A* Search

- Special case of best-first search
  - Best is by $f$-cost, where $f(s) = g(s) + h(s)$
  - Estimates total path cost through a node to the goal
- If heuristic is consistent, $f$-costs will be monotonically non-decreasing
A* search

• Complete?
• Optimal?
• Time complexity?
• Space complexity?

• Can we do better than A*?

Iterative-Deepening A*

• Perform depth-first iterative deepening on $f$-costs instead of $g$-costs
  • How do we update the bounds?
  • How do we get our initial bound?

Assume heuristic is distance from leaves
IDA*

• Complete?
• Optimal?
• Time complexity?
• Space complexity?

Where do heuristics come from?

• Exact solution to relaxed version of original problem
• Relax the constraints in the original problem to make it easier to solve
• Use solution as heuristic in original problem

Heuristics for pathfinding & tsp

• Pathfinding
  • Normally constrained to move on grid/graph
  • Cannot move through obstacles
  • Relax by allowing straight-line movement
• Traveling Salesman Problem
  • Must visit all cities in a tour
  • Relax by visiting all cities in minimum spanning tree
Weighted A* \[f = g + 10 \cdot h\]

A* with better heuristic

Optimized A*

Optimized A*
Heuristics for sliding-tile puzzle

- What heuristic would you use for the sliding-tile puzzle?
  - Manhattan distance

- Domain abstraction for pattern databases
  - See Figure 3.30
Heuristics for pathfinding & tsp

• Pathfinding
  • Normally constrained to move on grid/graph
  • Cannot move through obstacles
  • Relax by allowing straight-line movement
• Traveling Salesman Problem
  • Must visit all cities in a tour
  • Relax by visiting all cities in minimum spanning tree

Heuristics for sliding-tile puzzle

• What heuristic would you use for the sliding-tile puzzle?
  • Manhattan distance

• Domain abstraction for pattern databases
  • See Figure 3.30
Local search

- Not all problems require a path as output
- Examples?

N-queens problem

- Place \( n \) queens on a chess board in a manner such that no queen can capture another queen in a single move
- Possible problem defns?

Homework

- 3.14(a) DFS always expands as many states as A*.
- 3.14(b) \( h(n) = 0 \) is an admissible heuristic
- 3.14(c) A* is of no use in robotics with continuous actions/states
- 3.14(d) BFS is complete even with 0-cost actions
- 3.14(e) If a rook can move across the board in one step, is Manhattan distance admissible?
N-queens problem

- Place $n$ queens on a chess board in a manner such that no queen can capture another queen in a single move
- Possible problem defns?

Local search

- Chapter 3 algorithms all find paths
- If a path is not required, there is more flexibility in the types of algorithms that can be considered

Local Search

- Don’t necessarily need a open/closed list
  - Don’t need an order to visit states
  - Don’t (usually) need to worry about duplicate states
- State space might be infinite
- May only care about solution quality
  - No goal test or path cost
  - Can consider “landscape” of objective function

How would you solve this problem?
Hill Climbing / Greedy local search

- Find best neighbor of current state
- Move to neighbor
- Repeat
- What is the best neighbor?

Example: Local search in 8-queens problem

Failure

- What does it mean for hill-climbing to fail?
- What should we do when hill-climbing fails?
Variations on simple hill climbing

- Act stochastically
- Don’t generate all successors
- Restart from a random state

What about SAT?

- Satisfiability
  - Find set of variables to satisfy a boolean formula
  - \((x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (x_1 \lor x_2 \lor \neg x_3)\)
  - WalkSat (http://www.cs.rochester.edu/u/kautz/walksat/)

Simulated Annealing

- Combine hill-climbing with random walks
  - Choose a random move
    - If it’s better, always follow it
  - Otherwise, according to current temperature, randomly choose whether to take move

Genetic Algorithms

- Genetic algorithms attempt to simulate populations of creatures in order to solve problems
- Several variations:
  - Describe creatures with a “genome”
    - Genome is “executed” to find fitness
  - Creatures directly represent solutions
    - Fitness is evaluated directly
Genetic Algorithms

- Start with population of individuals
- Instead of taking local actions, the representations “breed” leading to:
  - Cross-over between representations
  - Random mutation of states
- Breeding only occurs between:
  - Best individuals from population
  - Or randomly according to fitness

8-queen example

- What is the genome / representation?
- What is the fitness function?
- How does representation matter?

Homework for next class: 5.18

Optional Exercise: Modified problem 4.3
Do not implement; just describe:

- 4.3 (a) How would you use hill-climbing to solve TSPs?
- 4.3 (b) How would you use genetic algorithms to solve a TSP?