

1 exercise

ALEXANDR MATVEYCHUK II-11

code(PYTHON 3):

```
import random
def get_fitness(chromosome):
    return chromosome.count(1)
def roulette_select(fs):
    p = random.uniform(0, sum(fs))
    #fs.reverse()
    for i, f in enumerate(fs):
        if p <= 0:
            break
        p -= f
    return i
def uniform_crossover(parents):
    child = []
    for x in range(1000):
        point = random.random()
        if point > 0.5:
            child.append(parents[0][x])
        else:
            child.append(parents[1][x])
    # print(child)
    return child
def single_point_crossover(parents):
    point = random.randint(0, 999)
    return [parents[0][point:] + parents[1][:point],
            parents[0][:point] + parents[1][point:]]
def main():
    chromosomes = [[random.randint(0, 1) for _ in range(1000)] for _ in
range(100)]
    chromosomes.sort(key=get_fitness, reverse=False)
    # fitness = [get_fitness(chromosome) for chromosome in chromosomes]
    new_chromosomes = []
    better_fitness = 0
    population = 0
    try:
        file = open("uniform_best_fitness.txt", 'w')
        file1 = open("uniform_average_fitness.txt", 'w')
        while population < 900:
            fitness = [get_fitness(chromosome) for chromosome in chromosomes]
            for _ in range(100):
                mom_index = roulette_select(fitness)
                pope_index = roulette_select(fitness)
                parents = [chromosomes[mom_index], chromosomes[pope_index]]
                new_chromosomes.append(uniform_crossover(parents))
                #new_chromosomes.extend(single_point_crossover(parents))
            new_chromosomes.sort(key=get_fitness, reverse=False)
            better_fitness = get_fitness(new_chromosomes[99])
            avg_fitness = sum(fitness) / len(fitness)
            population += 1
            chromosomes = new_chromosomes
            new_chromosomes = []
            file.writelines(str(better_fitness) + "\n")
            file1.writelines(str(avg_fitness) + "\n")
            print(better_fitness, population)
    finally:
        file.close()
        file1.close()
main()
```

Graph:

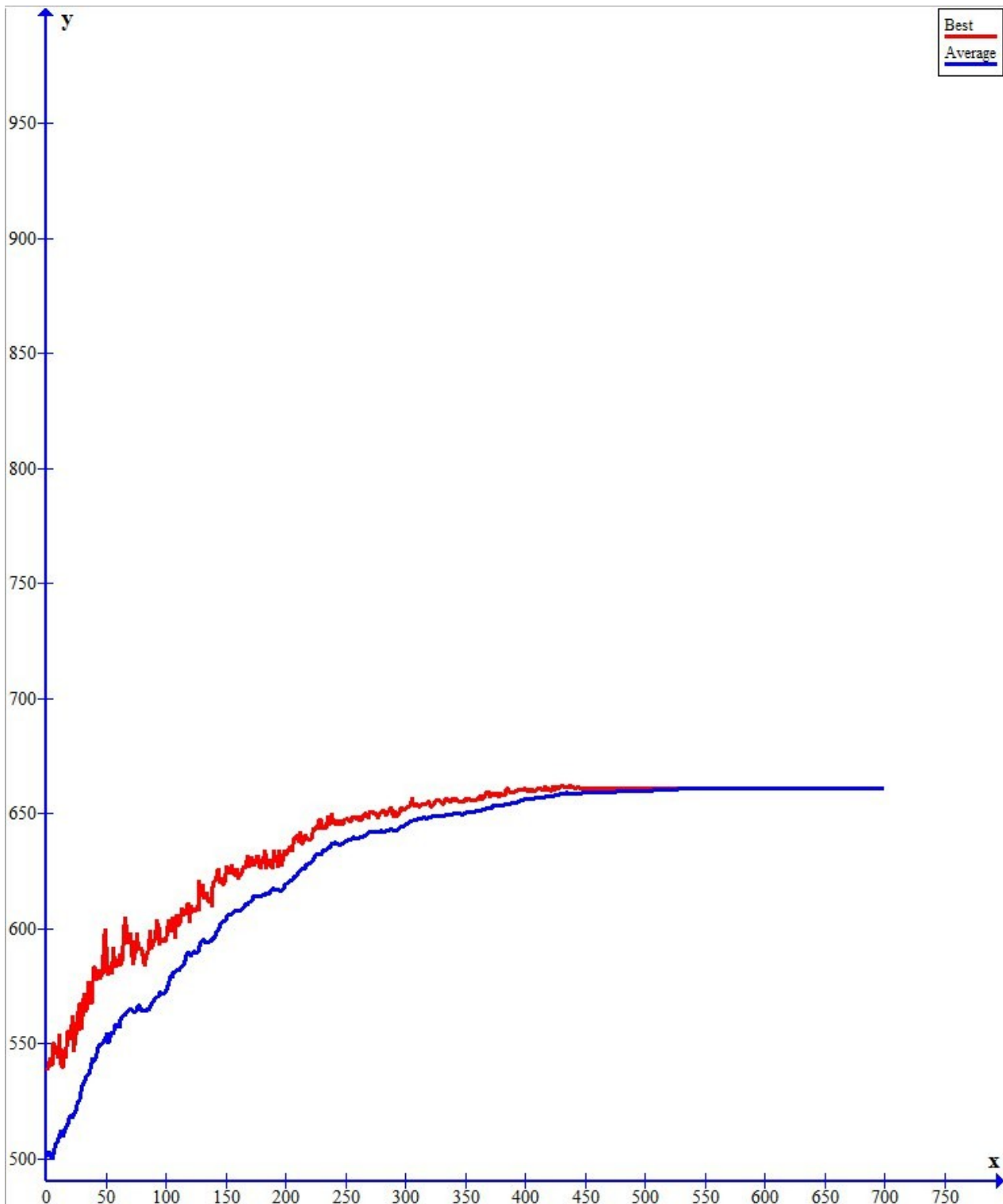
roulette selection and uniform crossover.

Population: 700

X : POPULATION

Y: FITNESS (RED LINE — BEST, BLUE LINE — AVERAGE)

MAX FITNESS: 661



roulette selection and one-point crossover.

Population: 700

X : POPULATION

Y: FITNESS (RED LINE — BEST, BLUE LINE — AVERAGE)

MAX FINTESS: 992

