

Problem komiwojażera

Adrian Stępień i Wojciech Młyńczak

6 kwietnia 2020

1 Zadanie 2

1.1 Opis zadania

Zadanie polega na implementacji lokalnego przeszukiwania dla zmodyfikowanego problemu komiwojażera. Lokalne przeszukiwanie my być zaimplementowane w wersjach stromej i zachłannej z dwoma różnymi rodzajami sąsiedztwa.

1.2 Opis zaimplementowanych algorytmów

1.2.1 Algorytm lokalnego przeszukiwania w wersji zachłannej dla zamiany wierzchołków

```
1 Wygeneruj losowe rozwiązanie.
2 Dopóki nowe rozwiązanie jest lepsze:
3     Oznacz, że nowe rozwiązanie jest gorsze.
4     Wygeneruj losową serię.
5     Dla każdego punktu w serii:
6         Oblicz dystans aktualnego rozwiązania.
7         Wygeneruj drugą losową serię.
8         Dla każdego punktu w drugiej losowej serii:
9             Jeżeli wybrany punkt jest taki sam jak punkt z
                pierwszej serii, to przejdź do następnego
                punktu z drugiej serii.
10        Oblicz deltę.
11        Jeżeli delta jest mniejsza od 0:
12            Zamień punkt z rozwiązania wskazany, przez
                pierwszą losową serię, na punkt z
                drugiej serii, który jest poza
                rozwiązaniem.
13        Oznacz, że nowe rozwiązanie jest lepsze.
```

1.2.2 Algorytm lokalnego przeszukiwania w wersji stromej dla zamiany wierzchołków

```
1 Wygeneruj losowe rozwiązanie.
2 Dopóki nowe rozwiązanie jest lepsze:
3     Oznacz, że nowe rozwiązanie jest gorsze.
4     Wyzeruj najlepszą deltę i najlepsze punkty do zamiany.
5     Wygeneruj losową serię.
```

```

6     Dla każdego punktu w serii:
7         Oblicz dystans aktualnego rozwiązania.
8         Wygeneruj drugą losową serię.
9         Dla każdego punktu w drugiej losowej serii:
10            Jeżeli wybrany punkt jest taki sam jak punkt z
                pierwszej serii, to przejdź do następnego
                punktu z drugiej serii.
11            Oblicz deltę.
12            Jeżeli delta jest lepsza od najlepszej delty:
13                Zamień najlepszą deltę na obliczoną deltę.
14                Zamień znalezione punkty z obecnymi
                    punktami do zamiany.
15                Oznacz, że nowe rozwiązanie jest lepsze.
16     Jeżeli najlepsza delta jest mniejsza od 0, to zamień
        znalezione dwa punkty ze sobą.

```

1.2.3 Algorytm lokalnego przeszukiwania w wersji zachłannej dla zamiany krawędzi

```

1     Wygeneruj losowe rozwiązanie.
2     Dopóki nowe rozwiązanie jest lepsze:
3         Oznacz, że nowe rozwiązanie jest gorsze.
4         Wyzeruj najlepszą deltę i najlepsze punkty do zamiany.
5         Wygeneruj losową serię.
6         Dla każdego punktu w serii:
7             Oblicz dystans aktualnego rozwiązania.
8             Wygeneruj drugą losową serię bez punktów z
                aktualnego rozwiązania.
9             Dla każdego punktu w drugiej losowej serii:
10                Jeżeli wybrany punkt jest taki sam jak punkt z
                    pierwszej serii, to przejdź do następnego
                    punktu z drugiej serii.
11                Oblicz deltę.
12                Jeżeli delta jest mniejsza od zera
13                    Zamień znalezione dwa punkty ze sobą.
14                    Oznacz, że nowe rozwiązanie jest lepsze.
15         Wygeneruj losową serię.
16         Dla każdego punktu w serii:
17             Oblicz dystans aktualnego rozwiązania.
18             Wygeneruj drugą losową serię.
19             Dla każdego punktu w drugiej losowej serii:
20                Jeżeli wybrany punkt jest taki sam jak punkt z
                    pierwszej serii lub pierwszy i drugi punkt
                    sąsiadują za sobą, to przejdź do następnego
                    punktu z drugiej serii.
21                Oblicz deltę.
22                Jeżeli delta jest mniejsza od zera
23                    Zamień znalezione dwa punkty ze sobą.
24                Oznacz, że nowe rozwiązanie jest lepsze.

