

# Inteligentne metody optymalizacji

## Laboratorium nr 5

24 stycznia 2024

Prowadzący: prof. dr hab. inż. Andrzej Jaskiewicz

Autorzy: **Krzysztof Martin** 141275  
**Bartosz Paliński** 145224

Zajęcia poniedziałkowe, 11:45.

Oświadczam/y, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższych autora/ów, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.

# 1 Opis zadania

Celem laboratorium jest implementacja implementacja hybrydowego algorytmu ewolucyjnego i porównanie go z metodami MSLS i ILS1 oraz ILS2 zaimplementowanymi na poprzednich laboratoriach.

W dalszej części sprawozdania przedstawione zostaną wyniki badań, wraz z analizą skuteczności poszczególnych metod oraz omówieniem wpływu zastosowanych mechanizmów na jakość wyników i czas przetwarzania.

Efektywność algorytmów była oceniana na podstawie instancji kroaA200 i kroB200 z biblioteki TSPLib. Do uruchomienia algorytmów wykorzystano losowe rozwiązania startowe.

## 2 Algorytmy

### 2.1 Hybrydowy algorytm ewolucyjny

- 1: *populationSize* = 20
- 2: *population* = []
- 3: **Powtarzaj** *populationSize* razy:
  - 4: wygeneruj losowe rozwiązanie - *tmpSolution*
  - 5: dokonaj optymalizacji rozwiązania *tmpSolution* korzystając z lokalnego przeszukiwania z pamięcią.
  - 6: **Jeżeli** *tmpSolution* nie istnieje w *population*:
  - 7:     Dodaj *tmpSolution* do *population*
- 8: **Powtarzaj**:
  - 9:     *doLocalSearch* = True/False
  - 10:     Wylosuj 2 rozwiązania A i B z populacji
  - 11:     Usuń z rozwiązania A wszystkie krawędzie, które nie występują w rozwiązaniu B
  - 12:     Usuń wierzchołki, które stały się wolne (zostały usunięte obie sąsiednie krawędzie)
  - 13:     Uruchom algorytm 2-żalu ważonego, aby uzupełnić powstałe rozwiązanie *Solution* o brakujące wierzchołki.
  - 14:     **Jeżeli** *doLocalSearch*:
    - 15:         Wynonaj algorytm lokalnego przeszukiwania na powstałym rozwiązaniu *Solution*
  - 16:     **Jeżeli** *Solution* jest lepsze niż najgorsze rozwiązanie w populacji *population*:
  - 17:         Zastąp najgorsze rozwiązanie w *population* rozwiązaniem *Solution*
  - 18:     **Jeżeli** upłynął maksymalny czas działania algorytmu:
  - 19:         Zwróć rozwiązanie *Solution* jako najlepsze rozwiązanie znalezione przez algorytm.

## 3 Wyniki eksperymentu obliczeniowego

Każdy algorytm został przetestowany 10 razy dla każdej instancji. Średnie, najlepsze oraz najgorsze wyniki są przedstawione w poniższej tabeli wraz z czasami wykonywania.

Tabela 1: Zestawienie wyników poszczególnych algorytmów

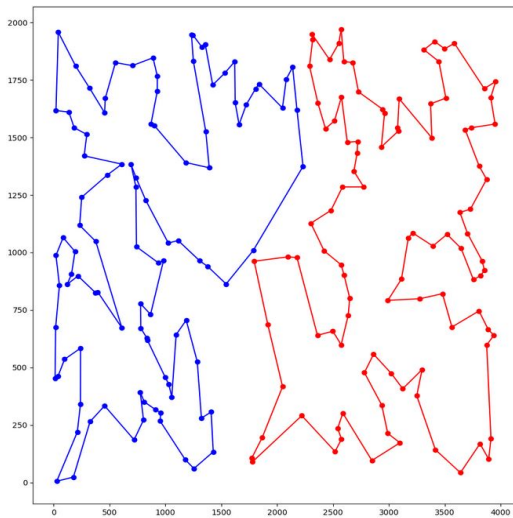
Lokalne przeszukiwanie	Instancja kroA200	Instancja kroB200
Hybrydowy ewolucyjny	32257 (31217 - 33302) - 24199 ms	32773 (30469 - 33863) - 23858ms
Hybrydowy ewolucyjny + LS	32477 (31192 - 33544) - 24199 ms	32551 (31202 - 33299) - 23858ms
MSLS z pamięcią	35845 (35459 - 36266) - 23982,2 ms	35573 (35160 - 35837) - 23748 ms
ILS1	35842 (35151 - 36559) - 24199 ms	36282 (35812 - 36663) - 23858 ms
ILS2	32915 (31510 - 34054) - 24199 ms	33091 (31390 - 37462) - 23858ms
ILS2+LS	35986 (34061 - 39211 ) - 24199 ms	34101 (32040 - 35314 ) - 23858ms

