Algorytm genetyczny

Kacper Grzesik 52684, Dominik Górski 52679

1. main.py

Główny plik zawierający algorytm genetyczny.

a) Podanie danych wejściowych

Nazwa zmiennej	Znaczenie
generate_population_size	populacja na generację
generation_limit	maksymalna ilość generacji
mutation_probability	szansa na mutację
file	plik do analizy
file_optimum	plik z optimum
verbose	czy printować output

```
# DATA
generate_population_size = 200
generation_limit = 100
mutation_probability = 0.2
verbose = False  # should print the output

file = open("large_scale/knapPI_1_100_1000_1", "r")  # file to analize
file_optimum = open("large_scale-optimum/knapPI_1_100_1000_1", "r")  # file with optimum
```

b) Ustalenie funkcji selekcji oraz krzyżowania w ewolucji

Odbywa się to przez komentowanie niepotrzebnych funkcji. Zostało to rozszerzone w podpunktach związanych z rysowaniem wykresów.

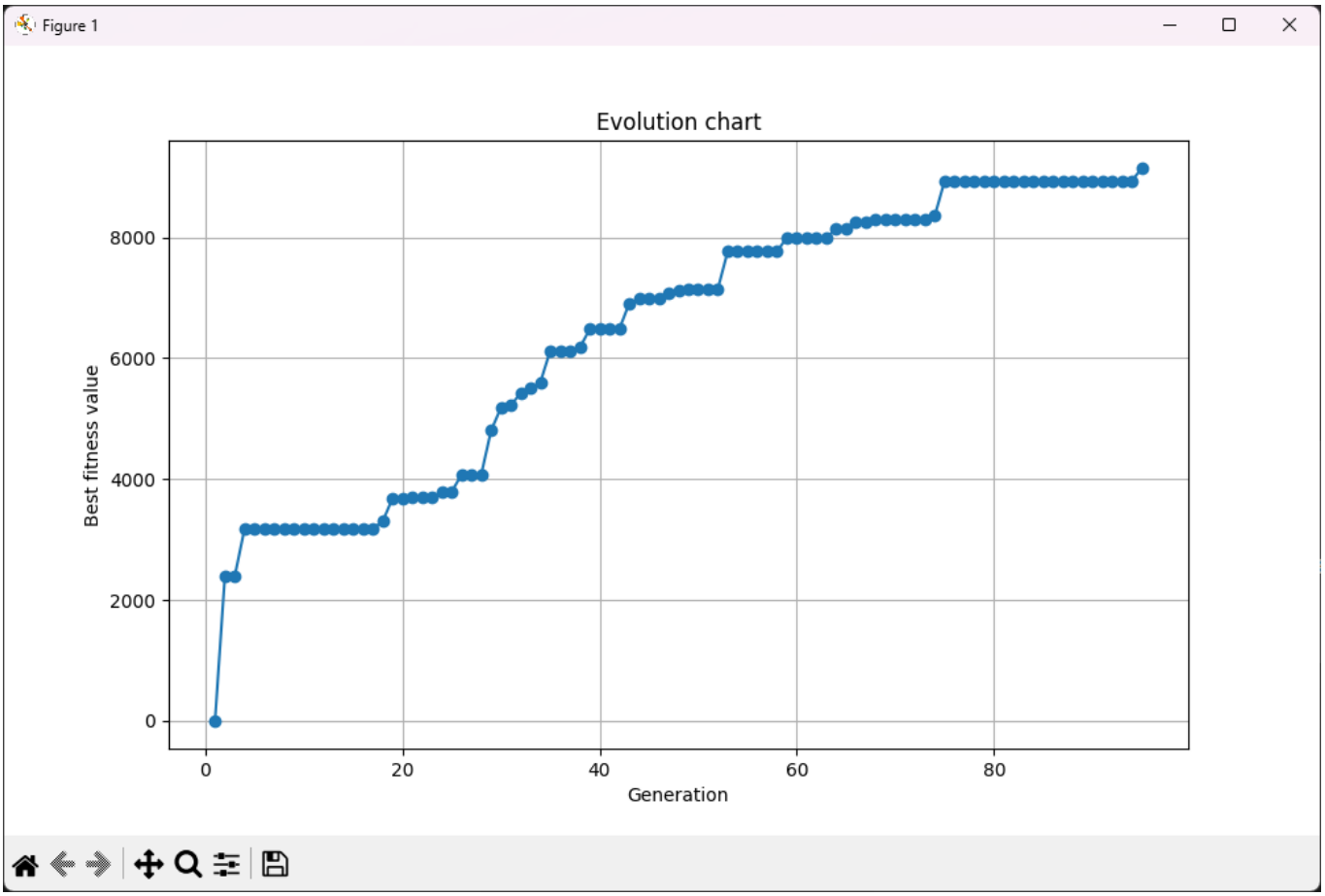
```python
# evolution
def run_evolution(
    populate_func,
    fitness_func,
    fitness_limit: int,  # if fitness of the best solution exceeds the limit, it's done
    selection_func = roulette_wheel_selection_pair,
    # selection_func = ranking_selection_pair,
    # selection_func = tournament_selection_pair,
    # crossover_func = single_point_crossover,
    crossover_func = two_point_crossover,
    mutation_func = mutation,
    generation_limit = generation_limit
):
```

c) Uzyskanie wyników (jeśli verbose = True)

```
Population size: 200
Mutation probability: 0.2
Number of generations: 100
Time: 6.397189499999513s
Best solution: [791, 874, 908, 569, 931, 726, 724, 641, 800, 992]
Item count: 10
Best solution value sum: 7956
Optimum: 9147
```

