Hardness of Approximating the Maximization Problem

Main Idea


MAX E3SAT
Gap-preserving Reduction
Label Cover (Maximization)

Short Recap 1 (Gap-preserving Reduction)

For a *(**Constraint satisfying**, **NP-Complete**)* Problem A:

- Distinguishing between ***instances*** with
  - **≥ U_A *satisfiable constraints*** and
  - **≤ L_A *satisfiable constraints*** in ***poly time***,
- will lead to ***solving decision versions of Problem A in poly time***.

Short Recap 1 (Gap-preserving Reduction)

If problem A is ***"gap-preserve reduced"*** to problem B

- We get similar upper (U_B) and lower (L_B) bounds for problem B, such that
  - $U_B = f(U_A)$
  - $L_B = f(L_A)$
- That is, the "***gap***"s [(U_A , L_A) and (U_B , L_B)] are ***preserved***

Short Recap 2 (MAX E3SAT)

$\boxed{\textbf{MAX}}\ \boxed{\textbf{E3}}\ \boxed{\textbf{SAT}}$

SAT: Satisfiability Problem (CNF)

E3: Exactly 3 literals per clause

MAX: Maximize number of clauses satisfied

Reduction (MAX E3 SAT to Label Cover (Max))

Given: a MAX E3 SAT problem

- X = set of variables
- C = set of clauses in an E3 CNF
  - m total clauses

$$X = \{x_1, x_2, x_3 \dots \dots x_n\}$$

$$\textbf{E3CNF: } \boxed{(x_1 \vee x_2 \vee x_3)} \wedge \boxed{(x_2' \vee x_5 \vee x_6')} \wedge \dots \wedge \boxed{(x_8' \vee x_9 \vee x_{10}')}$$

$$C = \{\boxed{C_1}, \boxed{C_2}, \dots, \boxed{C_n}\}$$

Reduce to: A Label Cover instance (maximization)

- Bipartite graph (V_1, V_2, E)
- Label sets L_1, L_2
- Edge Relations $R_{(u,w)}$ for all edges $(u,w) \in E$

Reduction (MAX E3 SAT to Label Cover (Max))

V_1 : Set of vertices $\boldsymbol{u_i}$ for each variable $\boldsymbol{x_i} \in \boldsymbol{X}$

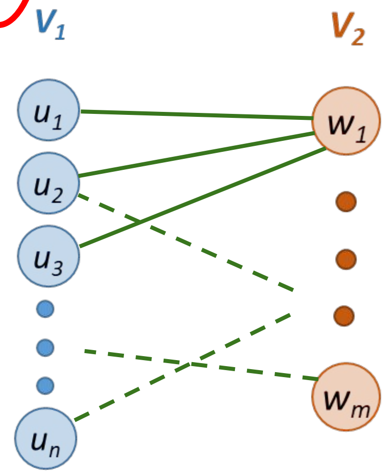
V_2 : Set of vertices $\boldsymbol{w_j}$ for each clause $\boldsymbol{C_j} \in \boldsymbol{C}$

E : Join vertices $\boldsymbol{u_i}$ and $\boldsymbol{w_j}$ if i^{th} variable, $\boldsymbol{x_i}$ occurs in jth clause, $\boldsymbol{C_j}$ ***(+vely or -vely)***

E3CNF: $(x_1 \vee x_2 \vee x_3) \wedge (x_2' \vee x_5 \vee x_6') \wedge \ldots \wedge (x_8' \vee x_9 \vee x_{10}')$

$\boldsymbol{X} = \{x_1, x_2, x_3 \ldots \ldots x_n\}$

$\boldsymbol{C} = \{C_1, C_2, \ldots, C_m\}$



Reduction (MAX E3 SAT to Label Cover (Max))

