Network Planner Performance

Optimus: Machine where we run "big" NP scenarios
32G RAM, Dual Quad Core, SSD (which didn't survive)
and yet...

Num Nodes	**Grid Size**	**Memory Consumption**	**Run Time**
937	3696	3G	339 minutes
5577	4641	36G	3966 minutes

That last scenario originally took even longer due to excessive paging.

Run Time

Problem:

Profiling revealed intersection test as the dominant time consumer. (Up to 99% as scenario size increases)

For each candidate segment that passes mvMax criteria, check intersection with every segment in network.

If 20k segments pass, and there are 1k segments in network: 20M intersection tests. (each is several steps)

Solution:

Introduce R-Tree to store/search network for intersections.

Now for each candidate segment, there are $log_b(K)$ intersection calls.

b is tree branching factor and K is the number segments in the network.

So for the same scenario above, if branching factor is 2, there are $\frac{1}{10}$th of the intersection tests.

Memory

Problem:

There are $\frac{(n)(n-1)}{2} = \binom{n}{2}$ candidate segments (n is the number of nodes in the scenario)

For 5k nodes, that's $>$ 12M candidate segments

Each candidate segment is a Python object, originally consuming 1k, for a total of 12G

Solution:

Represent candidate segments as a 3xN matrix

uint16, uint16, float32 → node index 1, node index 2, length

8 bytes each

So now the candidate segments for the scenario above consume 96M

Results

Before:

Num Nodes	Grid Size	Memory Consumption	Run Time
937	3696	3G	339 minutes
5577	4641	36G	3966 minutes

After:

Num Nodes	Grid Size	Memory Consumption	Run Time
937	3696	1G	27 minutes
5577	4641	1.2G	276 minutes

Practically, 10X memory and run-time reduction for “large” scenarios

Up to 60X memory reduction within modKruskal

More dramatic improvements as scenario size grows