Algorithm Complexity Considerations of Content Connectivity Problem Formulation

Giap Le: Ph.D. Candidate, Computer Networks Lab, UC Davis

Supervisors: Distinguished Prof. Mukherjee and Prof. Tornatore

Friday, October 11, 2019

ACK: NSF-JUNO2 CNS-1818972

UCDAVIS
NETWORKS RESEARCH LAB

Outline

- An overview of optimization problem
- A review of content connectivity problem formulation
- Heuristic/relaxation consideration for content connectivity problem formulation

Optimization Problem in Optical Networks

- Resource allocation problem
- Formulated as an optimization problem

Linear Programming (LP)

minimize

$$\boldsymbol{c}^T.\boldsymbol{x}$$

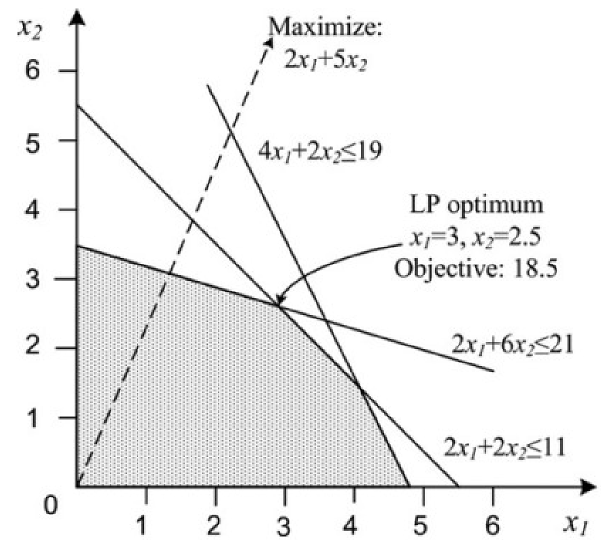
Subject to:

$$\boldsymbol{A}.\boldsymbol{x} \leq \boldsymbol{b}$$

Where, $\boldsymbol{A}$ is an $m \times n$ matrix, $\boldsymbol{x}^T = [x_1, x_2, \dots, x_n]^T$ is an $n \times 1$ vector of variables, and $\boldsymbol{c}$ and $\boldsymbol{b}$ are constants. Variables in $\boldsymbol{x}$ can take real values.

Linear Programming (LP)

- Efficient to find optimal solution
- Why?
  - Solution space = n-dimensional polyhedron
  - Optimal solution at one of vertices
  - Solver: enumeration among vertices



$$
\begin{aligned}
&\text{Maximize}: 2x_1 + 5x_2 \\
&\text{subject to} \\
&2x_1 + 6x_2 \le 21 \\
&4x_1 + 2x_2 \le 19 \\
&2x_1 + 2x_2 \le 11 \\
&x_1 \ge 0, x_2 \ge 0
\end{aligned}
$$

[1] E. A. Varvarigos and K. Christodoulopoulos, "Algorithmic Aspects in Planning Fixed and Flexible Optical Networks With Emphasis on Linear Optimization and Heuristic Techniques," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 681-693, Feb.15, 2014.

Linear/Mixed Integer Programming (ILP, MIP)

- Unfortunately, we cannot always formulate our problem as an LP

minimize

$$\boldsymbol{c}^T.\boldsymbol{x}$$

Subject to:

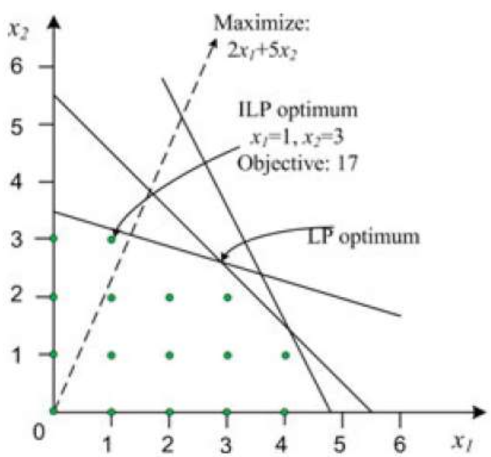
$$\boldsymbol{A}.\boldsymbol{x} \leq \boldsymbol{b}$$

Where, $\boldsymbol{A}$ is an $m \times n$ matrix, $\boldsymbol{x}^T = [x_1, x_2, \dots, x_n]^T$ is an $n \times 1$ vector of variables, and $\boldsymbol{c}$ and $\boldsymbol{b}$ are constants. Variables (or some) in $\boldsymbol{x}$ can only take integer numbers (e.g., number of wavelengths, number of transceivers, etc.).

