Searching algorithms

- Uninformed search : you don't know end state
- informed huristic search:
- local search

propositional predicate logic

Book 7.1 – 7.6 8.1 – 8.3

Breadth-First search — Exponetial memory
└ Que

Depth-First Search

Dijkstra' Alogrithm

Depth-Limited search — why we limited the search
DFS

Bidirectional search –
You know where the goal is
parallel Iterative Deeping searches

terminate

Uninformed search
Fast network search penrate system

Informed search baseline
hints + bruteforce search
hint is heuristic function h(n)

Euclidean distance

Greedy best search
shortest path not garentteed
fewest edges not garnteed.

A* Algorithm heuristic
F=G+H cost of node
cost required to reach the goal

Mark all source as unvistied

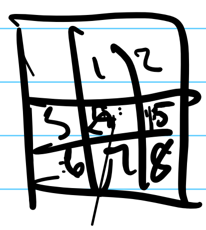
if tie go with huristics

min heap priority que

* Heuristics

Not all heuristics

could combining heuristics into meta h's



number of tiles

- never overestimate
  - L we never know that is the best solution

also be consistance

Each estimate must be less than or equal to true cost neighbor

Effective branching factor

Solve for b^* $N+1 = 1 + b^* + (b^*)^2 + \ldots + (b^*)^d$

$= 1.92$

Generating Heuristics

try to create relaxted problem

fewer constraint

A complex problem into

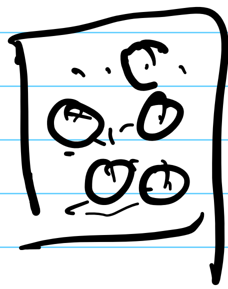
many smaller less constraint problem.

Composite Heuristics

aggregate $h(n) = \max\{h_1(n), h_2(n),$
$\ldots, h_K(n)\}$

possible downside is the time it takes on each heuristic

if the graph each pair of vertices



group them as landmark

$O(|V|^2)$ space

$O(|E|^3)$ time to compute
Fessible for 10,000 vertices, not feasible for 10 million

choose a subset of vertices lable them as landmarks
This is not admissible

this means its not accurate.

not guaranteed find actual shortest path

time saved calculation likely offsets the real time

Project 1

write program that takes JSON graph

each object is a vertex with name and collection of adjacent vertices path cost.

Your Program has 60 seconds to precompute an internal prepresention of the graph

Find landmarks, calculate shortest path.

Some of this time may be spent

analyzing the graph structure
You can customise how detailed
this proces is

The user will enter a relative path
to a file containing a number
of sourse, destination
length of path taken
the time it took calculate
the path
max me 1GB

Overview

These notes cover various search algorithms, their characteristics, heuristics, and implementation strategies in graph and path-finding problems.

Types of Search Algorithms

1. Uninformed Search

- No knowledge of end state
- Explores search space without specific guidance
- Examples:
  - Breadth-First Search (BFS)
    - Exponential memory usage
  - Depth-First Search (DFS)
  - Depth-Limited Search

2. Informed Search

- Uses heuristics to guide search
- Knows goal state or has additional information

Search Strategies

Breadth-First Search

- Explores all neighboring nodes before moving to next level
- High memory requirements

Depth-First Search

- Explores as far as possible along each branch before backtracking

Bidirectional Search

- Searches from both start and goal states simultaneously

Dijkstra's Algorithm

- Finds shortest path in weighted graphs

Heuristic Search Techniques

Greedy Best Search

- Selects path with lowest estimated cost
- Not guaranteed to find optimal solution

A* Algorithm

- Uses cost function: F = G + H
  - G: Cost to reach current node
  - H: Estimated cost to goal
- Uses heuristics like:
  - Euclidean distance
  - Manhattan distance

Heuristic Properties

Characteristics of Good Heuristics

- Never overestimate actual cost
- Consistent and admissible
- Each estimate must be ≤ true cost of neighbor

Generating Heuristics

- Create relaxed problems with fewer constraints
- Composite Heuristics: max{h1(n), h2(n), ..., hp(n)}

Landmark Heuristics

- Choose subset of vertices as landmarks
- Pros:
- Time saved in calculations
- Cons:
- Not always admissible
- Less accurate
- Increased computational complexity

Project Guidelines

1. Parse JSON graph
2. Precompute graph representation
3. Find landmarks
4. Calculate shortest paths

- Time Constraint: 60 seconds for preprocessing

Additional Considerations

- Effective heuristics balance:
- Computational efficiency
- Solution accuracy
- Not all heuristics are equally effective
- Combining heuristics can create meta-heuristics