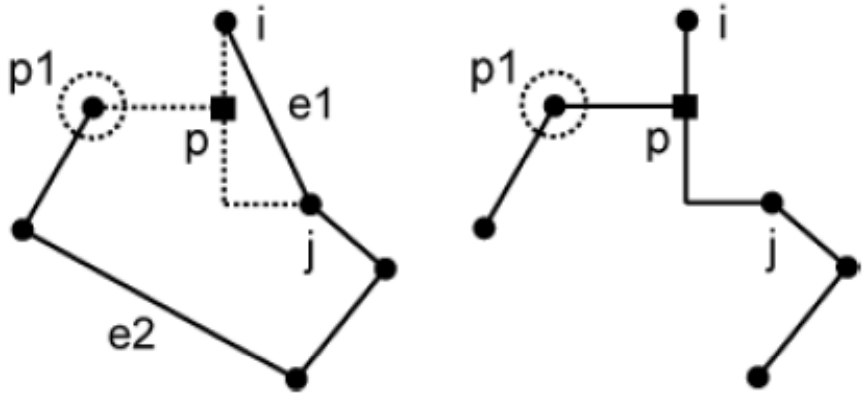
Borah's 1-Steiner Algorithm

James Yang

Algorithm Refresher

- Find EVERY Node / Edge Pair
  - For each pair, find the longest edge from the node to the edge
  - Compute gain for each pair
    - gain = length of longest edge – length from steiner point to node
- Sort the pairs in descending order
- For each pair,
  - Add steiner point (p)
  - Remove longest edge (e2)
  - Remove the edge (e1)
  - Connect all 3 nodes to the steiner point

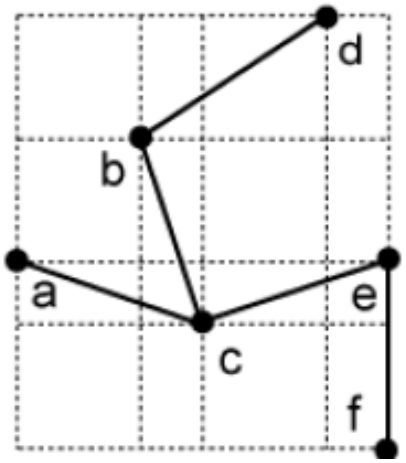


Demo

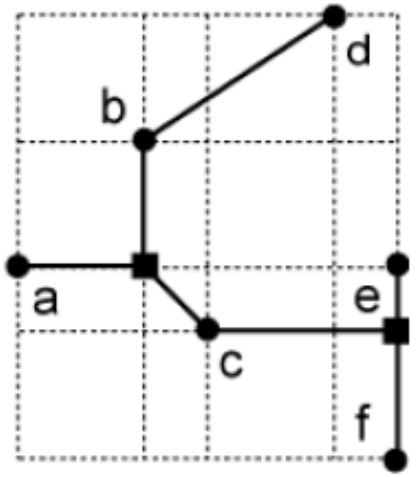
1. Dr. Lim's Example
2. 20 Random Points
3. 100 Random Points
4. 200 Random Points
5. 500 Random Points

Dr. Lim's Example

- Initial



- Final



Gain = 3

Demo (20, 100, 200 and 500 Nodes)

- Journal's Results

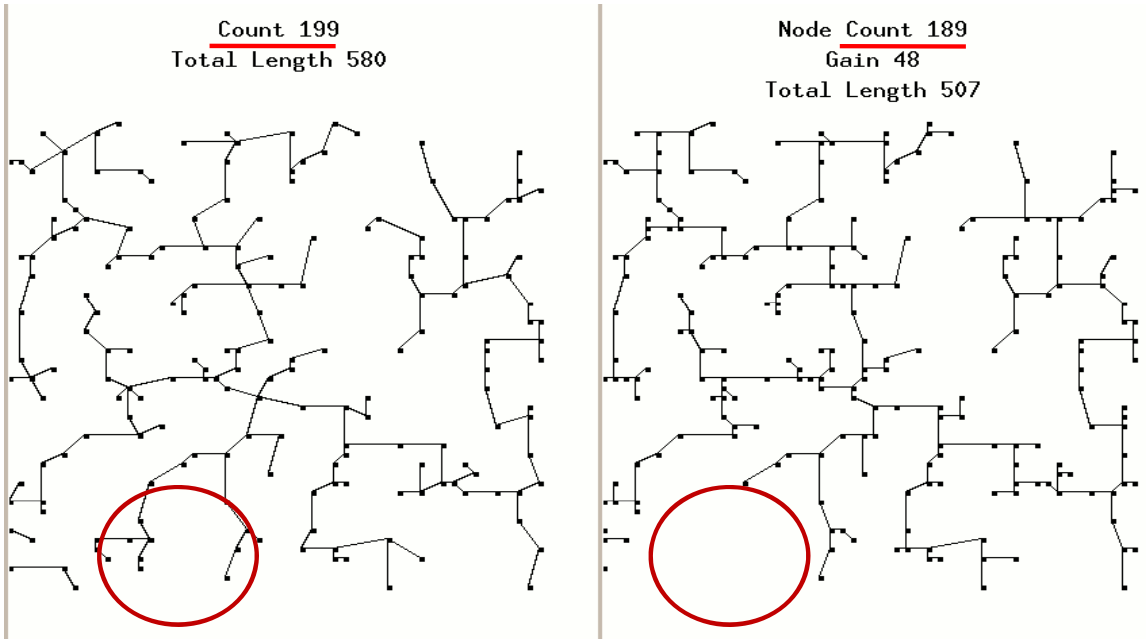
size	Passes Required			Avg. improv./pass (%)				Max. improv./pass (%)			
	Avg	Max	Min	1st	2nd	3rd	4th	1st	2nd	3rd	4th
6	1.41	3	1	8.88	.99	.02	0	19.69	11.21	5.19	0
10	1.7	3	1	9.15	.97	.01	0	15.04	4.65	.47	0
50	2.3	4	1	9.08	1.31	.06	.01	12.69	2.87	.61	.07
100	2.5	5	2	9.28	1.40	.07	.002	11.26	3.12	.49	.18
200	2.7	4	2	9.48	1.36	.07	.001	11.05	2.24	.37	.04
500	3.3	4	3	9.46	1.41	.05	.002	10.39	1.79	.15	.02

