Stable Paths Problem

Recall how BGP works

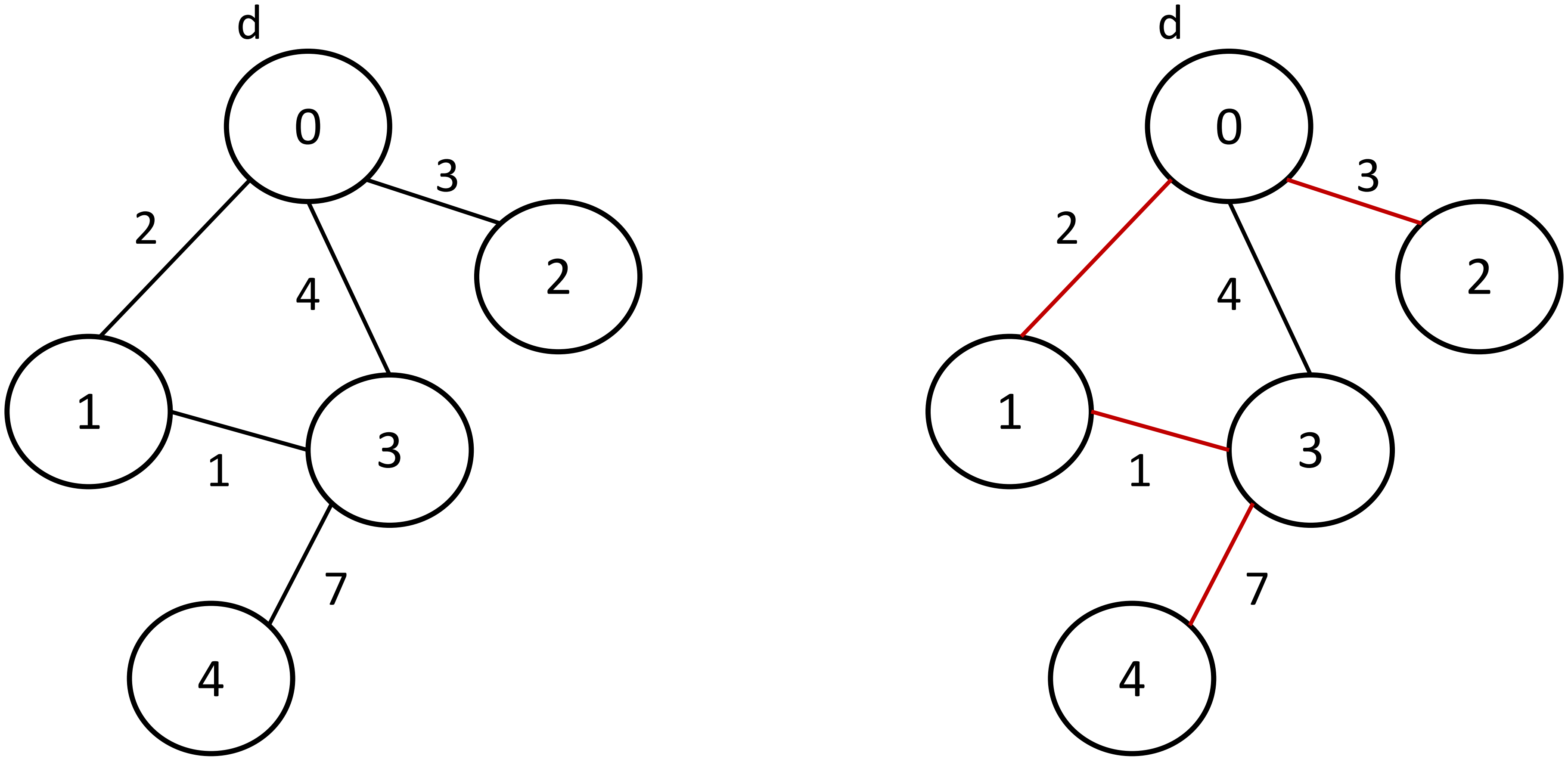
1. Path-vector routing protocol
   - Similar to distance vector, but routes carry the entire path
   - Each hop is an ASN (organization in internet, router in DC)

2. Independent organizations have different policy
   - Not necessarily shortest path
   - Could be least cost, cold-potato, avoid China, etc.

Policy in BGP

1. Export policy
   - Determines what to announce to whom
2. Import policy
   - Determine how to modify (or drop) incoming announcements
   - Can set the BGP local-preference field (highest priority in decision process)
3. Decision process
   - Determine “best” path among all those available

Distance Vector Routing (Shortest Paths)



Solution forms a shortest path tree

What problem is BGP solving?

RIP, OSPF, … → Shortest Paths Problem

BGP → ???

BGP created in 1989 to solve a pressing need.

Protocol dynamics not well understood at the time.

What problem is BGP solving?

RIP, OSPF, … → Shortest Paths Problem

BGP → Stable Paths Problem

“The Stable Paths Problem and Interdomain Routing”

Timothy G. Griffin and F. Bruce Shepherd and Gordon Wilfong (2002)

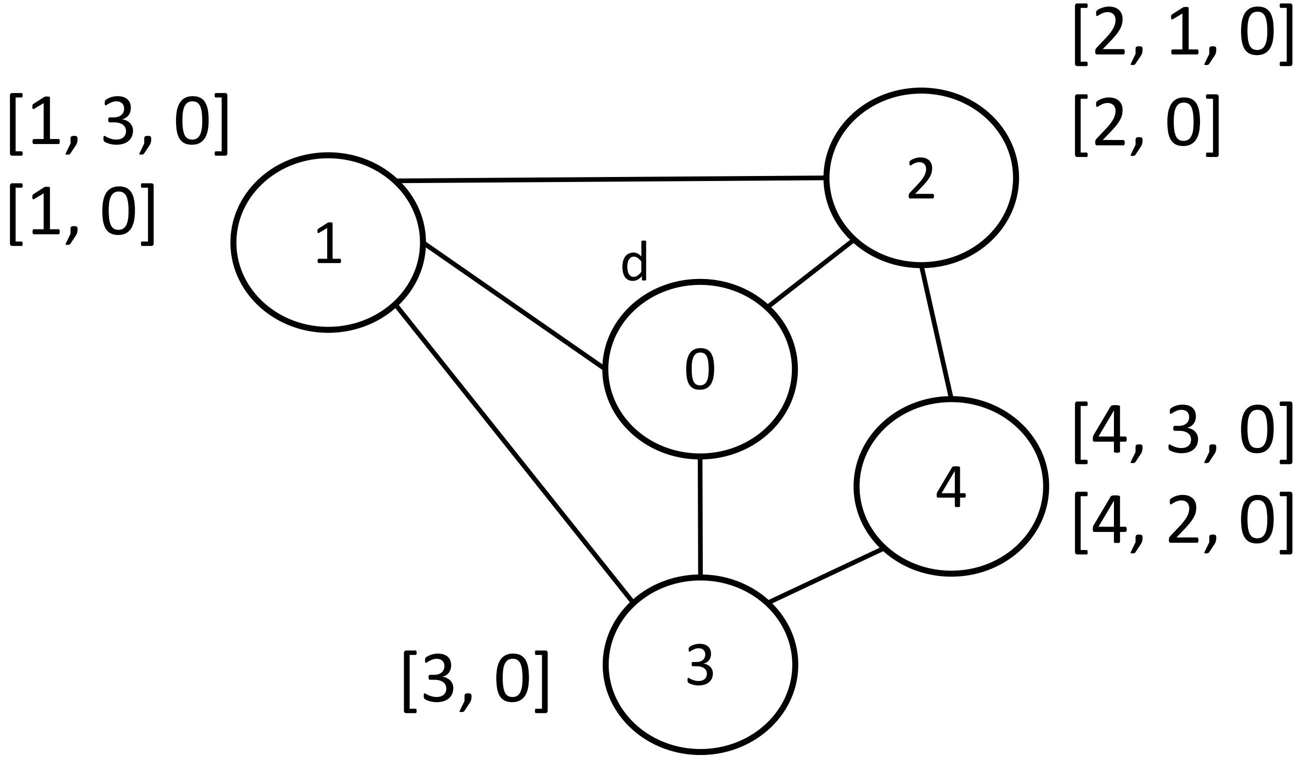
Insights from the paper

- BGP can (always, sometimes, never) converge.
- BGP will solve for a “stable tree”
  - A local equilibrium rather than a global optimum.
  - Stable tree: each node has no preference to change from its current path.
  - Related to Nash Equilibrium from game theory.
- Relationship as follows:
  - If no stable tree, then BGP will never converge.
  - If a unique stable tree, then BGP will converge optimally.
  - If multiple stable trees, then BGP may converge to any tree or may diverge.