$\boldsymbol{L_1}$: { T , F}

$\boldsymbol{L_2 : L_1 \times L_1 \times L_1}$

= { (T,T,T), (T,T,F), (T,F,T), (T,F,F),

(F,T,T), (F,T,F), (F,F,T), (F,F,F) }

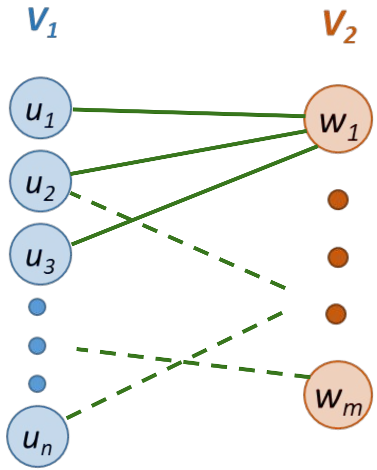
Intuition:

- L_1 : value of corresponding ***variable***
- L_2 : value of corresponding ***clause***

$\boldsymbol{V_1}$: represent variables

$\boldsymbol{V_2}$: represent clauses

E : represent variable belonging to clause



Reduction (MAX E3 SAT to Label Cover (Max))

$R_{(u,w)} \subset L_1 \times L_2$

- Only L_2 values that satisfy clause (7 values)
- L_1 value corresponding to variable value in clause
- Let $C_1 = (x_1 \vee x_2' \vee x_3')$ **[(F,T,T) will not satisfy]**
- So, the following edges exist:

- For edge (u_1, w_1),
  - $R_{11} = \{b_1, (b_1, b_2, b_3)\}$

    = { (T,(T,T,T)), (T,(T,T,F)), (T,(T,F,T)), (T,(T,F,F)),

    (F,(F,T,F)), (F,(F,F,T)), (F,(F,F,F)) }
  - ***No (F,(F,T,T))***

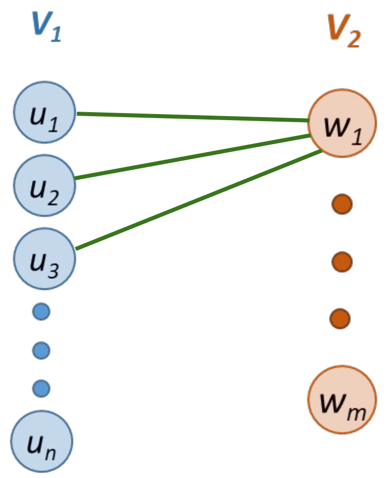
V_1 : represent variables

V_2 : represent clauses

E : represent variable belonging to clause

L_1 : represent variable assignment

L_2 : represent clause assignment



Reduction (MAX E3 SAT to Label Cover (Max))

Intuition:

- We see, edge connecting a ***clause*** with its ***first variable*** will allow ***R*** values in the form $(b_1, (b_1, b_2, b_3))$
- For edge (u_1, w_1), [w_1 represent $(x_1 \lor x_2' \lor x_3')$]
  - $R_{11} = \{ b_1, (b_1, b_2, b_3) \}$

    $= \{ (T,(T,T,T)), (T,(T,T,F)), (T,(T,F,T)), (T,(T,F,F)),$

    $(F,(F,T,F)), (F,(F,F,T)), (F,(F,F,F)) \}$
- similarly, an edge connecting a ***clause*** with its ***second variable*** will allow ***R*** values in the form $(b_2, (b_1, b_2, b_3))$
- and, an edge connecting a ***clause*** with its ***third variable*** will allow ***R*** values in the form $(b_3, (b_1, b_2, b_3))$

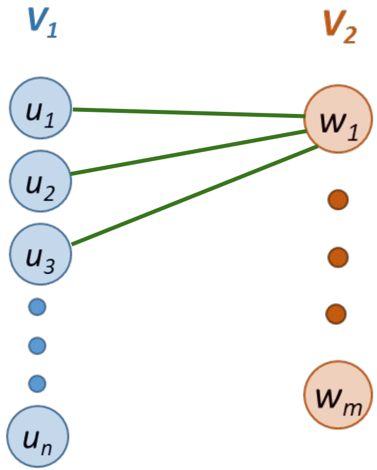
V_1 : represent variables

V_2 : represent clauses

E : represent variable belonging to clause

L_1 : represent variable assignment

L_2 : represent clause assignment



Reduction (MAX E3 SAT to Label Cover (Max))

Important Property:

Given an edge, if **a clause is labelled** only **<u>one value for the variable</u>** will satisfy that edge.

