2ex. Alexandr Matveychuk

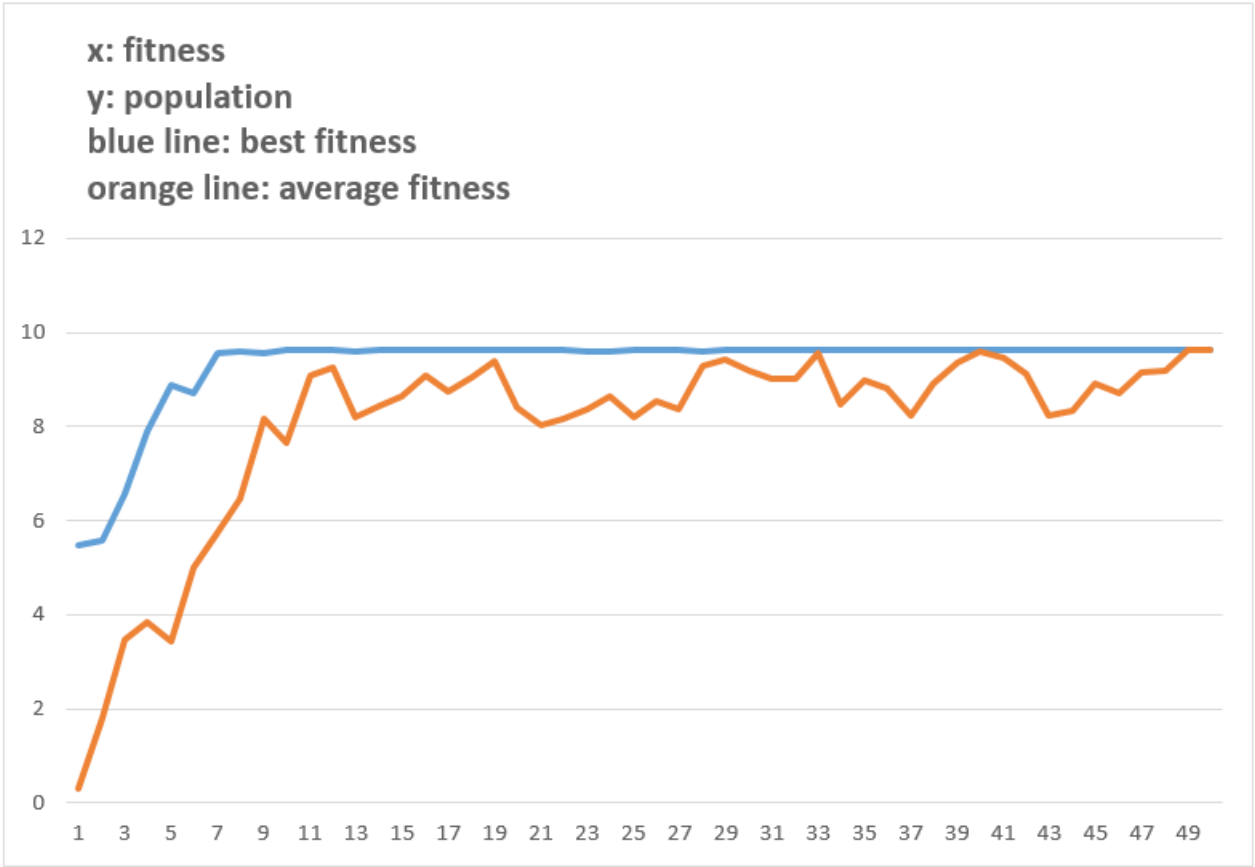
Maximization of 3-D Schefel function

code(Python3):

