ISE 533 Network Planning with Random Demand (SSN)

•••

Group 5: Yueru Zhang, Xuchen Shao, Yongjun Kim, Zhaoqi Xiao

Content

Problem Statement - Network Planning

Model - Stochastic Linear Programming (SLP)

Algorithm - Stochastic Decomposition (SD)

Simulation - Least Loaded Routing (LLR)

Conclusion

Suggestion

Problem Statement

Applications require higher bandwidth

Increase private lines demand & decrease the accessibility of network capacity

Demand ⬆️ → revenue ⬆️

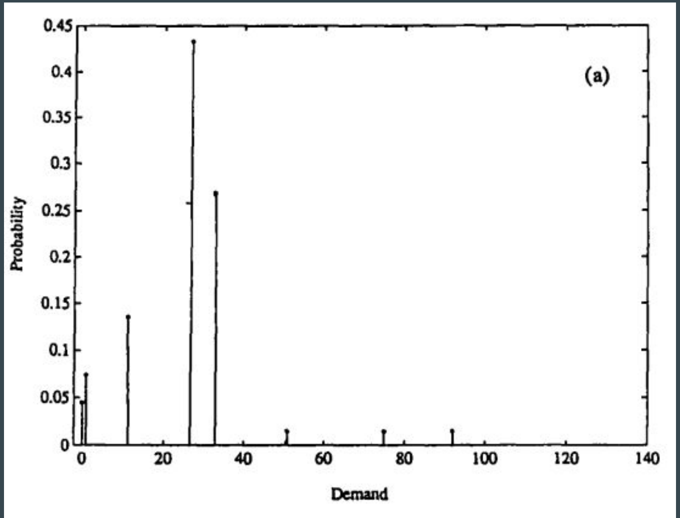
Network providers → bandwidth allocations among network links

→ maximize the number of requests served by the network

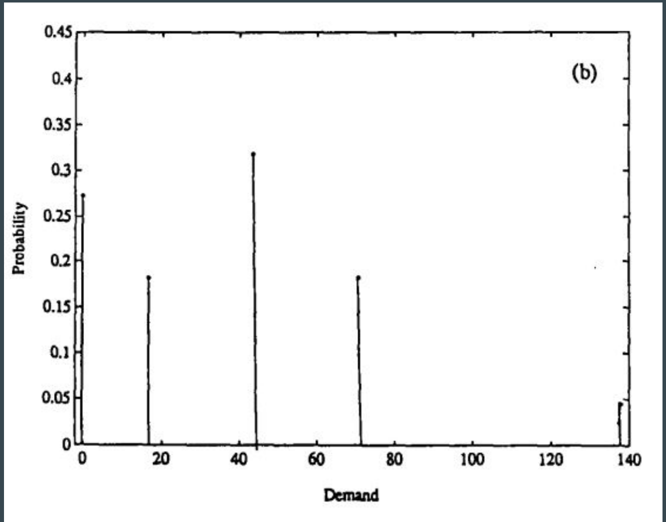
Preparation

The variability in the forecast error increases with the length of the planning period.

6-month demand forecast



2-year demand forecast



→ short planning horizon

Model for Network Planning with Random Demand - SLP

First stage - Provides link capacity

$$\begin{aligned} &\text{MIN} \quad E[h(x, \tilde{d})] \\ &\text{s.t.} \quad \sum_j x_j \le b \\ &\qquad\quad x \ge 0 \end{aligned}$$

n → the number of links that are to be considered for capacity expansion

$\mathbf{x} \in \mathbf{R}^n$ → a vector whose elements x_j denote the amount of additional capacity to be allocated to link j (j = 1 ..... n)

$\tilde{\mathbf{d}} \in \mathbf{R}^m$ → a multi-dimensional **random variable** that represents **demands** associated with the m point-to-point pairs served by the network

b → the total capacity (in units of DS1) that can be allocated throughout the network

Model for Network Planning with Random Demand - SLP

Second stage - Provides a model for the most efficient utilization of network capacity

$$h(x, d) = \text{MIN} \sum_{i=1}^{m} s_i$$

$$\text{s.t.} \quad \sum_{i} \sum_{r \in R(i)} A_{ir} f_{ir} \le x + e$$

$$\sum_{r \in R(i)} f_{ir} + s_i = d_i$$

$$f_{ir} \ge 0,\ s_i \ge 0,\ \forall i,\ r \in R(i)$$

Decision Variables:

f_{ir} ≈ the number of DS1 connections associated with pair i using route $r \in R(i)$

s_i ≈ the number of unserved DS1 requests associated with pair i

i → an index for point-to-point pairs, i = 1,..., m

R(i) → a set of routes that can be used for connections associated with point-to-point pair i

A_{ir}, $r \in R(i)$ → an incidence vector in R^n whose jth element, $a_{ir\,j}$ is 1 if link j belongs to route r and is 0 otherwise

e → denote a vector in R^n that lists the current (embedded) link capacities (DS1) in the network

