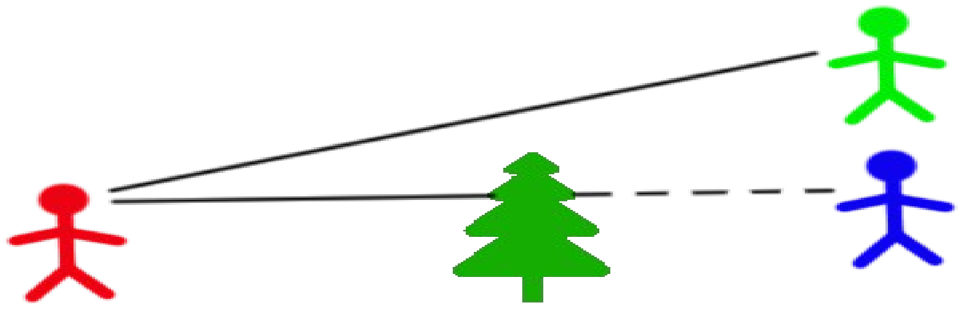
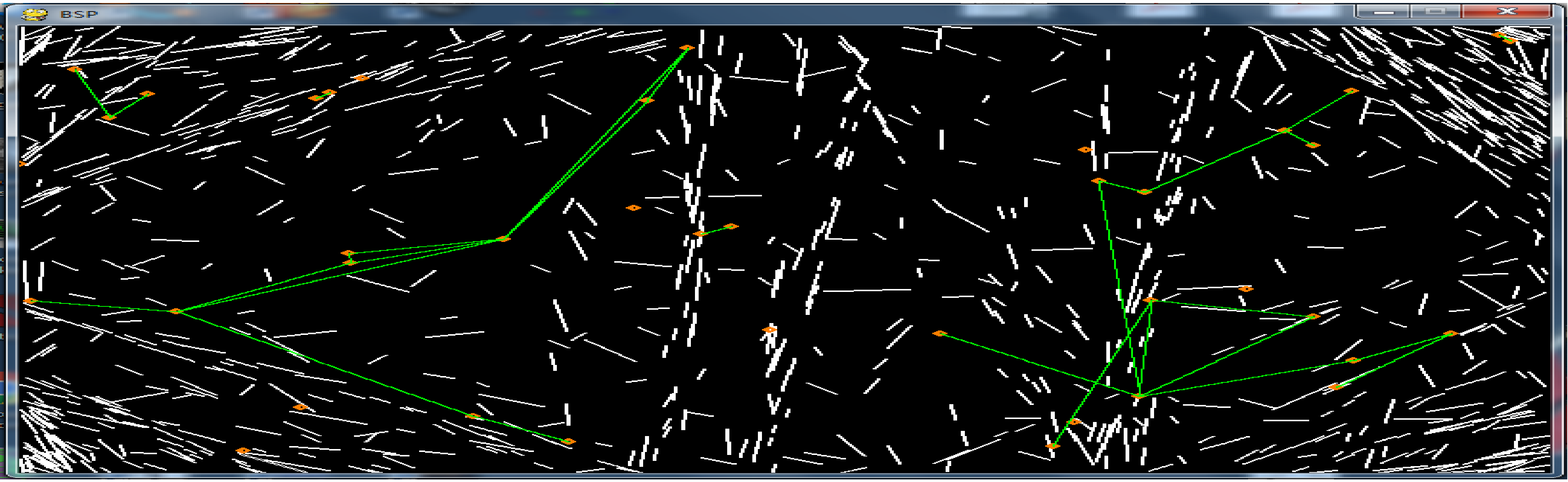
Line of Sight Detection

