CS 5003: Parameterized Algorithms

Lectures 16-17

Krithika Ramaswamy

IIT Palakkad

Reference Books: Parameterized Algorithms by Cygan et al. and Kernelization by Fomin et al.

Vertex Cover Above LP

Vertex Cover Above LP

Instance: A graph G on n vertices m edges and integer k
Question: Does G have a vertex cover of size at most k?
Parameter: k-lp(G)

lp(G) >= |M|

Vertex Cover Above Matching

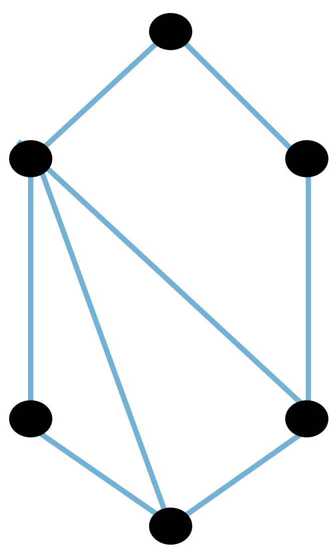
Instance: A graph G on n vertices m edges, integer k, a matching M
Question: Does G have a vertex cover of size at most k?
Parameter: k-|M|

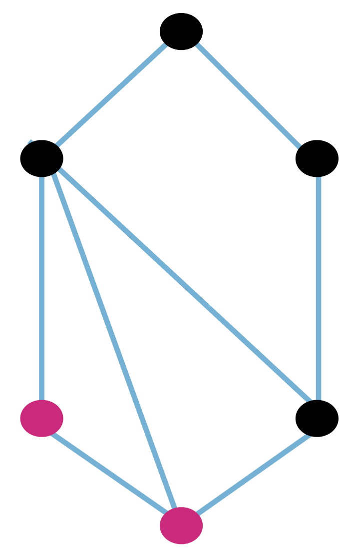
We have lp(G) >= M (think about it) and thus this makes sense.

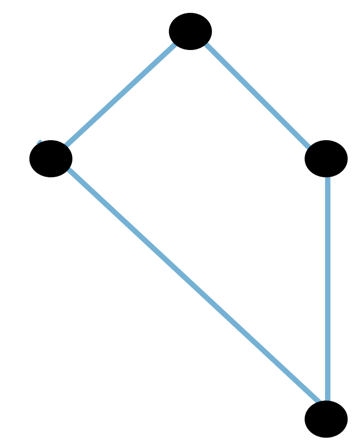
$4^{(k-lp(G))} n^{O(1)}$ time algorithm is a $4^{(k-|M|)} n^{O(1)}$ time algorithm

Odd Cycle Transversal

OCT - set of vertices that has at least one vertex of every odd length cycle







bipartite

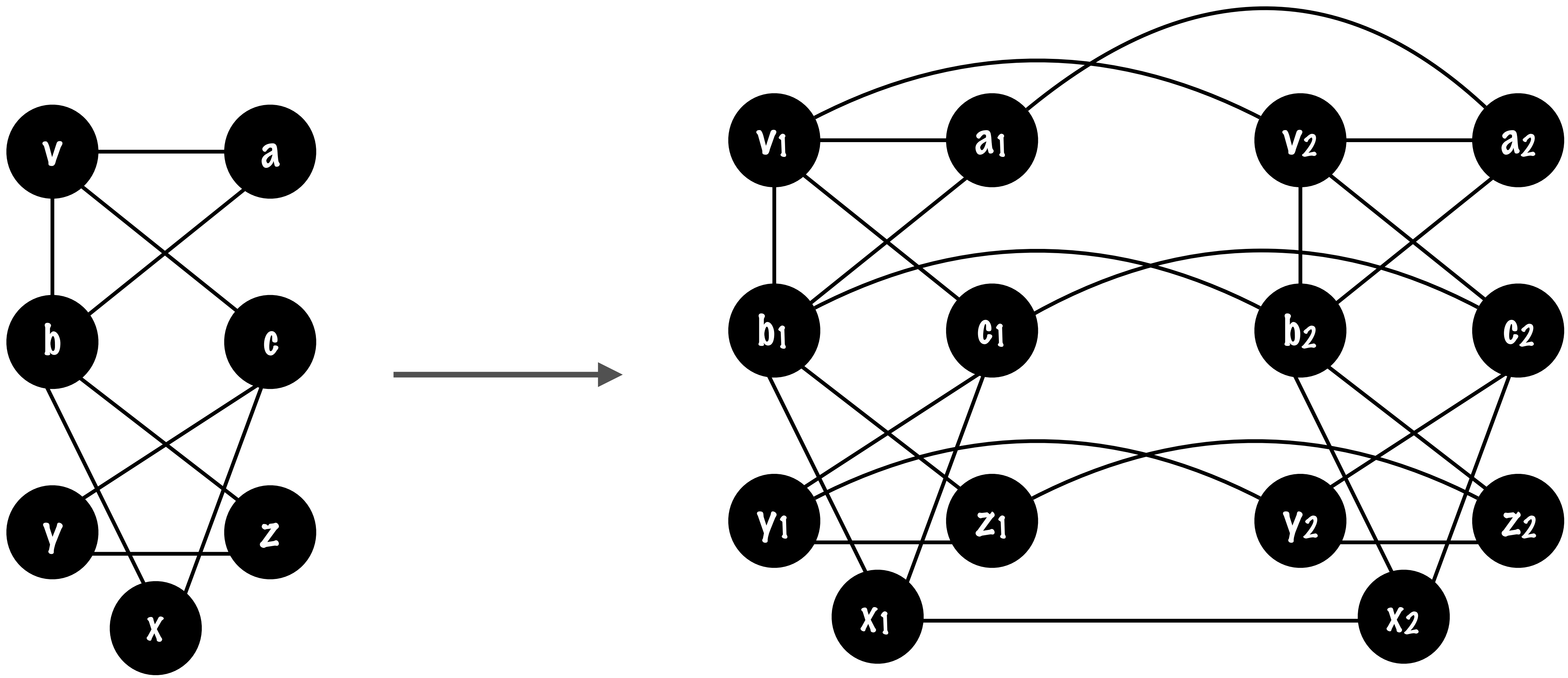
A graph is bipartite iff it has no odd cycle

forward dirn can be proved by contradiction, for reverse dirn, let v be any vertex in G, define X = {x | d_G(v, x) is even} and Y = {y | d_

Odd Cycle Transversal

<u>Instance:</u> A graph G on n vertices m edges and integer k
<u>Question:</u> Does G have an oct of size at most k?
<u>Parameter:</u> k