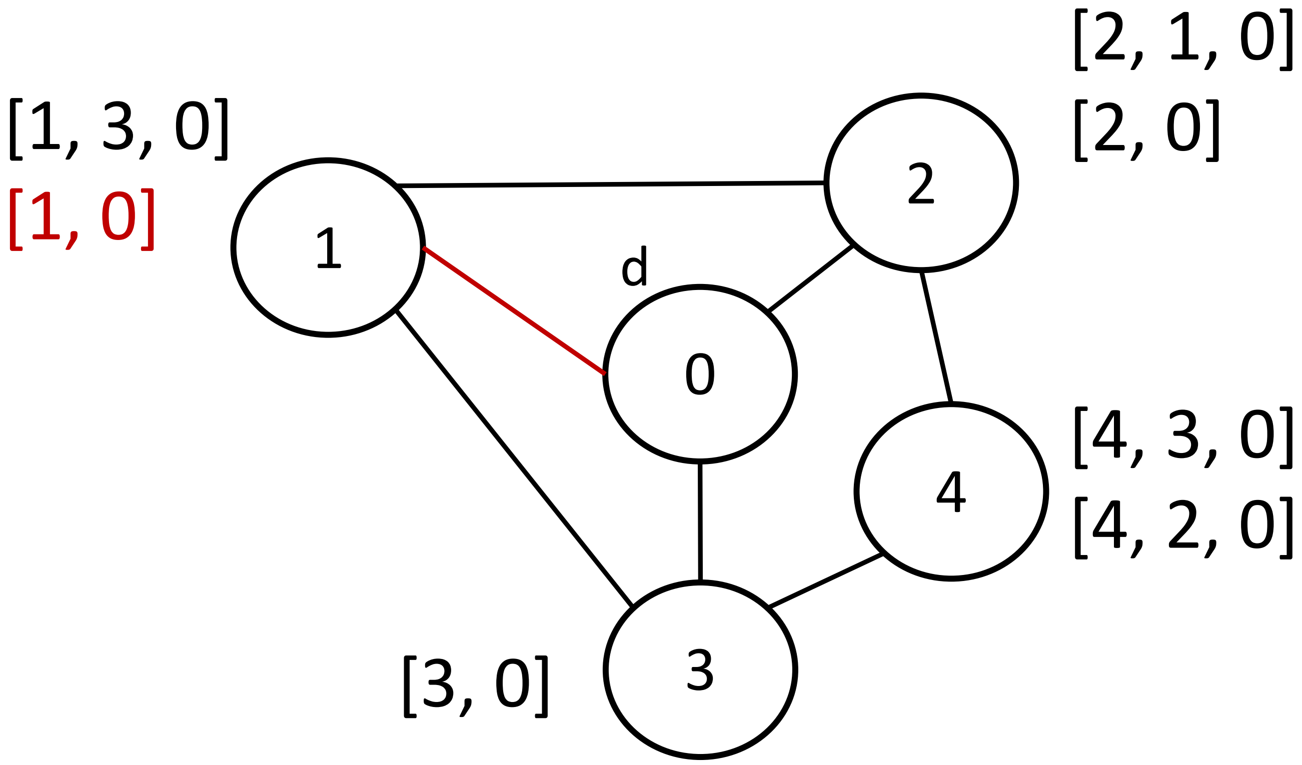
Simple Model of BGP

- Network topology is an undirected graph
- Each node list paths that aren’t filtered
    - List of paths is sorted by preference at that node

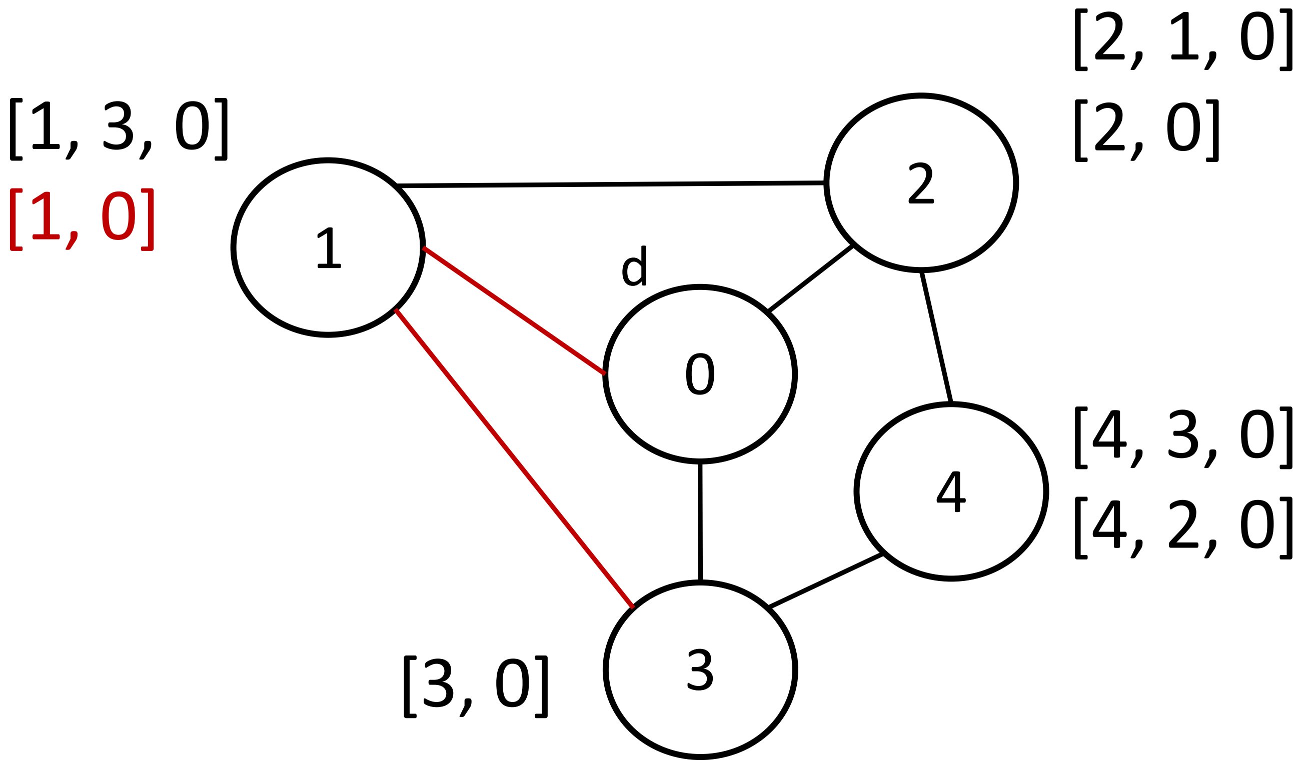
BGP will always converge (“good gadget”)



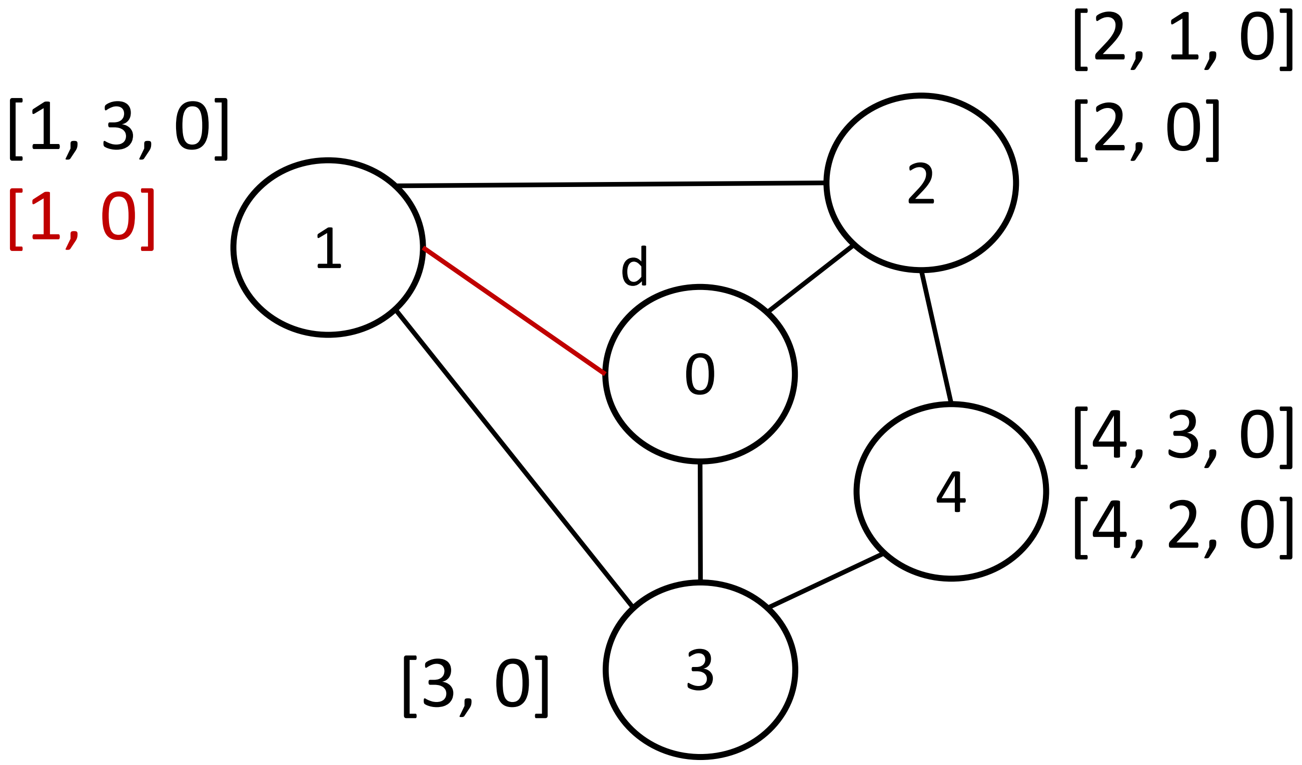
BGP will always converge (“good gadget”)



