

1

(b)

In the 30 runs plotted, we found the global maximum 6 times.

(c)

Since the probability of a certain member being selected as a parent is directly proportional to its fitness, it is impossible to assign a probability to a member with negative fitness. This can be corrected by finding the individual with the most negative fitness, if one exists, and adding the absolute value of this fitness to the fitness of every individual. This will guarantee every individual has a non-negative fitness.

(d)

The crossover function treats the decision variables as one 14-bit string, with only one crossover point. Since the decision variables are actually two distinct integers, we should treat each as a 7-bit string and perform crossover independently for each decision variable.

(e)

Each decision variable ranges from 0 to 127, inclusive. Therefore there are $128 * 128 = 16386$ unique combinations for the pair of decision variable. We can be certain we've found the optimal solution if we evaluate the objective function for each of the 16386 decision variable combinations.

2

(b)

In this problem tournament selection worked significantly better. I believe that it's because in this case, having a probability to pick a very bad parent (as can happen with roulette selection), really hurt the next population.

3

(a)

To convert to a minimization problem add the `min_cost` and `max_cost` and then subtract our actual fitness value. This causes small actual fitnesses to evaluate to large fitnesses, so by maximizing we're actually finding the smallest actual fitness.

(d)

It's difficult to conclude based on this that either algorithm is better or worse. Here we're testing both problems with certain parameters. The parameters chosen can make a large difference in these heuristic optimization problems, and we're not sure what parameters would make these algorithms better or worse.

We also don't know how these two algorithms perform on different cost functions. Every function is different, and some algorithms will do a better job with functions with certain properties.

Finally, the two algorithms might do significantly differently if the number of cost evaluations changed. Overall here we're comparing over a very small sample size (one problem), so we can't definitely say which algorithm is better (and honestly neither will ever be definitely better).