Algorithm - Stochastic Decomposition

- Reason to pick: 1. The total number of demand scenarios is too large($T = 5^{100}$). The team need to pick a algorithm based on sampling. 2. It has asymptotic properties. 3. It has a statistically motivated stopping rule.
- It provides piecewise linear approximations of $E[h(x, \tilde{d})]$
- For each iteration K of the method, the process of generating a cut involves two steps:
- (a) Using the current iterate x^K and a random sample d^K from demand distribution to solve

$$\boxed{\begin{aligned} h(x, d) = \text{MIN} \sum_{i=1}^{m} s_i \\ \text{s.t.} \quad \sum_{i} \sum_{r \in R(i)} A_{ir} f_{ir} \le x + e \\ \sum_{r \in R(i)} f_{ir} + s_i = d_i \\ f_{ir} \ge 0, s_i \ge 0, \forall i, r \in R(i) \end{aligned}}$$

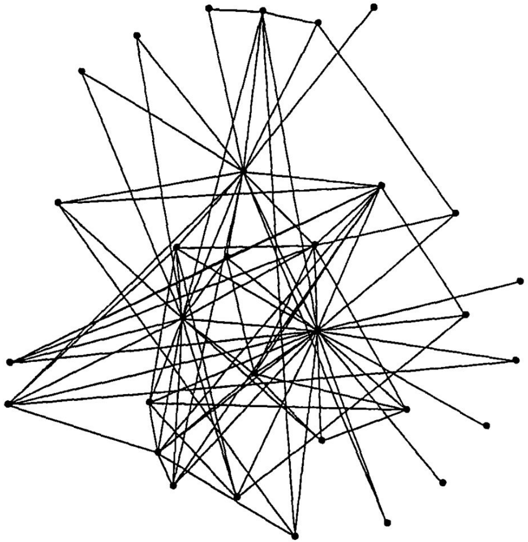
(b)Using previous $\{d^k\}^K_{k=1}$ and x^K to obtain an approximation of the sample mean function $(1/K) \sum_{k=1}^{K} h(x, d^k)$

Algorithm - Stochastic Decomposition

- Since the $\{d^k\}^K_{k=1}$ is random and the algorithm recursively calculates $(1/K)\sum_{k=1}^{K} h(x, d^k)$, the variance of objective function estimates can be reduced as the algorithm proceeds.

Network dimensions.

No. of nodes	31
No. of demand pairs	86
No. of random demands	82
R.V. information	independent
Outcomes per r.v.	5–10
Links	89
No. of routes	706



Stochastic program dimensions.

No. of variables in (**P**)	89
No. of rows in (**P**)	1
No. of variables in (**S**)	706
No. of rows in (**S**)	175

Algorithm - Stochastic Decomposition

- Since the network application is large, the optimality tests were not invoked in the first 1500 iterations. However, the last 400 iterations generally show the same recommended plans. The total computation time is approximately 10500 seconds.

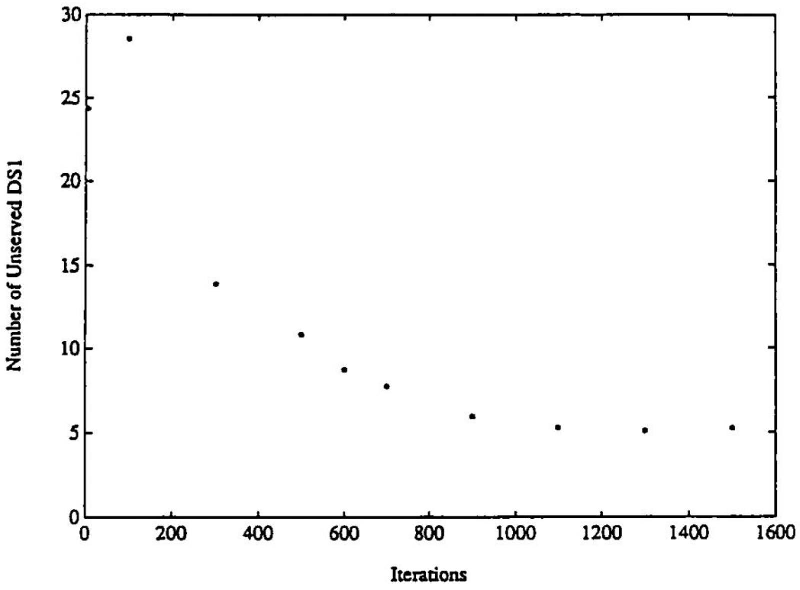


Fig. 3. Estimated objective values.

