

Rozszerzenia lokalnego przeszukiwania

Autorzy: Dominik Maćkowiak 151915, Szymon Szymankiewicz 151821

Opis zadania

Celem zadania jest rozwiązanie zmodyfikowanego problemu komiwojażera poprzez znalezienie dwóch rozłącznych cykli, obejmujących wszystkie wierzchołki. Każdy cykl powinien zawierać połowę wierzchołków (lub o jeden więcej w przypadku nieparzystej liczby).

Dane są dwie instancje ("kroA200" oraz kroB200") z biblioteki TSPLib, które zawierają informacje o położeniu konkretnych wierzchołków. Celem zadania jest utworzenie z każdej instancji dwa rozłączne cykle, tak aby ich łączna długość była jak najmniejsza.

Zaimplementowano i porównano trzy metaheurystyki:

- **MSLS** – wielokrotne lokalne przeszukiwanie z losowych punktów startowych,
- **ILS** – iteracyjne przeszukiwanie lokalne z niewielką perturbacją,
- **LNS** – przeszukiwanie z dużą perturbacją typu *Destroy-Repair*.

Jako lokalne przeszukiwanie wykorzystano wersję stromą z zamianą krawędzi. Dla ILS zastosowano perturbację przez wymianę kilku elementów, a dla LNS – usunięcie i odbudowę ok. 30% rozwiązania.

Każdą metodę uruchamiano 10 razy, MSLS wykonywało 200 iteracji, a ILS i LNS działały przez czas równy średniemu czasowi MSLS.

Algorytmy

MSLS

Zainicjalizuj najlepsze znalezione rozwiązanie jako puste

Powtarzaj określoną liczbę razy (200 razy):

- a. Wygeneruj losowe rozwiązanie początkowe, składające się z dwóch cykli
- b. Zastosuj lokalne przeszukiwanie do poprawy tego rozwiązania:
 - Wykorzystaj algorytm lokalnego przeszukiwania **w wersji stromej (steepest)**
- c. Jeśli uzyskane rozwiązanie jest lepsze (ma krótszą długość cykli) niż dotychczas najlepsze, zapamiętaj je jako nowe najlepsze

Po wykonaniu wszystkich prób zwróć najlepsze znalezione rozwiązanie oraz jego długość

Perturbacja

Perturbacja polega na wykonaniu z szansą $\frac{1}{3}$ jednej z akcji:

- Zamiana wierzchołków między cyklami
- Zamiana wierzchołków w jednym cyklu ($\frac{1}{2}$ szans na wybranie cyklu 1 lub 2)
- Zamiana krawędzi w jednym cyklu ($\frac{1}{2}$ szans na wybranie cyklu 1 lub 2)

Wykonujemy 3 takie perturbacje

ILS

Wygeneruj początkowe rozwiązanie (dwa cykle)

Popraw je za pomocą lokalnego przeszukiwania:

- Wykorzystaj algorytm lokalnego przeszukiwania **w wersji stromej (steepest)**

Zainicjalizuj to rozwiązanie jako najlepsze znalezione

Powtarzaj aż do osiągnięcia limitu czasu (średni czas MSLS dla aktualnej instancji):

- a. Utwórz kopię aktualnie najlepszego rozwiązania
- b. Wprowadź perturbacje:
 - Wykonaj **3** losowe operacje typu swap, aby wydostać się z lokalnego optimum
- c. Zastosuj ponownie lokalne przeszukiwanie **Steepest** na zmodyfikowanym rozwiązaniu
- d. Jeżeli nowe rozwiązanie ma lepszą długość (niższą wartość funkcji celu), zaakceptuj je jako nowe najlepsze

Po upływie limitu czasu zwróć najlepsze znalezione rozwiązanie oraz jego długość

Destroy

Wśród dwóch cykli, które są rozwiązaniem, wybieramy losowo 30% wierzchołków

Usuujemy z cykli te wybrane wierzchołki

Jeśli zdarzy się że cykl nie jest zamknięty to go domykamy

Repair

Przywracamy usunięte wierzchołki do cykli w kolejności malejącego żalu

1. Oblicz żal dla wstawienia wierzchołka do cyklu 1 i 2
2. Wybierz cykl z większym żalem (lub mniejszym wzrostem długości w przypadku remisu)
3. Wstaw wierzchołek do tego cyklu

Zamknij cykle, jeśli nie są zamknięte

LNS

Wygeneruj początkowe rozwiązanie (dwa cykle)

Popraw je za pomocą przeszukiwania lokalnego (**steepest**)

