Το Πρόβλημα Routing and Path Coloring και οι εφαρμογές του σε πλήρως οπτικά δίκτυα

Άρης Παγουρτζής

Ευχαριστίες: οι διαφάνειες αυτές βασίστηκαν εν μέρει στην παρουσίαση της διπλωματικής εργασίας του Στρατή Ιωαννίδη

Optical Fibers

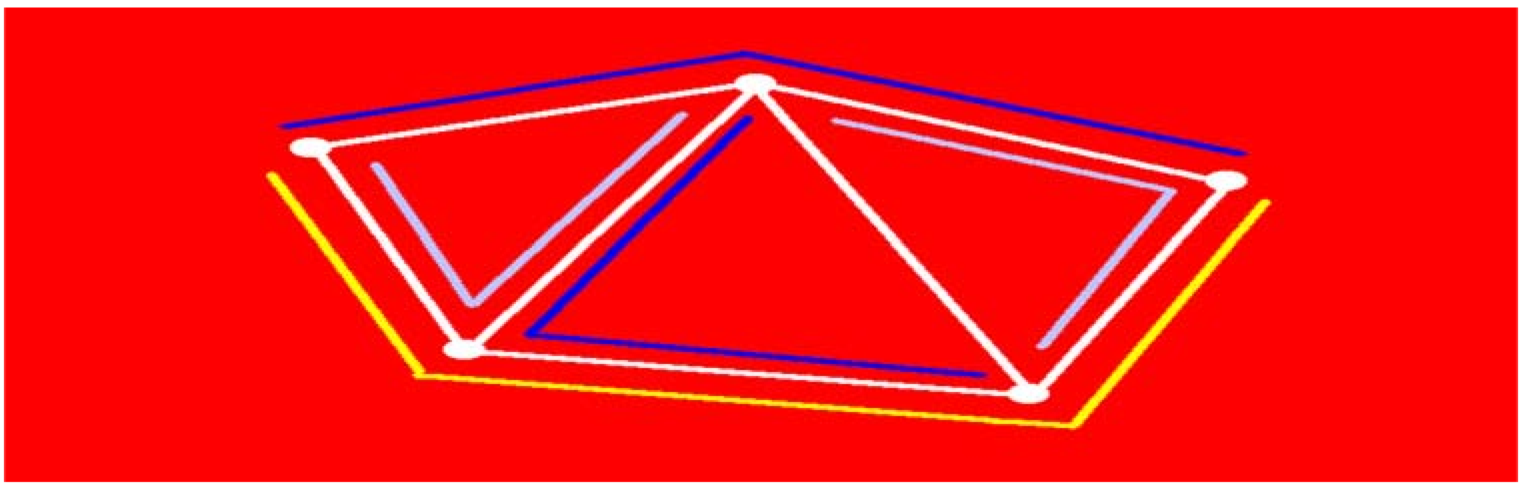
- High transmission rate
- Low bit error rate
- The bottleneck lies in converting an electronic signal to optical and vice versa

All-Optical Networks

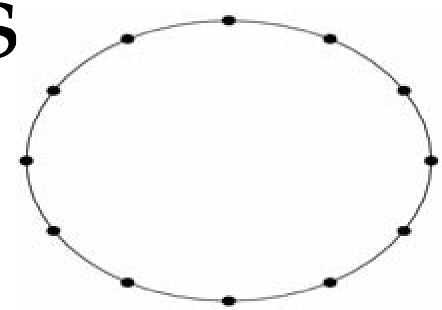
- All physical connections are optical
- Multiplexing is achieved through *wavelength division multiplexing (WDM)*: in each fiber multiple *colors* are used
- Switching on routers is done passively and thus more effectively (no conversion from electrical to optical)
- Two network nodes communicate using *one light beam*: a single wavelength is used for each connection

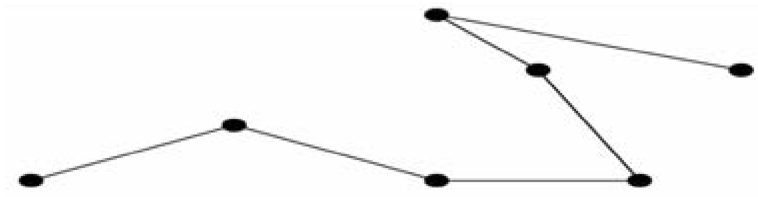
Graph Representation

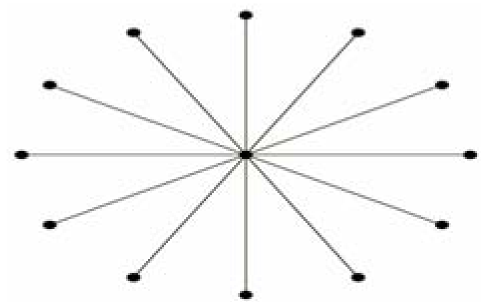
- All physical links are represented as graph edges
- Communication among nodes is indicated by paths
- Paths are assigned colors (wavelengths)
- Overlapping paths (i.e. sharing at least one edge) are assigned different colors