Observations:

1. poor starting point

2.strict descent in each iteration

is not guaranteed

Simulation - LLR

The routing algorithm - Least Loaded Routing

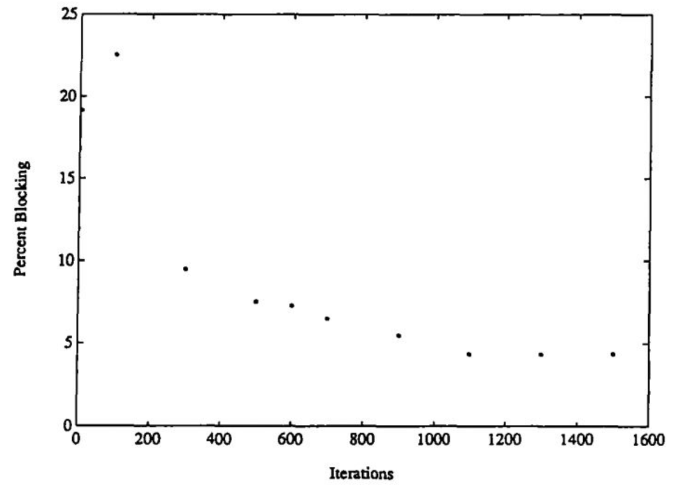
- Potential route with the least carried load
- Direct route (one-link) preferably
- Spare on route: unused capacity - “Trunk reservation”
- How to improve the performance?
    - Rerouting connections if holding time > threshold

Simulation - LLR

- Using percentage of blocked requests
- 750 times of simulation for each plan

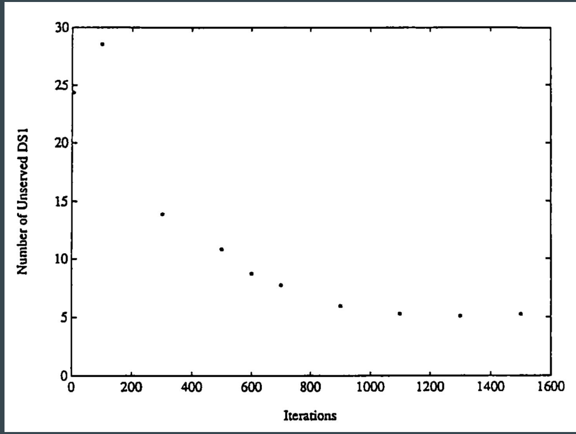
Simulation results.

Iteration	Est. mean	95% conf. interval
1	19.16	[18.29, 20.02]
100	22.54	[21.62, 23.47]
300	9.48	[8.83, 10.13]
500	7.52	[6.98, 8.08]
600	7.26	[6.73, 7.79]
700	6.49	[5.99, 6.99]
900	5.48	[4.98, 5.98]
1100	4.37	[3.9, 4.84]

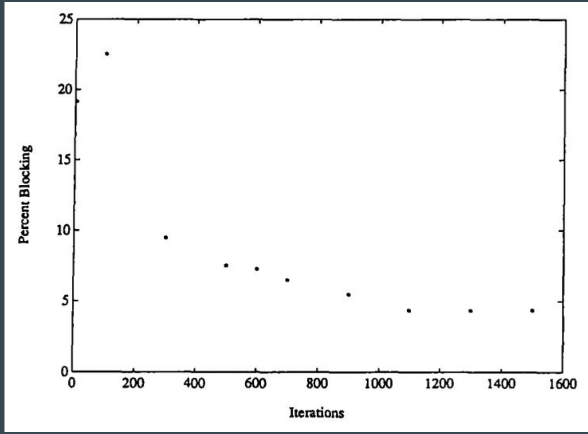


Simulation - LLR

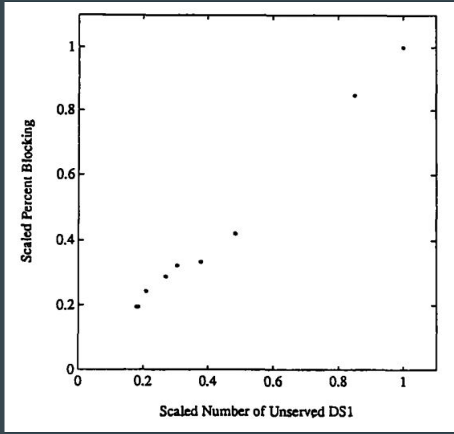
SLP-SD



Simulation-LLR



Model validation



Conclusion

Optimization model ———— Using network flow.

Computational complexity ———— SD algorithm ———— Save computational time.

The two stage SP ———— Static network flow model ———— Get a global optimal plan for capacity allocation.

The LLR ———— Dynamic model ———— Search for optimal solution over each case ————Greedy

Despite two methods are different, they are concomitant. It shows that the solution is reliable.

Improvement

1. The distribution of different pairs' demand should be a little different. We can set a normal distribution with different mean and variance for each pair's demand to better simulate the real life situation.

2. We can set up a penalty for longer route because longer route might cause larger possibilities for signal delay and line fault problem.

3. Improving LLR by setting maximum available rate for those links that can be used to connect more than one pair.

4. Try other simulation methods to see if the solution is still stable or not.

Thank you!