Size of net	*Batched 1-Steiner*
5	7.1 *msec*
6	13.3 *msec*
8	32.1 *msec*
10	59.6 *msec*
20	456 *msec*
50	6.53 sec
100	52 sec
200	395 sec
500	>1.5 hrs
1000	-

With 1994 Technology
Probably a Pentium I

Uh oh.

- Missing Nodes



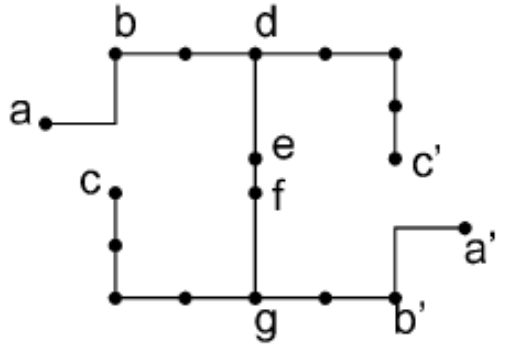
- H. Zhou's Claim

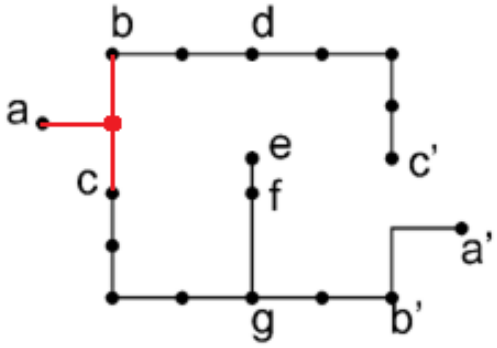
From "Efficient Steiner Tree Construction Based on Spanning Graphs"

  - Borah's Algorithm is "not totally correct"
    - .."we should avoid placing the correctness of an algorithm only on our intuitions"

What can go wrong?

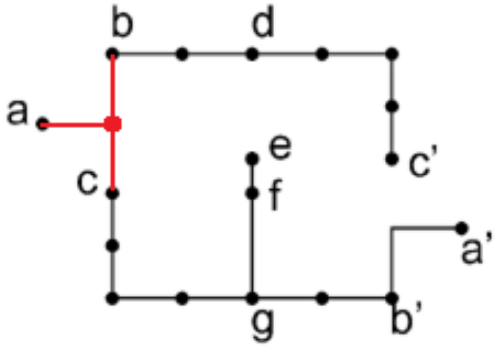
- Starting Pair
  - Edge : a to b
  - Node: c
- Longest Edges
  - d to e
  - f to g
  - We pick the closest edge
    - d to e

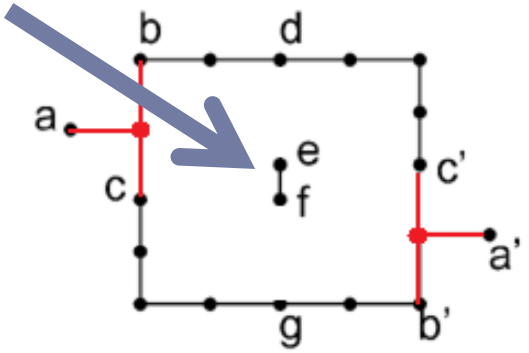




What can go wrong? (continued)

- Next Pair
  - Edge : a' to b'
  - Node: c'
- Longest Edges
  - d to e AND
  - f to g
  - We pick the closest edge for consistency – f to g





My Implementation

- Written in ANSI C and Python (GUI)
- No memory leak (checked using valgrind)
- Uses lots of CPU and Memory
  - CPU at 100% during the execution
  - For >500 nodes, it starts to use page files
- Challenges
  - Bidirectional Graph
  - Lots of book keeping
- Possible Improvements
  - Use an array or matrix based data structure
  - Apply Zhou's modifications to the algorithm to make it "correct"

End

Thank you