Tabela 2: Ilość iteracji poszczególnych algorytmów

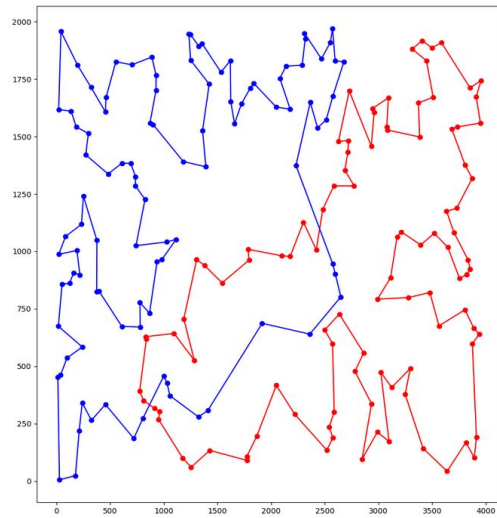
Lokalne przeszukiwanie	Instancja kroA200	Instancja kroB200
Hybrydowy ewolucyjny	324	329
Hybrydowy ewolucyjny + LS	281	295
MSLS z pamięcią	100	100
ILS1	6369	5952
ILS2	6470	6549
ILS2+LS	3798	3832

## 4 Wizualizacje najlepszych rozwiązań

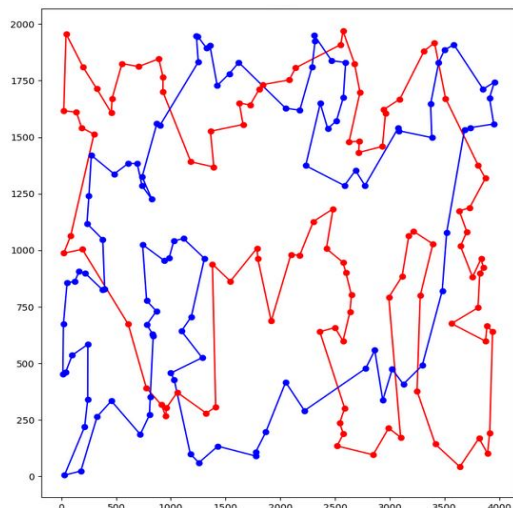
### 4.1 Instancja kroaA200



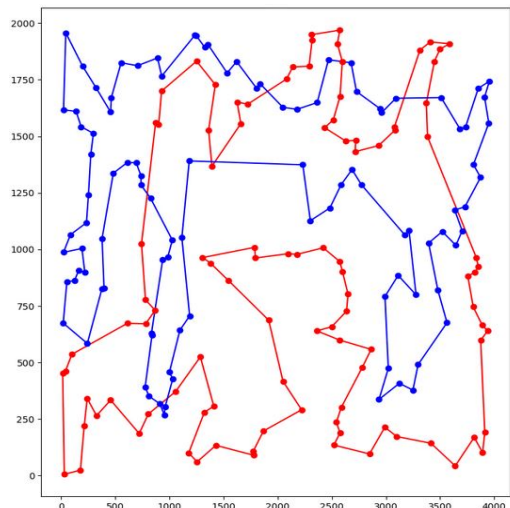
Rysunek 1: Hybrydowy algorytm ewolucyjny