UCDAVIS NETWORKS RESEARCH LAB

Why ILP and MIP are NP-Complete?

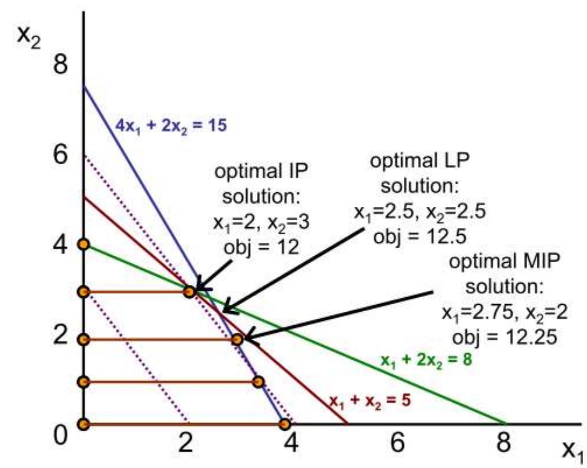
- NP-complete: no guaranteed optimal solution on polynomial time



ILP

$$\begin{aligned}&\text{Maximize}: 2x_1 + 5x_2\\&\text{subject to}\\&2x_1 + 6x_2 \le 21\\&4x_1 + 2x_2 \le 19\\&2x_1 + 2x_2 \le 11\\&x_1 \ge 0, x_2 \ge 0\\&x_1, x_2 \ \text{integers}\end{aligned}$$

Huge solution search space



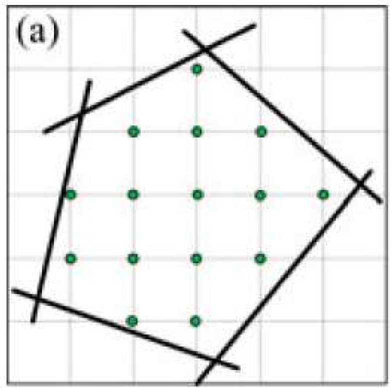
MIP

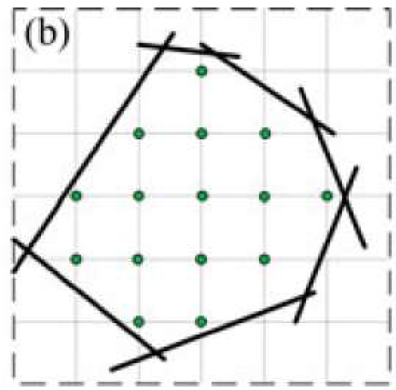
UCDAVIS NETWORKS RESEARCH LAB

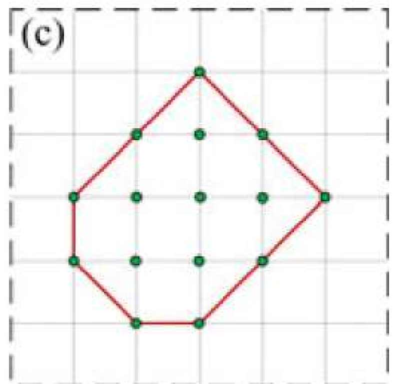
LP and ILP Connection: Convex Hull

- If an LP problem has an optimal integer solution, it is also solution to corresponding ILP