Zainicjalizuj rozwiązanie jako najlepsze

Powtarzaj aż do osiągnięcia limitu czasu (średni czas MSLS dla aktualnej instancji):

- a. Utwórz kopię najlepszego znalezionej rozwiązania
- b. **Destroy** - losowo usuń część wierzchołków z obu cykli (**30% wierzchołków**), pozostawiając resztę niezmienną
- c. **Repair** – odbuduj cykle, wstawiając usunięte wierzchołki ponownie, stosując **ważony 2-żal**
- d. Zastosuj lokalne przeszukiwanie (**steepest**) na naprawionym rozwiązaniu
- e. Jeżeli nowe rozwiązanie jest lepsze (ma mniejszą długość cykli), zaakceptuj je jako nowe najlepsze

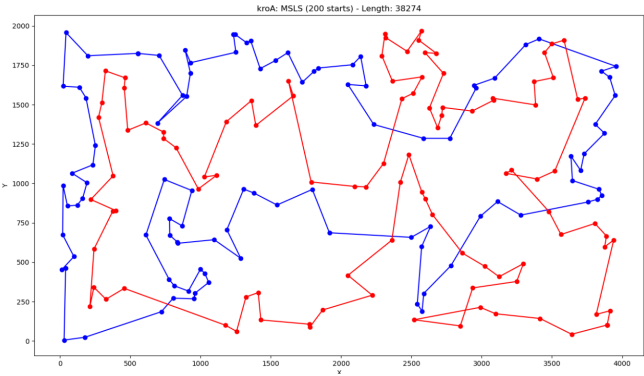
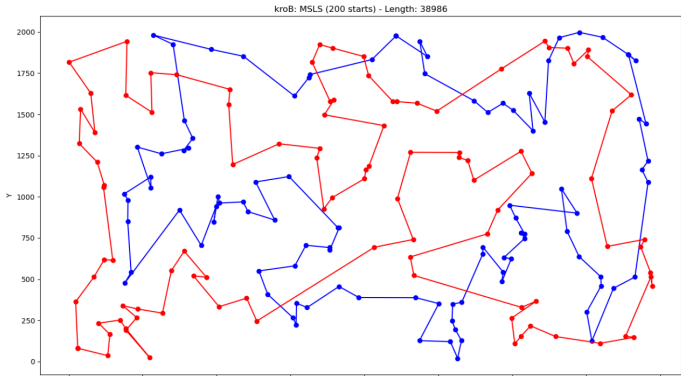
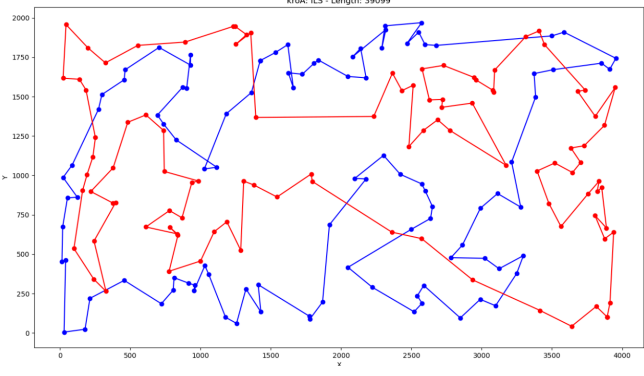
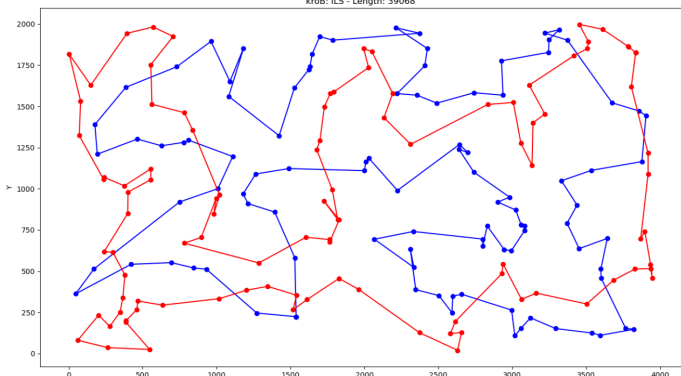
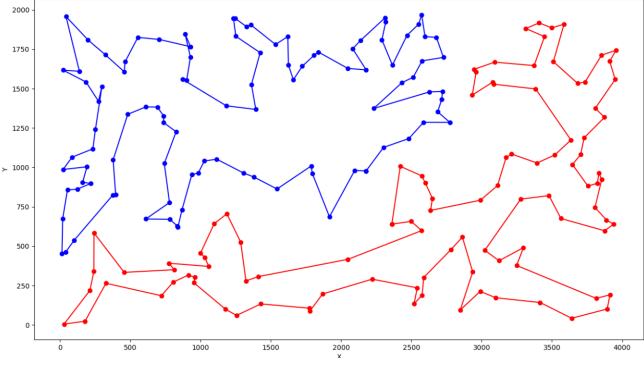
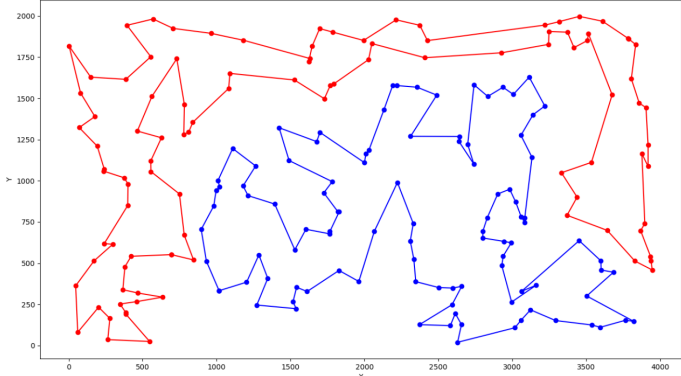
Zwróć najlepsze znalezione rozwiązanie oraz jego długość po upływie limitu czasu