OCT Reduces to VC Above LP



(G, k) OCT

(H, |V(G)|+k) Vertex Cover

G has an OCT of size k iff H has a VC of size |V(G)|+k

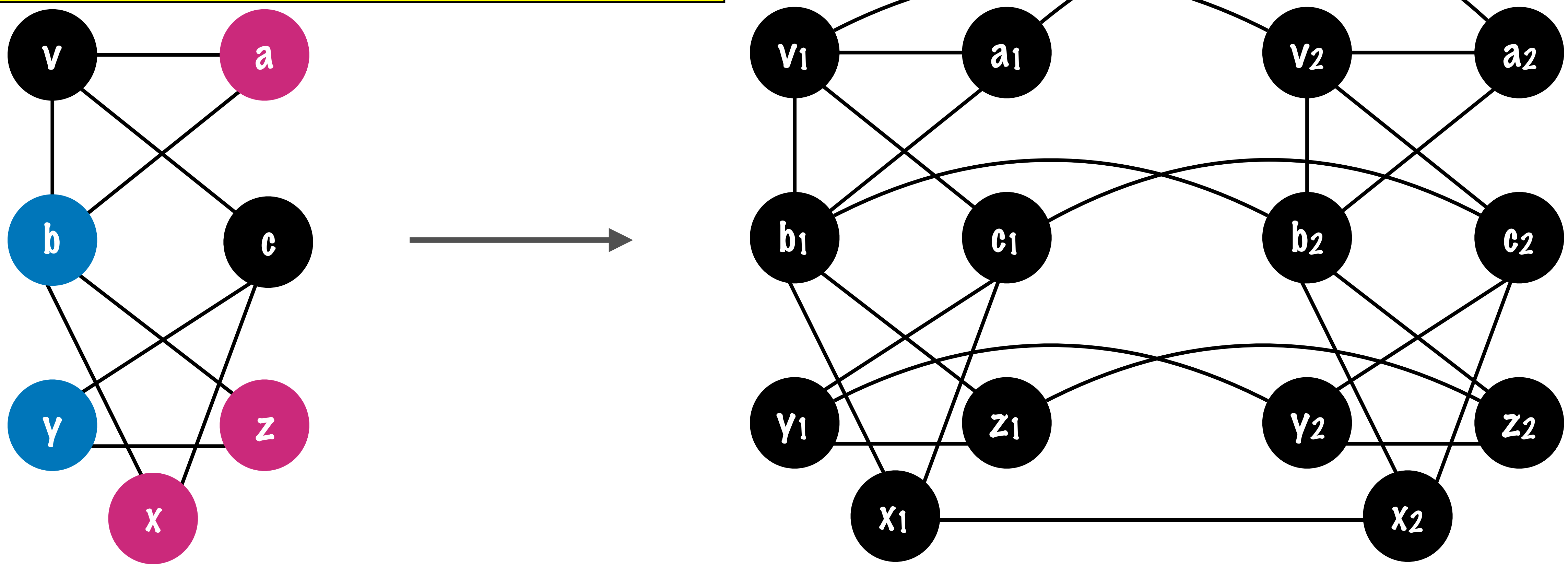
<=

<=

OCT Reduces to VC Above LP

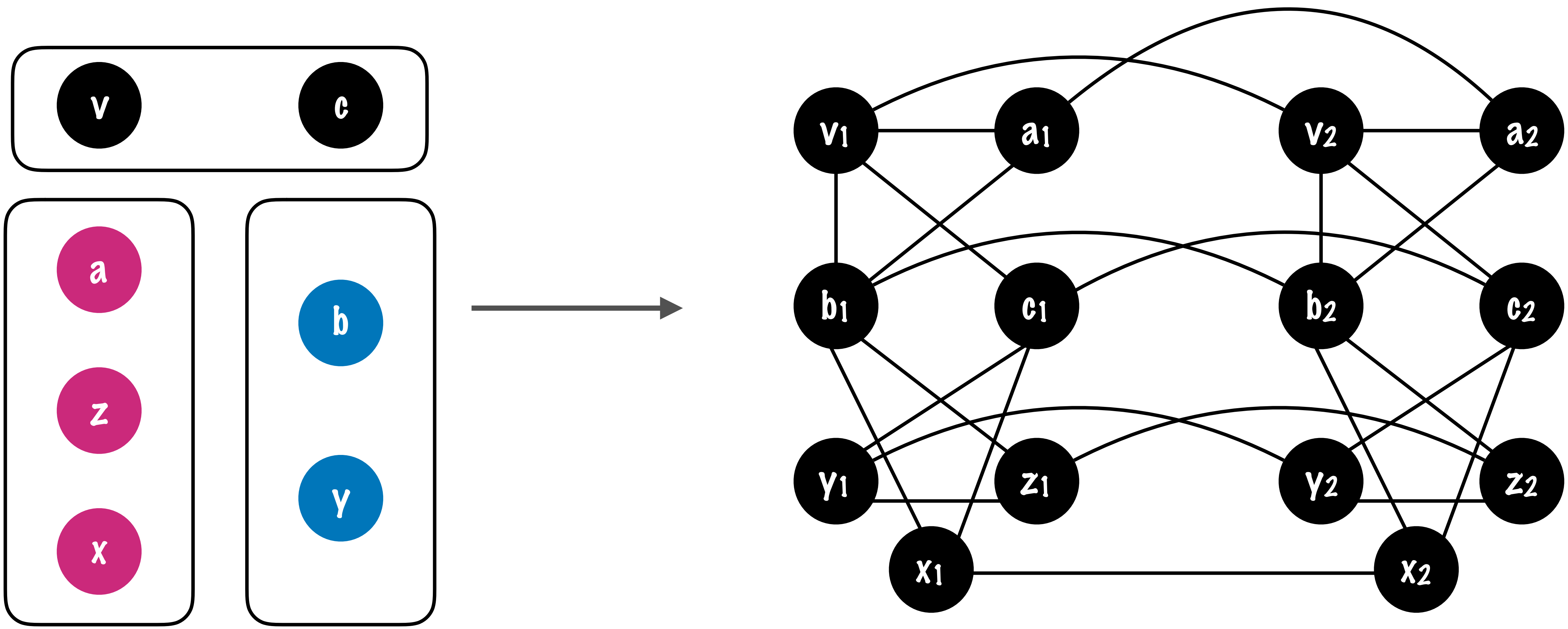
Suppose G has an OCT of size k

black denotes OCT and other 2 colors are for bipartite nes



OCT Reduces to VC Above LP

