Sprawozdanie z laboratorium nr 4

Inteligentne Metody Optymalizacji

Autorzy: Jakub Gołąb, Mariusz Hybiak

Wprowadzenie

Celem zadania było rozszerzenie implementacji lokalnego przeszukiwania stosując trzy metody:

- (MSLS) Multiple start local search
- (ILS1) Iterated local search - Iteracyjne przeszukiwanie lokalne z niewielką perturbacją
- (ILS2) Iterated local search - Iteracyjne przeszukiwanie lokalne z Large-scale neighborhood search, tj. większą perturbacją typu Destroy-Repair

Algorytmy

MSLS

```
function MSLS(distance_matrix, data, n_iter):
    best_cycle1 = NULL
    best_cycle2 = NULL
    best_total_length = INFINITY

    for n_iter:
      cycle1, cycle2 = random_cycle(data)
      cycle1, cycle2 = local_search(cycle1, cycle2, distance_matrix, data)
      length = calculate_cycles_length(cycle1, cycle2, distance_matrix)

      if length < best_length:
        best_cycle_1 = cycle1
        best_cycle_2 = cycle2
        best_length = length

    return best_cycle1, best_cycle2
```

ILS1

```
function ILS1(distance_matrix, data, time_MSLS):
    best_cycle_1, best_cycle_2 = random_cycle(data)
    best_cycle_1, best_cycle_2 = local_search(best_cycle_1, best_cycle_2, distance_matrix, data)
    best_length = calculate_cycles_length(best_cycle_1, best_cycle_2, distance_matrix)

    while time_elapsed <= time_MSLS:
        cycle1, cycle2 = little_perturbation(best_cycle_1, best_cycle_2)
        cycle1, cycle2 = local_search(cycle1, cycle2, distance_matrix, data)
        length = calculate_cycles_length(cycle1, cycle2, distance_matrix)
        if length < best_length:
            best_length = length
            best_cycle_1 = cycle1
            best_cycle_2 = cycle2

    return best_cycle_1, best_cycle_2

function little_perturbation(cycle1, cycle2):
    foreach cycle in set_of[cycle1, cycle2]:
        cycle = swap_random_nodes(cycle1, cycle2, 0.2)

    return cycle1, cycle2
```

ILS2

```
function ILS2(distance_matrix, data, time_MSLS):
    best_cycle_1, best_cycle_2 = random_cycle(data)
    best_cycle_1, best_cycle_2 = local_search(best_cycle_1, best_cycle_2, distance_matrix, data)
    best_length = calculate_cycles_length(best_cycle_1, best_cycle_2, distance_matrix)

    while time_elapsed <= time_MSLS:
        cycle1, cycle2 = severe_perturbation(best_cycle_1, best_cycle_2)
        cycle1, cycle2 = local_search(cycle1, cycle2, distance_matrix, data)
        length = calculate_cycles_length(cycle1, cycle2, distance_matrix)
        if length < best_length:
            best_length = length
            best_cycle_1 = cycle1
            best_cycle_2 = cycle2

    return best_cycle_1, best_cycle_2

function severe_perturbation(cycle1, cycle2, distance_matrix):
    original_size_cycle1 = length_of(cycle1)
    original_size_cycle2 = length_of(cycle2)

    random1 = get_random_nodes_of(cycle1)
    random2 = get_random_nodes_of(cycle2)

    cycle1 = swap_random_nodes_of(cycle1, random1)
    cycle2 = swap_random_nodes_of(cycle2, random2)

    # Repair
    cities = random1 + random2
    while(len(cities) > 0)
      if length_of(cycle1) < original_size_cycle1:
          greedy_cycle(cycle1, cities, distance_matrix)

      if length_of(cycle2) < original_size_cycle2:
          greedy_cycle(cycle2, cities, distance_matrix)

    return cycle1, cycle2
```

Wyniki eksperymentu obliczeniowego

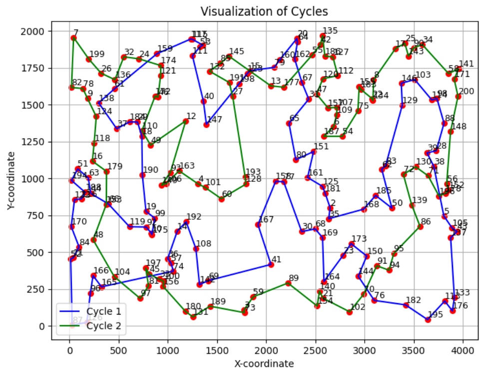
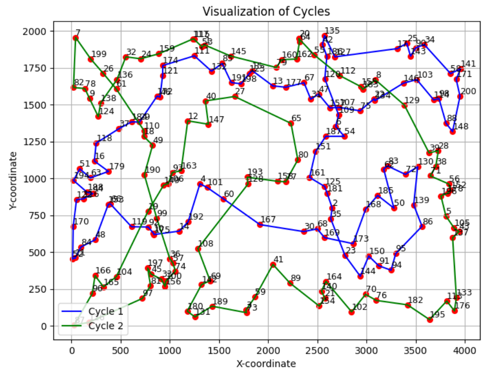
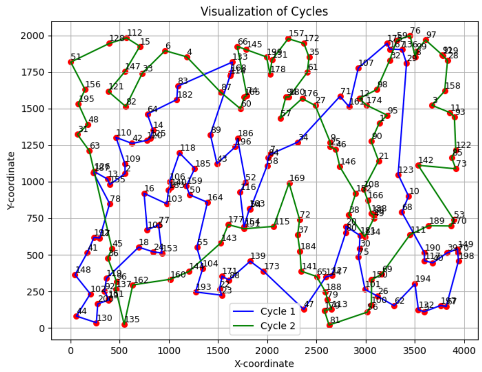
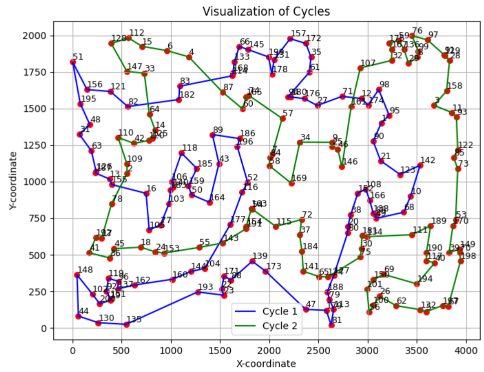
W tabeli przedstawiono sumy długości cykli dla każdej z metod dla obu instancji problemu oraz czasy wykonania.

