Chih-Ang Wang (AndrewID: chihangw)

1. **Textbook 2-1**

Since it’s a 8x8 board and each queen cannot check each others. Therefore, the first queen would have 8 choices to be positioned, 7 for the second, 6 for the third and so on. The phenotype space is 8! For the eight-queen problem.

1. **Textbook 3-1**

This problem is based on bit resolution. Since one bit could represent 0 or 1 and divide [0, 1] into two. With N bits, we have 2N segments. Therefore, to achieve a precision of 0.001 in distance, we need

0.001 = , so n = 10 !

In sum, 10 bits are needed at least to achieve this precision.

1. **Textbook 3-4**

Sine the chance that a bit will be flipped is 1/L, we know that the probability is (1-1/L) if not flipped. Besides, each time a bit will be flipped or not is independent from the previous one and the next one. Therefore, a binary chromosome with length L will not be changed if the probability is (1-1/L)L

1. **Textbook 3-5**

* 157/76 ≅ 2.07. Therefore, 2 copies of the best individual in the pool
* For the single best individual, its probability accounted in the Roulette Wheel algorithm (RW) would be . After 100 ronuds, the probability if that gene isn’t in the mating pool is .
* Since the probability of each chromosome is determined before the applied algorithm no matter it’s RW or SUS, the probability is the same.

1. **Textbook 3-6**

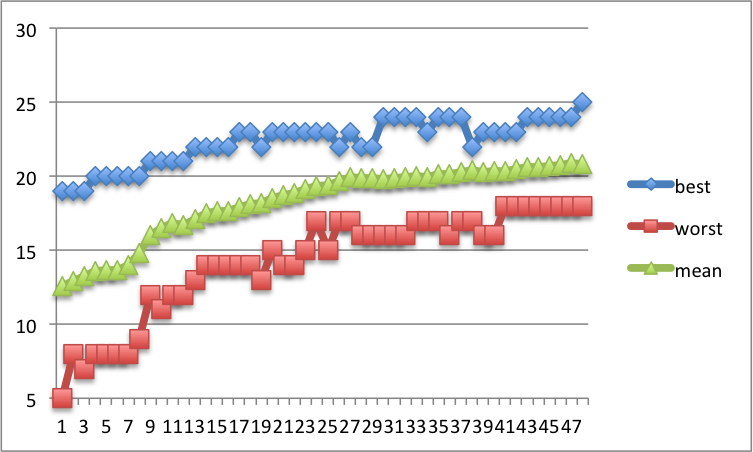
Since crossover doesn’t generate new chromosomes in the pool, the allele frequency will not change even after k operations.

1. **Textbook 3-7 (Programming)**

In this problem, I use C++ to implement the algorithm. The source file is named “gen\_ea.cpp”. To run the program, please enter “**g++ -o ea gen\_ea.cpp**” in your terminal to compile it first, then enter “**./ea**” to start running. One thing to note is that the **time** taken for each run will be displayed in your terminal, while the detailed min, max and mean values will be in a newly generated file called **report.txt**. You can see fitness changes in this file.

The following is the plot for ***fitness v.s time*** changes.

Y-axis: fitness X-axis: time



After ten runs, the mean and standard deviation of the time taken until achieving optimum is:

**Mean**: 0.0054929 seconds.

**Standard deviation**: 0.0027332 seconds.