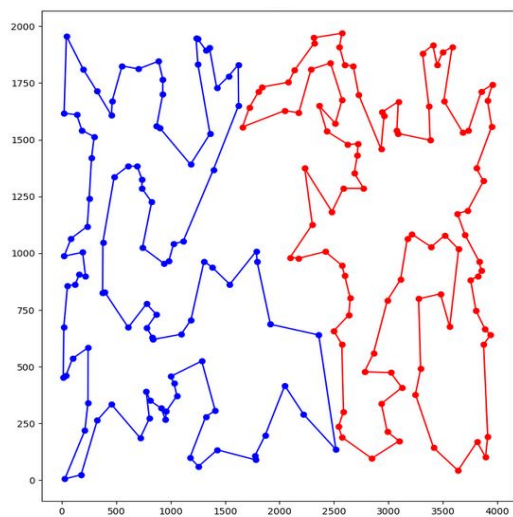
Rysunek 2: Hybrydowy ewolucyjny z LS



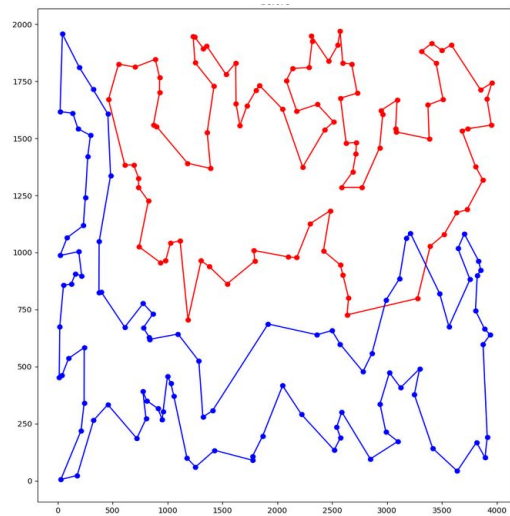
Rysunek 3: (MLSL)



