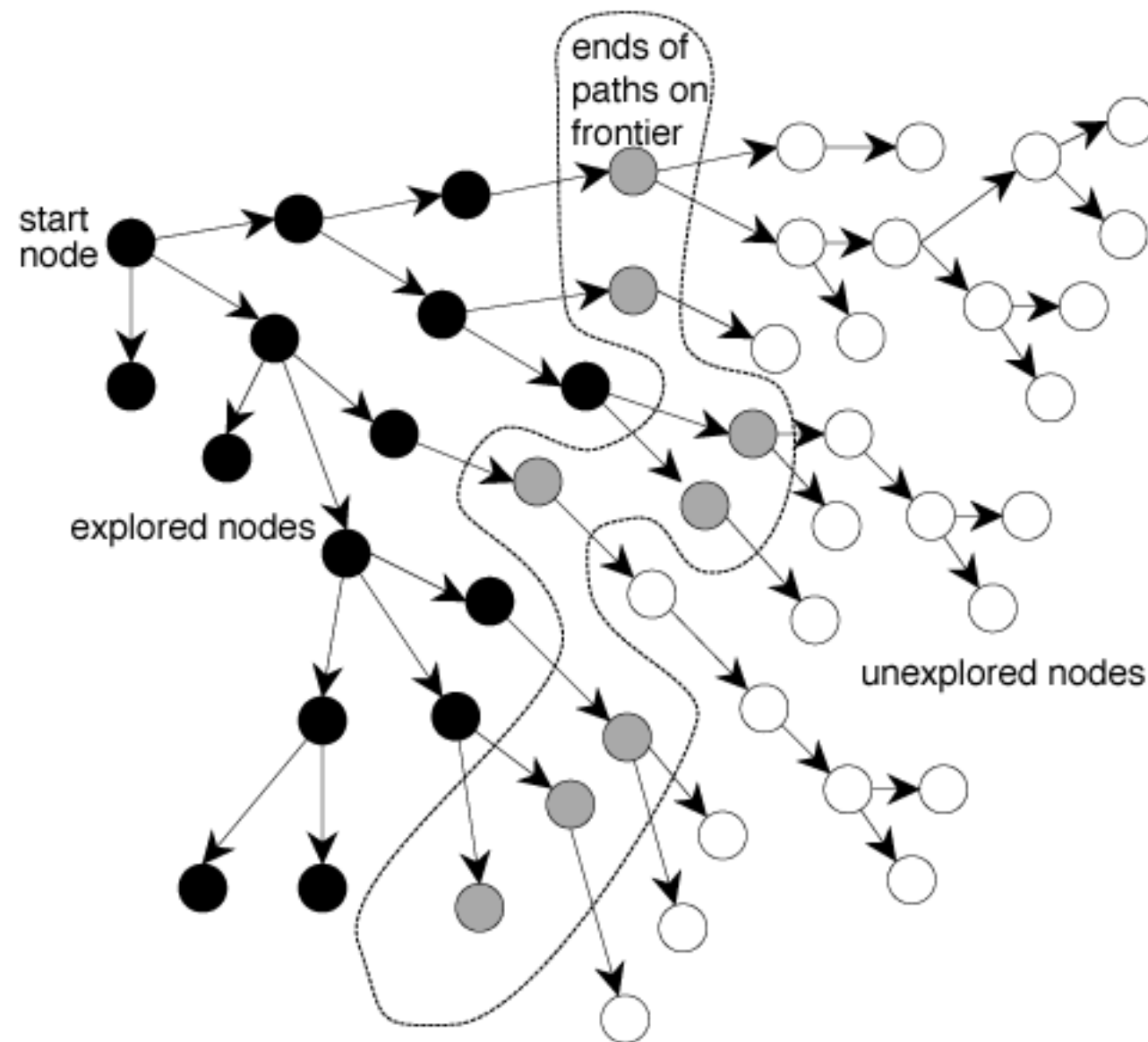


General Tree Search + Hyperheuristics

Wells Lucas Santo



Overview

- What's the General Tree Search
- Current Tree Search Space
- Working with Hyperheuristics
- Next Steps: Exhaustive Search, Evolution, Learning

General Tree Search (GTS)

- Simulate many different tree search algorithms
- Breaks up the tree search into seven generic steps
- Behaviors at each step specified by parameter
- Mix-and-match parameters to run different searches
- Implemented as controller in GVG-AI

Current Search Space

- Designed with **BFS, DFS, A*, MCTS** in mind
- Right now, 1620 possible tree searches
 - Not all combinations may make any sense

```
public class GTSPParams {  
    public static enum EXPLORATION    { FIRST, HIGH, LOW, LAST, UCT };  
    public static enum EXPANSION      { ALL, FIRST, RANDOM };  
    public static enum REMOVAL        { YES, NO };  
    public static enum SIMULATION     { RANDOM, NONE };  
    public static enum EVALUATION     { POINTS, WIN, WINLOSSPOINTS, DISTANCE };  
    public static enum BACKPROPAGATION { HIGH, LOW, INCREMENT };  
    public static enum SELECTION      { HIGH, LOW, MOSTVISITS };  
}
```

Working With Hyperheuristics

- GTS allows mix-and-match execution based on params
- Allows for wide variety of params to select from
- Hyperheuristic: Use heuristic to select which param to use for a particular game
 - Can use evolutionary computation or machine learning to get the algorithm to play “optimal” algorithm

Next Steps

Immediate Steps:

- Fix the code (the searches aren't running how they're supposed to)
- Develop small GVG-AI stages to easily test the algorithms
- Exhaustive search over the tree search space

Further Steps:

- Evolutionary computation to evolve the agent over multiple iterations of the game
- ML to learn what parameters work best with games of particular features

The Dream:

- Allow for in-game parameter switching based on situation (online hyperheuristic selection/generation)