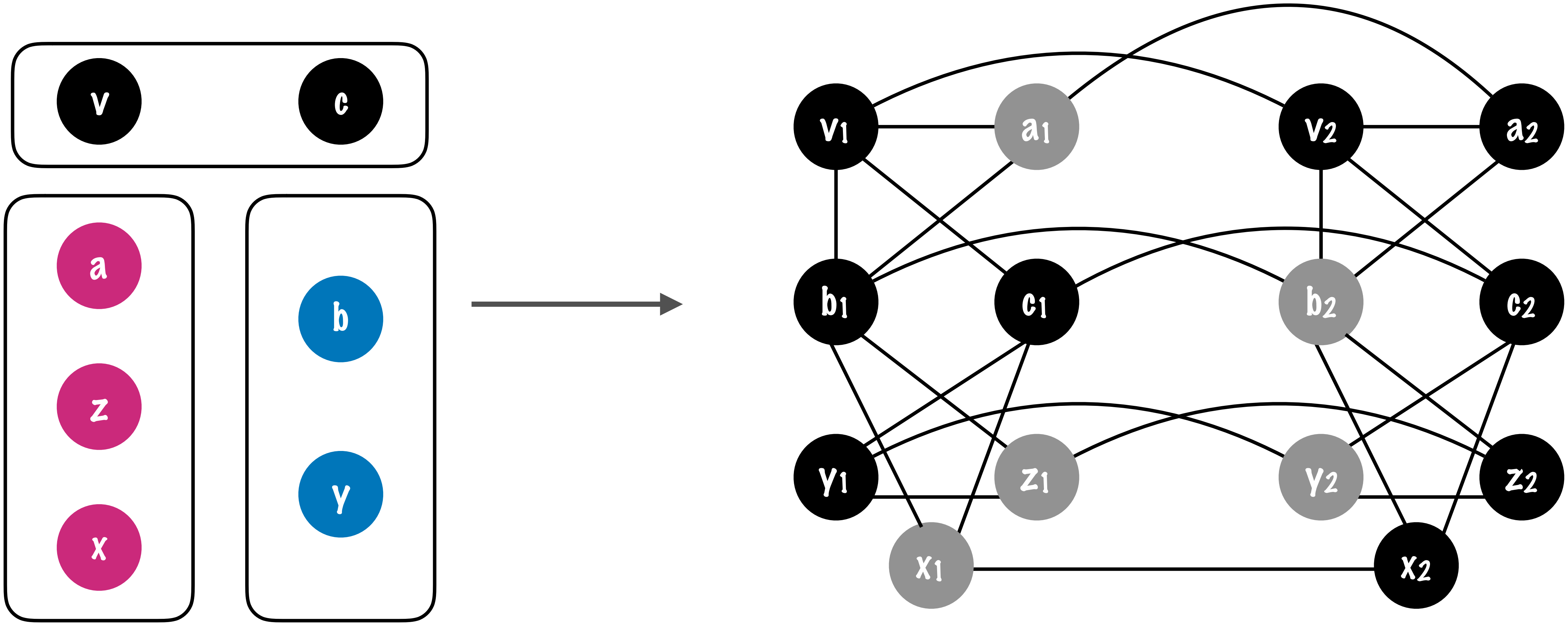
Suppose G has an OCT of size k



G has a bipartite graph of size |V(G)|-k

OCT Reduces to VC Above LP

Suppose G has an OCT of size k



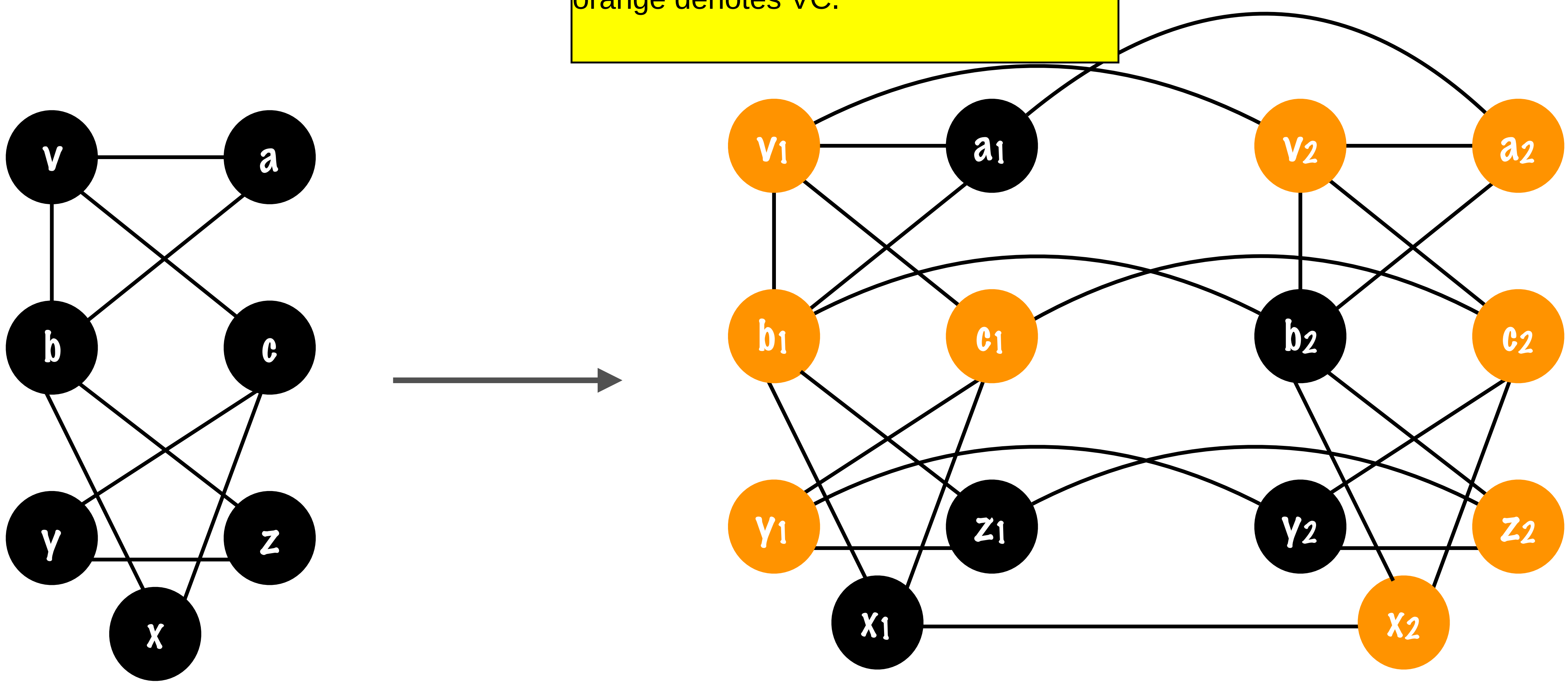
H has an IS of size |V(G)|-k

H has a VC of size |V(G)|+k

OCT Reduces to VC Above LP

Suppose H has a VC of size |V(G)|+k

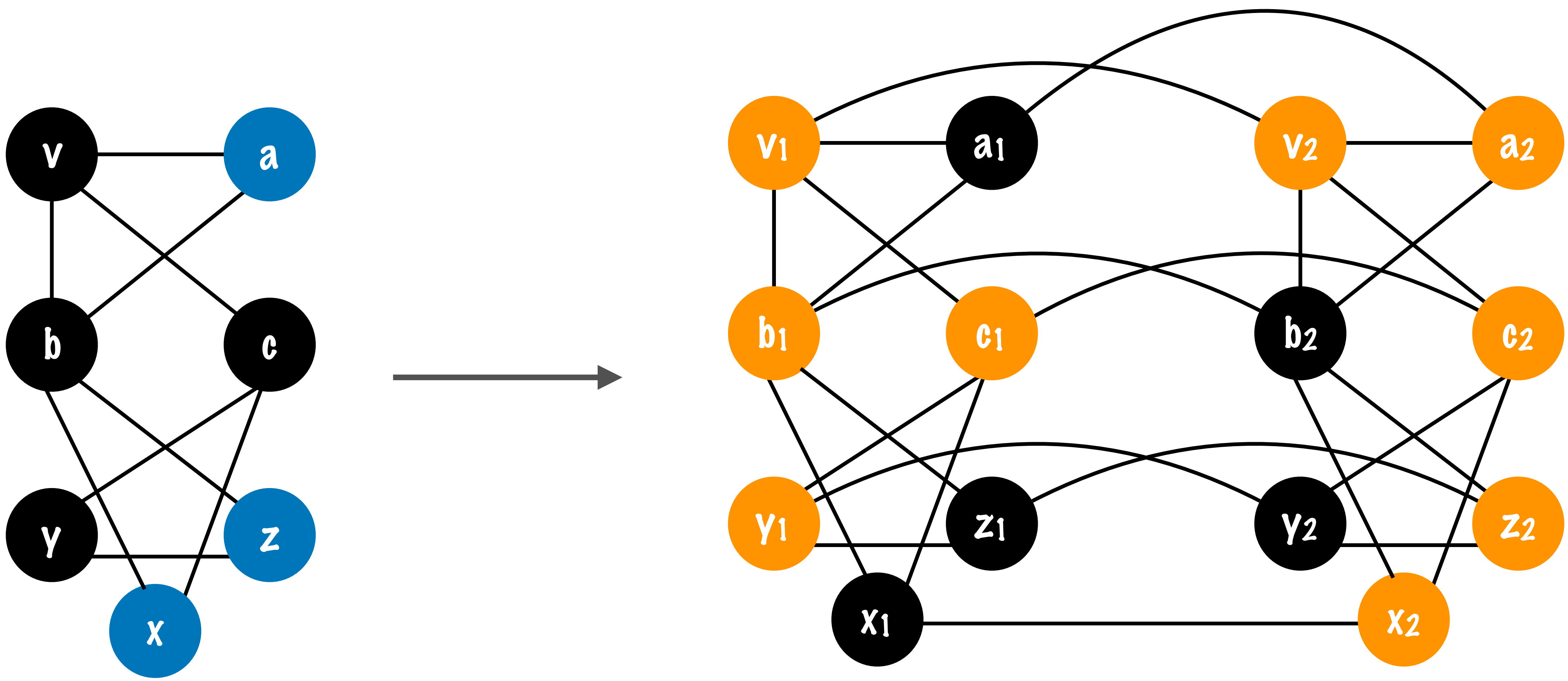
orange denotes VC.