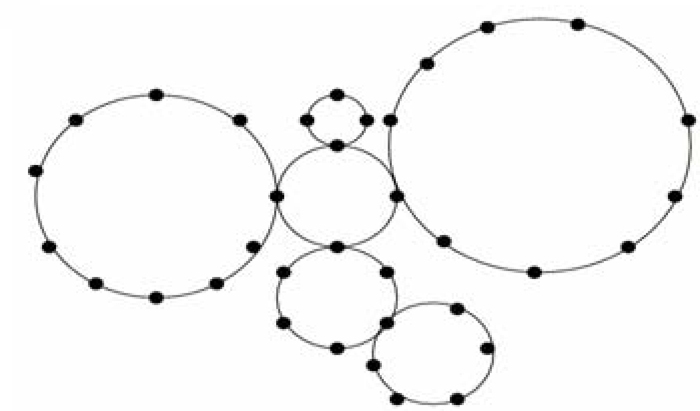


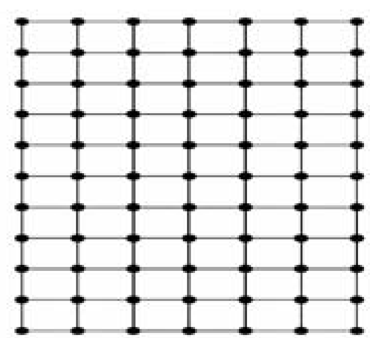
Graph Topologies

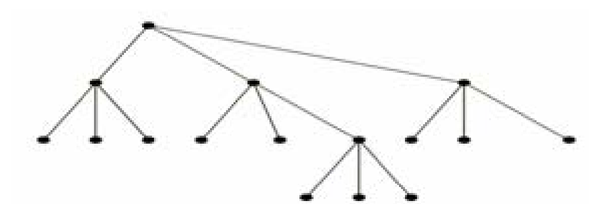












Graph Coloring (GC)

- *Input*: Graph G
- *Feasible solution*: Coloring of V using different colors for adjacent vertices
- *Goal*: Minimize the number of colors used, i.e. find chromatic number $\chi(G)$

- NP-hard
- There is no approximation algorithm of ratio n^{ε} for some $\varepsilon > 0$, unless P=NP (*polyAPX*-hard)
- Lower bound for $\chi(G)$: order (size) ω of maximum clique of G

Edge Coloring (EC)

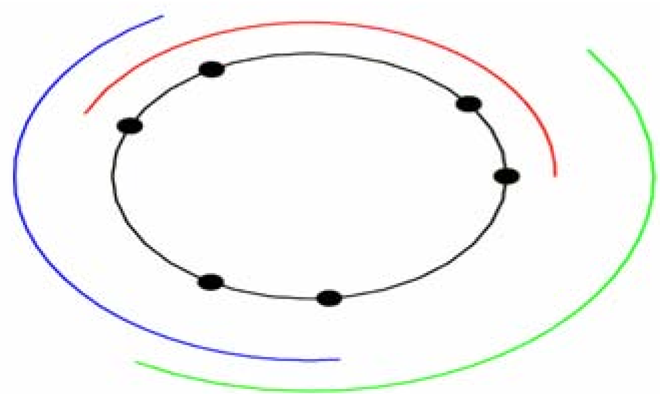
- *Input*: Graph *G*
- *Feasible solution*: Coloring of *E* using different colors for adjacent edges
- *Goal*: Minimize the number of colors used, i.e. find *chromatic index* $\chi'(G)$

- Lower bound for $\chi'(G)$: maximum degree $\Delta(G)$
- [Vizing'64]: $\chi'(G)$ is between $\Delta(G)$ and $\Delta(G)+1$ (simple graphs), and between $\Delta(G)$ and $3\Delta(G)/2$ (multigraphs)
- [Holyer'80]: NP-complete whether $\Delta(G)$ or $\Delta(G)+1$
- 4/3 -approximable in simple graphs and multigraphs
- Best possible (absolute) approximation unless P=NP

Path Coloring (PC)

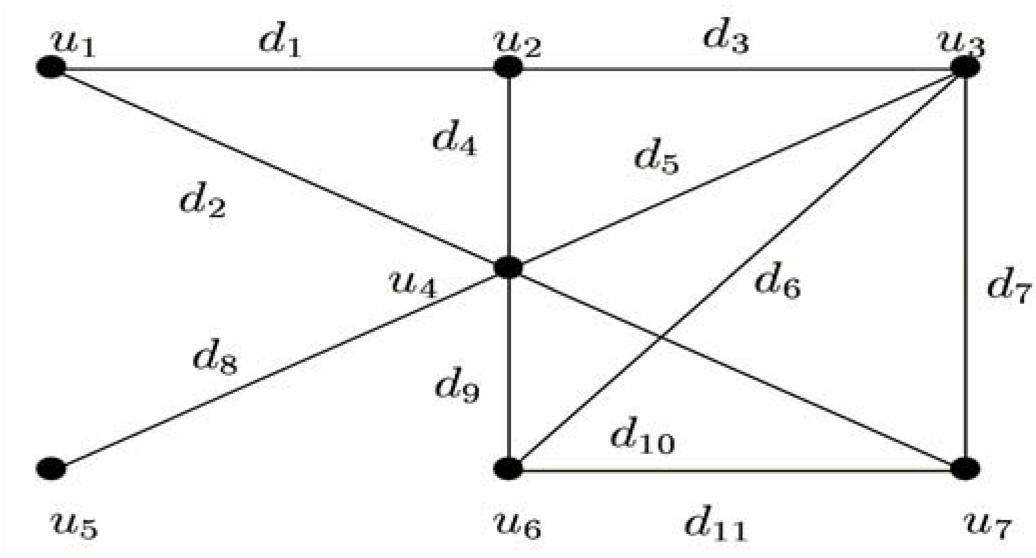
- *Input*: Graph *G*, set of paths *P*
- *Feasible solution*: Coloring of paths s.t. overlapping paths are not assigned the same color
- *Goal*: Minimize the number of colors used

