Midterm Notes

Thursday, November 21, 2024 4:39 PM

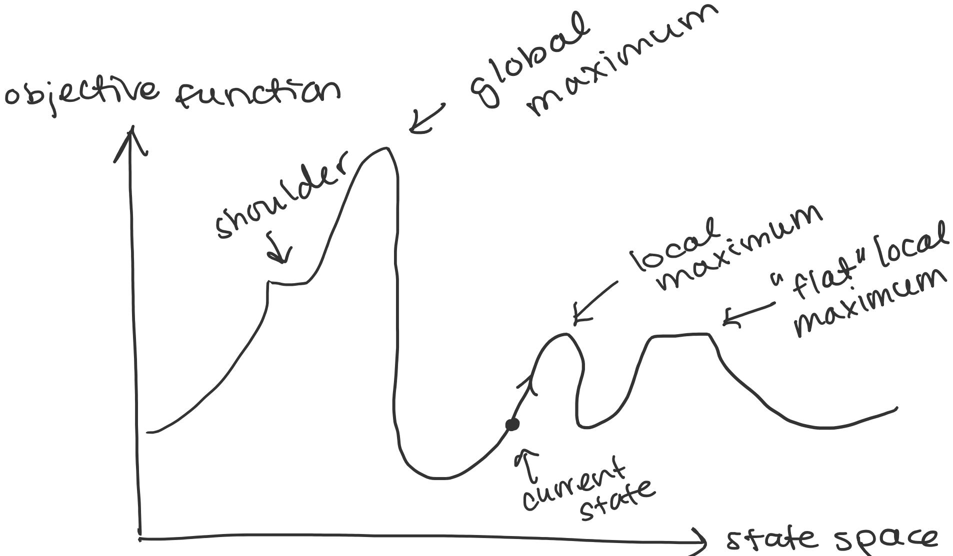
- General notes
  - Can use round(answer,6)
  - Order of importance for studying:
    - Optimization - random algorithms
    - CSP
    - Game playing
    - Bayes Net
    - Search
    - Probability

- Local search and optimization problems
  - Deterministic v non deterministic (page 63)
    - deterministic: if the next state of the environment is completely determined by the current state
      - Agent shouldn't have to worry about uncertainty in a fully observable deterministic environment
      - Vacuum world
    - non-deterministic: no dependency on current state - could have a different result every time - predicting not really possible
      - It environment is only partially observable it could appear nondeterministic
      - Most real situations are nondeterministic - impossible to keep track of all the un observable aspects
      - Taxi driving - can't predict traffic
  - Stochastic v nondeterministic
    - Sometimes considered synonymous
    - Stochastic = a model of the environment is stochastic if it explicitly deals with probabilities
    - non deterministic = if the possibilities are listed without being quantified (chance of rain tomorrow)
  - Discrete v. Continuous states
    - Applies to state of environment, the way time is handled,

precepts and actions of the agent
        - Discrete = chess = finite state
        - Continuous = taxi driving
            - Continuous state and continuous time problem
- Gradient descent → if elevation corresponds to cost, then the aim is to find the lowest valley- a global minimum (p. 129)
- Simulated annealing - stochastic local search
    - Try to combine hill climbing with a random walk in a way that yields both efficiency and completeness
    - Minimizing cost → gradient descent
    - Start by shaking hard and then gradually reduce the intensity of the shaking
    - Instead of picking the best more, however, it picks a random move
    - If the move improves the situation, it is always accepted
    - otherwise, the algorithm accepts the more with some probability less than- probability decreases exponentially with the "badness" of the move
    - Bad moves are more likely to be allowed at the start when t is high and they become more unlikely as t decreases
    - All the probability is concentrated on the global maxima which the algorithm will find with probability approaching 1
    - Factory scheduling and other large-scale optimization tasks
- Evolutionary algorithms
    - Can be seen as variants of stochastic beam search that are explicitly motivated by the by metaphor of natural selection in biology
    - Stochastic beam search - hill climbing- chooses successors with probability proportional to the successor's value - thus increasing diversity
    - Genetic algorithms (p. 137)
        - A method for solving both constrained and unconstrained optimization problems that is based on natural selection
        - Mixing number
        - Selection process
        - crossover point
        - mutation rate
        - Next gen sculling

            - Similar to stochastic beam search but with the addition of the crossover operation

- Adversarial search
    - Iterative deepening
    - When you want to keep depth first search from wandering down an infinite path
    - Supply a depth limit - treat all nodes at the depth as is they had no successors
    - Can make a poor choice for depth limit and fail to reach a solution
    - can't keep it from wasting time on redundant paths but can eliminate cycles
    - When you can't find a depth limit → iterative deepening search. (99)
    - iterative deepening search solves the problem by picking a good value for L by trying all values
        - Each iteration of iterative deepening search generates a new level - doesn't store all nodes in memory while iterative deepening does it by repeating the previous levels - thereby saving memory at the cost of more time
        - Preferred when state space is larger than can fit in memory and the depth of the solution is not known
- Structured CSP:
    - Constraint satisfaction problem
- A* search
    - Admissible is the key part here
- UCS (p95)
    - Dijkstra
    - Spreads out in Waves of uniform path cost instead of by uniform depth
    - Algorithm tests for goals only when it expands a node not when it generates a node
    - If we check for a goal upon generating a node rather than wen expanding the lowest- cost node tenure would have returned a higher - cost path
    - We want to prioritize exploring paths of lower cost before higher cost
    - First solution we find will have a cost that is at least as low as the cost of any other node in the frontier
    - Considers all paths systematically in order of increasing cost never getting caught going down a single infinite path



one dimensional state space landscape
elevation corresponds to the objective function
aim is to find global maximum (p. 129)