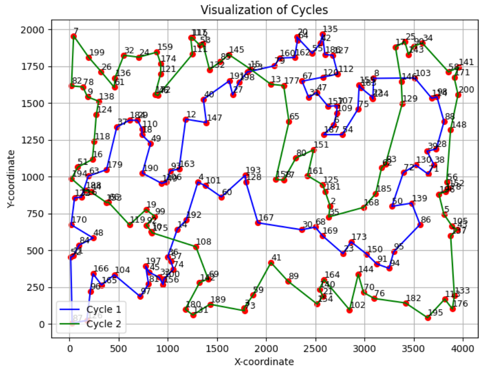
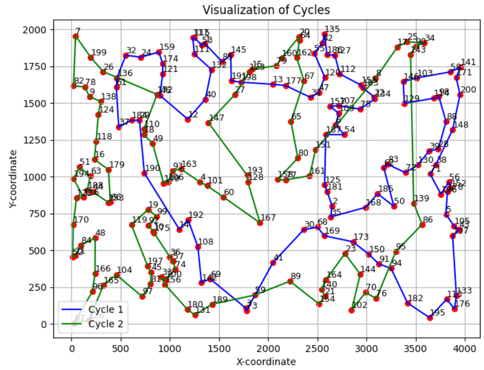
```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

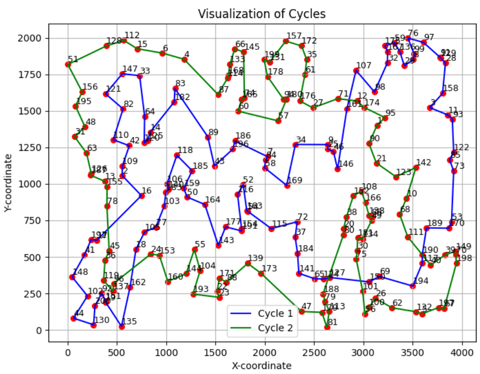
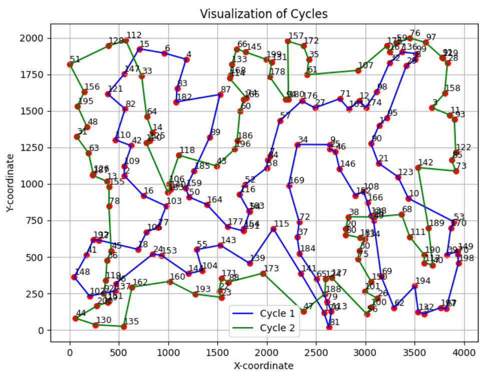
	Instance	Method	min time	mean time	max time	min length	mean length	max length
0	data/kroA200.tsp	ILS1	398.439000	401.822894	407.151261	35132.241805	36467.784606	37713.010272
1	data/kroA200.tsp	ILS2	390.630000	393.861476	398.987001	34584.205322	36029.483108	39188.511584
2	data/kroA200.tsp	ILS2a	401.236000	401.775724	402.434999	31287.750689	32032.848833	33039.638350
3	data/kroA200.tsp	MSLS	390.594031	393.835123	398.972001	35405.032448	36231.844409	36537.735383
4	data/kroB200.tsp	ILS1	396.210132	421.961807	477.018051	35305.112331	36128.036000	36766.275195
5	data/kroB200.tsp	ILS2	388.648000	413.776645	467.546440	33276.893733	36248.175578	39280.127460

	Instance	Method	min time	mean time	max time	min length	mean length	max length
6	data/kroB200.tsp	ILS2a	401.115705	401.606556	402.008951	32143.933715	32824.243708	34079.993762
7	data/kroB200.tsp	MSLS	388.607016	413.750936	467.529489	34859.235914	36378.274770	37180.009204

Wizualizacje najlepszych rozwiązań

	ILS1	ILS2
datakroA200.tsp		
datakroB200.tsp		

	ILS2a	MSLS
datakroA200.tsp		

	ILS2a	MSLS
datakroB200.tsp		

Wnioski

- Najlepsze wyniki uzyskano dla metody ILS2a, która jest wariantem ILS2 z przeszukiwaniem lokalnym.
- Wyniki dla metod są porównywalne z metodami z poprzednich laboratoriów.
- Rozmiar perturbacji nie pozwala na uzyskanie odseparowania cykli.

Kod programu

Kod programu znajduje się pod tym linkiem.