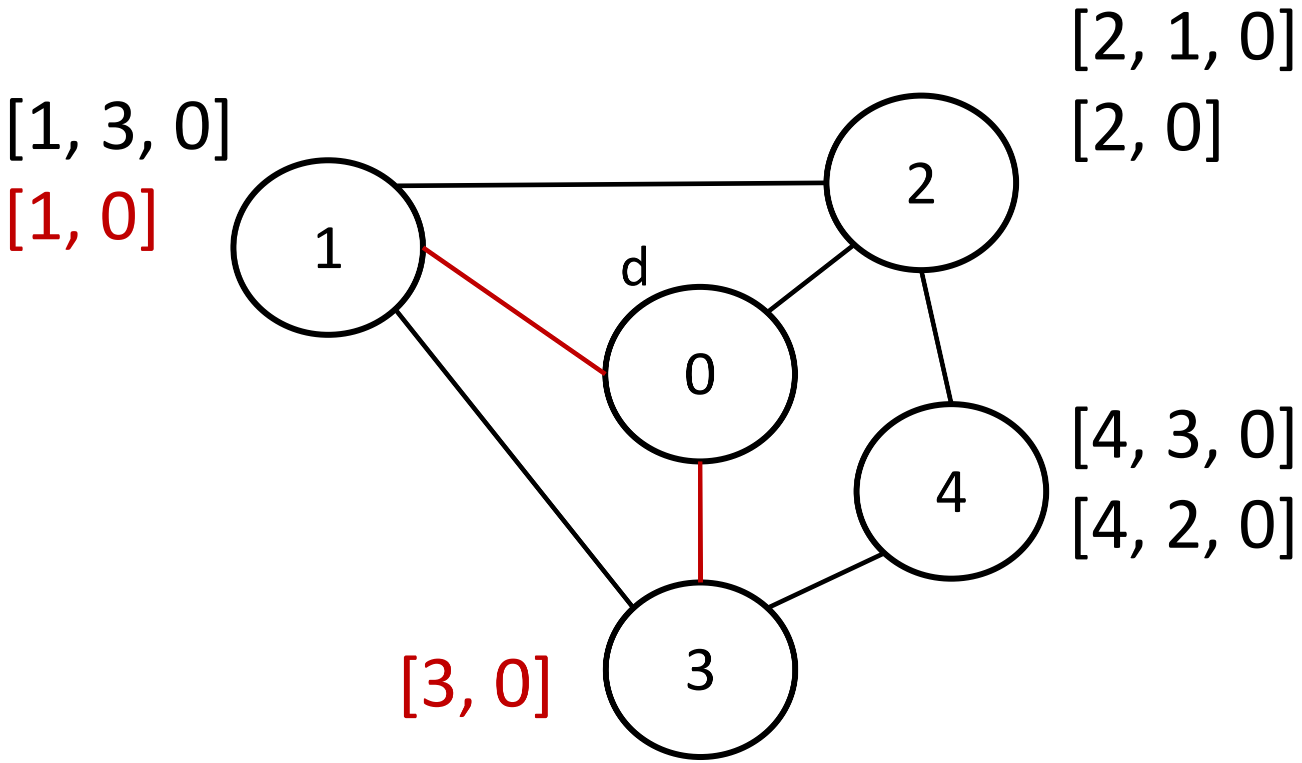
BGP will always converge (“good gadget”)



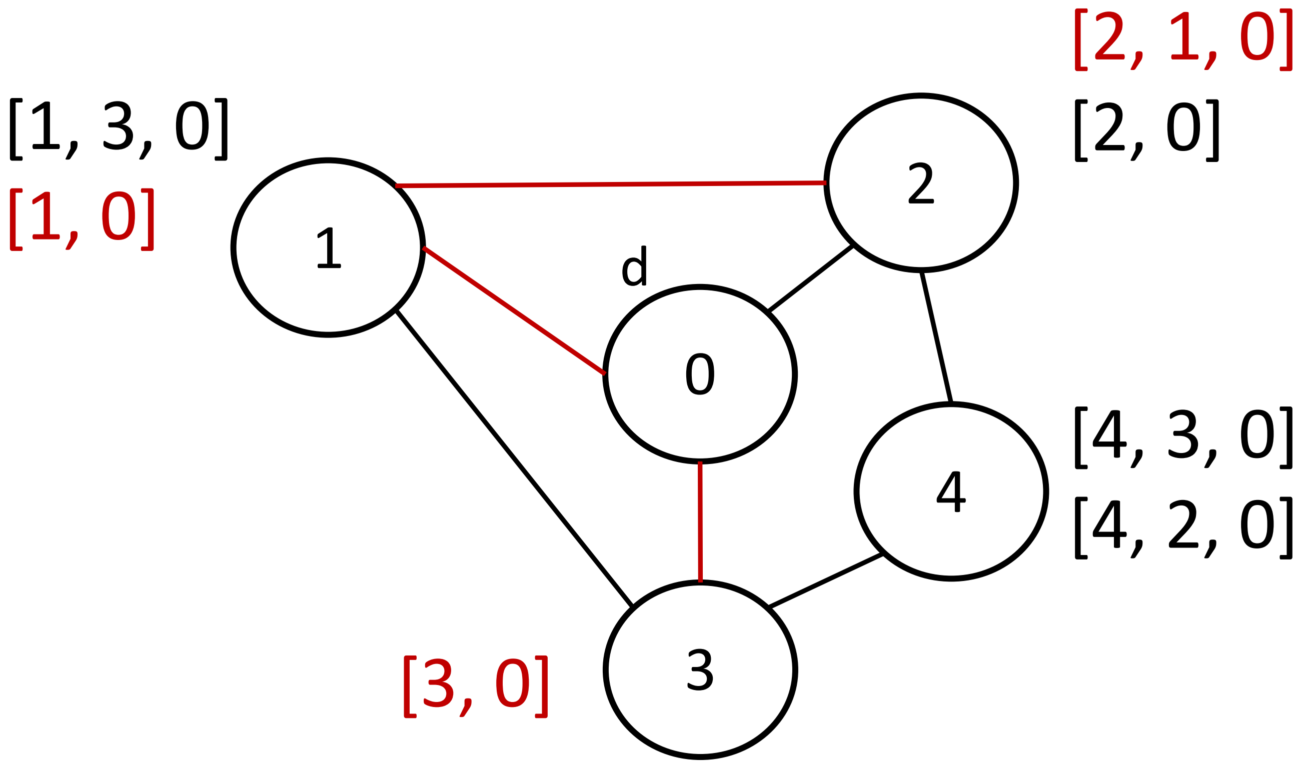
BGP will always converge ("good gadget")



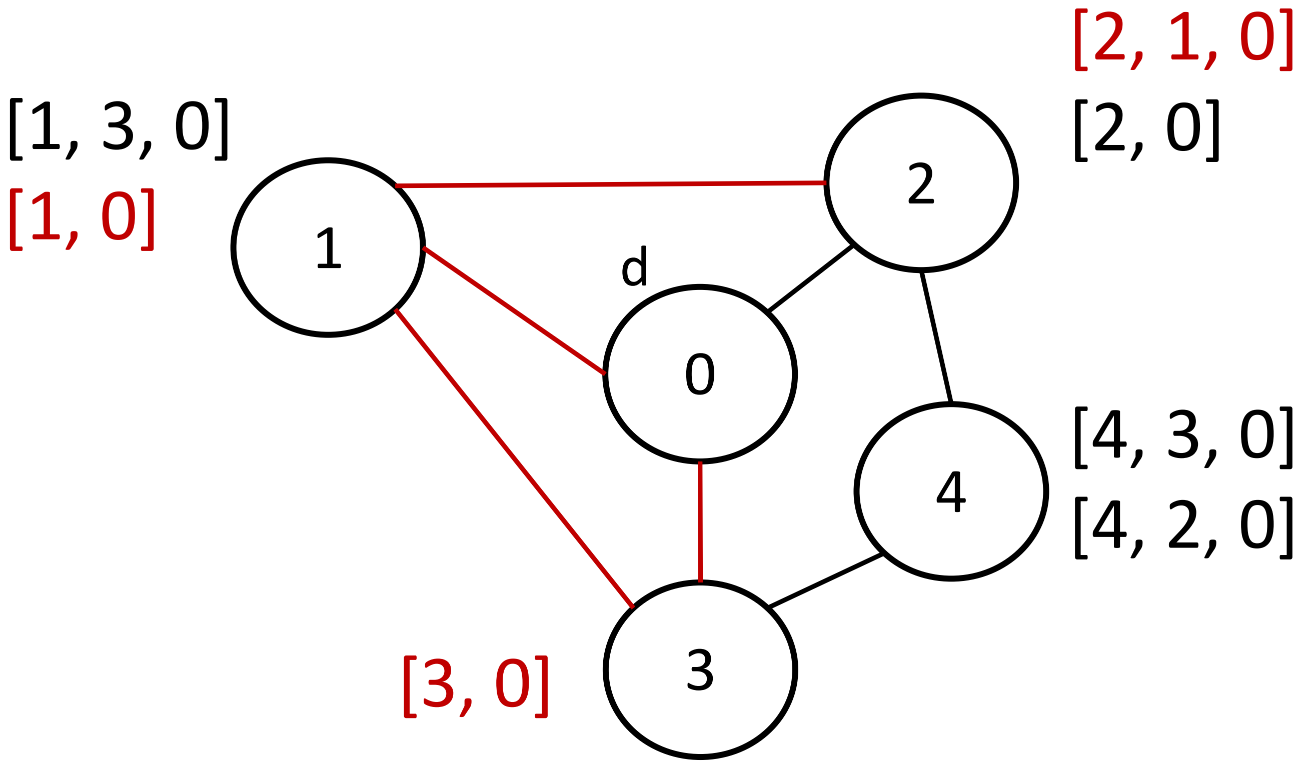
BGP will always converge (“good gadget”)