- LP can be used to estimate upper (or lower) bound of ILP problem





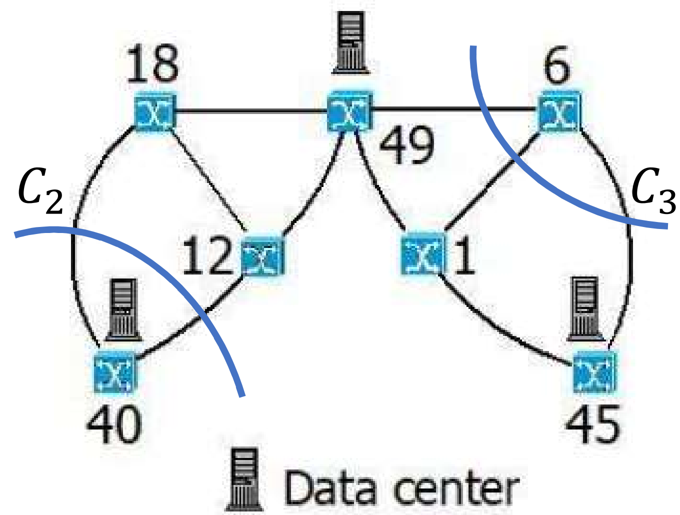


- ✓ Convex hull: minimal solution space including all feasible integer solutions
- ✓ We can rearrange ILP constraints to get a convex hull and solve the ILP as an LP
- ✓ Non-trivial tasks

UCDAVIS NETWORKS RESEARCH LAB

Network Cutset and Content Cutset

- Network Connectivity (NC) cutset:
  - ✓ C_2
  - ✓ Removal all links in C_2 violates NC
- Content Connectivity (CC) cutset:
  - ✓ C_3
  - ✓ Removal all links in C_3 disconnects node 6 from content
  - ✓ Nodes co-located with datacenters are content-connected



Logical topology

Content available at all DCs

Problem Statement

Given:

- ✓Logical topology
- ✓Physical topology

Objective:

- ✓Minimize network resource usage

Output:

- ✓Mapping with content connectivity after n link failures

Input Parameters

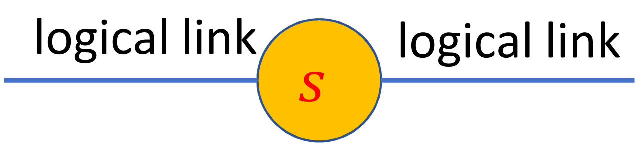
- $G_P(V_P, E_P)$: physical topology (graph)
- V_P: set of physical nodes
- E_P: set of physical links
- $G_L(V_L, E_L)$: logical topology (graph)
- V_L: set of logical nodes
- E_L: set of logical links
- D: set of Datacenter, $D \subset V_L$
- F_{ij}: number of fiber from i to j
- W: number of wavelength/fiber
- n: number of physical link failures
- P_n: set of n physical links
- C_{cc}: set of content-connected cutsets (next slides)

UCDAVIS NETWORKS RESEARCH LAB

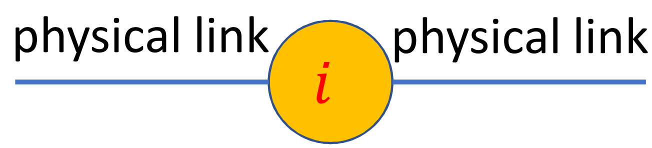
CC-n Existence

Theorem 1: Given $G_P(V_P, E_P)$, $G_L(V_L, E_L)$, and D, to find the mapping of G_L over G_P that guarantees CC-n, the following conditions must be satisfied:

- ✓ each logical node $s \in V_L - D$ has a nodal degree $\delta(s) \geq n + 1$, and
- ✓ each physical node $i \in V_P: i = s$ has a nodal degree $\delta(i) \geq n + 1$.