Ozair Shafiq

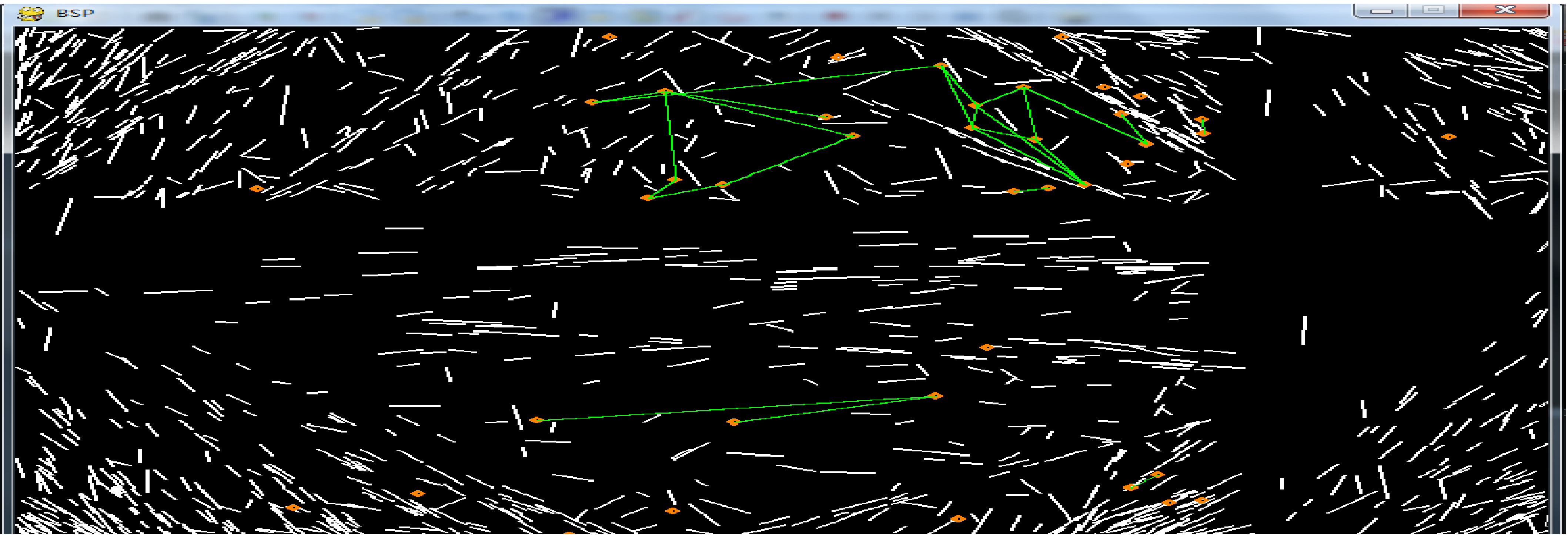


- **Input:** List of non-crossing line segments as obstacles in a 2D environment
- **Preprocessing:** Generate a Binary Space Partitioning tree from the list of obstacles
- **Algorithm:** Determine line of sight between two points by using the binary tree we generated
- **Output:** True/False value for whether there exists Line of Sight between two points
- **Restriction:** Generate a random set of obstacles using unifrom probability distribution
- **General case:** Generate a random set of obstacles using power law probability distribution