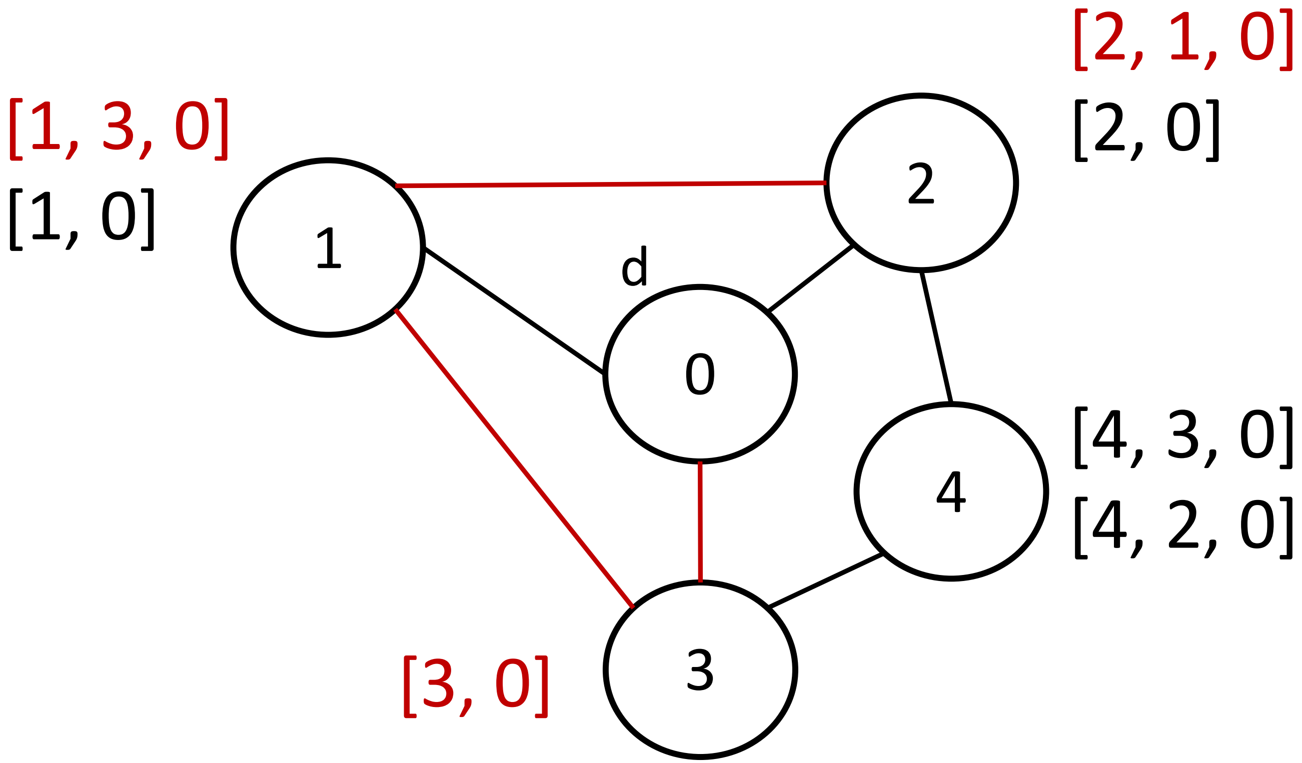
BGP will always converge (“good gadget”)



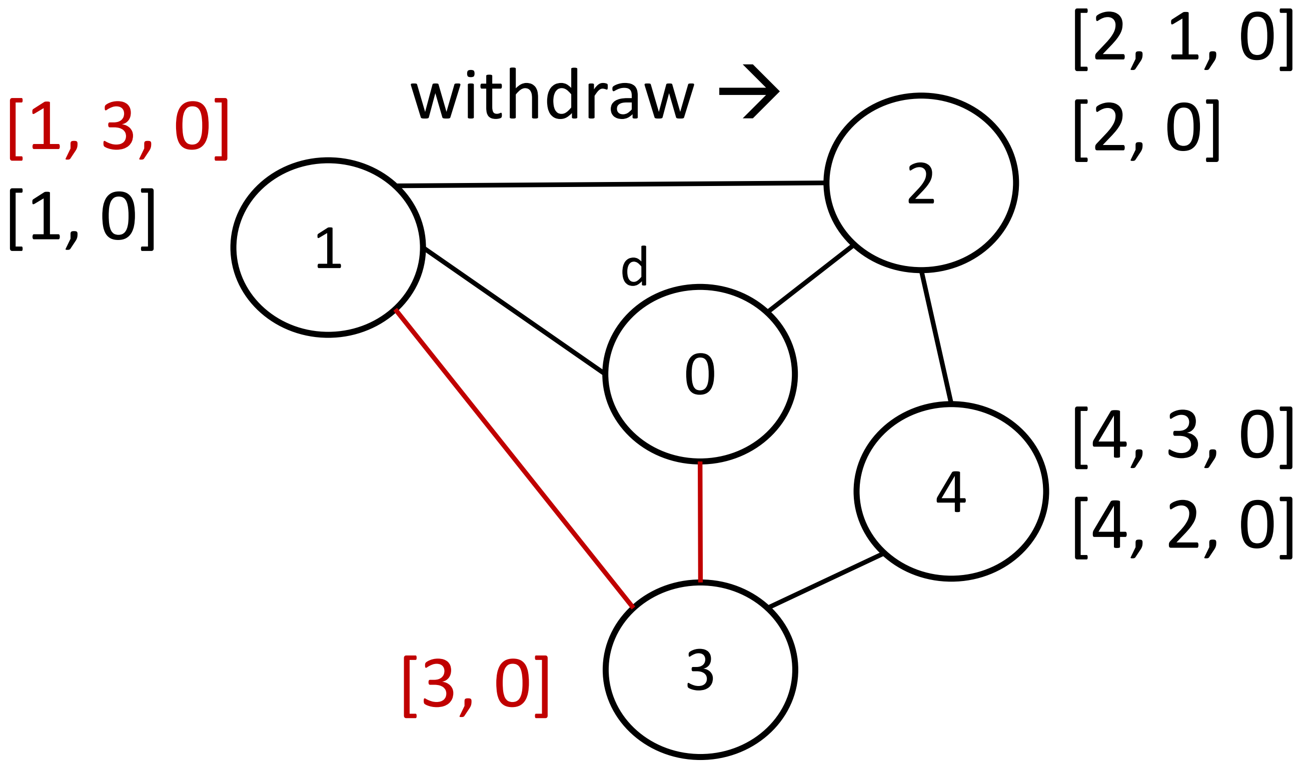
BGP will always converge (“good gadget”)



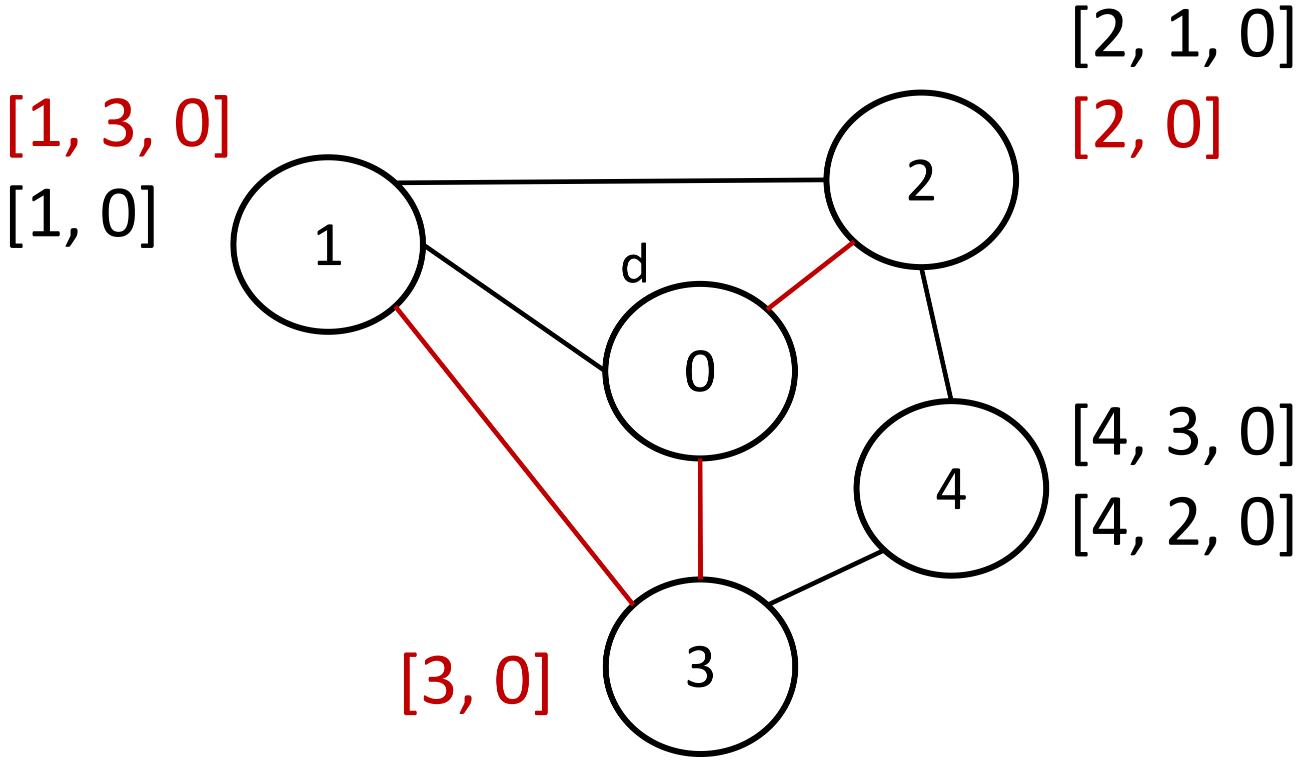
BGP will always converge (“good gadget”)