$\delta(s) = 2$, no CC-2 solution for node s



$\delta(i) = 2$, no CC-2 solution mapped over node i

UCDAVIS NETWORKS RESEARCH LAB

CC-n Enforcement

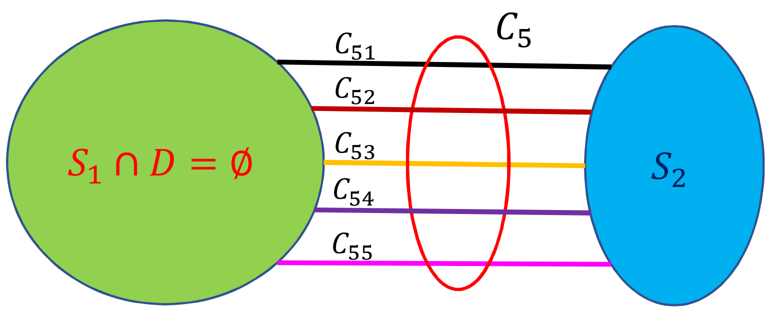
Theorem 2: Given $G_P(V_P, E_P)$, $G_L(V_L, E_L)$, D, let $P_n = \{\{P_n^k\}: \left|\{P_n^k\}\right| = n, \{P_n^k\} \in E_P\}$ be set of all possible combinations of n distinct physical links, and $C_{CC} = \{C_{CC}^l(S_l, V_L - S_l): S_l \cap D = \emptyset\}$ be set of logical topology content-connected cutsets where the removal of all logical links in each cutset C_{CC}^l disconnects G_L and divides V_L into two disjoint sets with one set without datacenters, the mapping of G_L over G_P is CC-n if and only if:

$$\sum_{ij \in P_n^k, st \in C_{CC}^l} f_{ij}^{st} \leq \left|C_{CC}^l\right| - 1, \forall P_n^k \in P_n, \forall C_{CC}^l \in C_{CC}.$$

UCDAVIS NETWORKS RESEARCH LAB

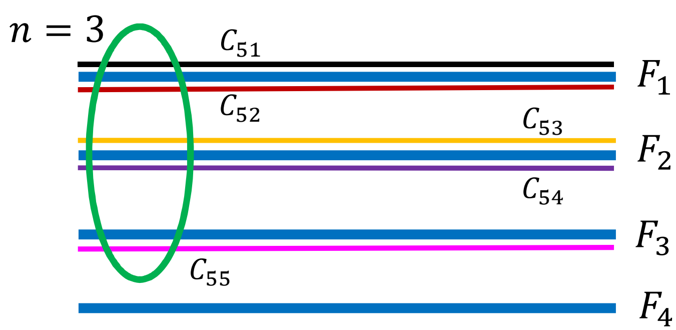
CC-n Enforcement

Theorem 2: Example $n = 3$ (survivable against 3 link failures)

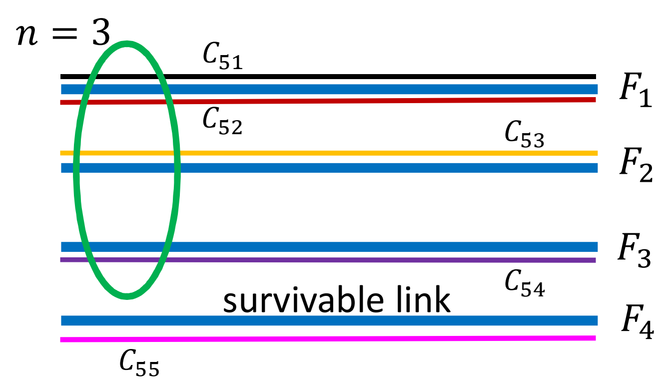


Content-Connected Cutset C_5

Physical links



CC-3 not survivable



CC-3 survivable

UC DAVIS NETWORKS RESEARCH LAB

Repeat for all possibilities in networks

Mathematical Formulations of CC-n Problem

Objective function:

$$\min \sum_{ij \in E_P,\, st \in E_L} f_{ij}^{st}$$

- ✓ Result in an ILP
- ✓ Lower complexity (compared to previous works)
- ✓ But still need relaxation/heuristic (for further publications)

Subject to:

Capacity Constr.

- $\sum_{st \in E_L} f_{ij}^{st} \le F_{ij} \times W, \forall ij \in E_P$

Flow Constr.