H has an IS of size |V(G)|-k

OCT Reduces to VC Above LP

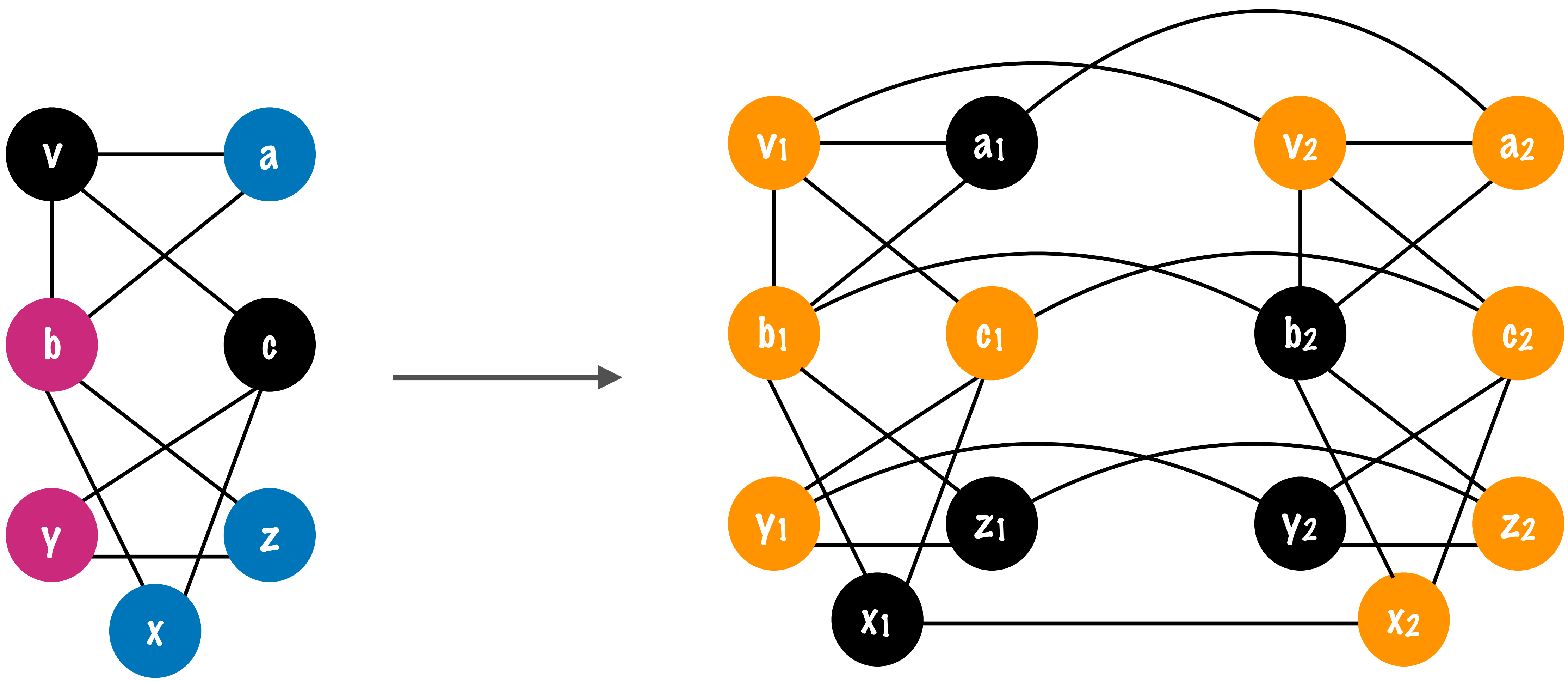
Suppose H has a VC of size |V(G)|+k



H has an IS of size |V(G)|-k

OCT Reduces to VC Above LP

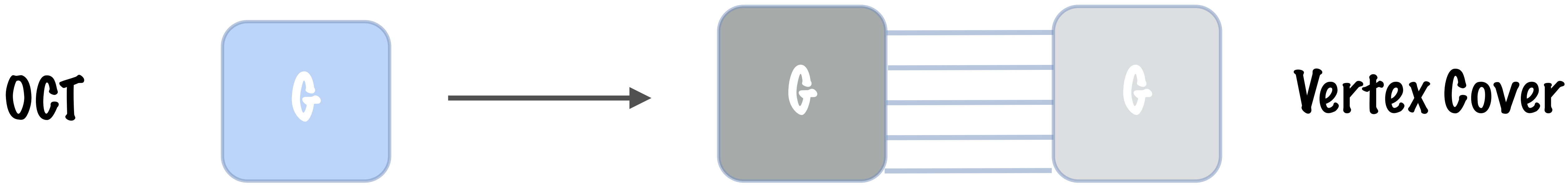
Suppose H has a VC of size |V(G)|+k



G has a bipartite graph of size |V(G)|-k

G has an OCT of size k

OCT Reduces to VC Above LP



G has an OCT of size k iff H has a VC of size |V(G)|+k

- To determine if G on n vertices has an OCT of size k,
  - Construct H from G
  - Matching M of size n in H
  - Determine if H has a VC of size n+k
    - Use the Vertex Cover Above LP algorithm on H that has matching M
      - ($4^{n+k-lpopt(H)}$ $n^{O(1)}$ time which is $4^{n+k-|M|=k}$ $n^{O(1)}$ time)

Iterative Compression

In search of better algorithm.

Does G have an odd cycle transversal of size at most k?

$O^*(f(k))$ algorithm

reduces to

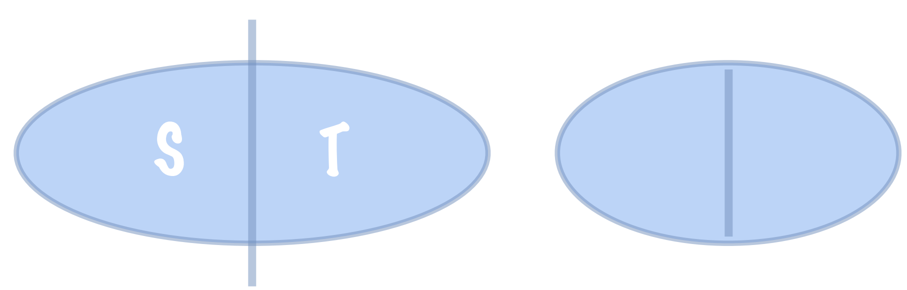
in 2^{k+1} time

Given an odd cycle transversal T of G, find a smaller disjoint odd cycle transversal