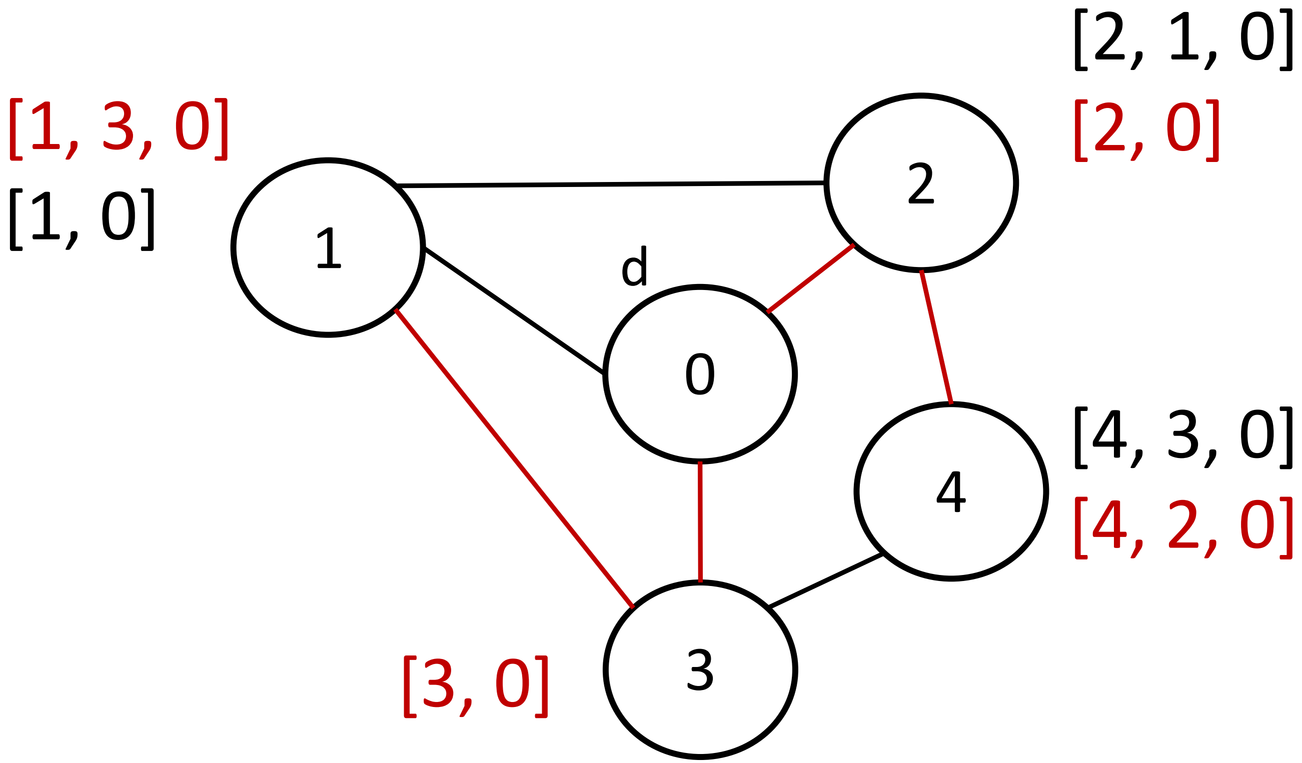
BGP will always converge ("good gadget")



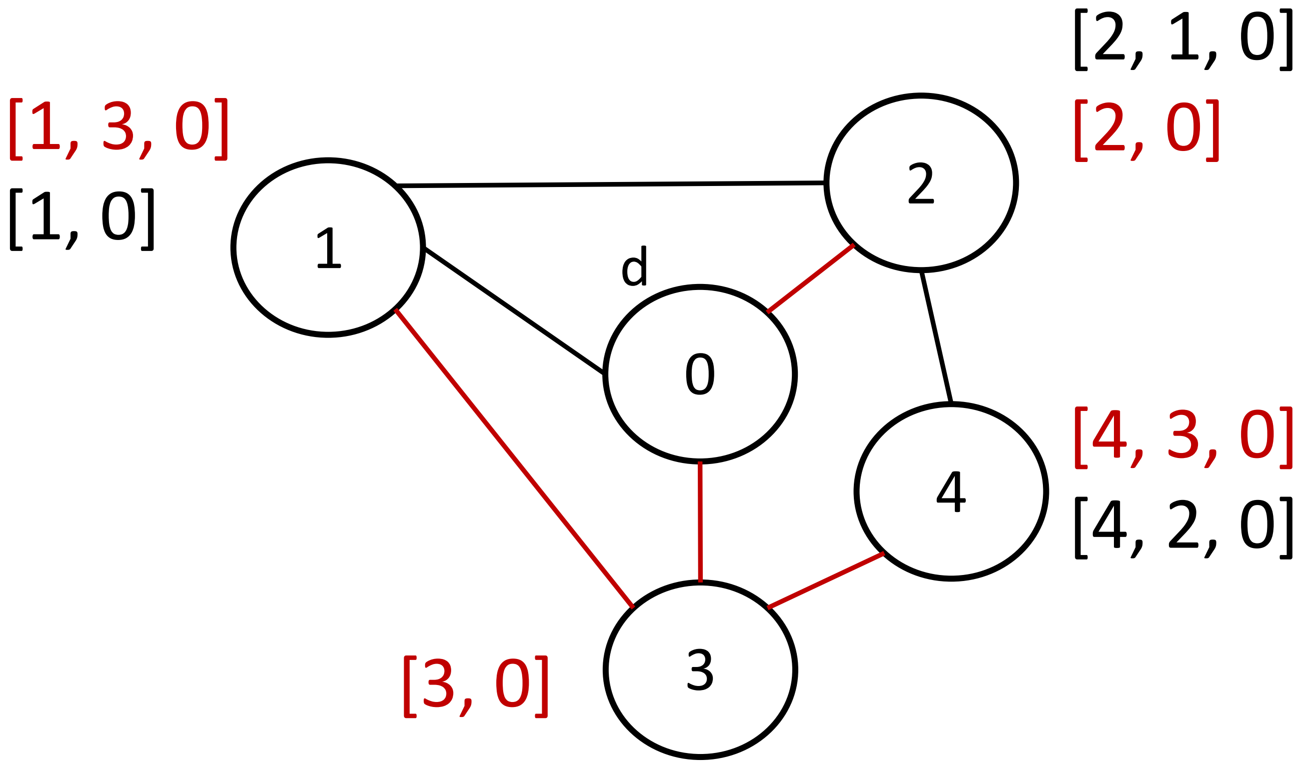
BGP will always converge (“good gadget”)



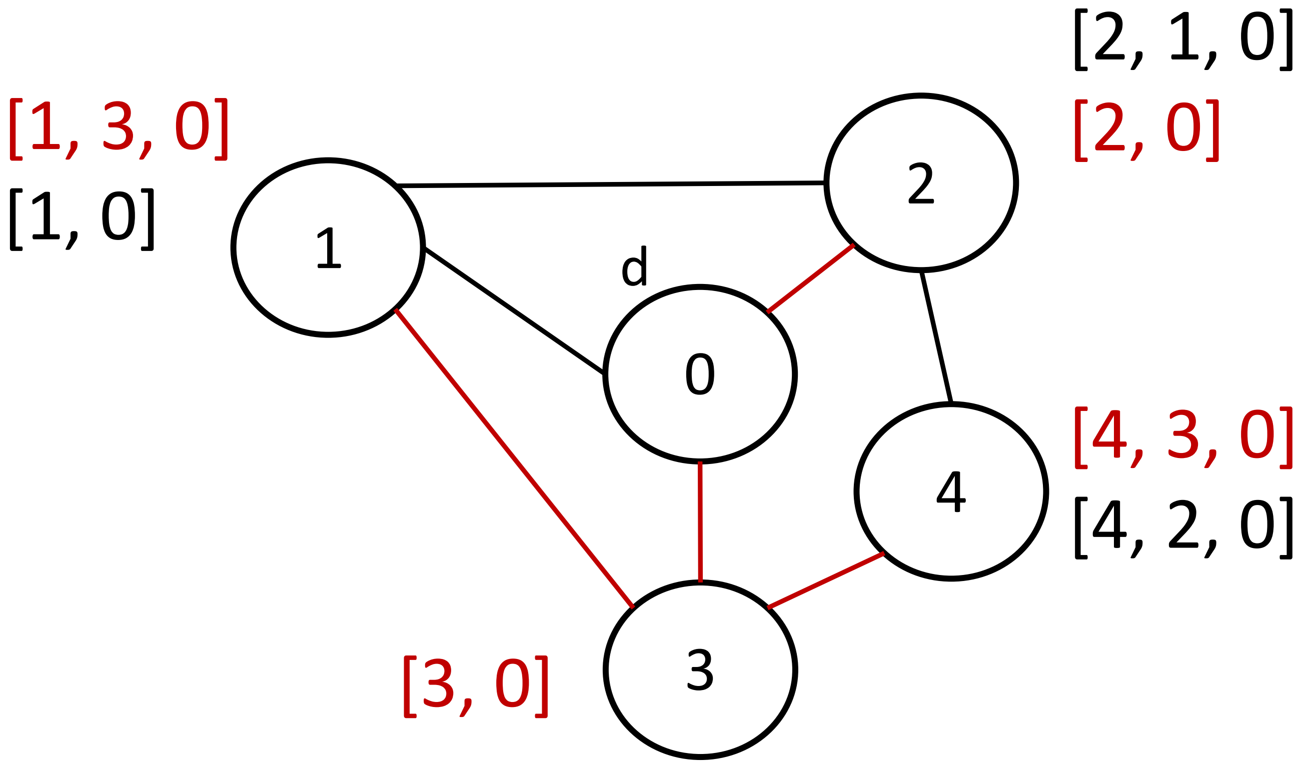
BGP will always converge ("good gadget")