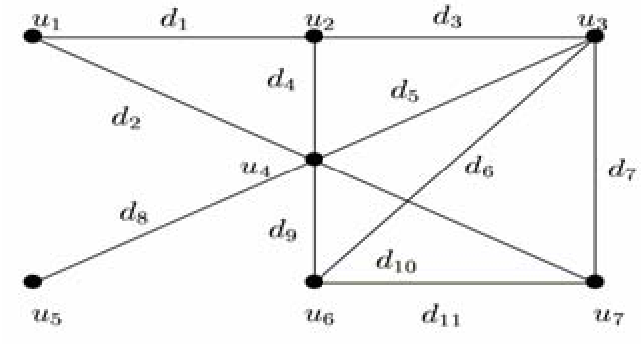
- Lower bound: maximum load L
- Reduces to GC by representing paths as vertices and overlapping paths as edges *(conflict graph)*
- Improved lower bound: order ω of the maximum clique of the conflict graph



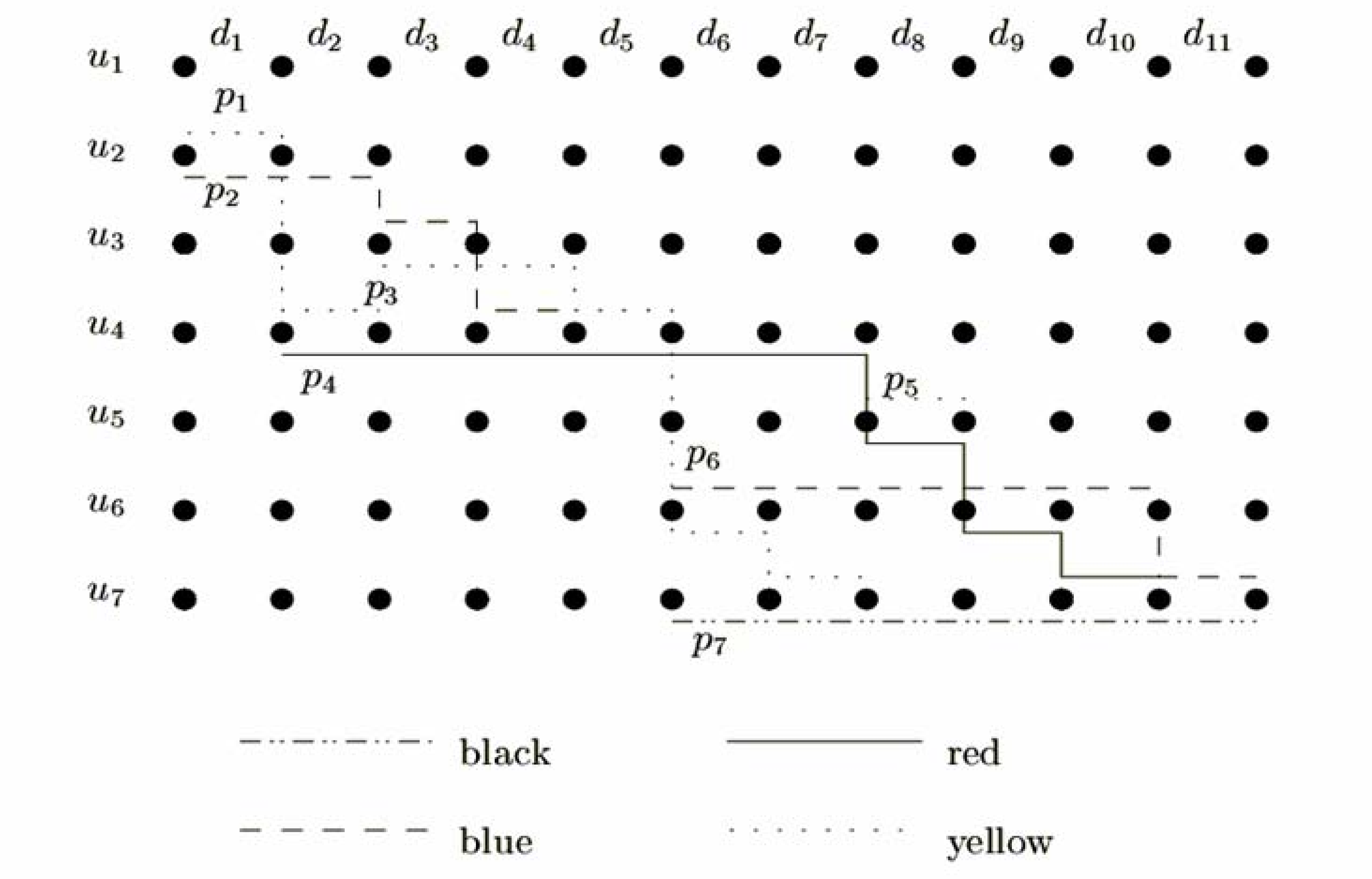
Path Coloring (PC)