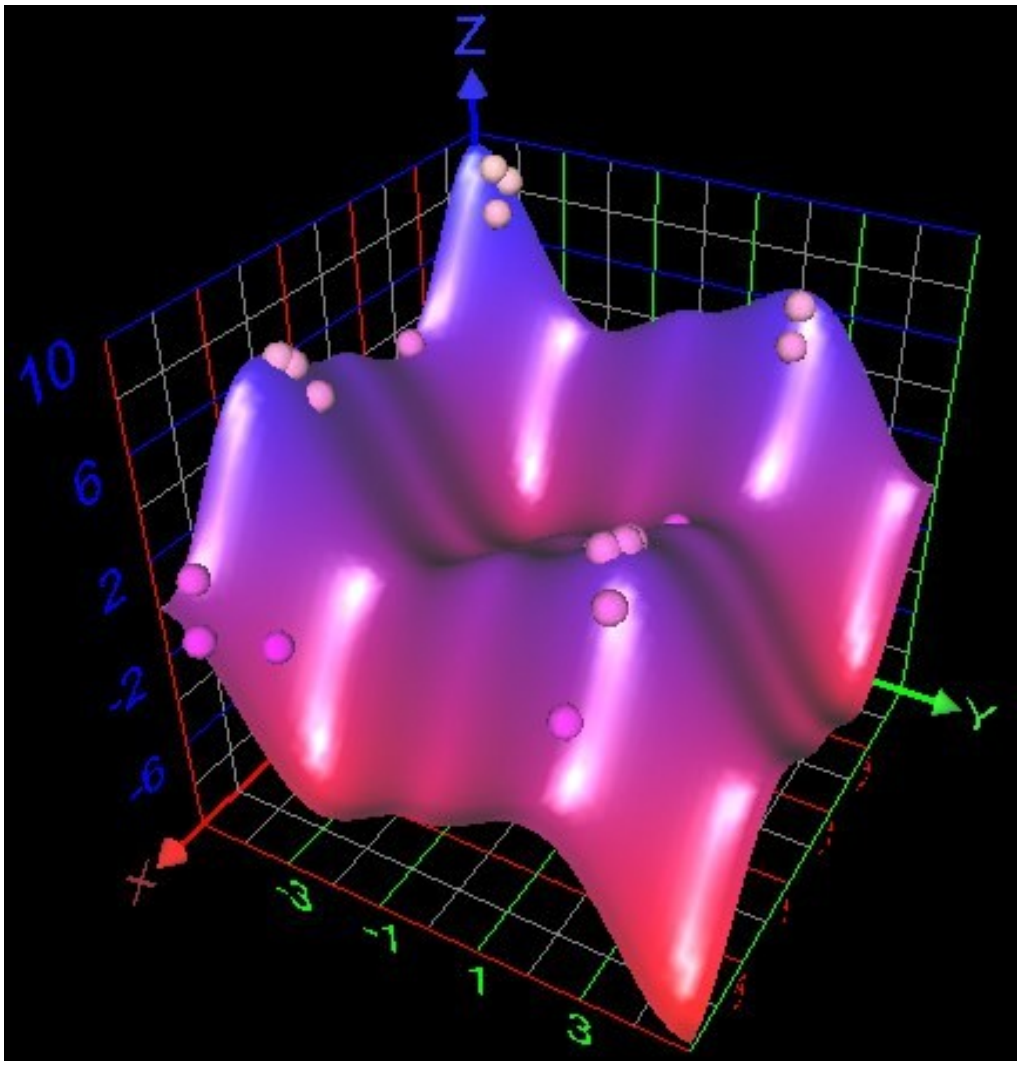
```
import random
import math
def get_fitness(chromosome):
    strChrom = ""
    for ch in chromosome:
        strChrom += str(ch)
    strChrom = strChrom[::-1]
    x1 = 5 * (int(strChrom[1:11], 2) / 1023)
    x2 = 5 * (int(strChrom[12:22], 2) / 1023)
    if strChrom[0] == "1":
        x1 = -x1
    if strChrom[12] == "1":
        x2 = -x2
    return x1 * math.sin(abs(x1)) + x2 * math.sin(abs(x2))
def points(chromosomes , file):
    #file = open("points.txt", "w+")
    strChrom = ""
    for chromosome in chromosomes:
        for ch in chromosome:
            strChrom += str(ch)
        strChrom = strChrom[::-1]
        x1 = 5 * (int(strChrom[1:11], 2) / 1023)
        x2 = 5 * (int(strChrom[12:22], 2) / 1023)
        if strChrom[0] == "1":
            x1 = -x1
        if strChrom[12] == "1":
            x2 = -x2
        x3 = x1 * math.sin(abs(x1)) + x2 * math.sin(abs(x2))
        wrLine = str(x1) + ", " + str(x2) + ", " + str(x3) + "\n"
        file.writelines(wrLine)    file.writelines("**********************\n")
def truncate_selection(chromosomes):
    return [chromosomes[random.randint(0, 9)],
            chromosomes[random.randint(0, 9)]]
def uniform_crossover(parents):
    child = []
    for x in range(22):
        point = random.random()
        if point > 0.5:
            child.append(parents[0][x])
        else:
            child.append(parents[1][x])
    # print(child)
    return child
def mutation(child):
    point = random.randint(0, 21)
    child[point] = (child[point] + 1) % 2
    return child
def main():
    chromosomes = [[random.randint(0, 1) for _ in range(22)] for _ in range(20)]
    chromosomes.sort(key=get_fitness, reverse=True)
    new_chromosomes = []
    avg_fitness = 0
    for ch in chromosomes:
        avg_fitness += get_fitness(ch)
    avg_fitness = avg_fitness / len(chromosomes)
```

```
    better_fitness = get_fitness(chromosomes[0])
    print(better_fitness, avg_fitness)
    populations = 0
    file = open("points.txt", "w+")
    while populations < 50:
        for _ in range(20):
            parents = truncate_selection(chromosomes)
            child = uniform_crossover(parents)
            child = mutation(child)
            new_chromosomes.append(child)
        new_chromosomes.sort(key=get_fitness, reverse=True)
        # for chr in new_chromosomes:
        #    print(get_fitness(chr))
        # print("____________________")
        better_fitness = get_fitness(new_chromosomes[0])
        for ch in new_chromosomes:
            avg_fitness += get_fitness(ch)
        avg_fitness = avg_fitness / len(chromosomes)
        populations += 1
        if populations== 1 or populations== 2 or populations== 5 or
populations== 8 or populations== 50:
            points(new_chromosomes, file)
        chromosomes = new_chromosomes
        new_chromosomes = []
        print(better_fitness, avg_fitness)
        # get_fitness(chromosomes[0])
        # print(new_chromosomes)
main()
```