- $\sum_{j:ji \in E_P} f_{ji}^{st} - \sum_{j:ij \in E_P} f_{ij}^{st} = \begin{cases} -1 \text{ if } i = s \\ 1 \quad \text{if } i = t \\ 0 \text{ otherwise} \end{cases}$

  $\forall i \in V_P, \forall st \in E_L$

CC-n Constr.

- $\sum_{ij \in P_n^k, st \in C_{cc}^l} f_{ij}^{st} \le \left|C_{cc}^l\right| - 1$

  $\forall P_n^k \in P_n, \forall C_{cc}^l \in C_{cc}$

UCDAVIS NETWORKS RESEARCH LAB

Problem Re-Statement

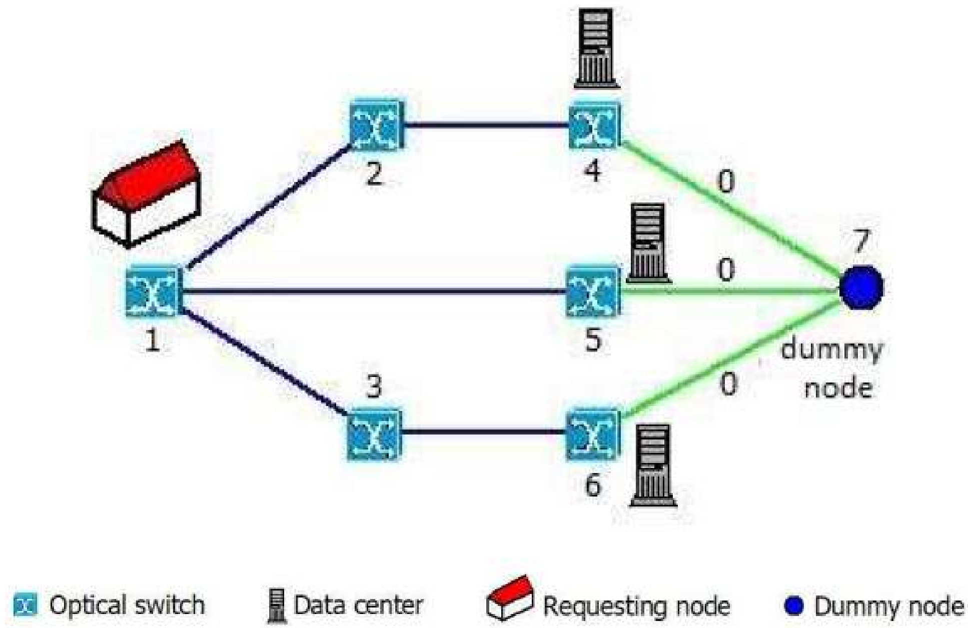
- Physical networks are fixed (infrastructure)
- Conventional: mapping a given logical topo over a physical one with CC
- Logical network can be flexible:
  - ✓Flexible number of datacenters: D
  - ✓Flexible logical links (lightpaths): number and connection
  - ✓Set of content requesting nodes: $V_L \setminus \mathrm{D}$

UC DAVIS NETWORKS RESEARCH LAB

How to Possibly Solve the CC Problem Faster?

- $V_L = \{1, 2, 3, 4, 5, 6\}$
- $D = \{4, 5, 6\}$
- Content requesting nodes:

  $$R = V_L \backslash \mathrm{D} = \{1, 2, 3\}$$
- Dummy node: $\{7\}$
- Establish n link-disjoint paths from node 1 to dummy node
- Nodes $\{2, 3\}$ must be used as transit nodes
- Repeat for all requesting nodes



UCDAVIS
NETWORKS RESEARCH LAB

Establish n-Edge Disjoint Paths

- k-shortet paths (Dijkstra, Bellman-Ford): No guarantee of disjointness
- Suurballe and Bhandari: primary and backup paths only (need extension)
- Ford-Fulkerson: efficient but we wish to learn more to build minimal logical topology

Establish n-Edge Disjoint Paths: Faster

- Set capacity for each physical link equal to 1
- Set flow for each logical link equal to 1
- Run shortest-path mapping (ILP without CC-n enforcement constraint)
- Solution may be sub-optimal
- Interesting question: how to build a k-edge (k = n+1)connected graph?