- Corresponding decision problem is NP-complete
- In general topologies the problem is *poly-APX*-hard
- Proof: Reduction of GC to PC in meshes [Nomikos'96]





u_1
d_1
u_2
d_3
u_3
d_4
d_5
d_2
u_4
d_6
d_7
d_8
d_9
d_{10}
u_5
u_6
d_{11}
u_7



d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11}
u_1
u_2
u_3
u_4
u_5
u_6
u_7
p_1
p_2
p_3
p_4
p_5
p_6
p_7
black
red
blue
yellow

Chain PC

- Optimally soluble in poly time with exactly L colors

Ring PC

- *Other name:* Arc Coloring
- NP-complete [GJMP 80]
- Easily obtainable approx. factor 2: Remove edge *e* and color resulting chain. Color all remaining paths that pass through e with new colors.

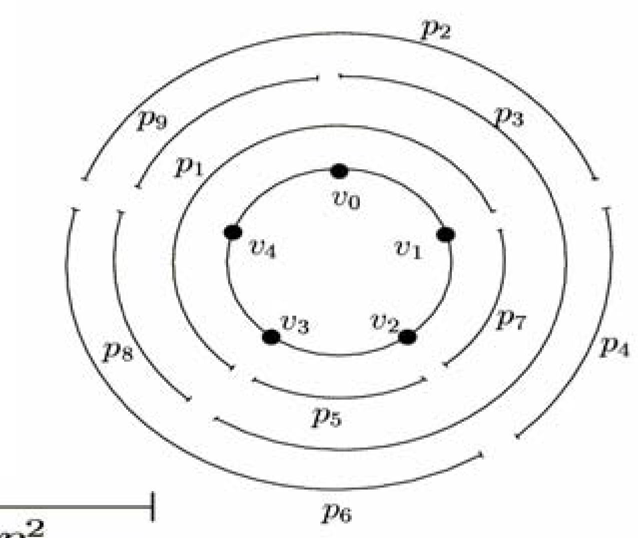
$$SOL_C \leq L$$

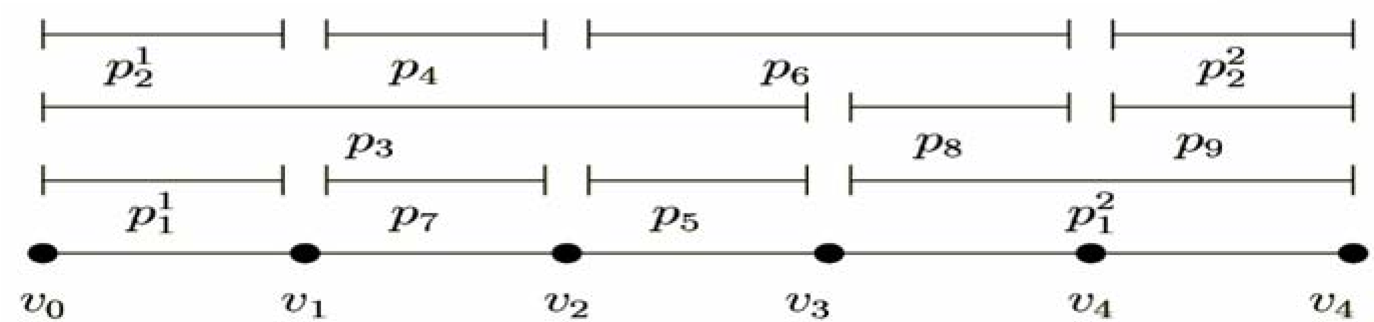
$$SOL \leq SOL_C + L - 1 \leq 2 \cdot OPT - 1$$

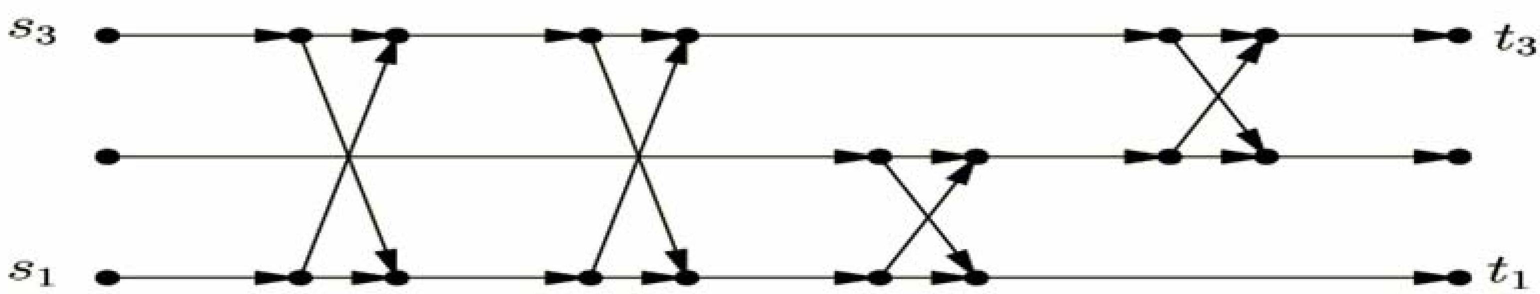
- W. K. Shih, W. L. Hsu: approx. factor 5/3
- I. Karapetian: approx. factor 3/2 (idea: use of maximum clique of conflict graph)