2. draw_chart.py

Generowanie wykresu na podstawie programu main.py.
Program musiał zostać zmodyfikowany tak, aby śledzić jednostkę o najwyższym fitnessie dla każdej ewolucji.



3. experiments.py

Program rysuje wykresy dla różnych parametrów funkcji algorytmu genetycznego za pomocą biblioteki matplotlib.pyplot

a) Dane początkowe

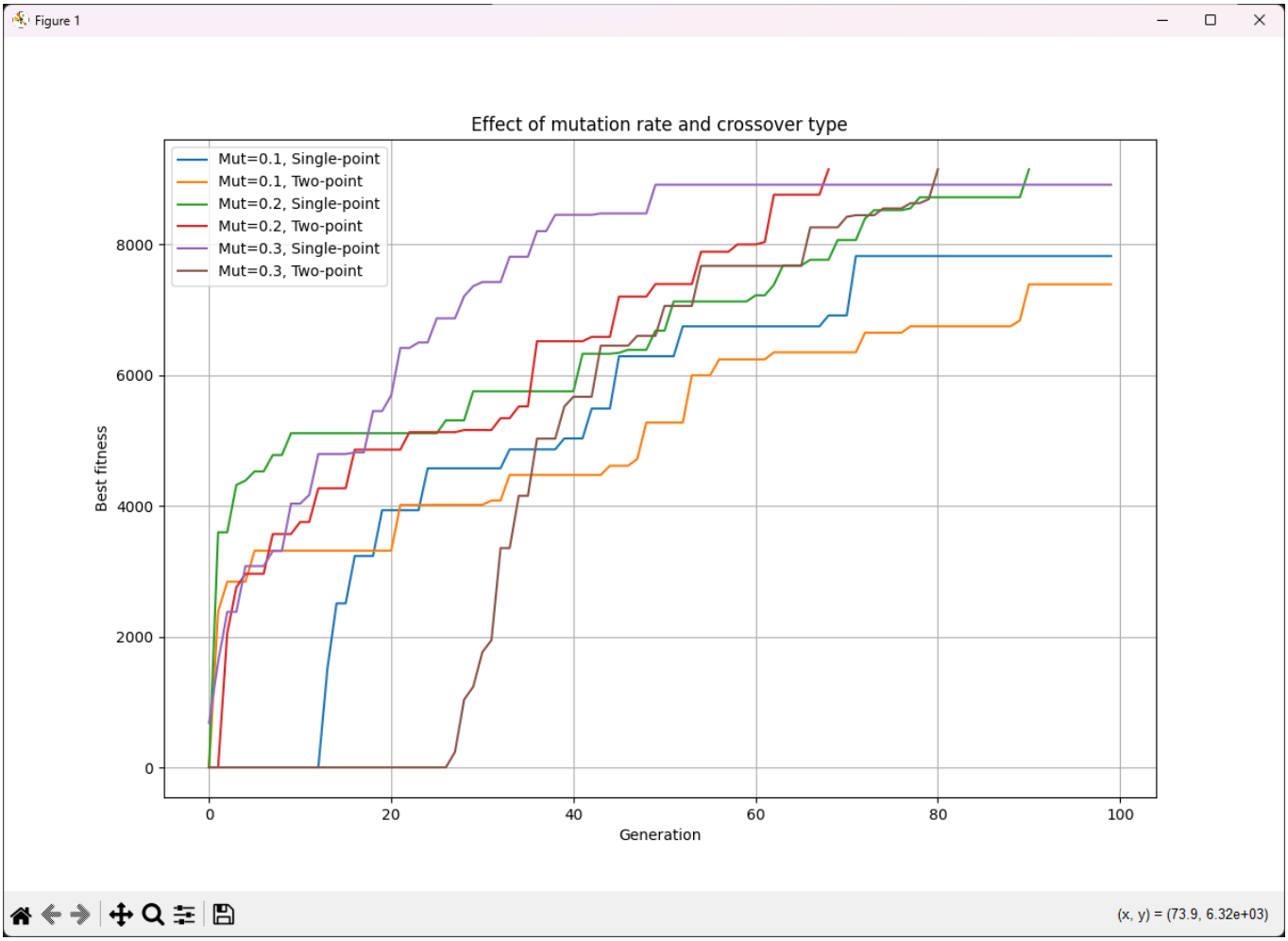
```python
# experiment parameters
crossover_methods = {
    "Single-point": single_point_crossover,
    "Two-point": two_point_crossover
}

mutation_rates = [0.1, 0.2, 0.3]

selection_methods = {
    "Roulette": roulette_wheel_selection_pair,
    "Ranking": ranking_selection_pair,
    "Tournament": tournament_selection_pair
}
```

b) Dane wyjściowe (wykresy)

