Efficiently Reaching the Largest Wireless Capacity with the Fewest Relays

Bai Liu*, Xiugang Wu†, Ayfer Özgür†

*Tsinghua University, †Stanford University

Introduction

Scenario

- Wireless communication
- **More relays → higher capacity**
- **Fewer relays → lower costs**
- Need to balance between capacity and cost

Problem Formulation

When all relays work, the capacity reaches maximum (*global maximum capacity*). However, we **don't have to apply all relays** to reach *global maximum capacity*.
Global network: a collection of subnetworks that reach *global maximum capacity*.
Optimal global network: the *global network* with fewest relays.
Problem: how to efficiently obtain optimal global network?

Existing Studies

No directly related work was found.
Calculation of capacity: [1] utilizes the submodularity property of the cut function and [2] proposes a method using dynamic programming approach.
Selection of relays: [2] applies simulated annealing approach, while [3] approximates the problem as a non-linear optimization problem.
Methods of existing studies are **time-consuming** for this problem.

Objectives

Theory

1. Explore structural characteristics of a *global network*.
2. Design an explicit criterion to decide the *optimal global network*.

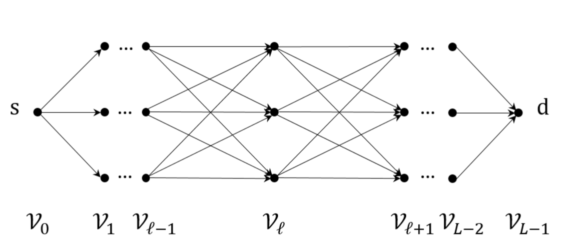
Practice

1. Develop efficient algorithms to obtain the *optimal global network*.
2. Efficiently obtain the new result when parameters change.

Model

Structure

- **Layered Gaussian relay network**
- source s → destination d
- L layers, with K_ℓ nodes in the ℓth layer
- Every edge: channel gain $H_{i,j}^{\ell}$
- $Y_j^{\ell+1} = \sum_{i=1}^{K_\ell} H_{i,j}^{l} X_i^{\ell} + Z_j^{\ell+1}$, $Z_j^{\ell+1}$ is i.i.d. $\mathcal{CN}(0,1)$



Capacity

Cut Ω: a subset of the network that satisfies $s \in \Omega$, $d \in \bar{\Omega}$.
For a given network $\mathcal{M}$, its **capacity** can be calculated as

$$C_{\mathcal{M}} = \min_{\{\Omega\}} \log\det\left(\mathbf{I} + \mathbf{H}_\Omega \mathbf{H}_\Omega^\dagger\right)$$

where $\mathbf{H}_\Omega$ denotes the MIMO channel matrix from nodes in Ω to nodes in $\bar{\Omega}$.
Using the characteristic of hierarchy, the equation can be reformulated as

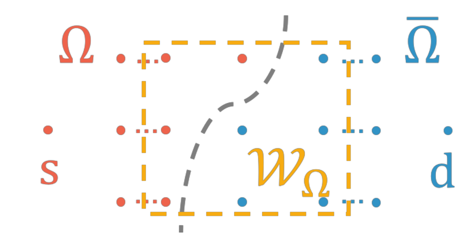
$$C_{\mathcal{M}} = \min_{\{\Omega\}} \sum_{\ell=1}^{L-1} \log\det\left(\mathbf{I} + \mathbf{H}_\Omega^{\ell} \mathbf{H}_\Omega^{\ell\dagger}\right)$$

where $\mathbf{H}_\Omega^{\ell}$ denotes the MIMO channel matrix from nodes $\Omega \cap \mathcal{V}_{\ell-1}$ to nodes in $\bar{\Omega} \cap \mathcal{V}_\ell$.

Theoretical Study

Definitions

Definition 1 Given a cut Ω and two nodes P, Q. If $P \in \Omega$, $Q \in \bar{\Omega}$ and Q is in the next layer of P, we define both P and Q as *working node*. Denote the set of working nodes as $\boldsymbol{\mathcal{W}}_\Omega$.