Ring PC

- V. Kumar: With high probability approx. factor 1.36
- Idea: Use of *multicommodity flow*

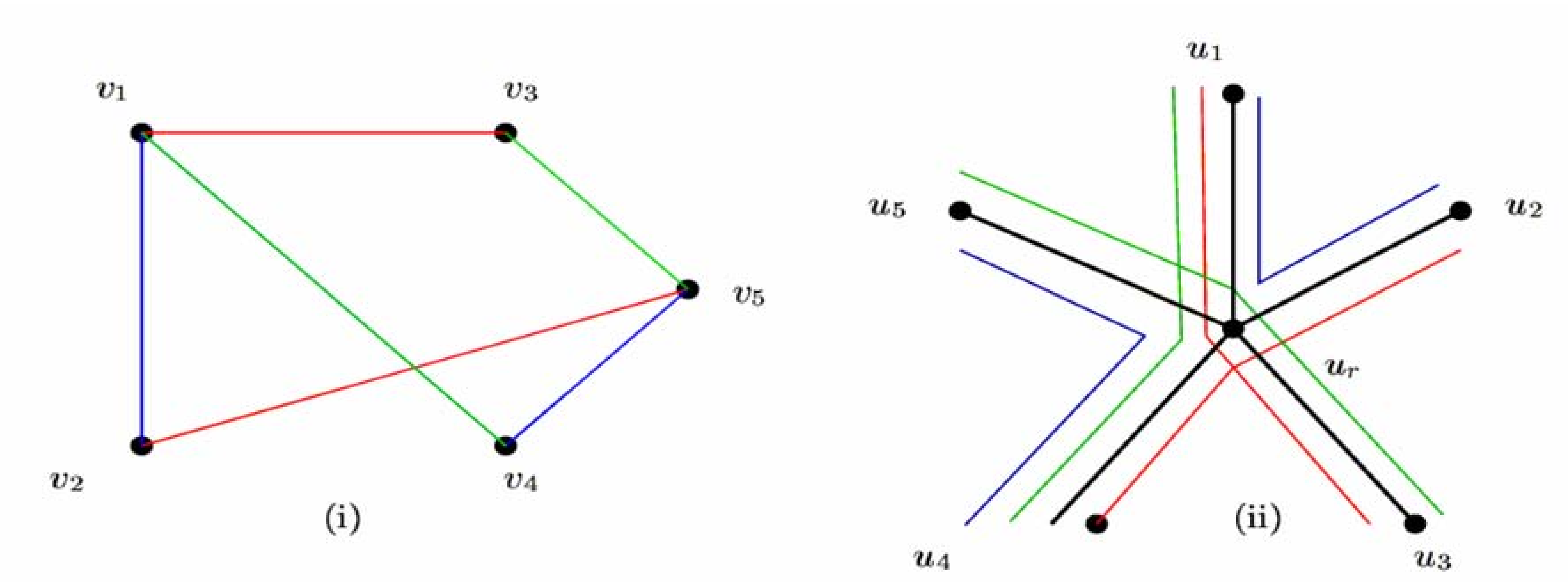






Star PC

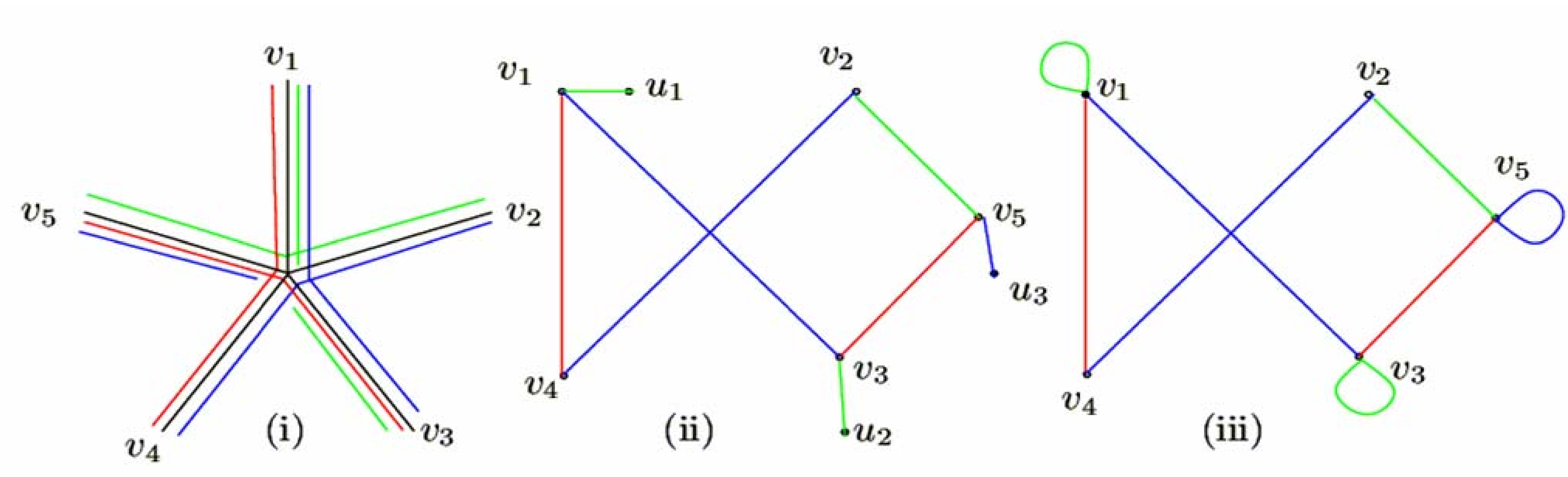
NP-completeness: Reduction of EC to Star PC