Wykresy dla różnych współczynników mutacji i krzyżowania:



Wykresy porównujące selekcję rankingową, ruletkową oraz turniejową, krzyżowanie jedno i dwupunktowe:

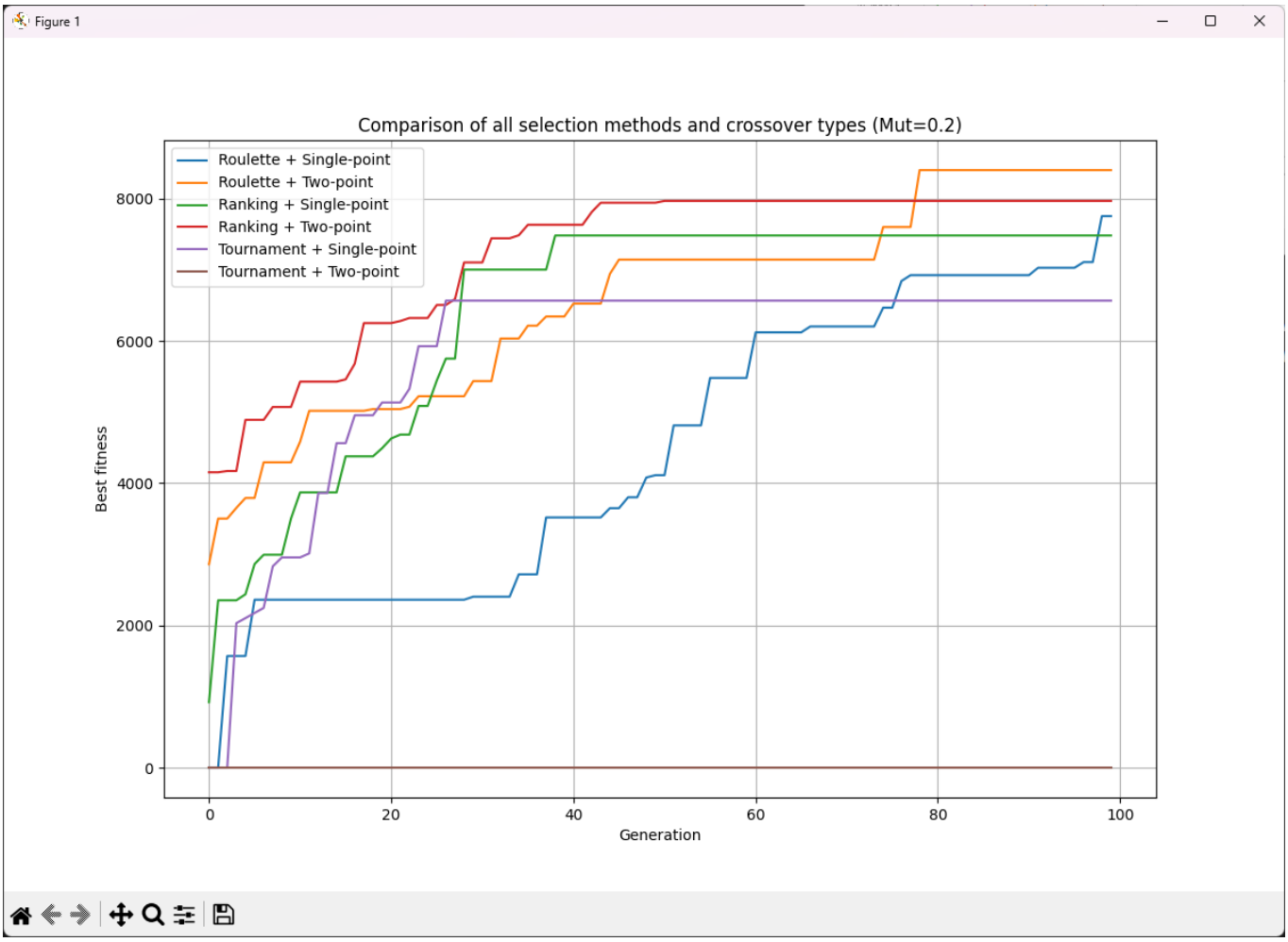


Figure 1
Comparison of all selection methods and crossover types (Mut=0.2)
Roulette + Single-point
Roulette + Two-point
Ranking + Single-point
Ranking + Two-point
Tournament + Single-point
Tournament + Two-point
Best fitness
Generation
0
2000
4000
6000
8000
0
20
40
60
80
100