* Mathematical definition of the problem
* Related work (bibliography)
* Description of your solution
* Data sets used
* Analysis of the results and comparison with existing tools
* Conclusions
* Possible directions for future research
* Instructions on how to compile and execute your program
* List of references

# Background

## Types of Tree

### Ordered tree and unordered tree

Drawing unordered trees with common optimization goals, for example, drawing unordered trees with minimum width using level-based approach is NP-complete, as proved by K. Marriott[[1]](#footnote-0)

### Binary tree and general tree

## Optimization Goals

Optimizations goals may conflict with each other. In practice, less important goals compromise to important goals. For example, if we want the tree to be as beautiful as possible, then we may accept larger area or slower time.

### Time

Linear

Polynomial

NP-complete

### Area

We want the area to be as narrow and short as possible.

log(n)

Linear

### Aesthetic Requirements

Different aesthetic requirements are applied to different application scenarios. Some common aesthetic rules are below.

Edges do not interact.

Parent node is centered among child nodes.

Same subtrees have same structures.

## Approaches

According to A. Rusu’s summary[[2]](#footnote-1), there are five approaches.

Level-Based Approach

H-V Approach

Path-Based Approach

Ringed Circular Layout Approach

Separation-Based Approach

# Implementations and Analysis

## Level-Based Approach

Nodes of same depth are located at same horizontal level.

We consider ordered tree.

We will review several algorithms from a historical view based on an excellent review on github[[3]](#footnote-2), D. Knuth’s paper[[4]](#footnote-3), C. Wetherell and A. Shannon’s paper[[5]](#footnote-4), E. Reingold and J. Tilford’s paper[[6]](#footnote-5), J. Walker’s paper[[7]](#footnote-6), and C. Buchheim, M. J Unger, and S. Leipert’s paper[[8]](#footnote-7).

### Knuth’s Algorithm

|  |  |
| --- | --- |
| Tree Type | Binary and general ordered trees |
| Time | O(n) |
| Area | O(n) |
| Aesthetic Outcomes | Outcome 1: Edges do not intersect  Outcome 2: Same level nodes stay at same horizontal line |

### Wetherell and Shannon’s Algorithm 1

|  |  |
| --- | --- |
| Tree Type | Binary and general ordered trees |
| Time | O(n) |
| Area | O(n) |
| Aesthetic Outcomes | Outcome 1: Edges do not intersect  Outcome 2: Same level nodes stay at same horizontal line  Outcome 3: Minimize the width given outcome 1 and outcome 2 |

### Wetherell and Shannon’s Algorithm 2

Adjust location of subtrees at each node

|  |  |
| --- | --- |
| Tree Type | Binary and general ordered trees |
| Time | O(n^2) |
| Area | O(n) |
| Aesthetic Outcomes | Outcome 1: Edges do not intersect  Outcome 2: Same level nodes stay at same horizontal line  Outcome 3: Parent node is centered among child nodes.  Outcome 4: Minimize the width given outcome 1, 2 and 3 |

### Wetherell and Shannon’s Algorithm 3

Use mod to store distance-to-move. Then do one more traversal to adjust the location of each node.

|  |  |
| --- | --- |
| Tree Type | Binary and general ordered trees |
| Time | O(n) |
| Area | O(n) |
| Aesthetic Outcomes | Outcome 1: Edges do not intersect  Outcome 2: Same level nodes stay at same horizontal line  Outcome 3: Parent node is centered among child nodes.  Outcome 4: Minimize the width given outcome 1, 2, and 3 |

### Reingold and Tilford’s Algorithm

|  |  |
| --- | --- |
| Tree Type | Binary and general ordered trees |
| Time | O(n) |
| Area | O(n) |
| Aesthetic Outcomes | Outcome 1: Edges do not intersect  Outcome 2: Same level nodes stay at same horizontal line  Outcome 3: Parent node is centered among child nodes  Outcome 4: Same subtrees should have same geometric structure  Outcome 5: Minimize the width given outcome 1, 2 , 3, and 4 |

We implement a time complexity O(n^2) algorithm on binary ordered tree.

### C. Buchheim et al’s Algorithm

|  |  |
| --- | --- |
| Tree Type | Binary and general ordered trees |
| Time | O(n) |
| Area | O(n) |
| Aesthetic Outcomes | Outcome 1: Edges do not intersect  Outcome 2: Same level nodes stay at same horizontal line  Outcome 3: Parent node is centered among child nodes  Outcome 4: Same subtrees should have same geometric structure  Outcome 5: Child nodes of same parent node are evenly spaced  Outcome 6: Minimize the width given outcome 1, 2 , 3, 4, and 5 |

We have no implementation for this.

## H-V Approach

# Conclusion

Trade off

1. Marriott, Kim, and Peter J. Stuckey. "NP-Completeness of Minimal Width Unordered Tree Layout." Graph Algorithms and Applications 5 (2006): 295-312. Print. [↑](#footnote-ref-0)
2. Rusu, Adrian. "Tree drawing algorithms." (2004). [↑](#footnote-ref-1)
3. https://llimllib.github.io/pymag-trees/ [↑](#footnote-ref-2)
4. Knuth, Donald E. "Optimum binary search trees." Acta informatica 1.1 (1971): 14-25. [↑](#footnote-ref-3)
5. Wetherell, Charles, and Alfred Shannon. "Tidy drawings of trees." IEEE Transactions on software Engineering 5 (1979): 514-520. [↑](#footnote-ref-4)
6. Reingold, Edward M., and John S. Tilford. "Tidier drawings of trees." IEEE Transactions on software Engineering 2 (1981): 223-228. [↑](#footnote-ref-5)
7. Walker, John Q. "A node‐positioning algorithm for general trees." Software: Practice and Experience 20.7 (1990): 685-705. [↑](#footnote-ref-6)
8. Buchheim, Christoph, Michael Jünger, and Sebastian Leipert. "Improving Walker’s algorithm to run in linear time." International Symposium on Graph Drawing. Springer Berlin Heidelberg, 2002. [↑](#footnote-ref-7)