Approximation ratio: at least 4/3

Star PC: Approximability

Reduction of Star PC to EC in multigraphs



Approximation ratio: 4/3

Tree PC

Recursive Algorithm

if tree is a star **then**

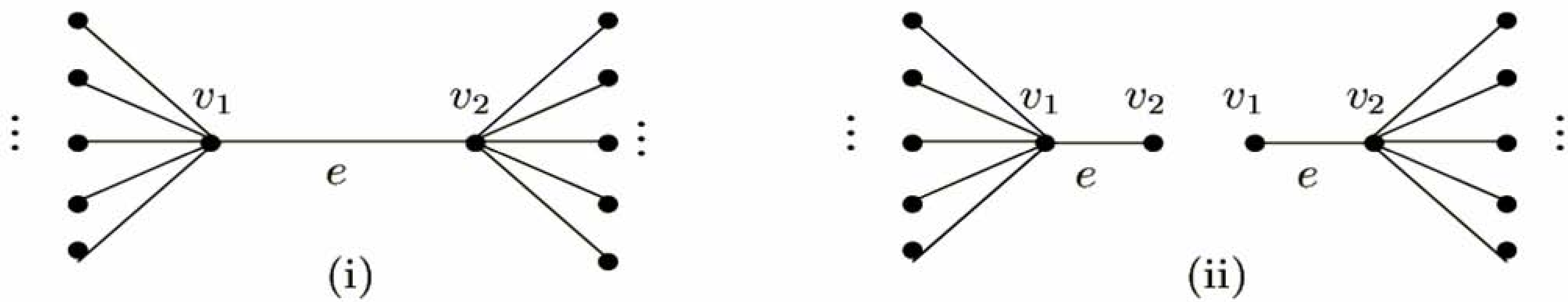
color it approximately

else

- Subdivide the tree by "breaking" one of its internal edges
- Color the resulting subtrees
- Join sub-instances by rearranging colors

Tree PC (ii)

$$P_1 = \{q = p \cap T_1 \mid p \in P\}$$
$$P_2 = \{q = p \cap T_2 \mid p \in P\}$$



Approximation ratio equal to the one achieved by the approximate Star PC algorithm, thus 4/3

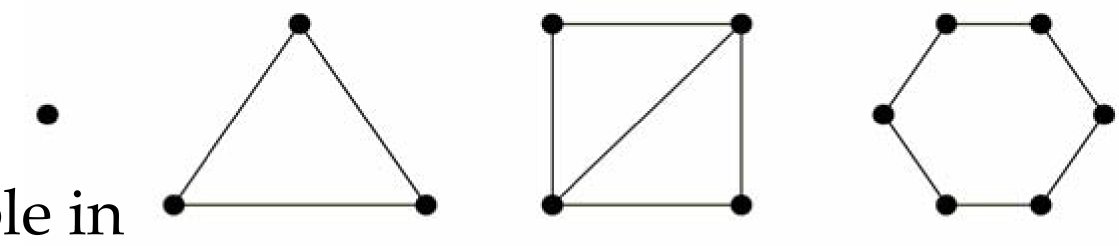
Bounded Degree Tree PC

- Trees of bounded degree are reduced by the above reduction to multigraphs of bounded size

