CS540 Introduction to Artificial Intelligence
Lecture 18

Young Wu
Based on lecture slides by Jerry Zhu and Yingyu Liang

July 25, 2019

Genetic Algorithm

Description

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

1. Cross over.
2. Mutation.

Reproduction Probability

Definition

- Each state in the population has probability of reproduction proportional to the fitness. Fitness is the opposite of the cost: higher cost means lower fitness. Use F to denote the fitness function, for example, $F(s) = \frac{1}{f(s)}$ is a valid fitness function.

$$p_i = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}, i = 1, 2, ..., N$$

- A pair of states are selected according to the reproduction probabilities (using CDF inversion).

Cross Over

Definition

- The states need to be encoded by strings.
- Cross over means swapping substrings.
- For example, the children of 10101 and 01010 could be the same as the parents or one of the following variations.

$$(11010, 00101), (10010, 01101)$$
$$(10110, 01001), (10100, 01011)$$

Mutation

Definition

- The states need to be encoded by strings.
- Mutation means randomly updating substrings. Each character is changed with small probability q, called the mutation rate.
- For example, the mutated state from 000 could stay the same or be one of the following.

$$\text{one of } 001, 010, 100, \text{ with probability } q\,(1-q)^2$$

$$\text{one of } 011, 101, 110, \text{ with probability } q^2\,(1-q)$$

$$\text{and } 111, \text{ with probability } q^3$$

Cross Over, Modifications

Definition

- The previous cross over method is called 1 point cross over.
- It is also possible to divide the string into N parts. The method is called N point cross over.
- It is also possible to choose each character from one of the parents randomly. The method is called uniform cross over.

Mutation, Modifications

Definition

- For specific problems, there are ways other than flipping bits to mutate a state.

1. Two-swap: ABCDE to EBCDA
2. Two-interchange: ABCDE to EDCBA

Genetic Algorithm SAT Example

Discussion

Genetic Algorithm TSP Example

Discussion

Fitness Example

Quiz (Graded)

- Fall 1999 Final Q5
- Which ones (multiple) of the following states have the highest reproduction probability?
- The fitness function is $5A + 3BC - D + 2E$.
- A: $(1, 1, 0, 1, 1)$
- B: $(0, 1, 1, 0, 1)$
- C: $(1, 1, 0, 0, 0)$
- D: $(1, 0, 1, 1, 1)$
- E: $(1, 0, 0, 0, 0)$

Genetic Algorithm, Part I

Algorithm

- Input: state space S represented by strings s and cost function f or fitness function F.
- Output: $s^{\star} \in S$ that minimizes $f(s)$.
- Randomly generate N solutions as the initial population.

$$s_1, s_2, ..., s_N$$

- Compute the reproduction probability.

$$p_i = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}, i = 1, 2, ..., N$$

Genetic Algorithm, Part II

Algorithm

- Randomly pick two states according to p_i, say s_a, s_b. Randomly select a cross over point c, swap the strings.

$$s'_a = s_a\,[0...c)\; s_b\,[c...m)$$
$$s'_b = s_b\,[0...c)\; s_a\,[c...m)$$

- Randomly mutate each position of each state s_i with a small probability (mutation rate).

$$s'_i\,[k] = \begin{cases} s_i\,[k] & \text{with probability } 1-q \\ \text{random} & \text{with probability } q \end{cases}, k = 1, 2, ..., m$$

- Repeat with population s'.

Variations

Discussion

- Parents can survive.
- Use ranking instead of $F(s)$ to compute reproduction probabilities.
- Cross over random bits instead of chunks.

Genetic Algorithm Performance

Discussion

- Use hill-climbing first.
- State design is the most important.
- In theory, cross over is much more efficient than mutation.

Lion Game Example, Part I

Quiz (Participation)

- There are N lions, ordered by size, $i = 1, 2, 3, ..., N$, and a bunny. N takes an integer between 1 and 10 with equal probability (known to all the lions). Each lion i can choose to jump out and eat the slightly smaller lion $i - 1$, or stay hidden, and only lion 1 can eat the bunny. Each lion prefers eating to staying hungry to being eaten.
- What is the probability that the bunny is eaten?
- A: 0, B: $\frac{1}{3}$, C: $\frac{1}{2}$, D: $\frac{2}{3}$, E: 1

Lion Game Example, Part II

Quiz (Participation)

Pirate Game Example, Part I

Quiz (Participation)

- 5 pirates got 100 gold coins. Each pirate takes a turn to propose how to divide the coins, and all pirates who are still alive will vote whether to accept the proposal or reject the proposal, kill the pirate, and continue to the next round. Use strict majority rule for the vote, and use the assumption that if a pirate is indifferent, he or she will vote reject.
- How will the first pirate propose?
- A: $(0, 0, 0, 0, 100)$
- B: $(20, 20, 20, 20, 20)$
- C: $(94, 0, 1, 2, 3)$
- D: $(97, 0, 1, 0, 2)$
- E: $(98, 0, 1, 0, 1)$

Pirate Game Example, Part II

Quiz (Participation)

Wage Competition Example, Part I

Quiz (Participation)

- An applicant can produce 20 dollars worth of products per hour. Two profit-maximizing companies compete to hire the applicant. Company 1 makes an offer first, then company 2 sees the offer and makes another offer. The applicant goes to the company that offers a higher wage, and in case of a tie, he or she will flip a fair coin to decide.
- What should the offers be (hourly wage, only integer amounts allowed)?
- A: $(20, 20)$
- B: $(20, 19)$
- C: $(19, 20)$
- D: $(19, 19)$
- E: None of the above.

Wage Competition Example, Part II

Quiz (Participation)

Game Tree

Motivation

- The initial state is the beginning of the game.
- There are no goal states, but there are multiple terminal states in which the game ends.
- Each successor of a state represents a feasible action (or a move) in the game.
- The search problem is to find the terminal state with the lowest cost (or usually the highest reward).

Adversarial Search

Motivation

- The main difference between finding solutions of games and standard search problems or local search problems is that part of the search is performed by an opponent adversarially.
- Usually, the opponent wants to maximize the cost or minimize the reward from the search. This type of search problems is called adversarial search.
- In game theory, the solution of a game is called an equilibrium. It is a path in which both players do not want to change actions.

Backward Induction

Motivation

- Games are usually solved backward starting from the terminal states.
- Each player chooses the best action (successor) given the (already solved) optimal actions of all players in the subtrees (called subgames).