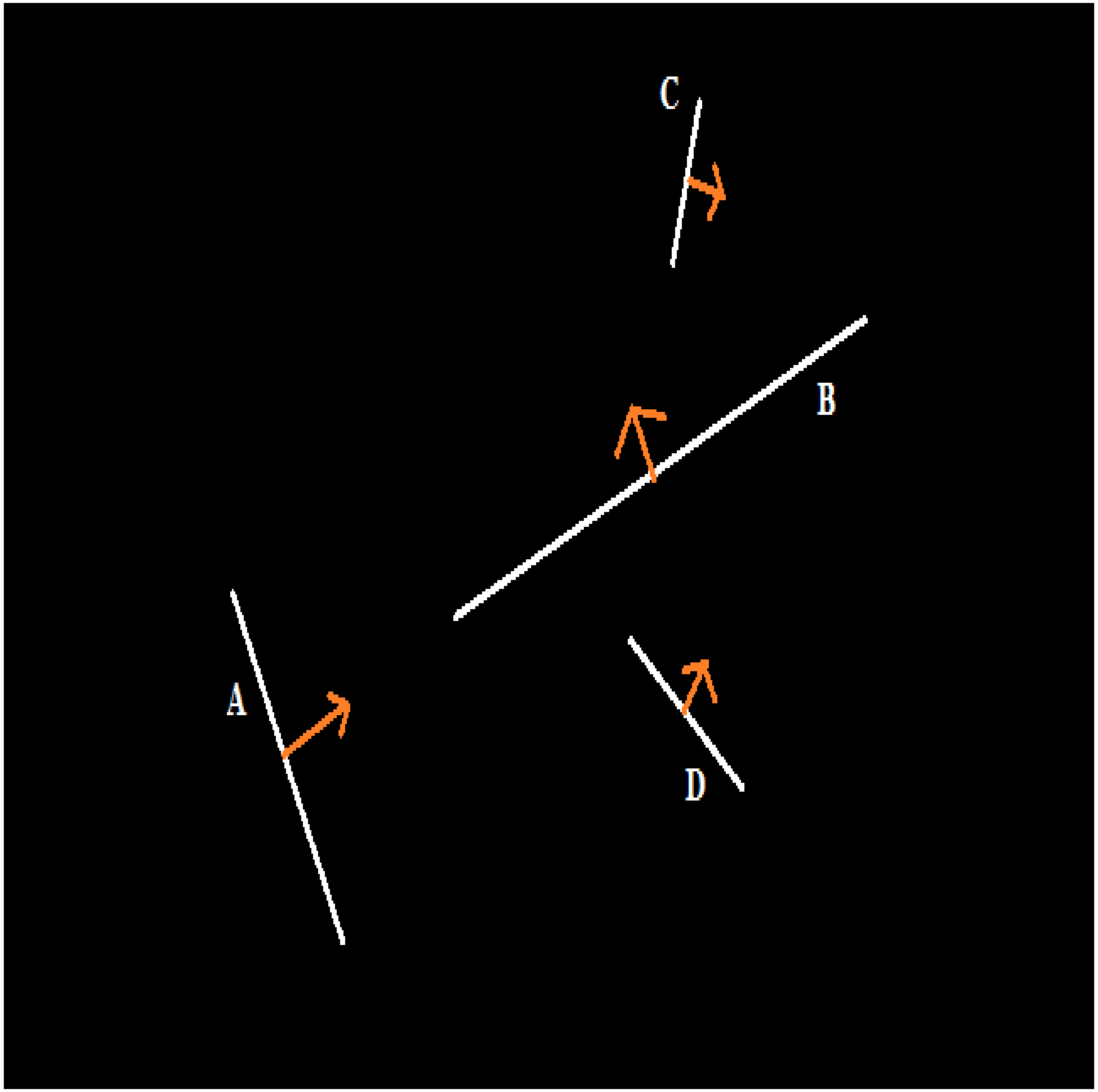
Unifrom



Power Law



Binary Space Partitioning Example



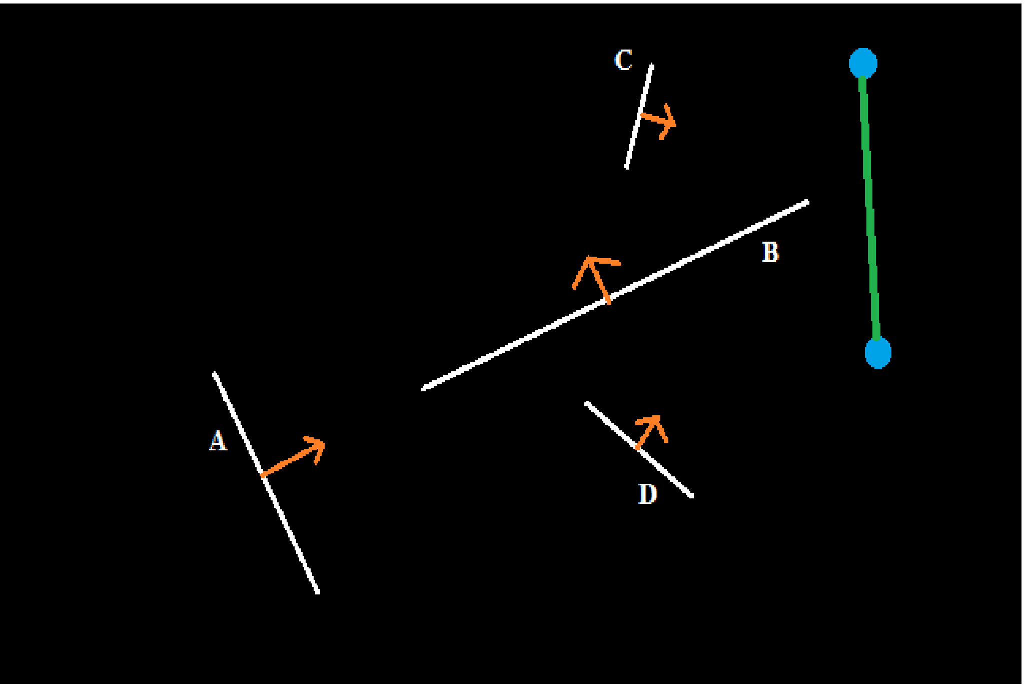