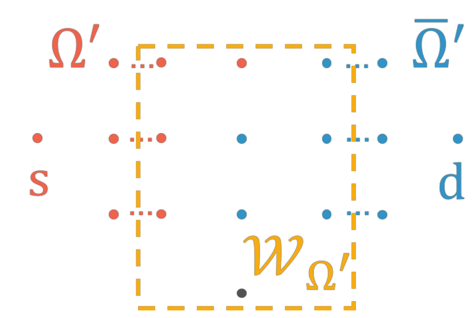
Definition 2 Define the *cutset* of a given *cut* Ω as

$$C(\Omega) = \log\det\left(\mathbf{I} + \mathbf{H}_\Omega \mathbf{H}_\Omega^\dagger\right)$$

Definition 3 Denote the *bottleneck cut* as $\Omega^* = \arg\min \mathrm{C}(\Omega)$.

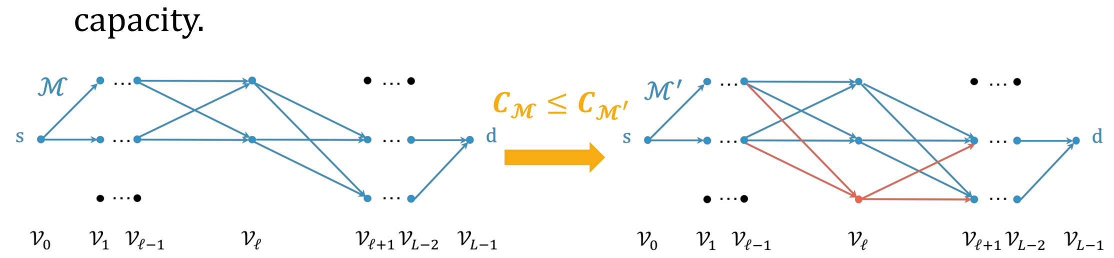
Properties

New nodes in $\mathcal{W}$
Adding a new node into a $\mathcal{W}$ will strictly **increase** its *cutset*.
$C(\Omega') > C(\Omega)$.



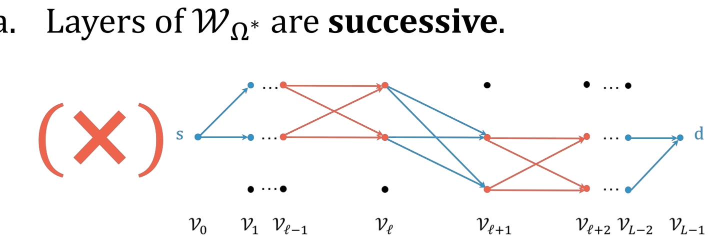
New nodes in $\mathcal{M}$

a. When the node number of subnetwork increases, the capacity of the optimal subnetwork $\mathcal{M}^*$ will **not decrease**.
b. Adding new nodes into a subnetwork $\mathcal{M}$ will **not decrease** its capacity.



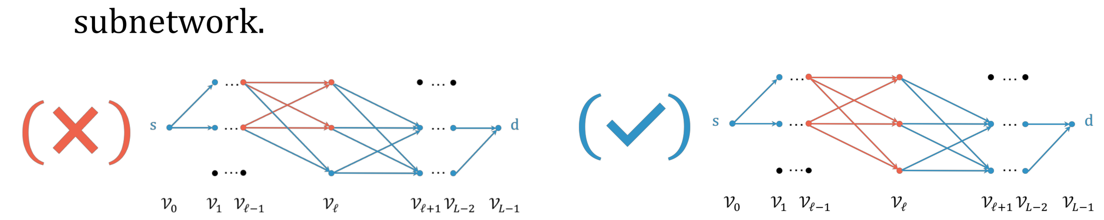
Bottleneck

a. Layers of $\mathcal{W}_{\Omega^*}$ are **successive**.



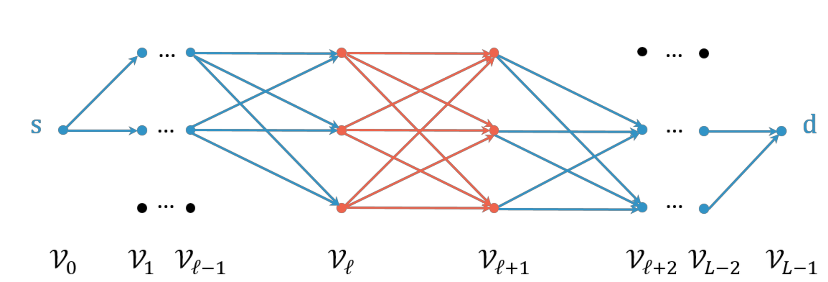
Denote red nodes as working nodes

b. $\mathcal{W}_{\Omega^*}$ takes up **all** the nodes in corresponding layers of the subnetwork.