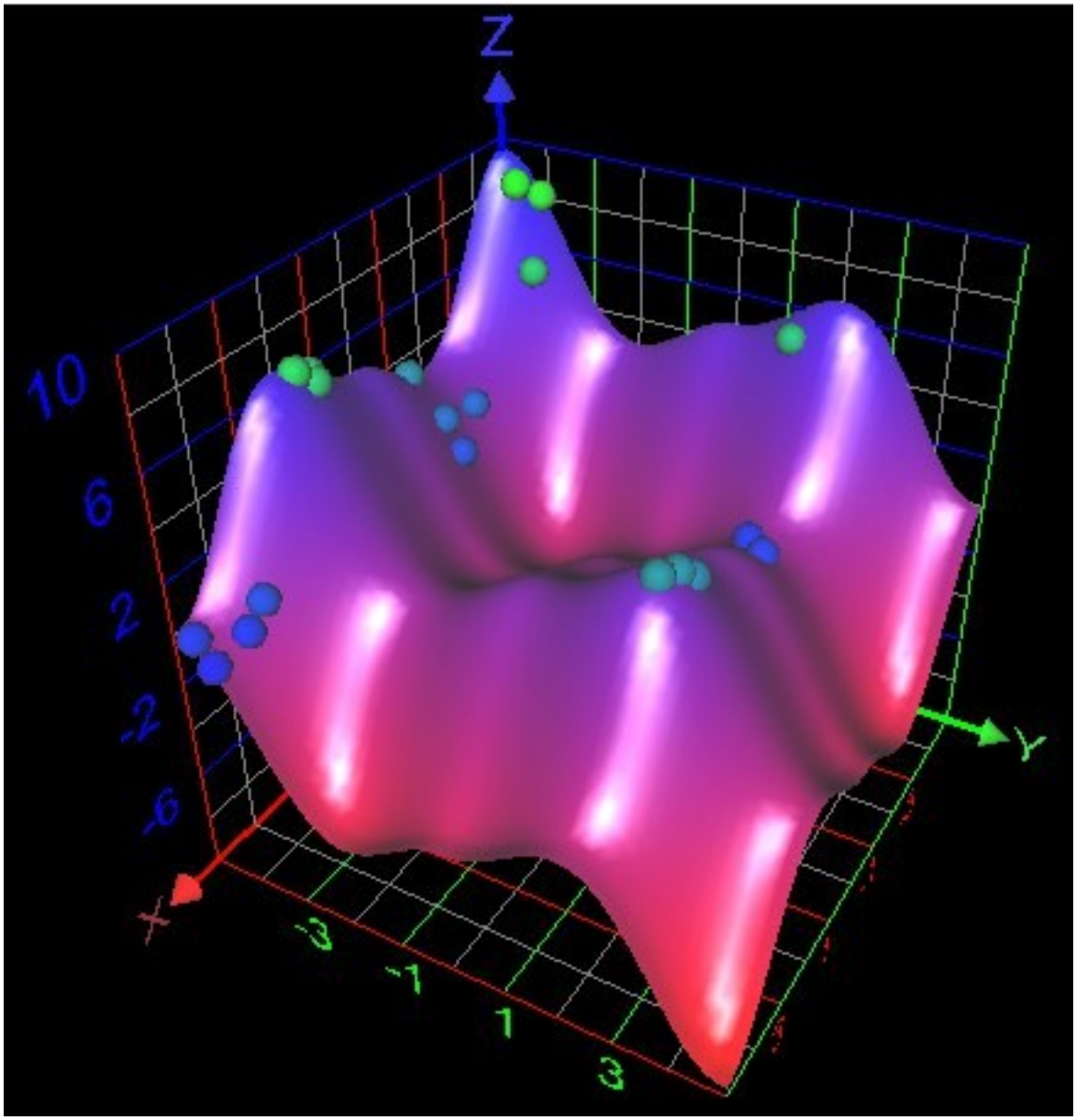
Graphs:



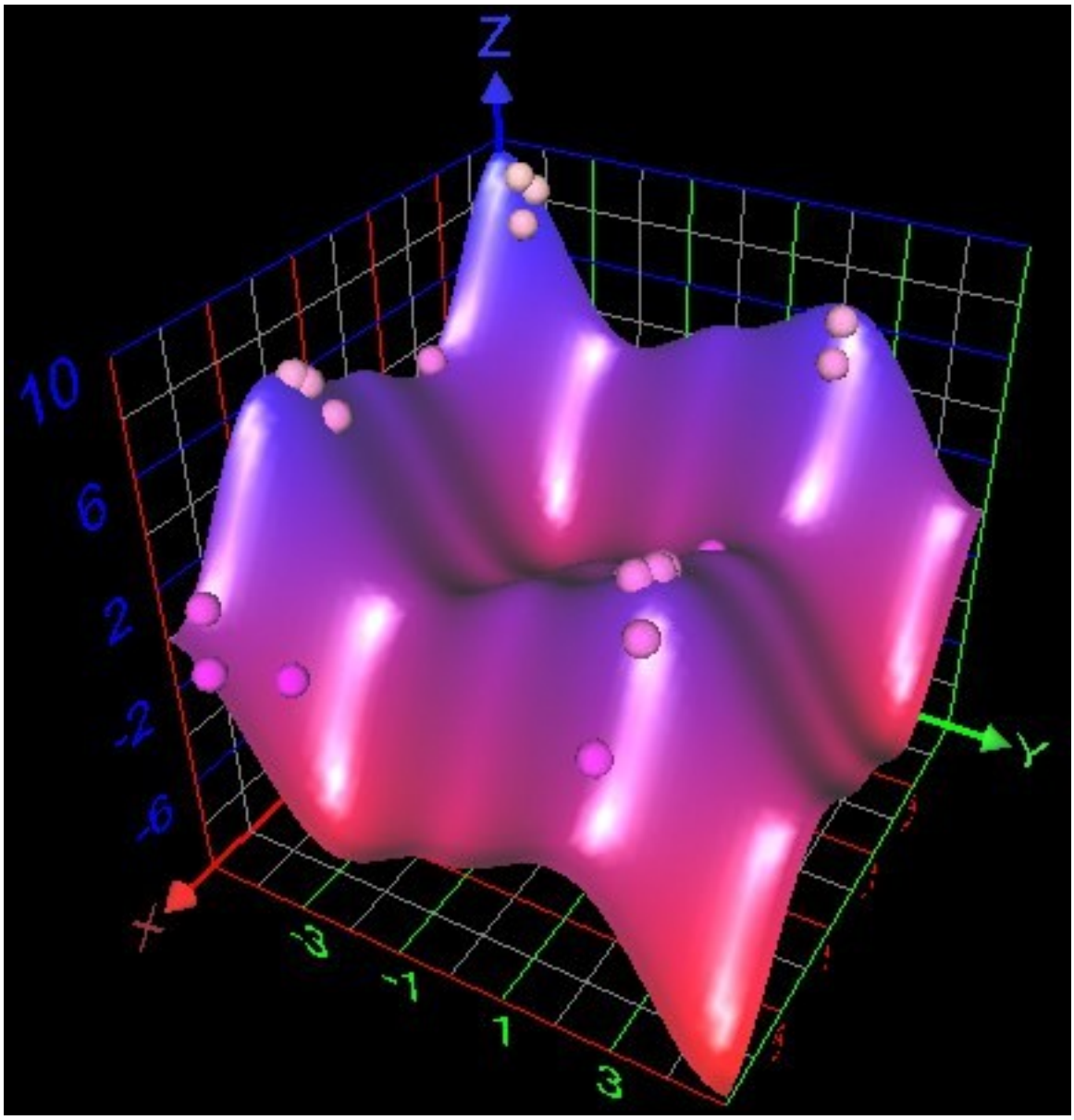
Population count: 50
1st population:



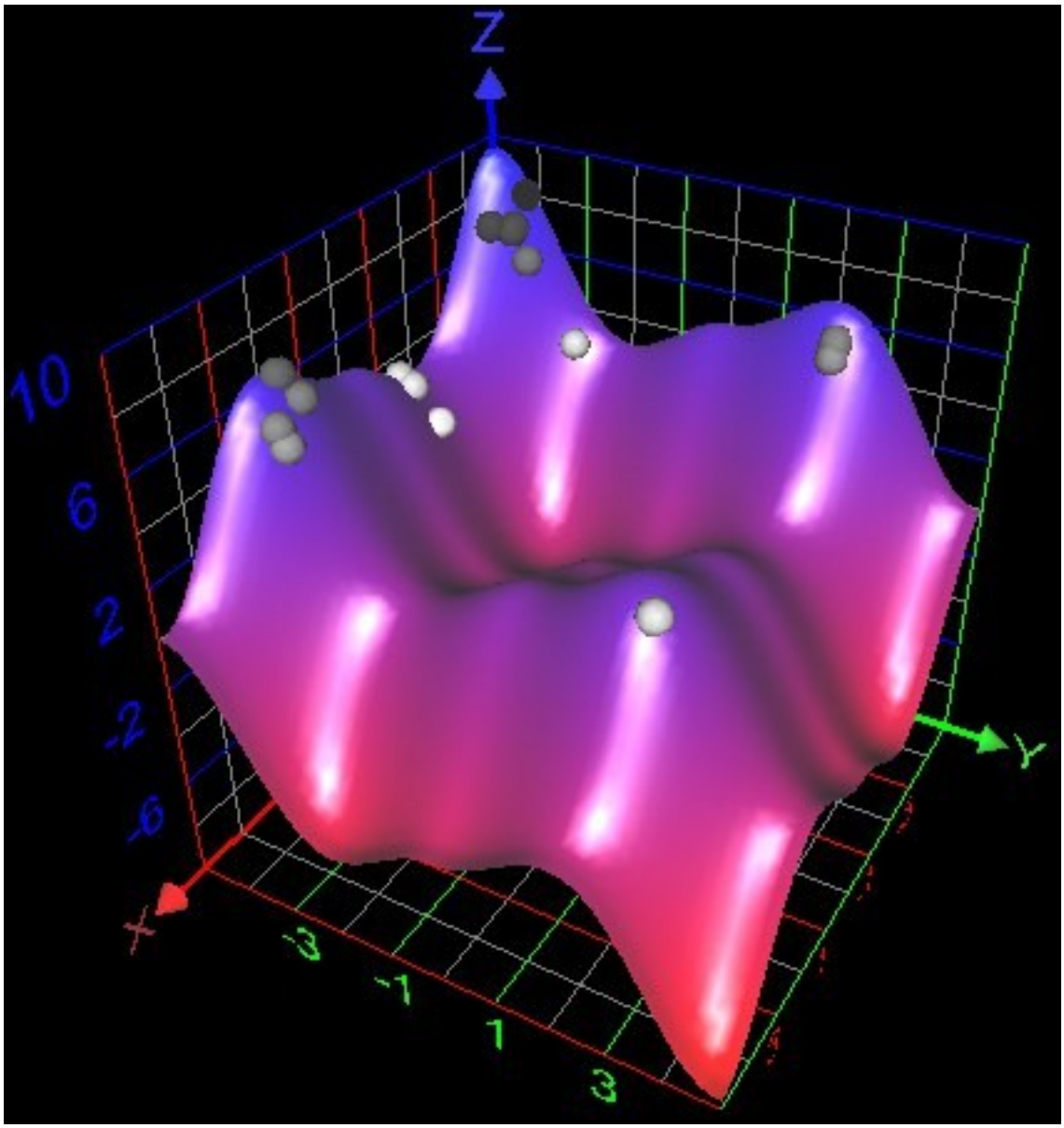
population num.10:



population num.25:



population num.40:



population num.50: