

CS341 SYSTEMS BIOLOGY I

LECTURE 19

GENETIC ALGORITHMS FOR PARAMETER-FITTING

Tournament Selection. Recall that we have implemented one useless selection operator (uniform) and one useful selection operator (truncation). Truncation selection works by choosing the breeding pool from just the best individuals of the previous generation. We noticed when we ran the algorithm that the parameter sets became quite similar quite quickly. This is good if we think our first random guess is close to a great solution. But if we want to allow the algorithm to explore more parameter space, then it pays to maintain a more diverse population. Maybe a high-cost individual has some redeeming qualities! We still want to apply pressure by ensuring that the breeding pool is somehow better than the population as a whole, but it may pay off to include a few high-cost individuals. One way to do this is with tournament selection. In tournament selection, we have a tournament size (e.g. 2 or 5) that determines the number of competitors in each “tournament”. To select an individual for the breeding pool, we begin by selecting a group of individuals randomly from the entire population. The best individual of that group (tournament) is selected to join the breeding pool. If we have a small tournament size, then we are more likely to end up with high-cost individuals than if we have a large tournament size. See Figure 1 for an illustration.

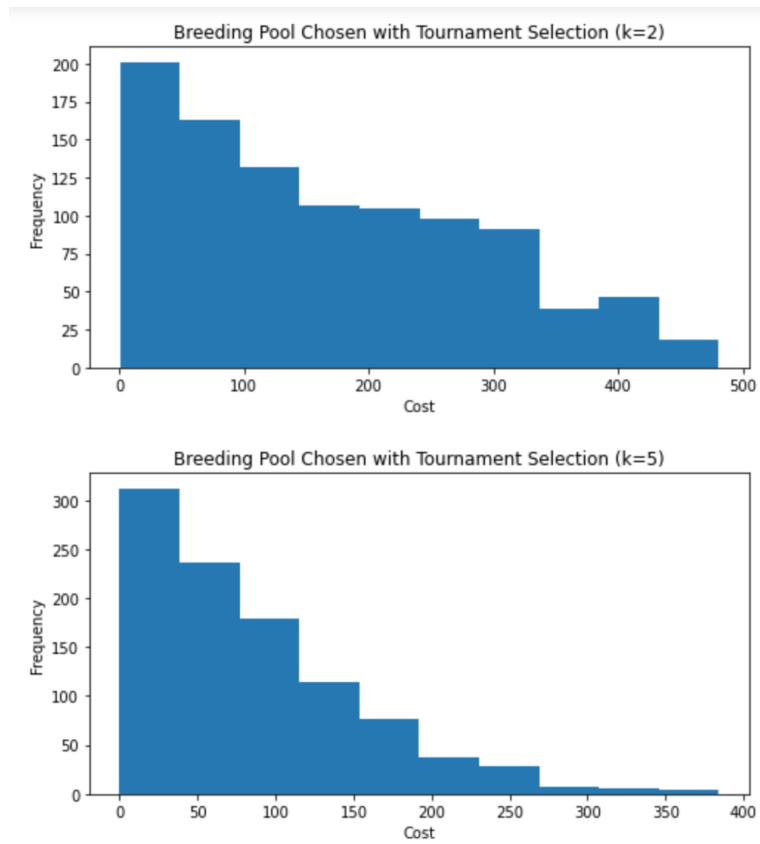


FIGURE 1. Histogram of costs of breeding pool. The previous generation had evenly spaced costs ranging from 0.1 to 500. Using tournament selection, our distribution of costs favors low-cost individuals. With a tournament size of 2, we retain more high-cost individuals than with a tournament of size 5.

Cost function for Goodwin Oscillator. Your project explores the ability of the genetic algorithm to find parameters for Goldbeter’s fly clock model. Let’s use lecture time to see how it performs on Goodwin’s oscillator. We develop a cost function that

- Simulates the model for 500 hours to remove the transient.
- Simulates the model for 100 hours to use for computing the cost.
- Compute the period of the second simulation. Call it τ .
- Compute the difference in amplitudes between the first 50 h and second 50 h of the simulation (we don’t want the oscillations to be growing or shrinking in amplitude). Sum the amplitudes of all the states in the first half and call it a_1 and sum the amplitudes in all the states in the second half and call it a_2 .
- Compute the amplitudes of the all states (we don’t want any to be too small). Call them a .