Rysunek 4: ILS1

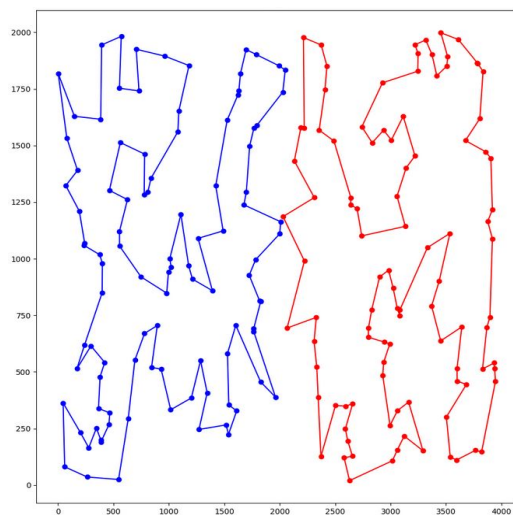


Rysunek 5: ILS2 wykorzystujący 2-żal

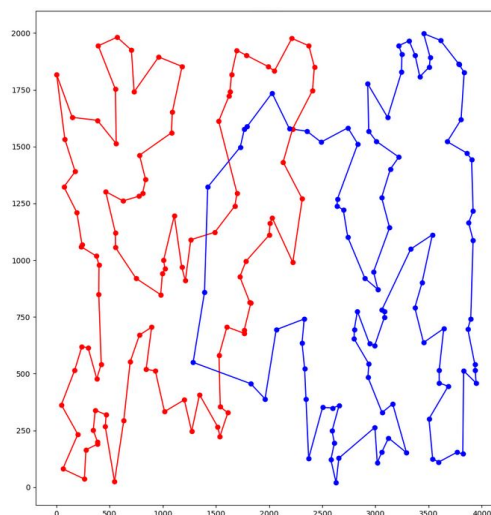


Rysunek 6: ILS2 z lokalnym przeszukiwaniem

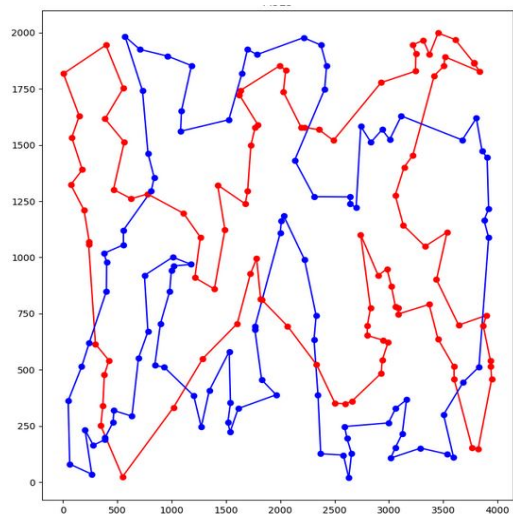
## 4.2 Instancja kroaB200



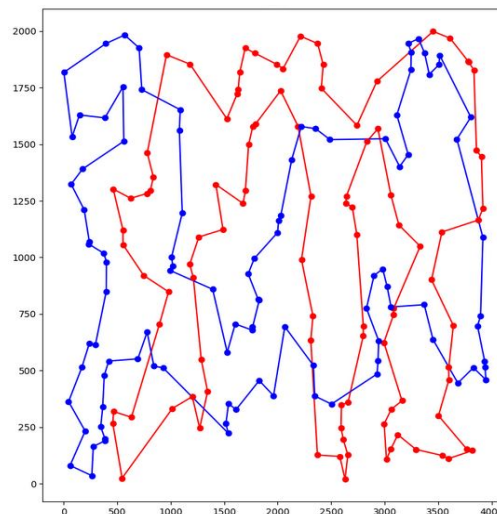
Rysunek 7: Hybrydowy algorytm ewolucyjny



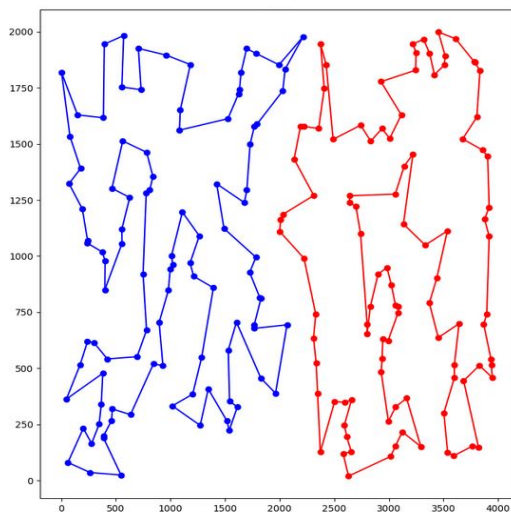
Rysunek 8: Hybrydowy ewolucyjny z LS



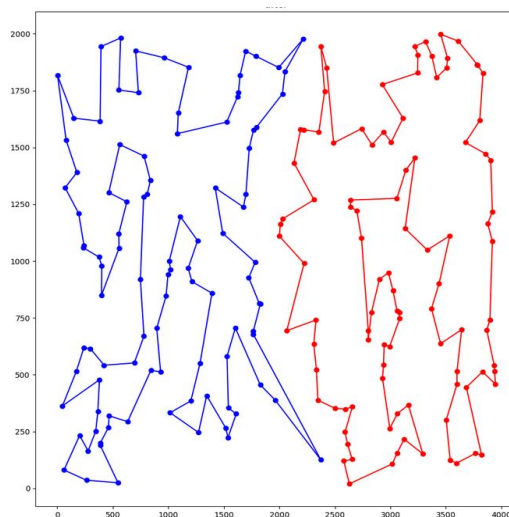
Rysunek 9: (MLSL)



Rysunek 10: ILS1



Rysunek 11: ILS2 wykorzystujący 2-żal



Rysunek 12: ILS2 z lokalnym przeszukiwaniem

## 5 Wnioski

- Hybrydowy algorytm ewolucyjny osiąga lepsze wyniki zarówno dla wartości minimalnej oraz średniej niż algorytmy MSLS, ILS1 oraz ILS2.
- Hybrydowy algorytm ewolucyjny wykorzystujący lokalne przeszukiwanie osiągnął najlepszy wynik dla instancji kroA200 wynoszący 31192 oraz kroB200 wynoszący 30469.
- Najlepsze rozwiązanie dla instancji kroA200 oraz kroB200, które uzyskano za pomocą hybrydowego algorytmu ewolucyjnego z lokalnym przeszukiwaniem, zawierają 2 cykle, które nie są od siebie w pełni odseparowane przestrzennie.
- Najlepsze rozwiązania hybrydowego algorytmu ewolucyjnego, zawierają cykle, które nie posiadają krawędzi przecinających się. Krawędzie przecinające się w danym cyklu są znacznikiem nieoptymalności danego rozwiązania. Najlepsze rozwiązania hybrydowego algorytmu ewolucyjnego nie zawierają takich nieoptymalności.

## 6 Kod programu

Kod źródłowy dostępny on-line: <https://github.com/barosz0/IMO/tree/main/lab5>