$O^*(f(|T|))$ algorithm

- Assume by induction that G has an OCT of size k+1
  - Base case: any subgraph on k+3 vertices has an OCT of size k+1
- Guess its intersection S with a smaller solution (2^{k+1} choices)

k+1 OCT

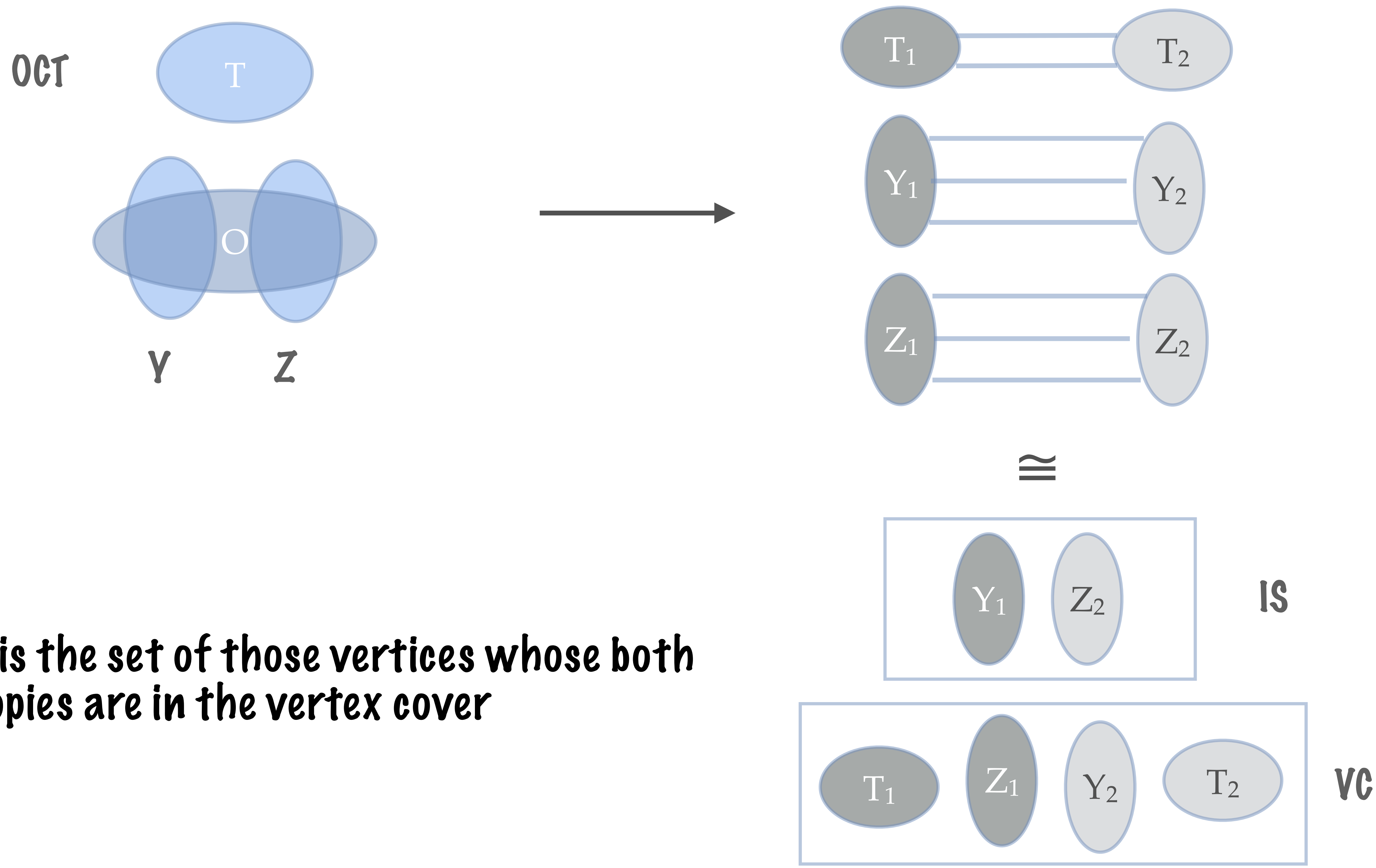


bipartite

Solve disjoint compression step: given an OCT, find a smaller disjoint OCT

Disjoint Compression for OCT

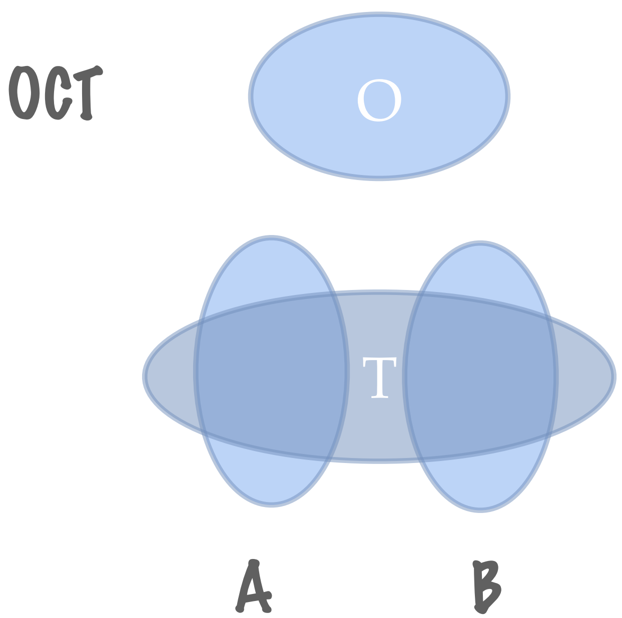
Given an OCT T, find a smaller disjoint OCT O