- EC in bounded size multigraphs can be solved optimally in polynomial time

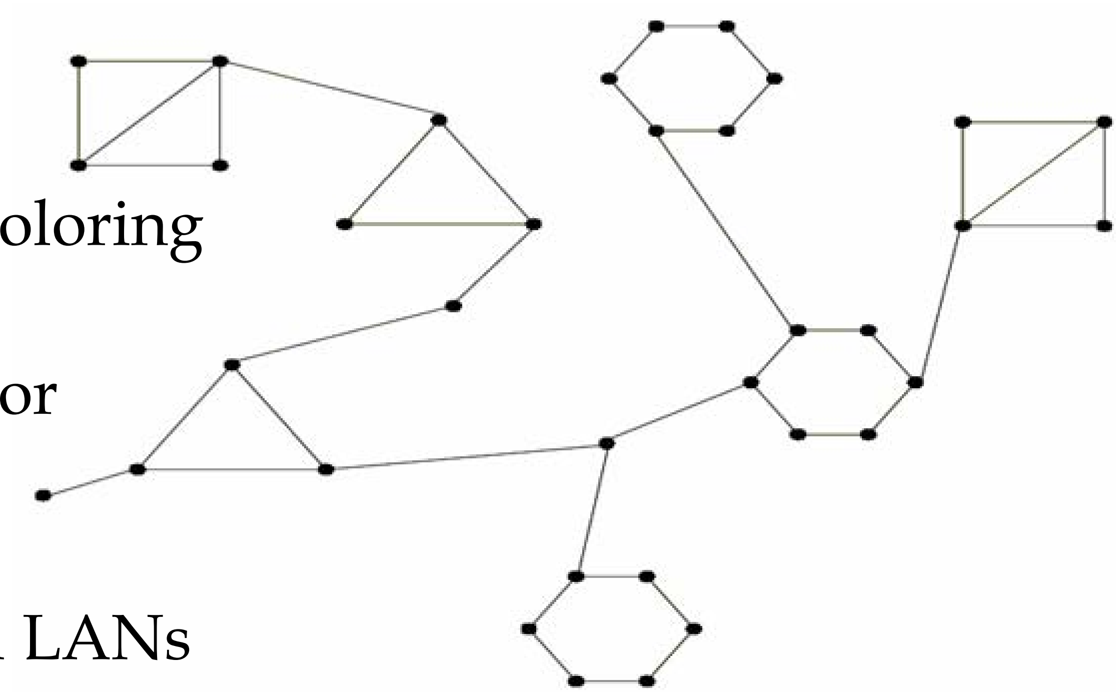
Generalized Tree (*S*,*d*) PC

- Finite set of graphs S
- Tree of degree at most *d*
- Optimally (exactly) soluble in polynomial time

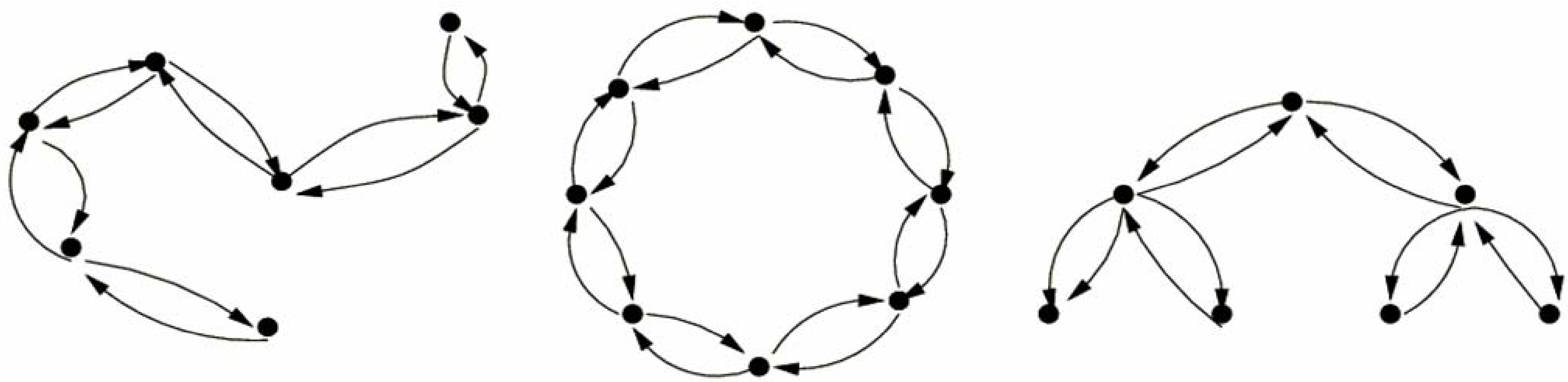


Idea:

- Since graphs are finite, coloring can be done in $|P|^{f(S,d)}$
- Recursive algorithm, color rearrangement
- Application: Backbone Networks of customized LANs