BGP will always converge ("good gadget")

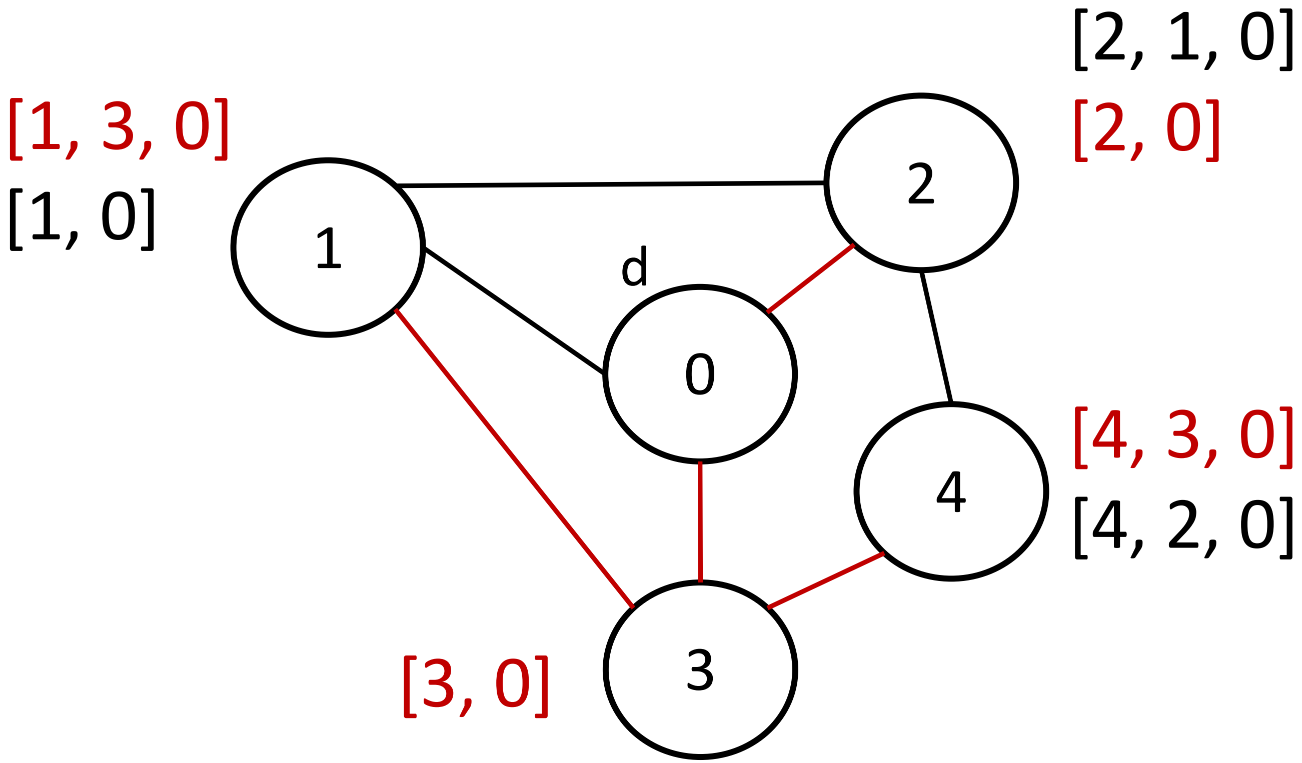


BGP will always converge ("good gadget")



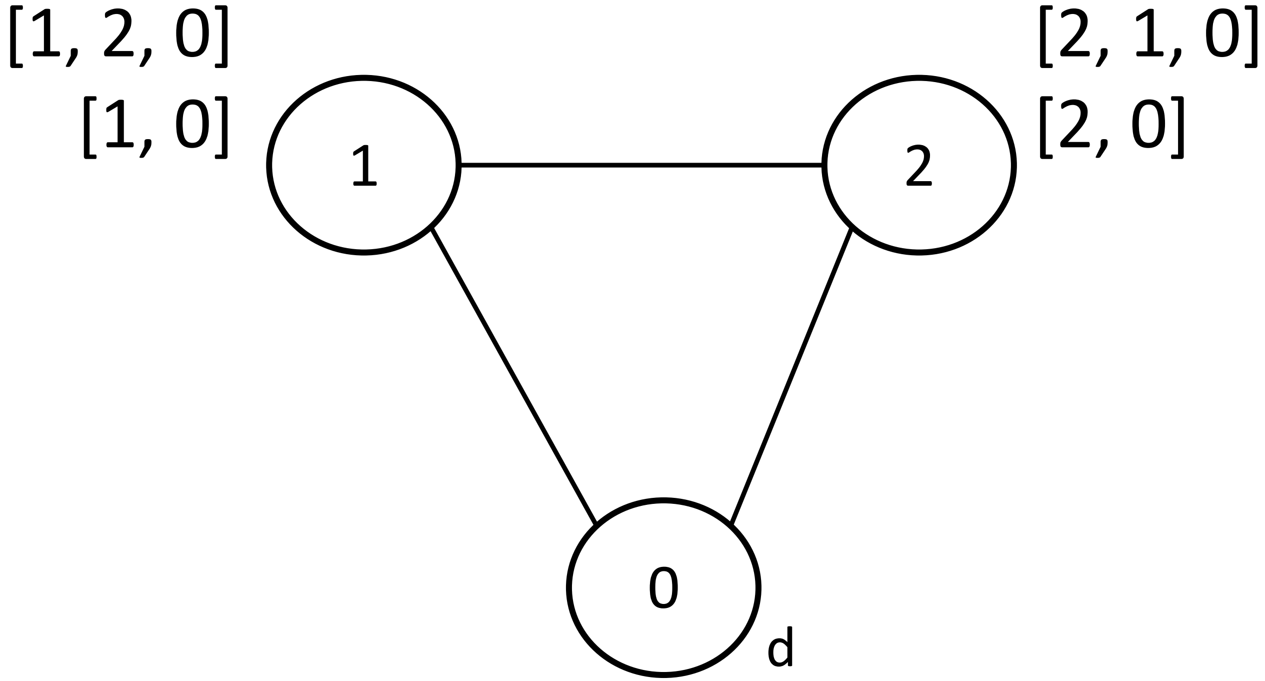
stable solution: each node is locally stable (no better option)

BGP will always converge (“good gadget”)

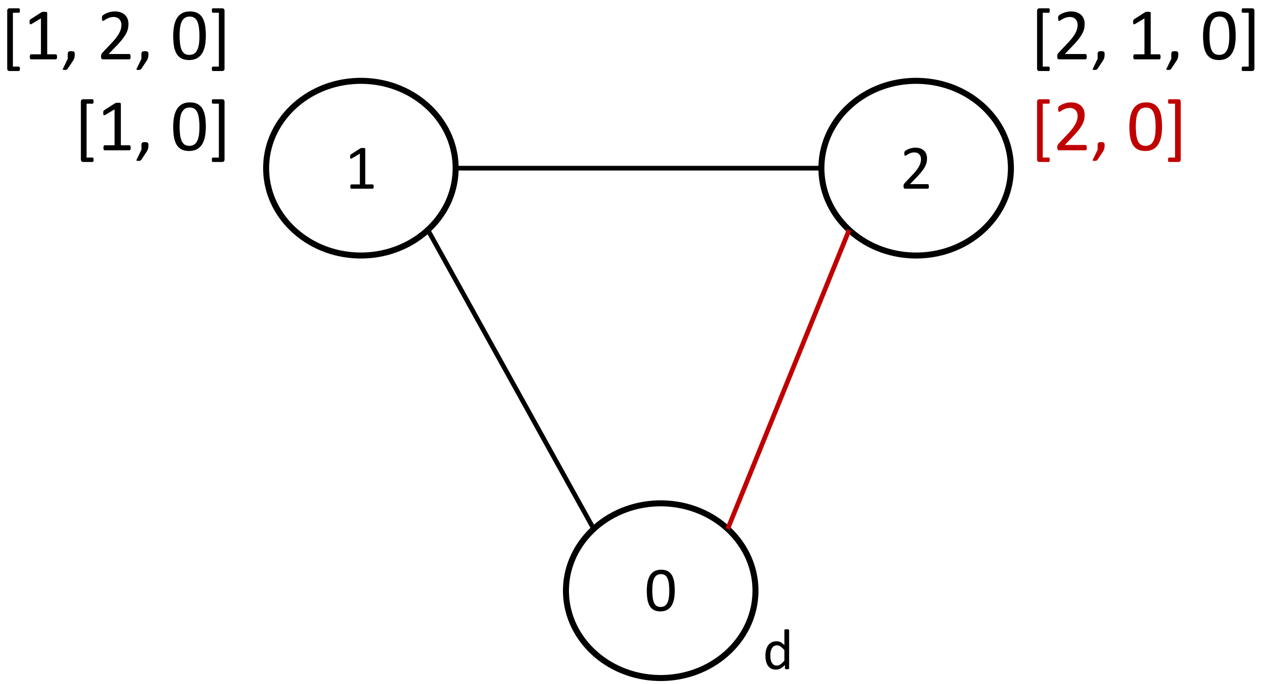


solution is *unique* (any asynchronous order would lead here)