Wyniki eksperymentu obliczeniowego

Przeprowadziliśmy eksperyment, który polegał na zmierzeniu długości dwóch utworzonych cykli. Dla każdej instancji przeprowadziliśmy 100 testów, gdzie każdy działał przez 65 sekund (średni czas MSLS). Obliczyliśmy średnią długość z tych testów, oraz wybraliśmy odcinek najdłuższy i najkrótszy.

	kroA200			kroB200		
	średnia	min	max	średnia	min	max
MSLS		38274			38986	
ILS		39099			39068	
LNS		37526			36702	

Wizualizacja dla najlepszych rozwiązań

	kroA200	kroB200
MSLS	 <p>kroA: MSLS (200 starts) - Length: 38274</p> <p>This plot shows the MSLS solution for kroA200. The x-axis represents the number of vertices (v) from 0 to 4000, and the y-axis represents the number of edges (e) from 0 to 2000. Two paths are shown: a blue path and a red path. The blue path is generally higher on the y-axis than the red path, indicating a higher edge count for the same number of vertices.</p>	 <p>kroB: MSLS (200 starts) - Length: 38986</p> <p>This plot shows the MSLS solution for kroB200. The x-axis represents the number of vertices (v) from 0 to 4000, and the y-axis represents the number of edges (e) from 0 to 2000. Two paths are shown: a blue path and a red path. The blue path is generally higher on the y-axis than the red path, indicating a higher edge count for the same number of vertices.</p>
ILS	 <p>kroA: ILS - Length: 39099</p> <p>This plot shows the ILS solution for kroA200. The x-axis represents the number of vertices (v) from 0 to 4000, and the y-axis represents the number of edges (e) from 0 to 2000. Two paths are shown: a blue path and a red path. The blue path is generally higher on the y-axis than the red path, indicating a higher edge count for the same number of vertices.</p>	 <p>kroB: ILS - Length: 39068</p> <p>This plot shows the ILS solution for kroB200. The x-axis represents the number of vertices (v) from 0 to 4000, and the y-axis represents the number of edges (e) from 0 to 2000. Two paths are shown: a blue path and a red path. The blue path is generally higher on the y-axis than the red path, indicating a higher edge count for the same number of vertices.</p>
LNS	 <p>kroA: LNS (regret repair) - Length: 32512</p> <p>This plot shows the LNS (regret repair) solution for kroA200. The x-axis represents the number of vertices (v) from 0 to 4000, and the y-axis represents the number of edges (e) from 0 to 2000. Two paths are shown: a blue path and a red path. The blue path is generally higher on the y-axis than the red path, indicating a higher edge count for the same number of vertices.</p>	 <p>kroB: LNS (regret repair) - Length: 33429</p> <p>This plot shows the LNS (regret repair) solution for kroB200. The x-axis represents the number of vertices (v) from 0 to 4000, and the y-axis represents the number of edges (e) from 0 to 2000. Two paths are shown: a blue path and a red path. The blue path is generally higher on the y-axis than the red path, indicating a higher edge count for the same number of vertices.</p>

Wnioski

Kod programu: <https://github.com/szymon240/imo-lab4>