Minimum Split

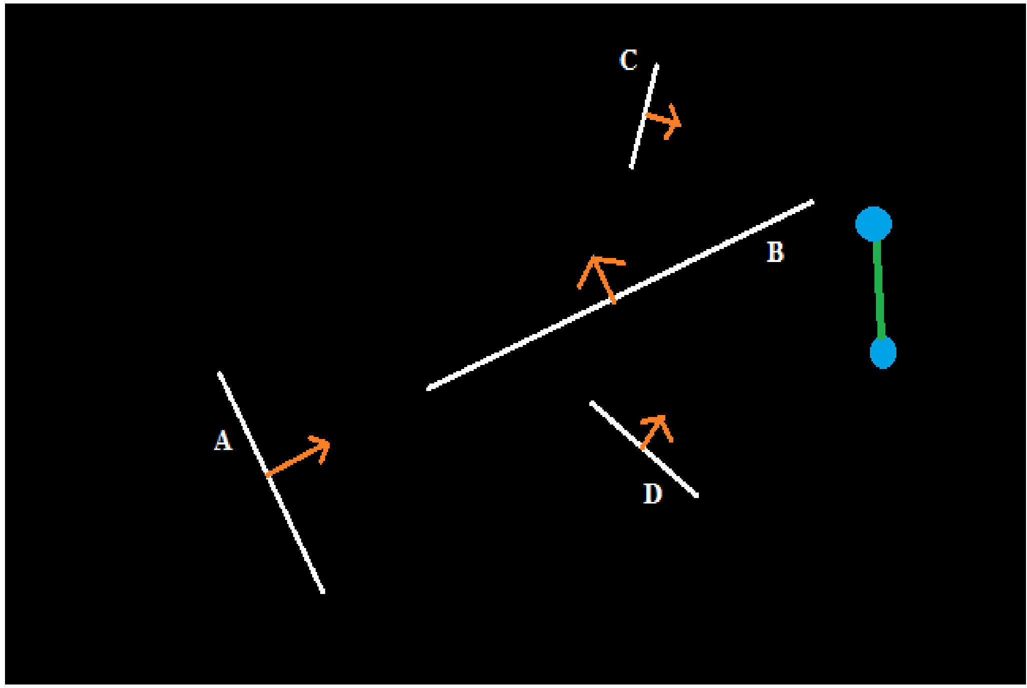
$$\begin{tikzcd}
A \arrow[dr] & & \\
& B \arrow[dl] \arrow[dr] & \\
C & & D
\end{tikzcd}$$