The cost is computed so that the cost will be small when the period is close to 24 h, so that the cost will be small if all elements of a are greater than 0.1, and so that the cost will be small if a_1 is close to a_2 . Since we expect $a_1 - a_2$ to be small, we multiply it by 100 to increase the weight of its effects on the cost. See Figures 2, 3, 4 for graphical representations.

$$cost = (\tau - 24)^2 + \text{sum}(\exp \frac{\log(0.001)}{0.1} a) + (100(a_2 - a_1))^2$$

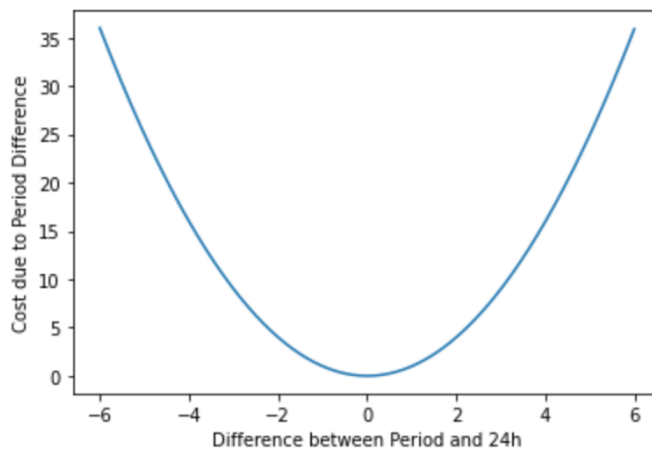


FIGURE 2. The cost due to the period is quadratic.

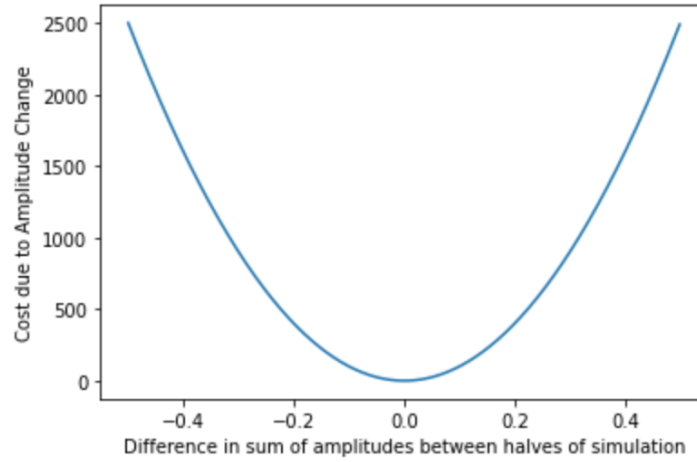


FIGURE 3. The cost due to amplitude changing over time is quadratic.

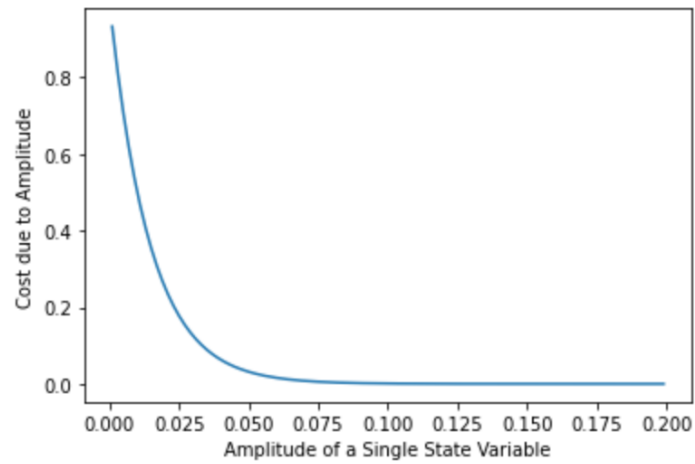


FIGURE 4. The cost due to amplitude is exponential. The cost for an amplitude of 0.1 is 0.001, with smaller amplitudes leading to exponentially larger costs.