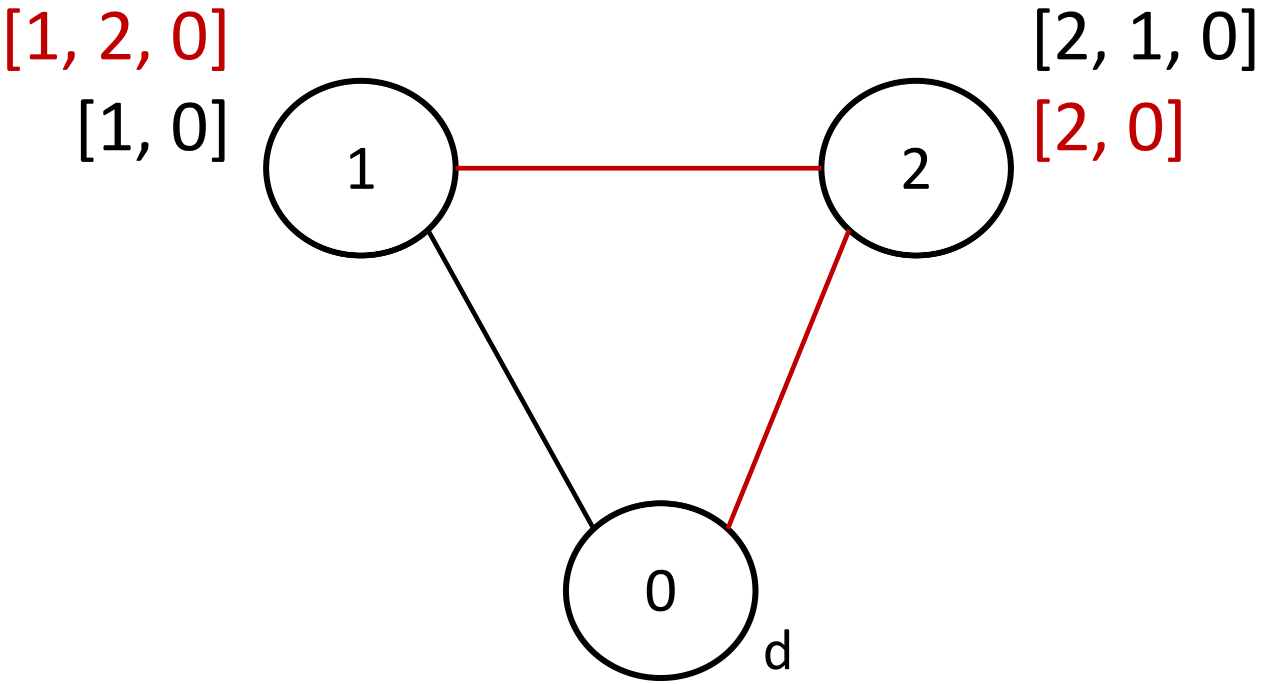
BGP *may* converge (“diagree”)



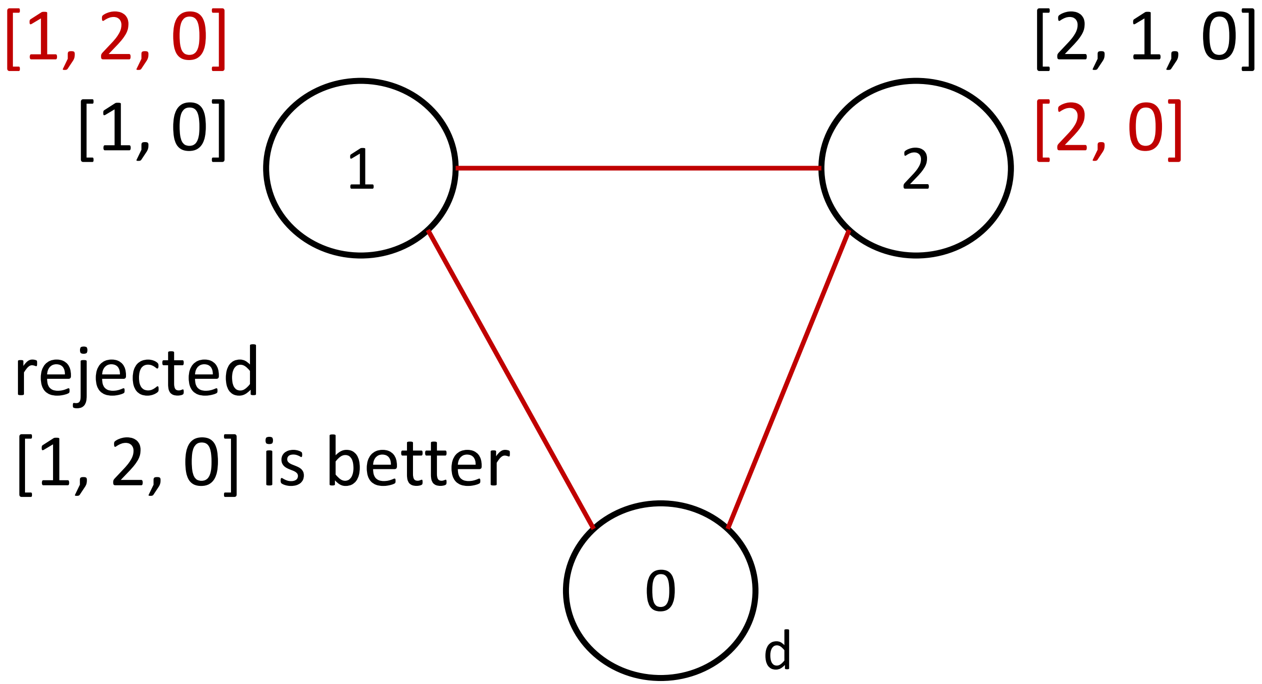
BGP *may* converge (“diagree”)



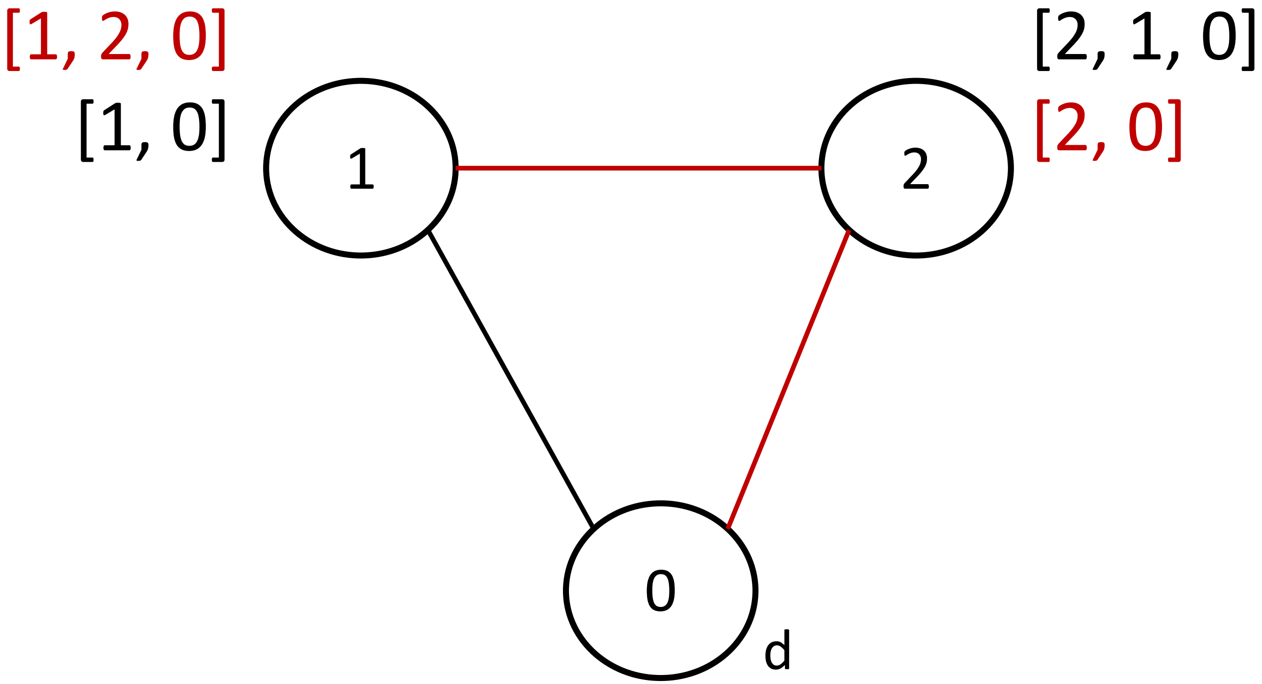
BGP *may* converge (“diagree”)



BGP *may* converge (“diagree”)