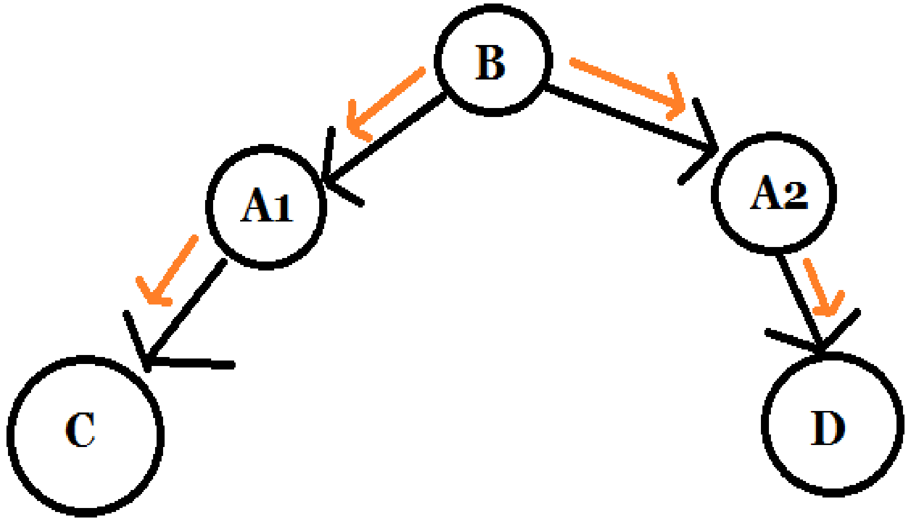
Even Divide

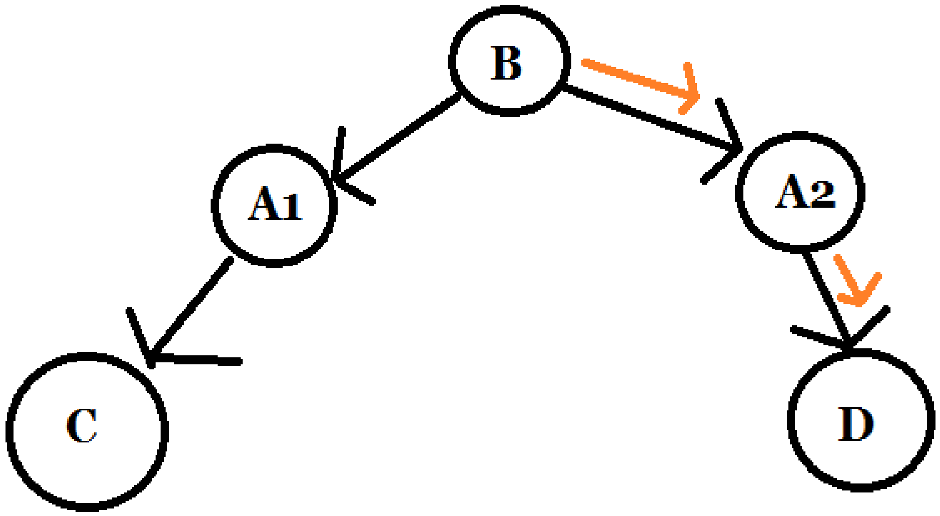
$$\begin{tikzcd}
& & B \arrow[dl] \arrow[dr] & & \\
& A1 \arrow[dl] & & A2 \arrow[dr] & \\
C & & & & D
\end{tikzcd}$$