Criterion

A subnetwork is a *global network*. ⟺ **$\mathcal{W}_{\Omega^*}$ takes up all the nodes in corresponding layers of the whole network.**



The subnetwork is *global network*

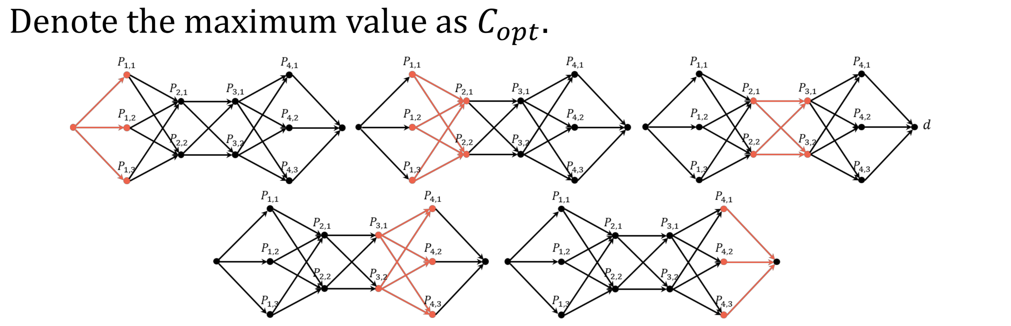
It is not necessary to calculate through the whole network.
This criterion can reduce the time complexity of judgement by at least a half.

Algorithms

Find *optimal global network*

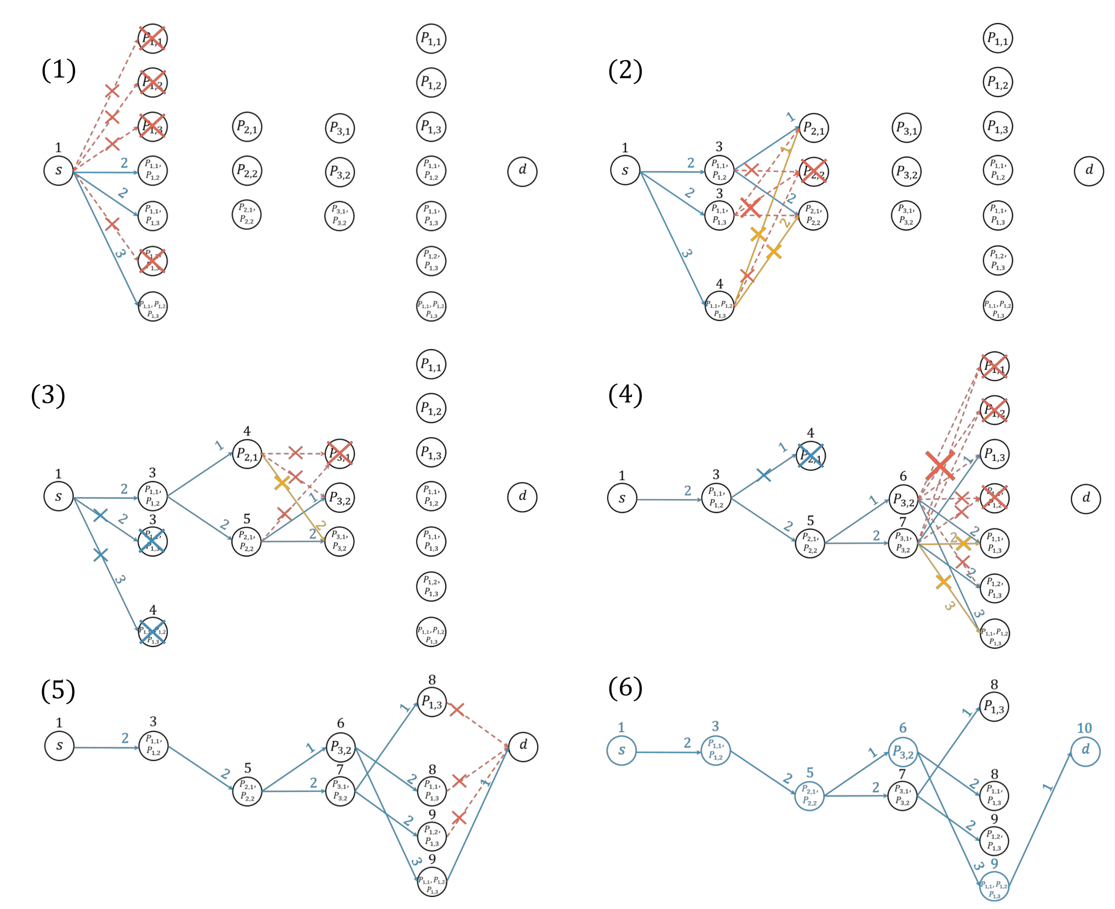
Step 1: Calculate global maximum capacity