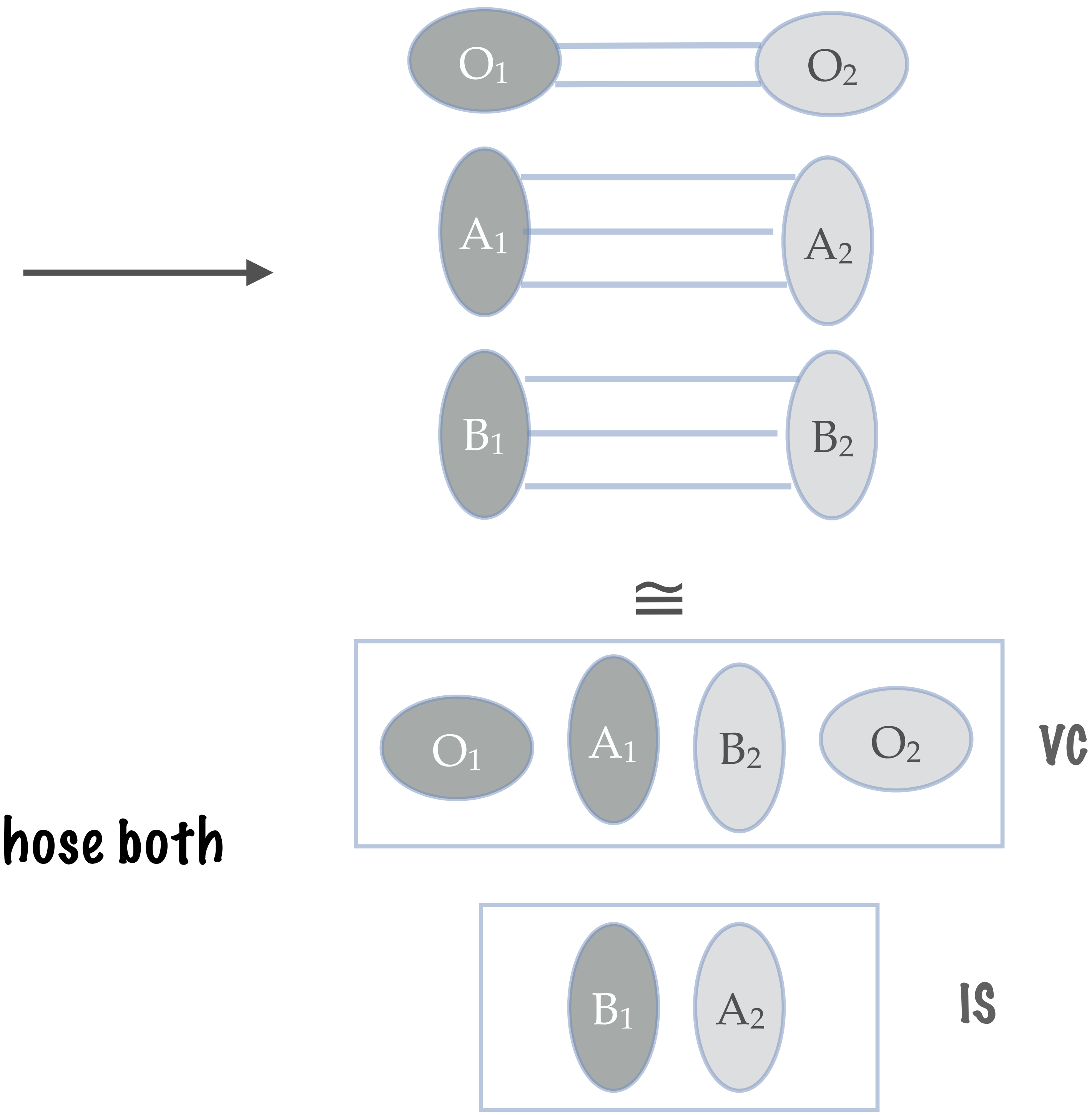


T is the set of those vertices whose both copies are in the vertex cover

Disjoint Compression for OCT

Given an OCT T, find a smaller disjoint OCT O

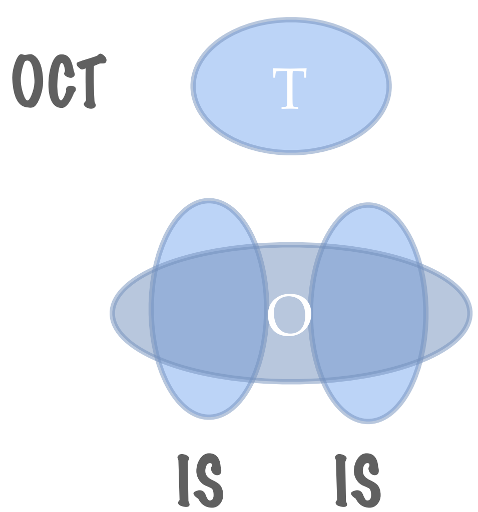


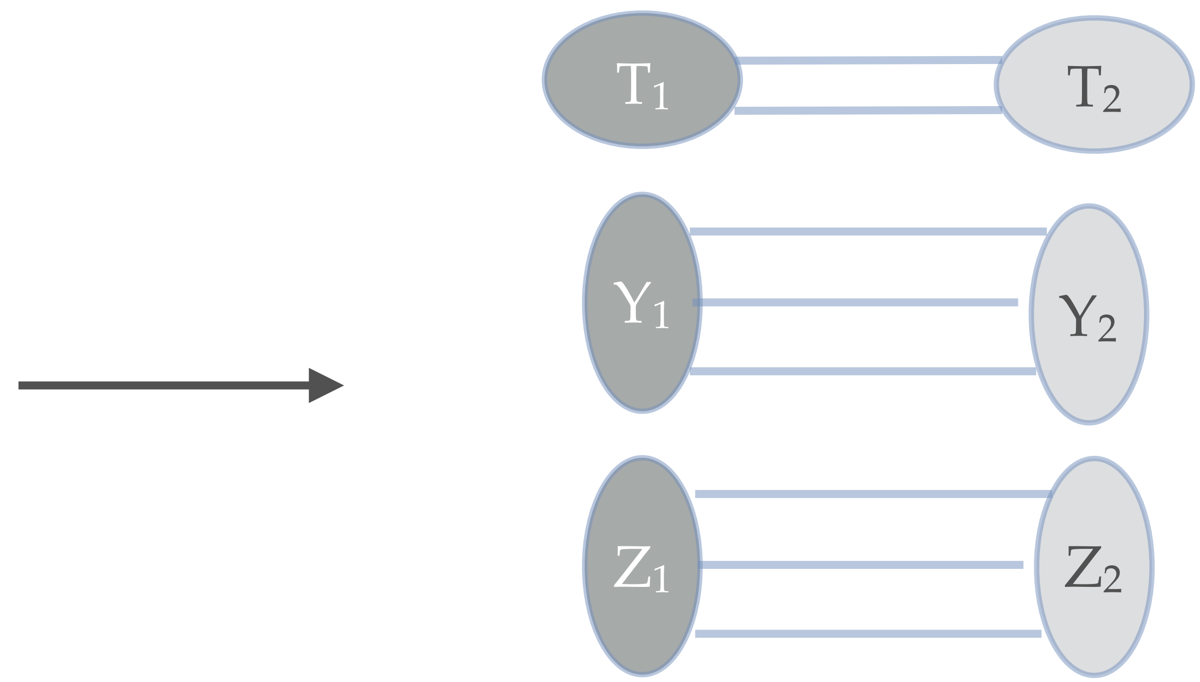


O is the set of those vertices whose both copies are in the vertex cover

Disjoint Compression for OCT