Binary Tree Traversal

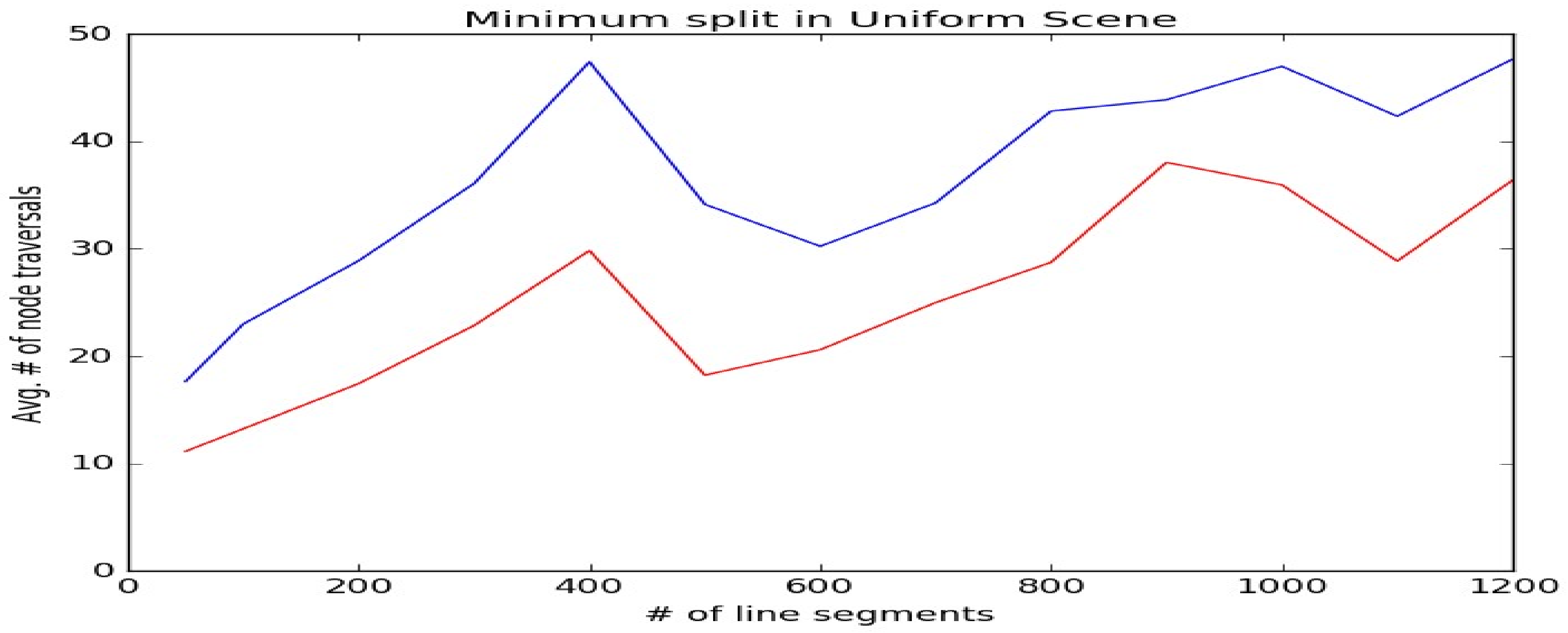


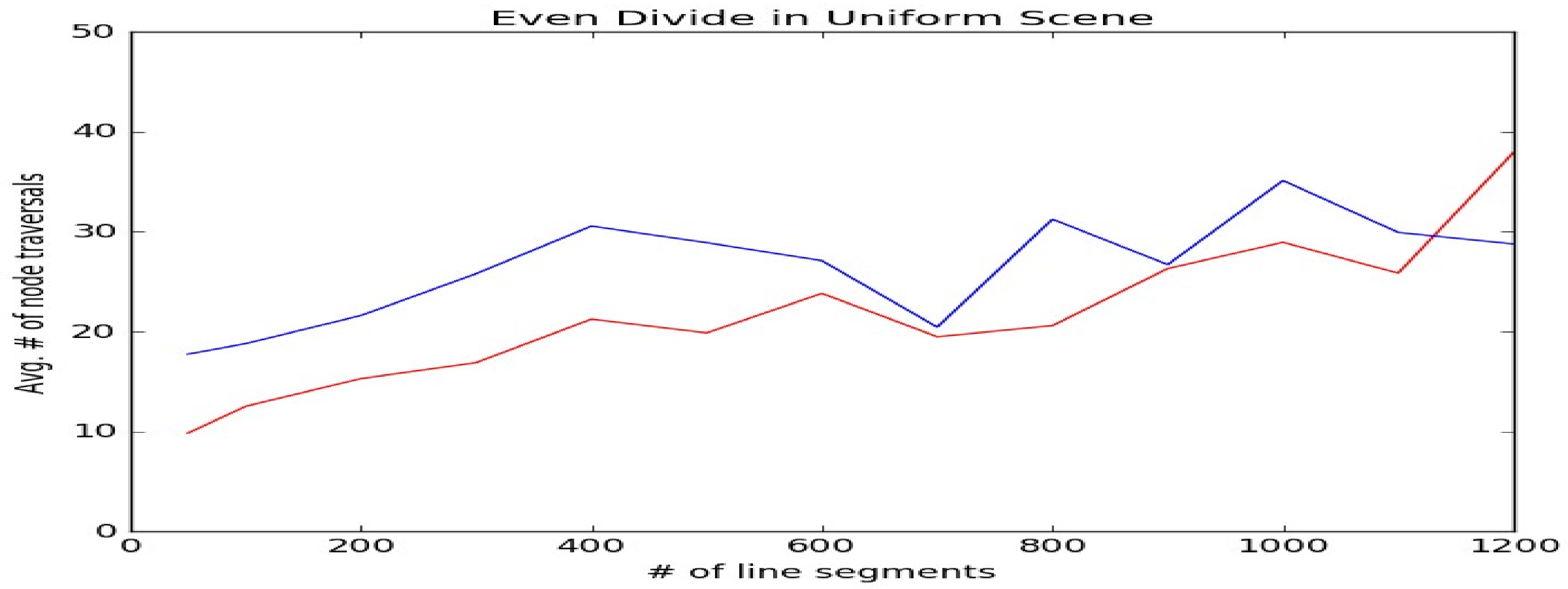


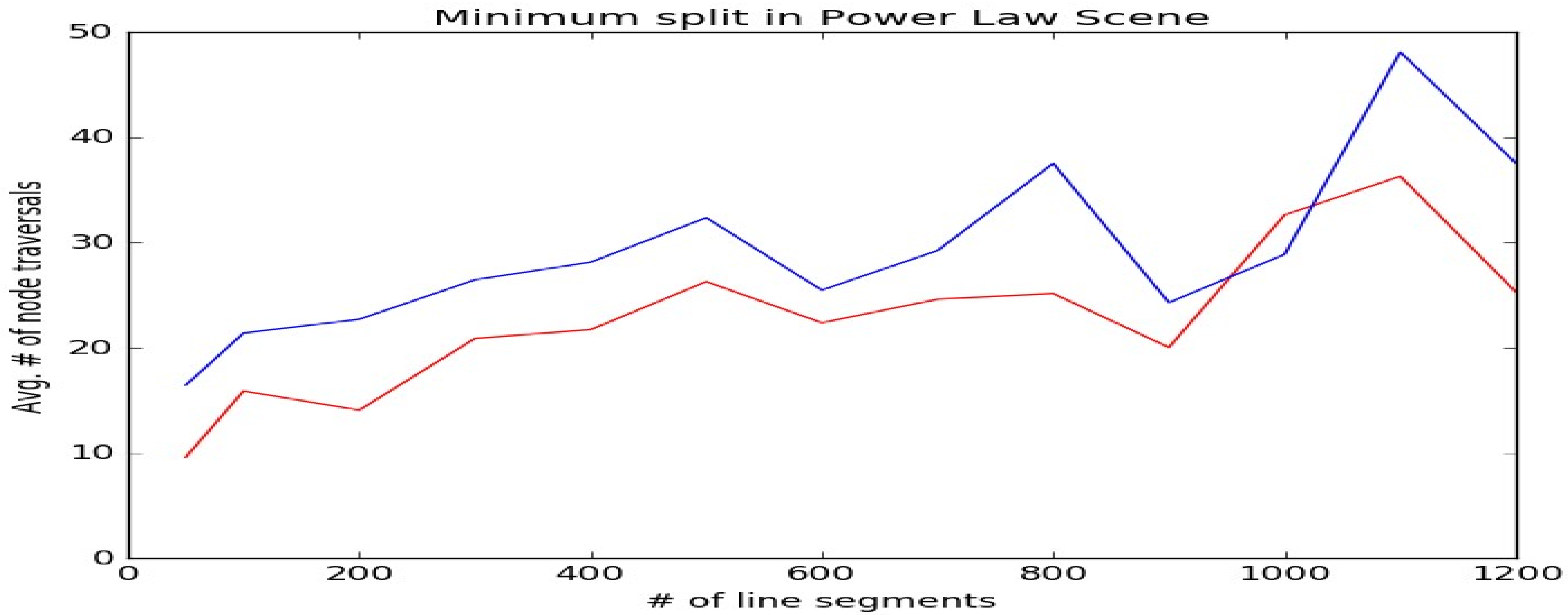


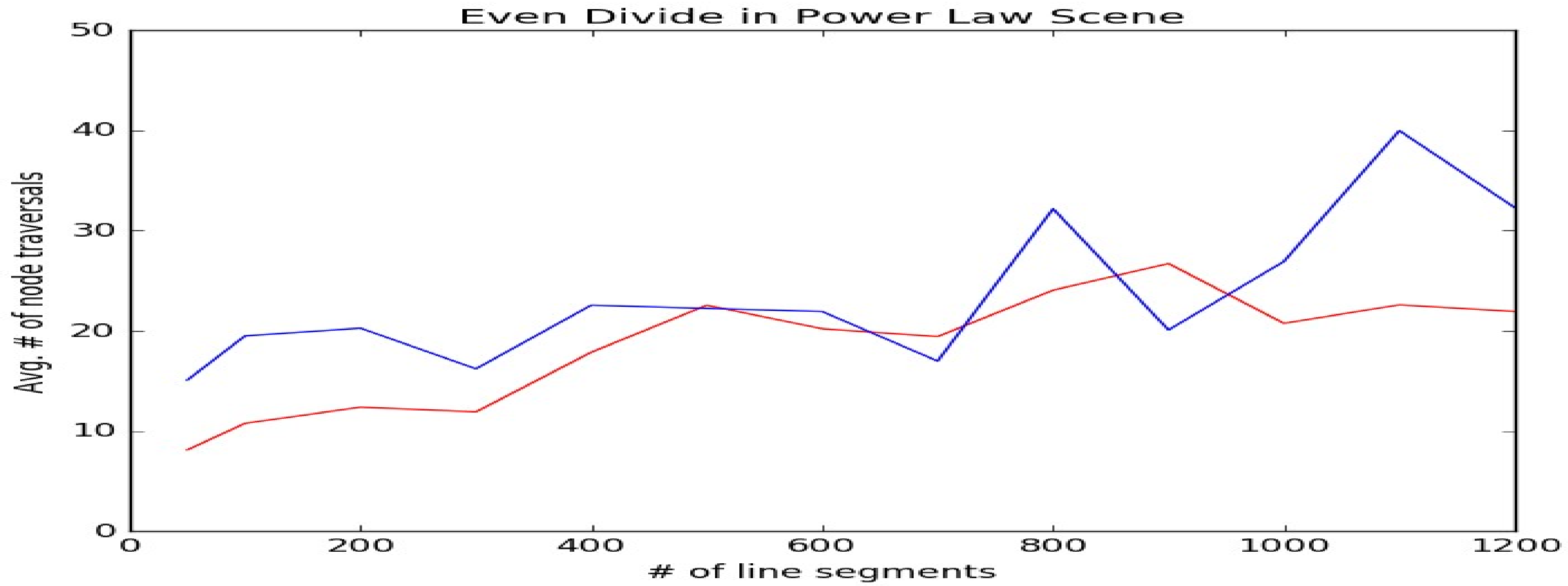


log(m) ≤ O(m) ≤ m, however, we expect to remain close to log(m)


Minimum split in Uniform Scene
Avg. # of node traversals
# of line segments
0
10
20
30
40
50
0
200
400
600
800
1000
1200


Even Divide in Uniform Scene
Avg. # of node traversals
# of line segments
0
10
20
30
40
50
0
200
400
600
800
1000
1200


Minimum split in Power Law Scene
Avg. # of node traversals
# of line segments
0
10
20
30
40
50
0
200
400
600
800
1000
1200


Even Divide in Power Law Scene
Avg. # of node traversals
# of line segments
0
10
20
30
40
50
0
200
400
600
800
1000
1200

Conclusion

- BSP with minimum split performs worse than BSP with even division on average on both cases
- However, BSP with even division takes more time to build and consumes more memory due to spliting line segments