1. For every ℓ, calculate the **maximum *cutset*** between layer ℓ and layer ℓ+1.
2. Denote the maximum value as C_{opt}.

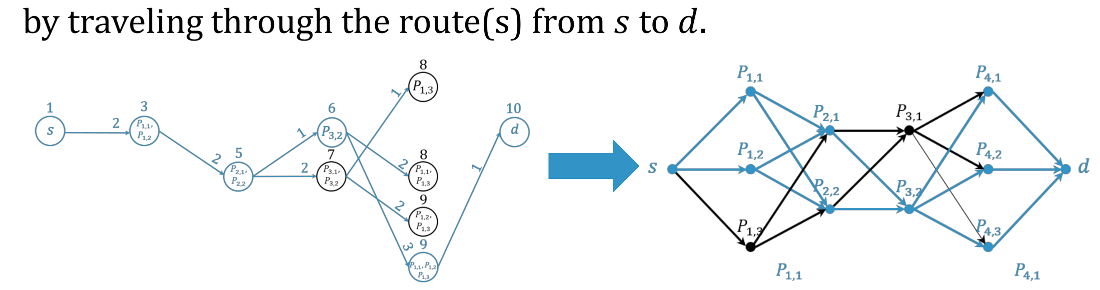


Step 2: Dynamic programming

1. Draw the figure below, in which every circle represents a selection of nodes.
2. Calculate the *cutset* between possible pairs of circles respectively from **current** layer and **previous** layer.
3. Connect the circles whose *cutset* is **not larger than** C_{opt} (solid line).
4. Delete nodes that have no connection with previous layer (marked with ×) or have nowhere to go (marked with ×) .
5. From s to the chosen circle in current layer, delete the paths that are not the **shortest** (marked with ×).
6. Move to next layer.
7. When d is reached, end the algorithm.



Step 3: Construct the optimal global network
From the result in Step 2, we can easily get the *optimal global network* by traveling through the route(s) from s to d.



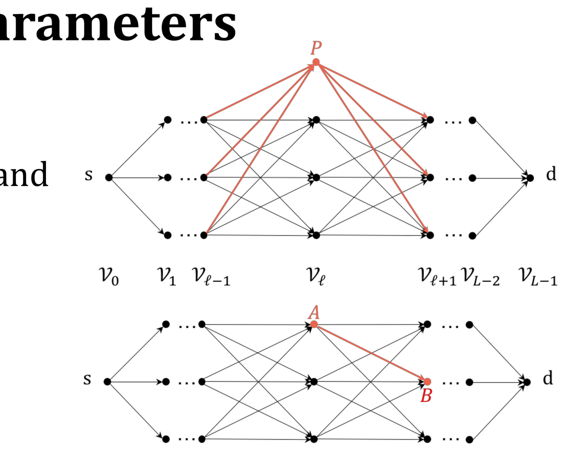
Deal with dynamic parameters

Adding new node *P*
With the property of dynamic programming, only layer $\ell - 1$, ℓ and $\ell + 1$ need to be searched.

Changing channel gain H_{AB}
If H_{AB} satisfies

$$a|H_{AB}|^2 - 2|b||H_{AB}|\cos\left(\theta_{H_{AB}} - \theta_b\right) + c - \min\{d_1, d_2\} \ge 0^*$$

the result will **not change**.