UCDAVIS NETWORKS RESEARCH LAB

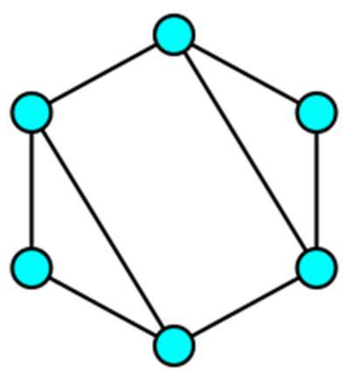
Minimal Logical Graph

- Number of nodes (i.e., offices requesting for content): fixed
- Number of datacenters in logical graph: flexible (upper bound = number of DCs available in physical infrastructure)
- Most important: how graph is connected
- How to define a minimal logical graph: depending on operator's need

UCDAVIS NETWORKS RESEARCH LAB

k-Edge Connected Graph

- Link failure protection
- Minimum logical graph: k-edge connected graph
- $k = n + 1$



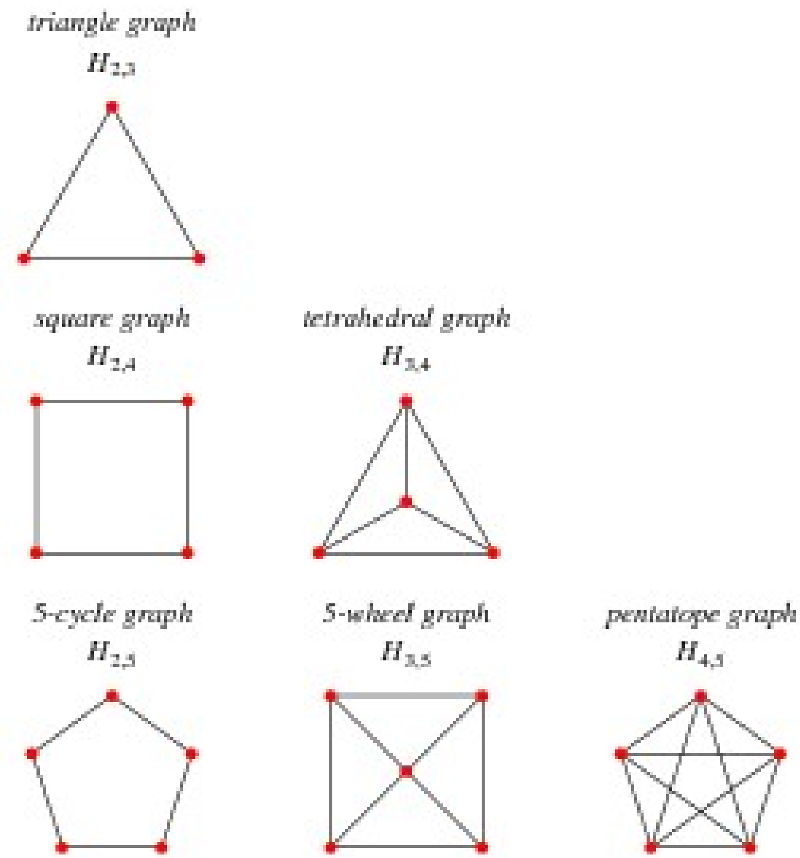
- ✓ k-edge connected graph: k = minimum number of edges whose removal disconnects graph
- ✓ Example in figure: 2-edge connected graph

UCDAVIS NETWORKS RESEARCH LAB

k-Edge Connected with Min. Number of Edges

- First: number of lightpaths
- Second: consider number of datacenters
- Harary graph (k, v):

Lower bound = $\text{ceil}(\frac{kv}{2})$



UCDAVIS NETWORKS RESEARCH LAB

Summary and Ongoing Research

- ILP: done
- Heuristic/relaxation: almost done
- Minimal logical graph: done
- Number of datacenters: investigating
- Location of datacenters: investigating