CS540 Introduction to Artificial Intelligence

Lecture 19

Young Wu

Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

August 2, 2022

Search vs. Local Search

Motivation

- Some problems do not have an initial state and a goal state.
- Every state is a solution. Some states are better than others, defined by a cost function (sometimes called score function in this setting), $f(s)$.
- The search strategy will go from state to state, but the path between states is not important.
- There are too many states to enumerate, so standard search through the state space methods are too expensive.

Local Search

Motivation

- Local search is about searching through a state space by iteratively improving the cost to find an optimal or near-optimal state.
- The successor states are called the neighbors (sometimes move set).
- The assumption is that similar (nearby) solutions have similar costs.

Local Search Application

Motivation

- Optimization problems (gradient descent methods are all local search methods)
- Traveling salesman
- Boolean satisfiability (SAT)
- Scheduling

Hill Climbing (Valley Finding)

Description

- Start at a random state.
- Move to the best neighbor state (one of the successors).
- Stop when all neighbors are worse than the current state.
- The idea is similar to gradient descent.

Hill Climbing

Algorithm

- Input: state space S and cost function f.
- Output: $s^\star \in S$ that minimizes $f(s)$.
- Start at a random state s_0.
- At iteration t, find the neighbor that minimizes f.

$$s_{t+1} = \operatorname*{argmin}_{s \in s'(s_t)} f(s)$$

- Stop when none of the neighbors have a lower cost.

$$\text{stop if } f(s_{t+1}) \leqslant f(s_t)$$

Hill Climbing Performance

Discussion

- It does not keep a frontier, so no jumping and no backtracking.
- It is simple, greedy, and stops at a local minimum.

Random Restarts

Discussion

- A simple modification is picking random initial states multiple times and finding the best among the local minima.

First Choice Hill Climbing

Discussion

- If there are too many neighbors, randomly generate neighbors until a better neighbor is found.
- This method is called first choice hill climbing.

Walk SAT Example

Discussion

- Pick a random unsatisfied clause.
- Select and flip a variable from that clause:

1. With probability p, pick a random variable.
2. With probability $1 - p$, pick the variable that maximizes the number of satisfied clauses.

- Repeat until the solution is found.
- Walk SAT uses the idea of stochastic hill climbing.

Simulated Annealing

Description

- Each time, a random neighbor is generated.
- If the neighbor has a lower cost, move to the neighbor.
- If the neighbor has a higher cost, move to the neighbor with a small probability.
- Stop until bored.
- It is a version of Metropolis-Hastings Algorithm.

Annealing

Definition

- The annealing process of heated solids.
- Anneal: to subject (glass or metal) to a process of heating and slow cooling to toughen and reduce brittleness.
- Alloys manage to find a near global minimum energy state when heated and then slowly cooled.

Acceptance Probability

Definition

- The probability of moving to a state with a higher cost should be small.

1. Constant: $p = 0.1$
2. Decreases with time: $p = \frac{1}{t}$
3. Decreases with time and as the energy difference increases: $p = \exp\left(-\frac{|f(s') - f(s)|}{T(t)}\right)$

- The algorithm corresponding to the third idea is called simulated annealing. The Temperature function $T(t)$ should be a decreasing in time t (iteration number).

Temperature

Definition

- T represents temperature which decreases over time. For example, the temperature can change arithmetically or geometrically.

$$T(t+1) = \max\{T(t)-1, 1\}, T(0) = \text{ large}$$
$$T(t+1) = 0.9T(t), T(0) = \text{ large}$$

- High temperature: almost always accept any s'.
- Low temperature: first choice hill climbing.

Simulated Annealing

Algorithm

- Input: state space S, temperature function T, and cost function f.
- Output: $s^{\star} \in S$ that minimizes $f(s)$.
- Start at a random state s_0.
- At iteration t, generate a random neighbor s', and update the state according to the following rule.

$$s_{t+1} = \begin{cases} s' & \text{if } f(s') < f(s_t) \\ s' & \text{with probability } \exp\left(-\dfrac{|f(s') - f(s_t)|}{T(t)}\right) \\ s_t & \text{otherwise} \end{cases}$$

Simulated Annealing Performance

Discussion

- Use hill-climbing first.
- Neighborhood design is the most important.
- In theory, with infinitely slow cooling rate, Simulated Annealing finds global minimum with probability 1.

Genetic Algorithm

Description

- Start with a fixed population of initial states.
- Find the successors by:

1. Cross over.
2. Mutation.