* Expressions of a, b, c, d_1, d_2 are omitted due to space limit

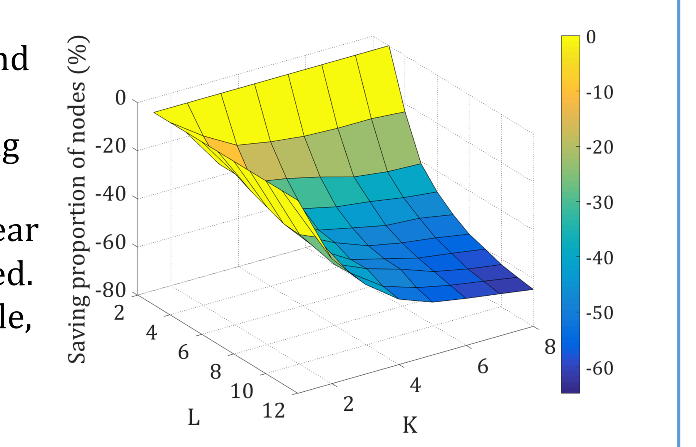
Results

Environment

Individual channel gains were chosen to be i.i.d. according to $10\log_{10}\left(\left|H_{i,j}^{\ell}\right|^2\right) \sim U[0,35]$, $\angle H_{i,j}^{\ell} \sim U[0,2\pi]$.
Tests were executed in MATLAB for Mac R2015b.

Necessity

When L (number of layers) and K(number of nodes in each middle layer) grow, the saving becomes more significant.
When $L = 12$ and $K = 8$, near **70%** of the nodes can be saved.
In the network with large scale, only **a fraction of** nodes are required to achieve *global maximum capacity*.



Time complexity

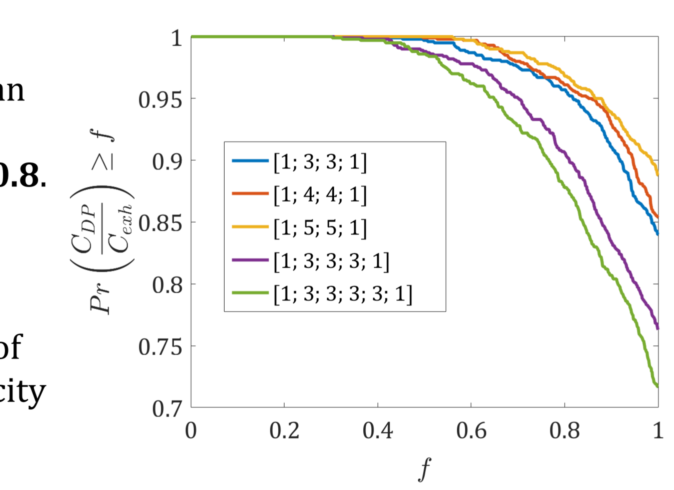
We compared the time complexity between dynamic programming approach and exhaustive search approach.
With L and K growing, the **acceleration multiple** grows **exponentially**.

Table: Acceleration multiple of dynamic programming approach

	$L = 3$	$L = 4$	$L = 5$	$L = 6$
$K = 1$	0.96	1.01	1.02	0.97
$K = 2$	2.25	4.82	8.66	23.99
$K = 3$	4.32	9.12	53.12	363.09
$K = 4$	4.22	25.07	273.83	4910.98

Accuracy

Generally, the algorithm can reach global maximum capacity with probability **0.8**.
The algorithm performs better when K is large.
In the worst case, the algorithm can reach 80% of the global maximum capacity with probability 0.9.



Conclusions

Network Properties

1. Only a fraction of relays are necessary to reach maximum capacity, which brings significantly saving.
2. We proposed and proved several properties of network structure.
3. We strictly proved an theorem, which provides an **explicit criterion** to decide the *optimal global network*, not necessary to calculate through the whole network.

Implementation

1. We designed an efficient algorithm that utilizes dynamic programming approach and network properties which can **speed up** the solution **exponentially**.
2. We took advantage of the layered characteristic of the algorithm, and proposed methods that can calculate the new result efficiently (**warm start**) when parameters change.

References

[1] F. Parvaresh and R. H. Etkin, "Efficient capacity computation and power optimization for relay networks," CoRR, vol. abs/1111.4244, 2011.
[2] Ritesh Kolte and Ayfer Özgür. Fast near-optimal subnetwork selection in layered relay networks. In Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on, pages 1238–1245. IEEE, 2014.
[3] Siddhartha Brahma, Ayan Sengupta, and Christina Fragouli. Efficient subnetwork selection in relay networks. In 2014 IEEE International Symposium on Information Theory, pages 1927–1931. Ieee, 2014.