

Lesson 10 (Search)

Optimal: Guaranteed to find the best path/shortest path

Complete: Guaranteed to find the goal

Breadth-first search: Optimal and Complete. frontier search space is 2^n in binary search tree. Expands closest first.

Depth-first search: Not optimal/ not complete. frontier search space is n . Expands farthest first, then back propagate.

Cheapest search (uniform cost search): Optimal and Complete. similar to bfs but considers weighted cost of the paths.

Greedy Search: Uses knowledge of the distance toward the goal. Searches for the next node closest to the goal. Is not optimal because there can be obstacles that make the search go a round-about path.

A* search: combining elements of Greedy and uniform cost search.

- Uses the minimum of $f = g + h$ where g is path cost and h is estimated distance to the goal.

- g helps keep the path short, and h keeps focused on the goal.

- Not optimal depending on the heuristic h . In order to be optimal, h should be admissible/optimistic/or never overestimate the true cost of reaching the goal.

- An admissible heuristic can be generated by relaxing the problem (removing constraints of the problem). Taking the max of all the generated admissible heuristic can give an optimal heuristic.

- Solving problems with search works best when:

- fully observable

- set of actions are known

- discrete: finite number of actions

- deterministic: results of an action is known

- Static: only our action changes the world

Lesson 11 (Simulated Annealing)

- Random restart is a good way to escape the local maximum when trying to find the global maximum.

- Simulated Annealing is a great way to escape local optimums with just the right amount of randomness so that it doesn't overstep and oscillate around an optimum. It starts with big 'Temperature', then slowly reduce the 'Temperature' as it will converge to global maximum. Lower temperature will cause less change in randomization of selecting the next solution. The randomness is added in when a probability is calculated for selecting a potential worse solution in hill climbing, which will allow the agent to explore more of the search space and escape local optimums. This is a probabilistic technique used for approximating global optimum when search space is discrete.

- Local Beam Search takes in the information of neighbors.

- Genetic Algorithm is used to take the best parents based on a fitness function to carry

on the traits of the those parents.

-Mutation is randomization in genetic algorithm.

Lesson 12 (Constraint Satisfaction)

Backtracking search: Search down the tree and backtrack one level up when there is a dead-end. There are different ways to optimize backtracking:

-Least Constraining value: Choose the value that constrains the future moves the least.

-Minimum Remaining Value(MRV): Choose the value with the fewest remaining legal moves. (Trying the most obvious choices)

-Degree Heuristic: In cases of tie for MRV, choose the variable that has the most constraints surrounding it. (Try to solve the more complex variable earlier than later)

Forward Checking: An improved version of backtracking. When choosing a value for a variable, check to see if this will end up creating no legal moves for one of the other variables in the future.

-Iterative algorithms are good at solving problems that have many solutions.

Lesson 13 (Logic and Reasoning)

$P \rightarrow Q$ is only false if P is true and Q is false

$P \leftrightarrow Q$ is only true if both of them are equivalent

$P \vee Q$ is true if either P or Q is true

$P \wedge Q$ is true if both P and Q is true

A 'sentence' is some combination of the above logical statements.

A 'model' is all the possible values for a set of propositional variables.

Truth tables are used to map out the model and all the possible outcome given a set of sentences.

A 'valid' sentence is one that is true in every possible model.

A 'satisfiable' sentence is one that is true in some model, but not necessarily all models.

An 'unsatisfiable' sentence is one that is false for every model.

-Propositional logic can only handle True and False values. It has no capabilities to deal with uncertainty like probability theory. Secondly, it can only talk about events that happen. It cannot talk about objects that have properties or the relation between them. Third, there is no shortcut (i.e. to quickly describe a world of many states). First order logic is used to address problem number 2 and 3.

-First order logic: Deal with objects, relationships, and functions.

-Constants: Variables and relations representation (i.e. A , B , x , y)

-Functions: Mapping from objects to objects.

-Relations: Description of how object relates to the world/other objects.

Quantifiers: $\forall x$ represents "for all" and $\exists x$ represents "there exists" where ' x ' is the variable.

- Typically when using $\exists x$, we do not use a conditional arrow \rightarrow
- \leftrightarrow is biconditional/equivalent sign representing “if and only if”

Lesson 14 (Planning)

Bounded solutions: A solution that is guaranteed to reach the goal in a finite number of steps.

Backward (Regression) search: Starts at goal and then search backward. (Look at definitions of possible actions that can result in conditions of the goal state)

Plan space search: Searching the space of plans rather than space of states. Forward searching is more popular now because we can acquire good heuristics.

-An action schema consist of the action, precondition, and the effect of doing the action.

Situational Calculus comprise of three main elements:

- Actions: Actions that can be taken by the agent, including a special predicate

Poss() to indicate when the action is executable.

i.e. move(x,y)

-Situations: A history of action occurrences that represent the current situation of the dynamic world.

Not to be confused with ‘states’.

- Fluents: Statements whose truth value may change. i.e. is_carrying(o, s)