Eg.

- Edge: (u_1, w_1)
- $R_{11} = \{ b_1, (b_1, b_2, b_3) \}$

  $= \{ (T,(T,T,T)), (T,(T,T,F)), (T,(T,F,T)), \boxed{(T,(T,F,F)),}$

  $(F,(F,T,F)), (F,(F,F,T)), (F,(F,F,F)) \}$
- Clause Label: (T, F, F)
- Variable label must be T

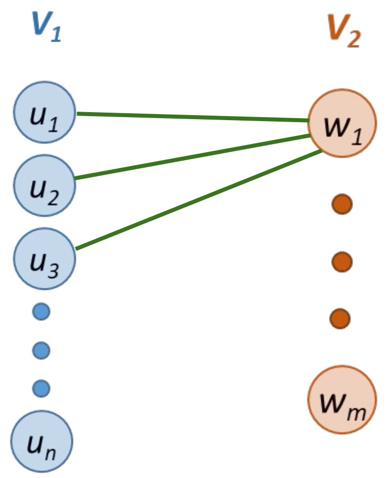
V_1 : represent variables

V_2 : represent clauses

E : represent variable belonging to clause

L_1 : represent variable assignment

L_2 : represent clause assignment



Proof that this is a Gap-Preserving Reduction

- If there are **m clauses** in the E3SAT, there are **3m edges** in the label cover instance (per clause 3 edges to its 3 variables)

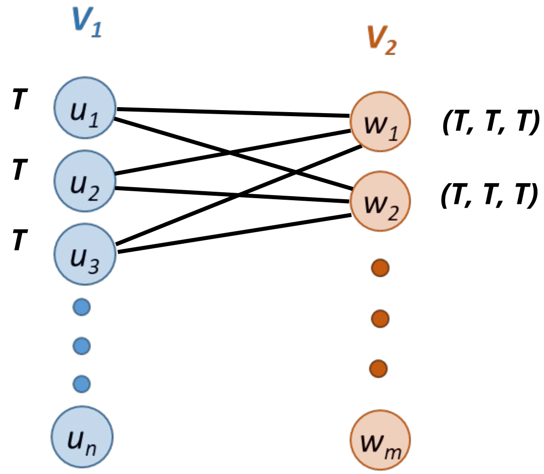
- ***Claim:***
  - ***Given:*** An assignment to variables of E3SAT st. ***k* clauses** satisfied
  - ***Can get:*** A solution to corresponding label cover that satisfies ***3k + 2(m-k)*** edges

Proof that this is a Gap-Preserving Reduction

V_1 : represent **variables**
V_2 : represent **clauses**
E : represent **variable belonging to clause**
L_1 : represent **variable assignment**
L_2 : represent **clause assignment**

- Example:
  - E3SAT: $(x_1 \vee x_2 \vee x_3) \wedge (x_1' \vee x_2' \vee x_3') \ldots$
  - Given Assignment: $x_1 = x_2 = x_3 = T$

- For all variables
  - Label each $v_i \in V_1$ with value of ***corresponding variable*** x_i
- *For all clauses*
  - Primarily assign all $w_i \in V_2$ with triplet of values of constituent ***variables***



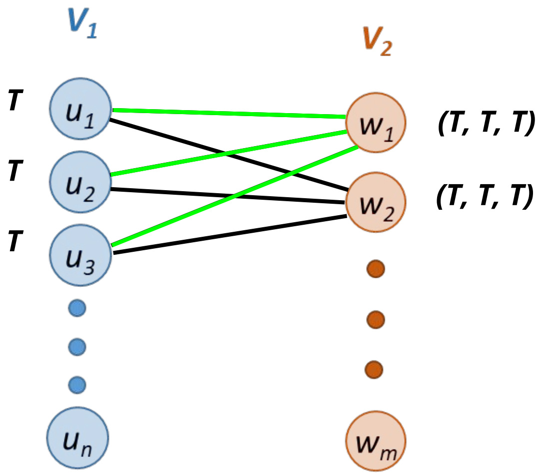
Proof that this is a Gap-Preserving Reduction