Given an OCT T, find a smaller disjoint OCT O

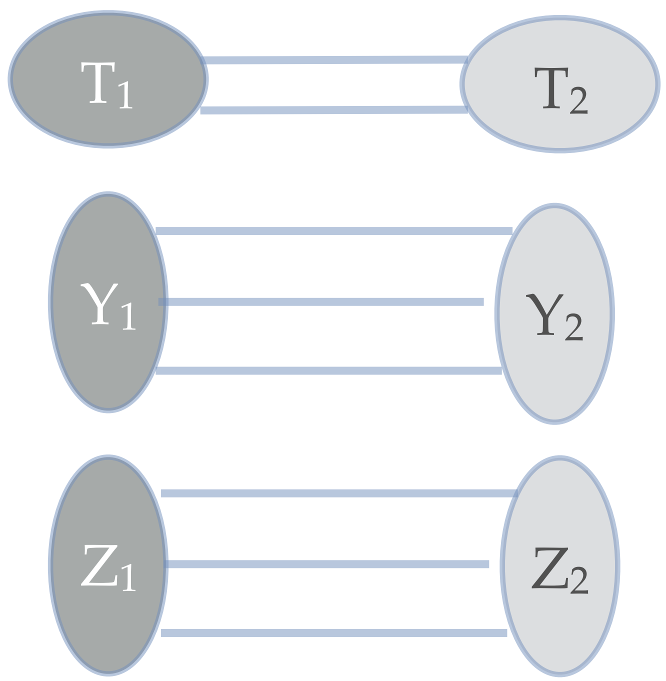


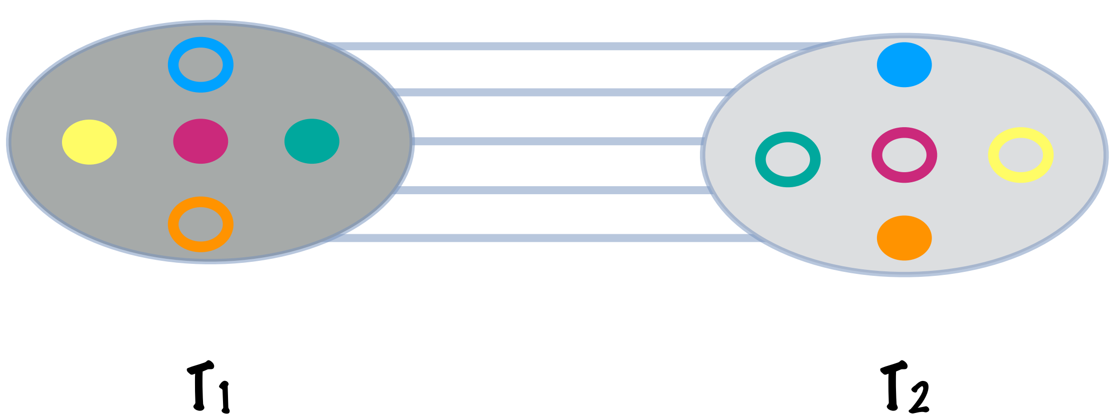


Find a min vertex cover that covers the edges across T_1 and T_2 exactly once

Disjoint Compression for OCT

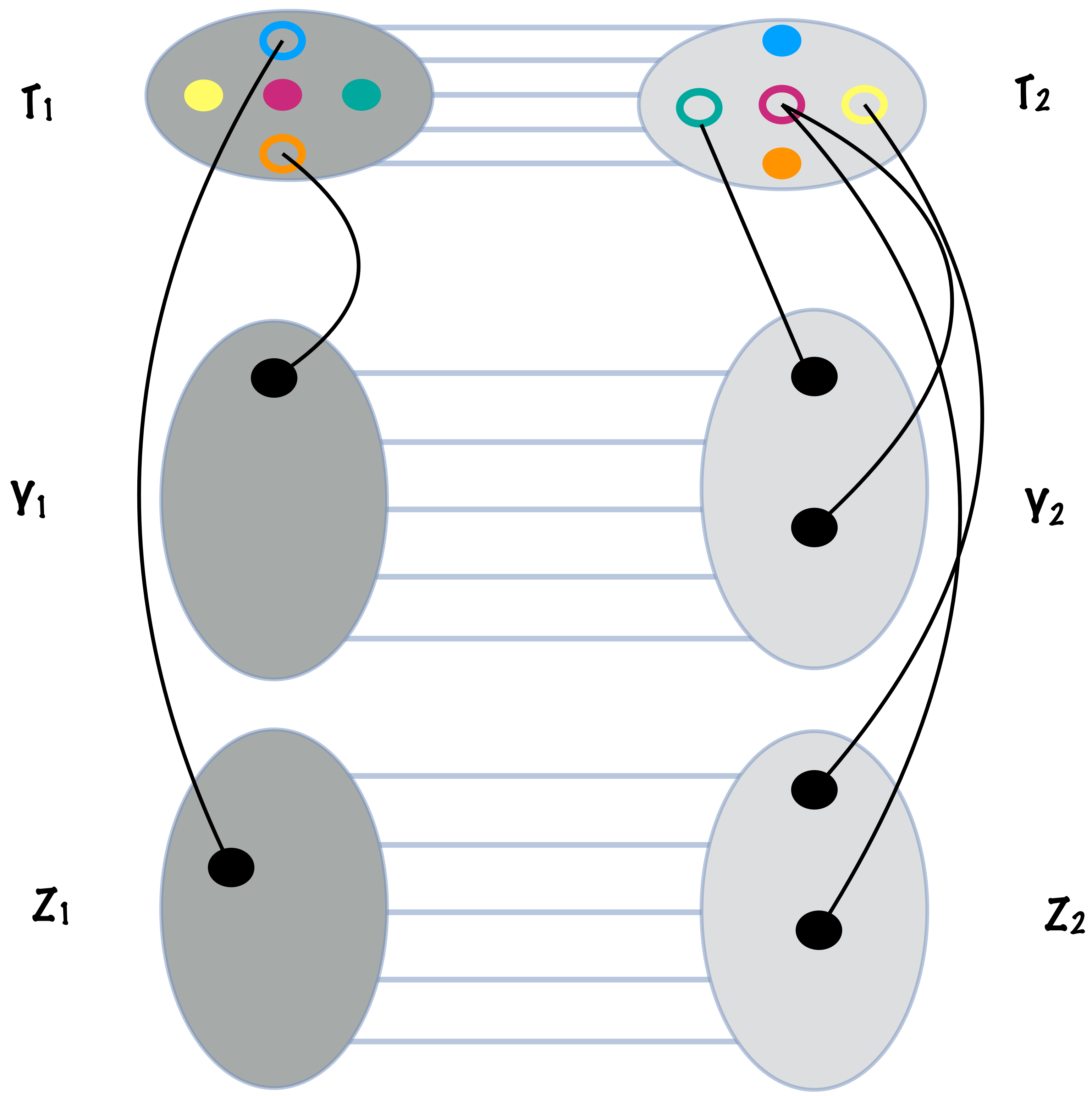
Find a min vertex cover that covers the edges across T_1 and T_2 exactly once