Directed Graphs

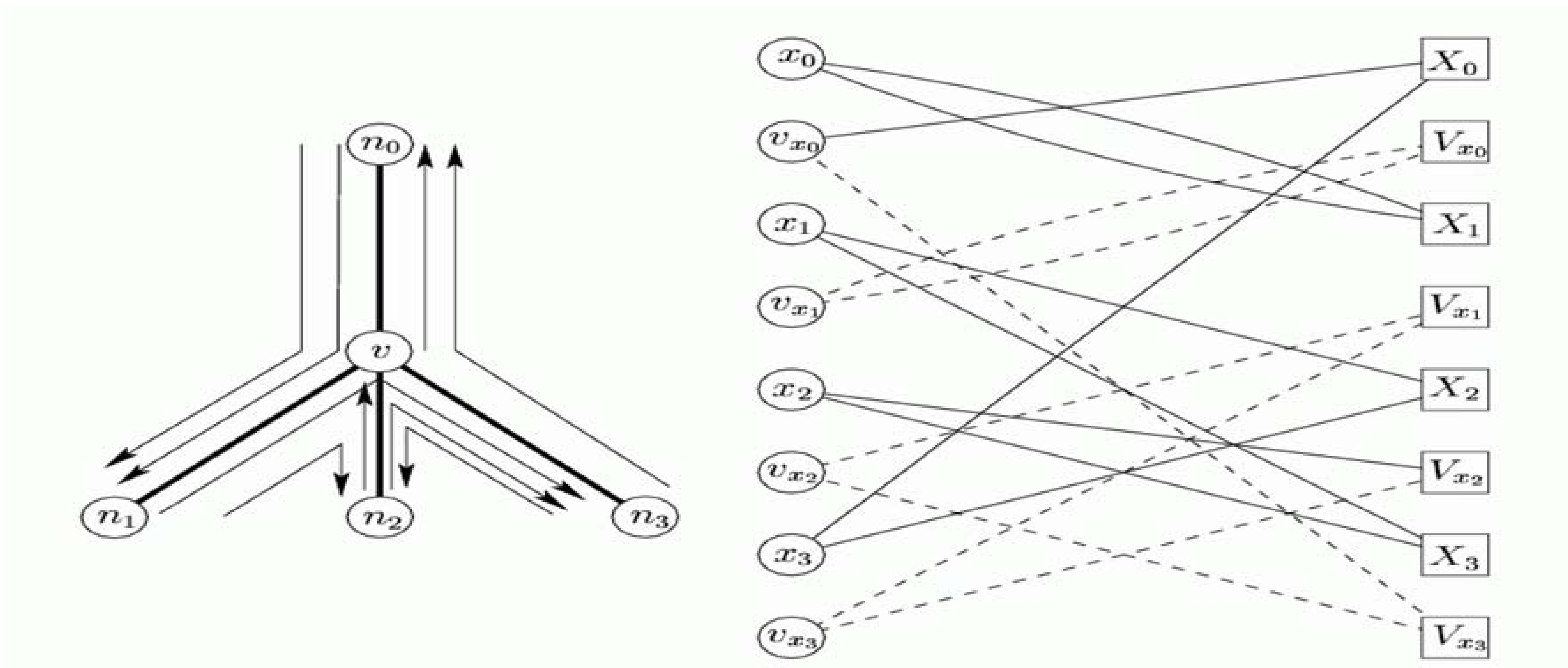


PC in directed graphs

- D-Chain PC: Reduced to two undirected instances
- D-Ring PC: As above
- D-TreePC:
  - Approximable within a 5/3 factor. Lower bound of 4/3; the 5/3 algorithm best possible among all greedy algorithms [Erlebach, Jansen, Kaklamanis, Persiano'97]
  - Not optimally soluble in bounded degree trees

PC in directed graphs

- D-Star PC: Optimally soluble in poly time!



Routing and Path Coloring (RPC)

- *Input*: Graph G, set of requests $R \subseteq V^2$
- *Feasible solution*: Routing of requests in R via a set of paths P and color assignment to P in such a way that overlapping paths are not assigned the same color
- *Goal*: Minimize the number of colors used

In acyclic graphs (trees, chains) RPC and PC coincide

Ring RPC

"Cut-a-link" technique [Raghavan-Upfal'94]

- Pick an edge e
- Route all requests avoiding edge e
- Solve chain instance with L colors

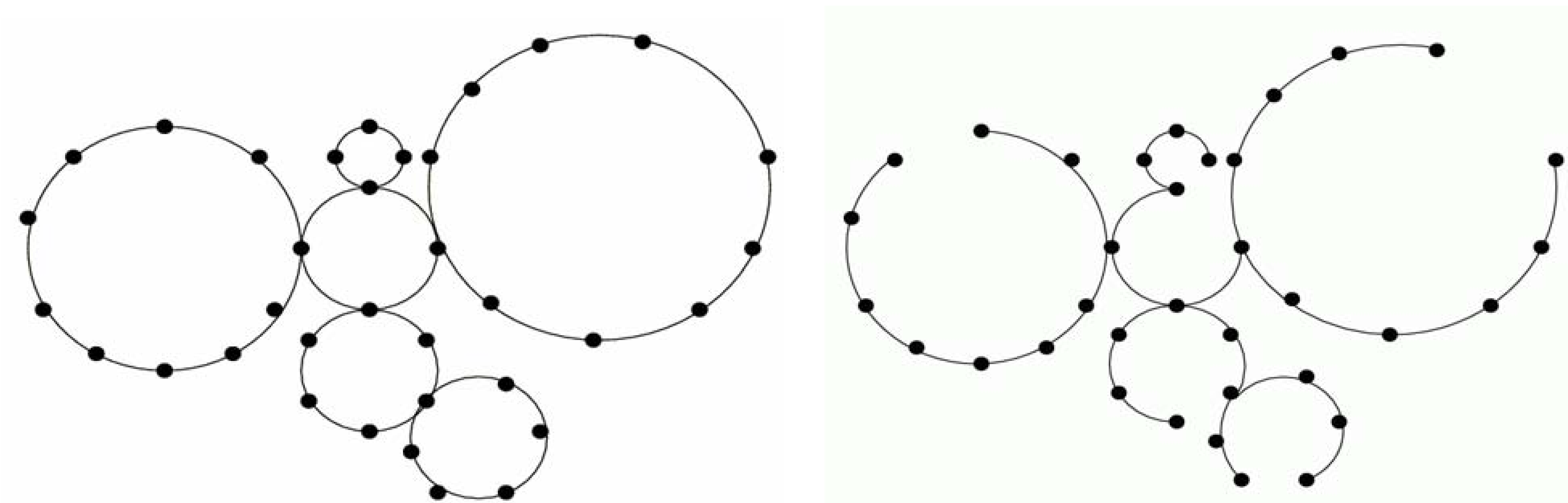
Thm: The above is a 2-approximation algorithm

Proof: $L <= 2\ L_{opt} <= 2\ OPT$

V. Kumar: 1.68-approximation with high probability

Tree of Rings RPC

Approximation ratio 3



RPC in (bi)directed topologies

- In acyclic topologies PC and RPC coincide

- In rings there is a simple 2-approximation algorithm.

- In trees of rings the same as before technique gives approximation ratio 10/3 (=2 x 5/3)