- For satisfied clauses:
  - The aforementioned assignment of labels ensure **3 satisfied edges** each
  - If number of satisfied clauses **= k,** then **3k** edges are satisfied

- For unsatisfied clauses:
  - Initially no edges satisfied
  - Flip 1 element of tuple
  - 2 edges satisfied per tuple
  - 2 (m-k) edges satisfied overall

E3SAT: $(x_1 \vee x_2 \vee x_3) \wedge (x_1' \vee x_2' \vee x_3') \ldots$

Given Assignment: $x_1 = x_2 = x_3 = T$



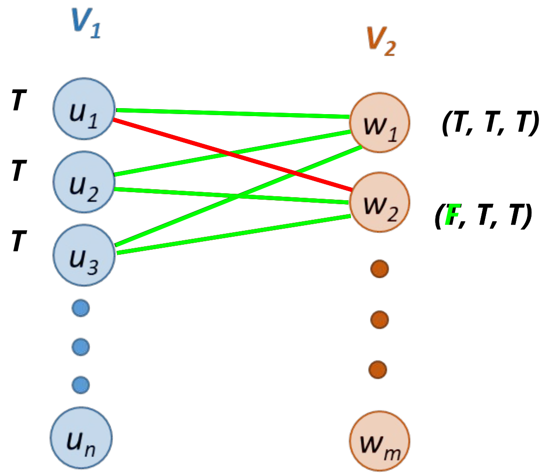
Proof that this is a Gap-Preserving Reduction

- For unsatisfied clauses:
  - Initially no edges satisfied
  - Flip 1 element of tuple
  - 2 edges satisfied per tuple
  - 2 (m-k) edges satisfied overall

- Example:
  - $C_2 = (x_1' \vee x_2' \vee x_3')$ evaluates to false for ***(T,T,T)***
  - So, R_{12}, R_{22}, R_{32} does not have ***(T, (T,T,T))***
- Suppose, change w_2 label to ***(F,T,T)***
  - R_{12} does not have ***(T, (F,T,T))***
  - But R_{22} has ***(T,(F,T,T))*** and R_{32} has ***(T,(F,T,T))***

E3SAT: $(x_1 \vee x_2 \vee x_3) \wedge (x_1' \vee x_2' \vee x_3')$ ...

Given Assignment: $x_1 = x_2 = x_3 = T$



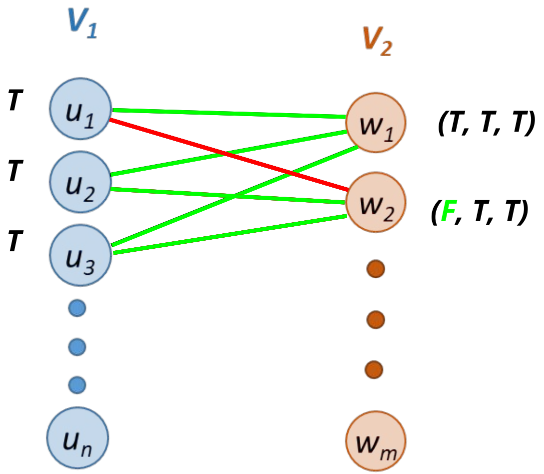
Proof that this is a Gap-Preserving Reduction

- Thus total edges satisfied
  - From sat clauses + unsat clauses

    = ***3k + 2(m-k)*** *[Proved]*

E3SAT: $(x_1 \lor x_2 \lor x_3) \land (x_1' \lor x_2' \lor x_3')$ ...

Given Assignment: $x_1 = x_2 = x_3 = T$



Proof that this is a Gap-Preserving Reduction

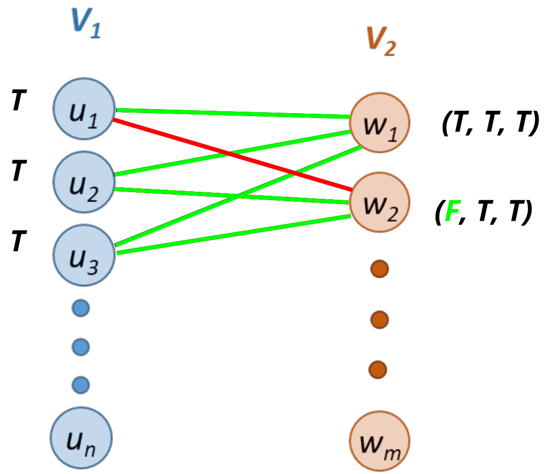
E3SAT: $(x_1 \vee x_2 \vee x_3) \wedge (x_1' \vee x_2' \vee x_3')$ …