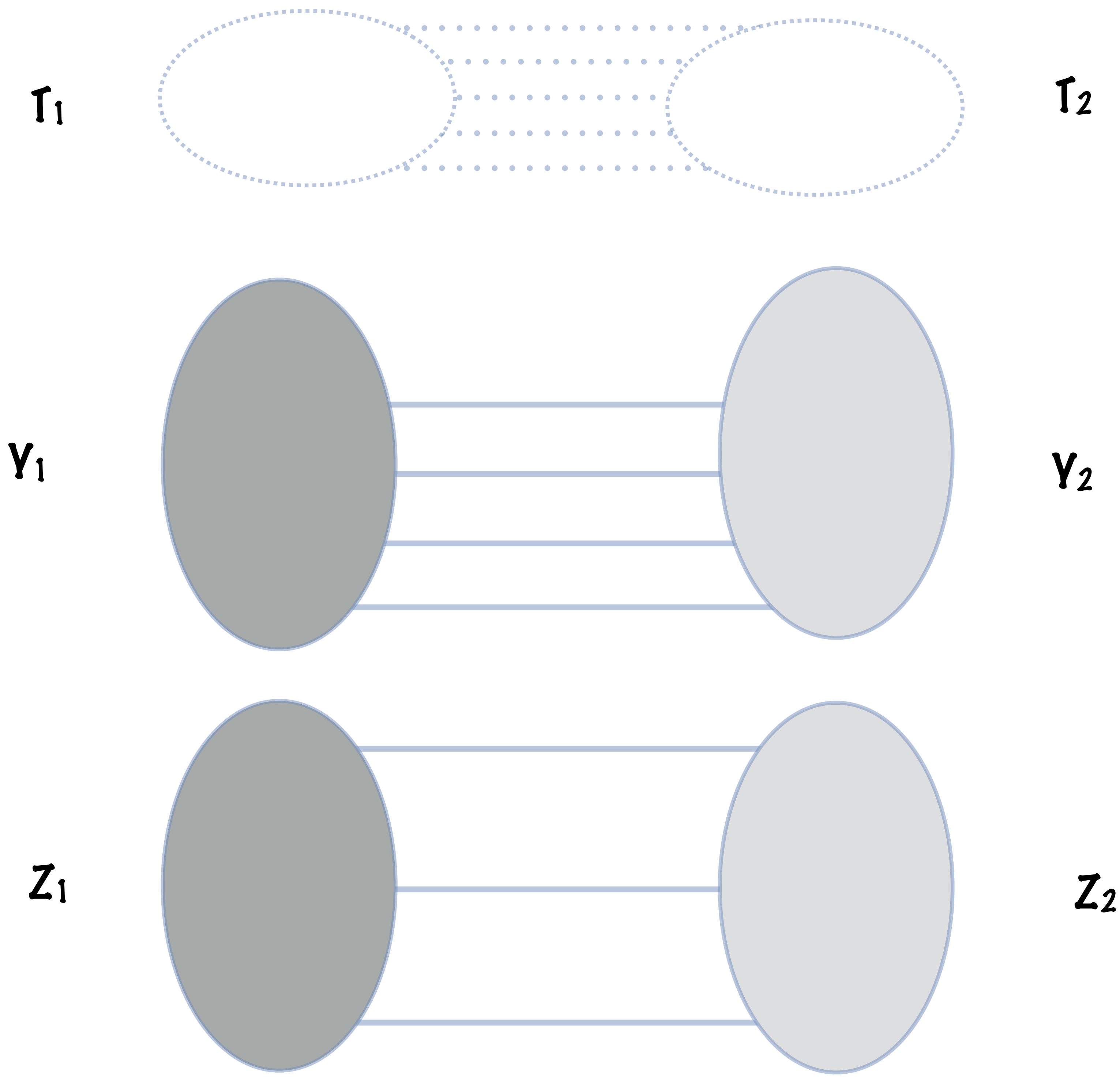
$2^{|T|}$ choices

Disjoint Compression for OCT



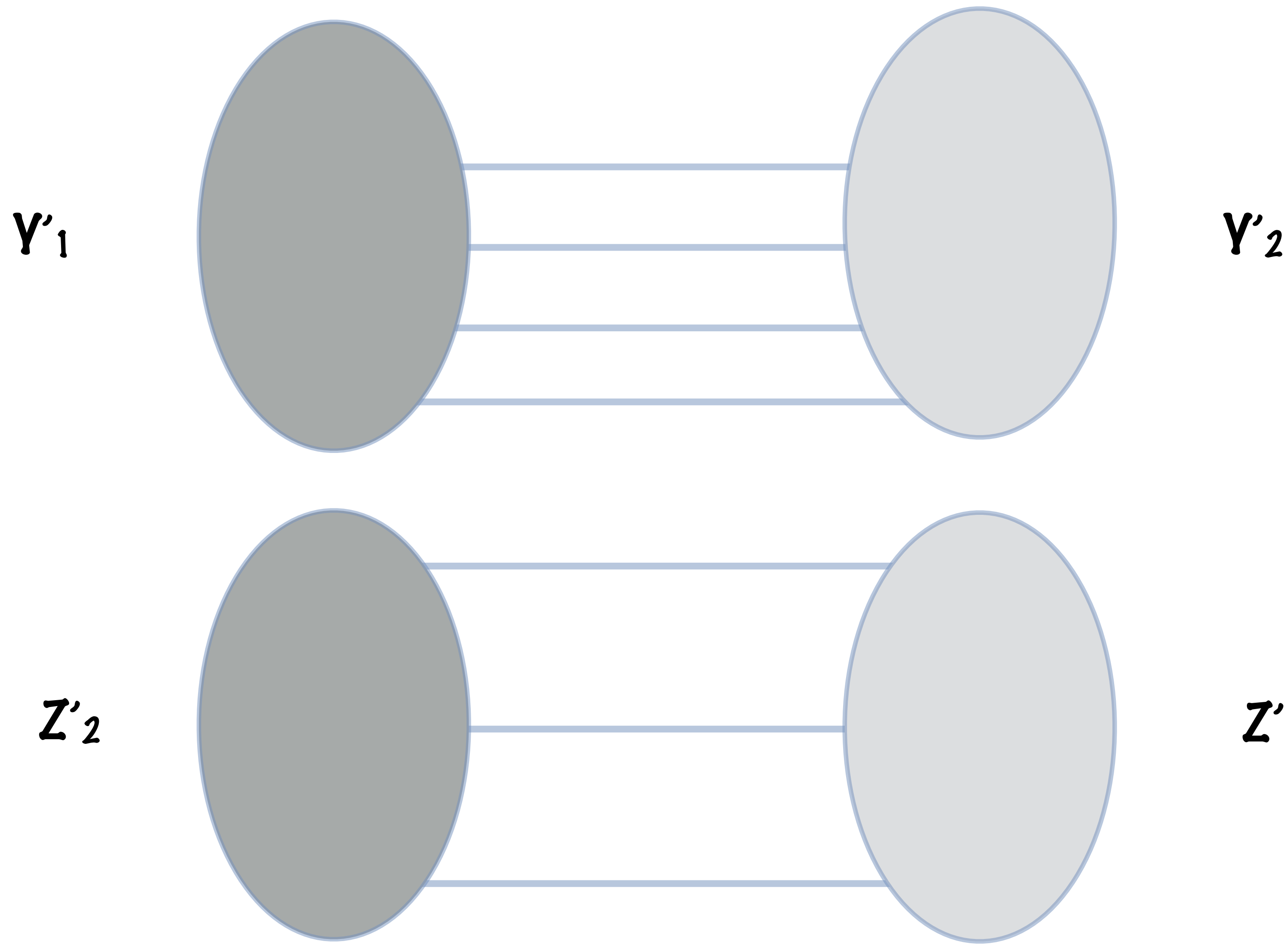
Disjoint Compression for OCT

Find a min vertex cover



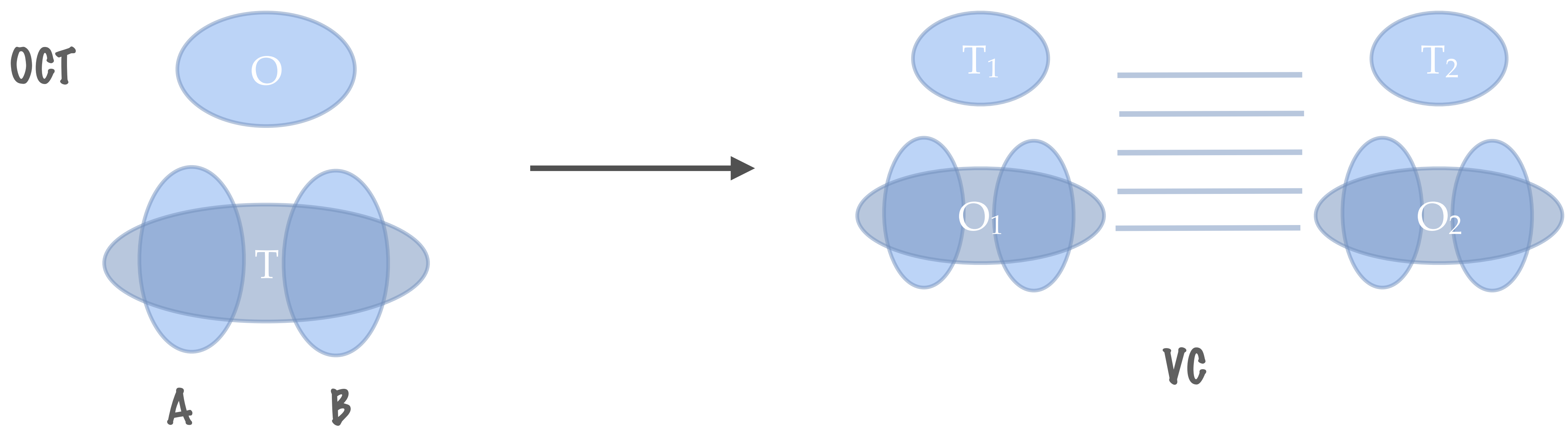
Disjoint Compression for OCT

Find a min vertex cover in a bipartite graph

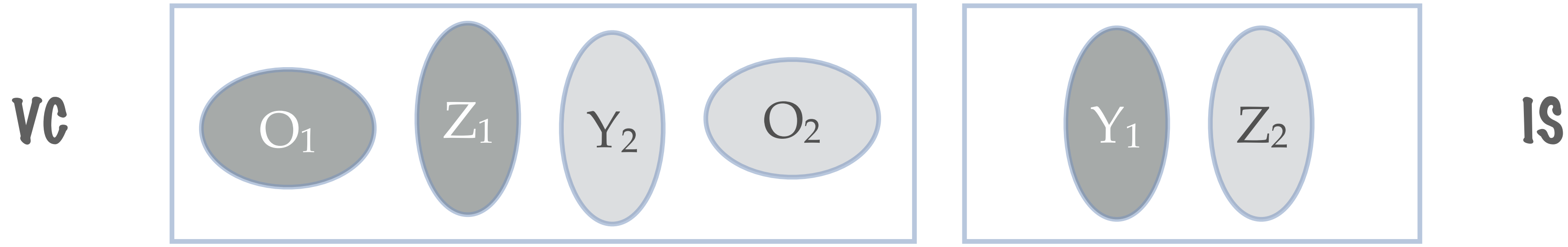


Polynomial time

Disjoint Compression for OCT



Find a min vertex cover that covers the edges across T_1 and T_2 exactly onc



$O^*(3^k)$ algorithm for OCT