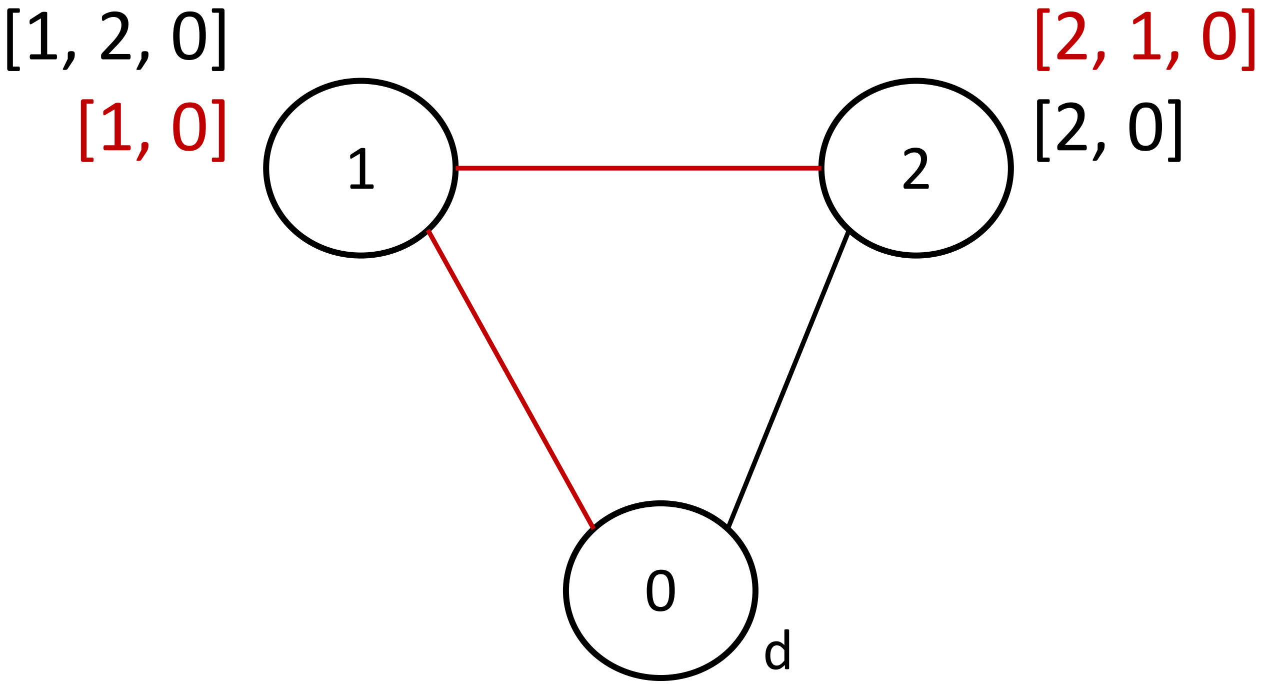


BGP *may* converge (“diagree”)



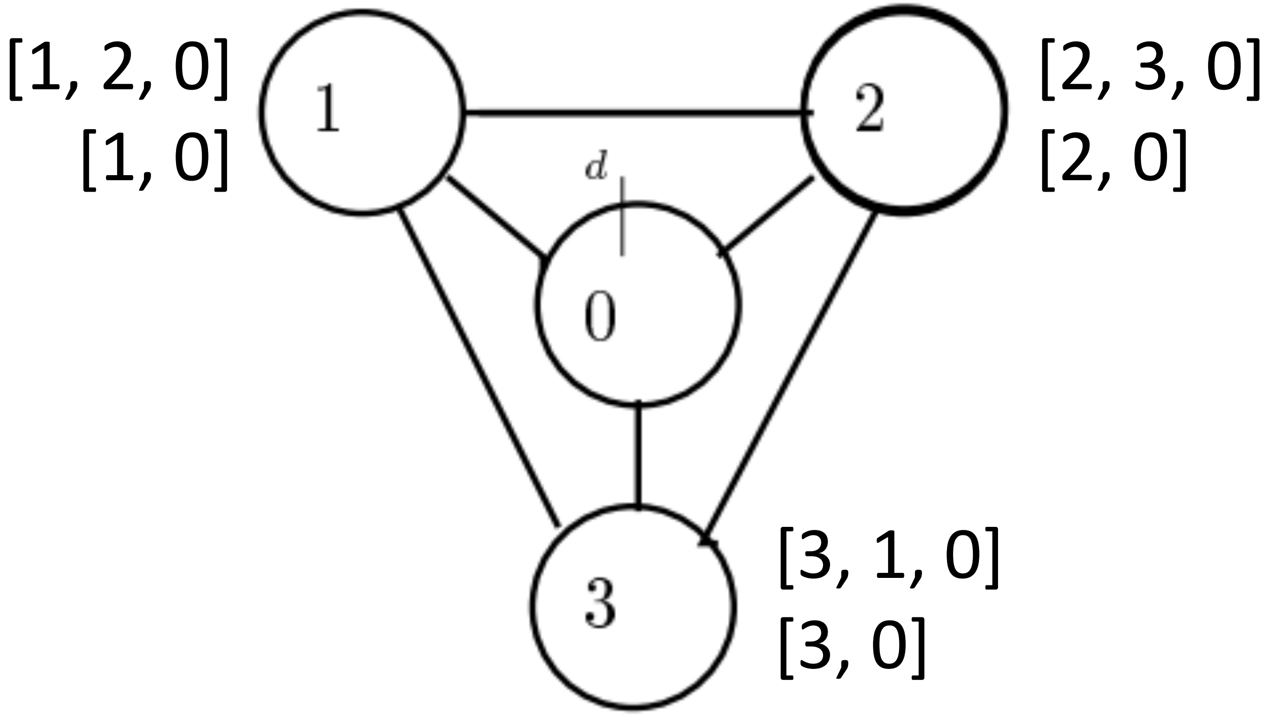
stable solution: each node is locally stable (2 never hears [2, 1, 0])

BGP *may* converge (“diagree”)



Another symmetric solution

BGP *never* converges (“bad gadget”)



[1, 2, 0]
[1, 0]

[2, 3, 0]
[2, 0]

[3, 1, 0]
[3, 0]

No stable tree for the bad gadget exists

Determining convergence is NP-Complete

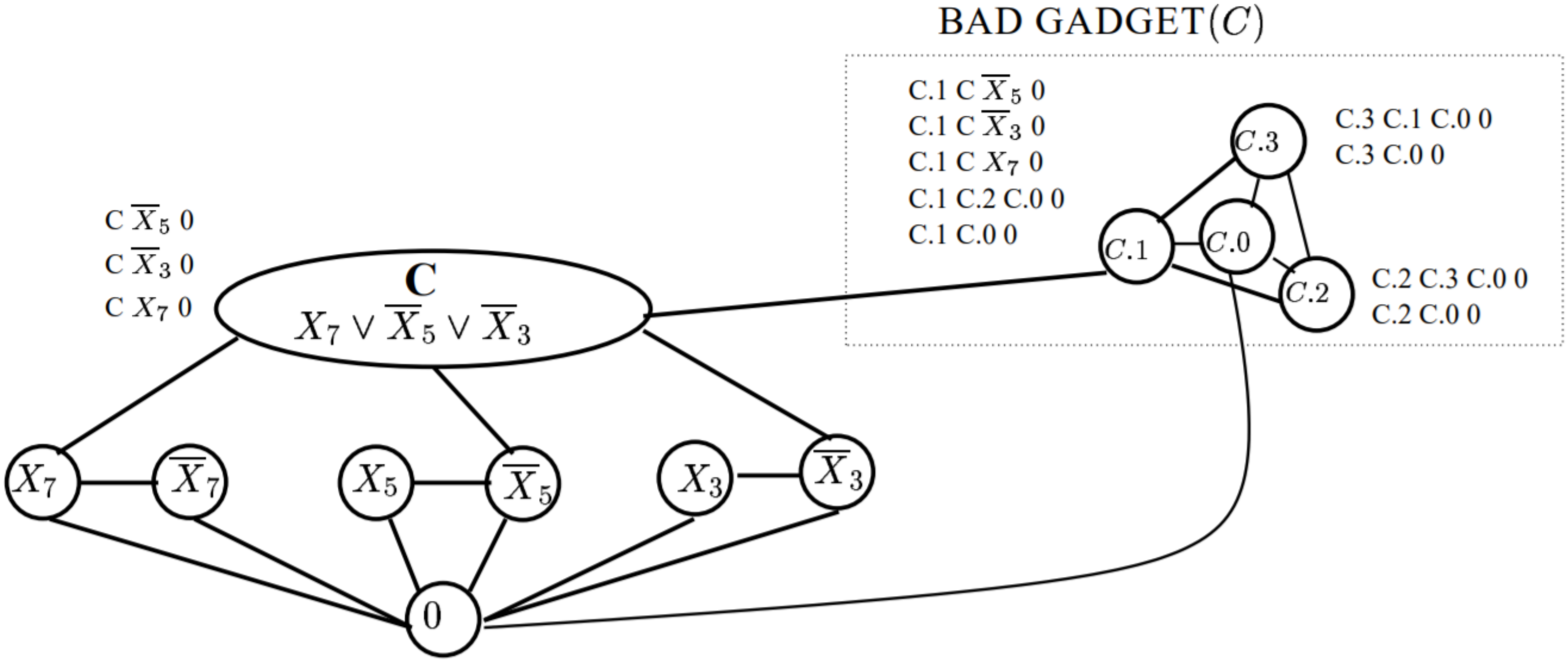


Fig. 7. Example of construction for clause $C = X_7 \vee \overline{X_5} \vee \overline{X_3}$

Paper for next week: Minesweeper

1. Stable paths is a tool for studying protocol convergence.
   - Simple mathematical model for routing solutions.
   - BGP converges iff a stable tree.

2. Paper insight: can we leverage this for verification?
   - Verifying all asynchronous executions of BGP → Hard!
   - Verifying all stable trees → Less Hard!
   - Can we use SMT since already NP-Complete?