Given Assignment: $x_1 = x_2 = x_3 = T$

Conversely,

- Each $w_j \in V_2$ with 3 satisfied incident edges MUST correspond to a satisfied clause $\mathbf{C_j}$
- If a label cover problem is solved with **3k + 2(m-k) satisfied edges**, then
- we can construct a corresponding solution to the E3SAT problem instance with **3m clauses satisfied**

Thus completing the reduction



4. Some Theorems

Theorem 16.25

Statement: If for any constant α > $\frac{23}{24}$, there is an α - approximation algorithm for the maximization version of the Label Cover problem, then **P = NP**

Proof:

- Hard to distinguish between MAX E3SAT problems where all *m* clauses satisfiable and $\left(\frac{7}{8}+\delta\right)m$ clauses are satisfiable
- From reduction above, hard to distinguish between [3m + 2 (m-m)] = ***3m edges*** and *[3* $\left(\frac{7}{8}+\delta\right)m$ *+ 2 (m -* $\left(\frac{7}{8}+\delta\right)m$ *)]* = $\left(\frac{23}{8}+\delta\right)m$ edges

- Fraction = ($\left(\frac{23}{8}+\delta\right)m$ */ 3m*) = $\left(\frac{23}{24}+\frac{\delta}{3}\right)$

- Thus hard to get an approx algo with α > $\frac{23}{24}$ ***[Proved]***

Other Theorems

Theorem 16.26. *If for a particular constant $\alpha < 1$ there is an α-approximation algorithm for the maximization version of the label cover problem on (5,3)-regular instances, then* $\mathrm{P} = \mathrm{NP}$.

Remark*: Similar reduction as before*

Theorem 16.27. *If for a particular constant $\alpha < 1$ there is an α-approximation algorithm for the maximization version of the label cover problem on 15-regular instances, then* $\mathrm{P} = \mathrm{NP}$.

Remark*: Follows from a construction based on* ***Theorem 16.26***

Other Theorems

Theorem 16.28. *There is a constant $c > 0$, such that for any label cover instance I with m edges and $L = |L_1| + |L_2|$ total labels, if $\mathrm{OPT}(I) = |E|(1-\delta)$, then $\mathrm{OPT}(I') \leq |E'|(1-\delta)^{\frac{ck}{\log L}}$.*

Remark: *Proof deemed "Highly non-trivial" and thus not provided*

Theorem 16.29. *There is a constant $c > 0$, such that for any positive integer k, we cannot distinguish between d-regular instances I of the maximization label cover problem in which $\mathrm{OPT}(I) = |E|$ and $\mathrm{OPT}(I) = |E|(1-\delta)^{\frac{ck}{\log 10}}$ for some constant $\delta > 0$ unless each problem in NP has an algorithm running in time $O(n^{O(k)})$.*

Remark: *Follows from applying **Theorem 16.28** for a fixed k*

Corollary 16.30. *There is no α-approximation algorithm for any constant $\alpha \leq 1$ for the maximization version of the label cover problem unless $\mathrm{P} = \mathrm{NP}$.*

Remark: *Follows from applying **Theorem 16.29***

Other Theorems

Theorem 16.31. *For any* $\epsilon > 0$*, there is no* $2^{-\log^{1-\epsilon} m}$*-approximation algorithm for the maximization version of the label cover problem with* d*-regular instances unless* NP *has quasipolynomial-time algorithms.*

Remark: *Uses* ***Theorem 16.29***

Theorem 16.32. *There is no* $(\frac{1}{32}\log N)$*-approximation algorithm for the unweighted set cover problem (where* N *is the size of the ground set of elements) unless each problem in* NP *has an algorithm running in time* $O(n^{O(\log\log n)})$.