```

1.2.4 Algorytm lokalnego przeszukiwania w wersji stromej dla zamiany krawędzi

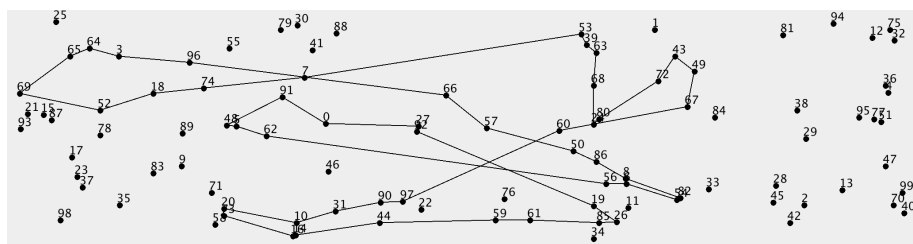
```
1 Wygeneruj losowe rozwiązanie.
2 Dopóki nowe rozwiązanie jest lepsze:
3   Oznacz, że nowe rozwiązanie jest gorsze.
4   Wyzeruj najlepszą deltę i najlepsze punkty do zamiany.
5   Wygeneruj losową serię.
6   Dla każdego punktu w serii:
7     Oblicz dystans aktualnego rozwiązania.
8     Wygeneruj drugą losową serię bez punktów z
       aktualnego rozwiązania.
9     Dla każdego punktu w drugiej losowej serii:
10      Jeżeli wybrany punkt jest taki sam jak punkt z
        pierwszej serii, to przejdź do następnego
        punktu z drugiej serii.
11      Oblicz deltę.
12      Jeżeli delta jest lepsza od najlepszej delty:
13        Zamień najlepszą deltę na obliczoną deltę.
14        Zamień znalezione punkty z obecnymi
          punktami do zamiany.
15        Oznacz, że nowe rozwiązanie jest lepsze.
16   Wygeneruj losową serię.
17   Dla każdego punktu w serii:
18     Oblicz dystans aktualnego rozwiązania.
19     Wygeneruj drugą losową serię.
20     Dla każdego punktu w drugiej losowej serii:
21      Jeżeli wybrany punkt jest taki sam jak punkt z
        pierwszej serii lub pierwszy i drugi punkt
        sąsiadują za sobą, to przejdź do następnego
        punktu z drugiej serii.
22      Oblicz deltę.
23      Jeżeli delta jest lepsza od najlepszej delty:
24        Zamień najlepszą deltę na obliczoną deltę.
25        Zamień znalezione punkty z obecnymi
          punktami do zamiany.
26      Oznacz, że nowe rozwiązanie jest lepsze.
27   Jeżeli najlepsza delta jest mniejsza od 0, to zamień
       znalezione dwa punkty ze sobą.
```

1.3 Wyniki pomiarów

Pomiar	Wartość średnia	Wartość minimalna	Wartość maksymalna	Średni czas [ms]	Minimalny czas [ms]	Maksymalny czas [ms]
Zamiana wierzchołków w wersji stromej dla kroA100	18116.54	14316.00	16646.00	701	436	925
Zamiana wierzchołków w wersji zachłannej dla kroA100	16747.61	13694.00	16207.00	116	71	224
Zamiana krawędzi w wersji stromej dla kroA100	12545.69	10769.00	11889.00	685	517	858
Zamiana krawędzi w wersji zachłannej dla kroA100	13469.44	11091.00	14262.00	56	26	92
Zamiana wierzchołków w wersji stromej dla kroB100	17721.50	14033.00	15635.00	717	504	1108
Zamiana wierzchołków w wersji zachłannej dla kroB100	16873.84	12300.00	15330.00	106	72	158
Zamiana krawędzi w wersji stromej dla kroB100	12541.73	11199.00	13471.00	665	502	835
Zamiana krawędzi w wersji zachłannej dla kroB100	13261.47	11477.00	13227.00	60	25	115

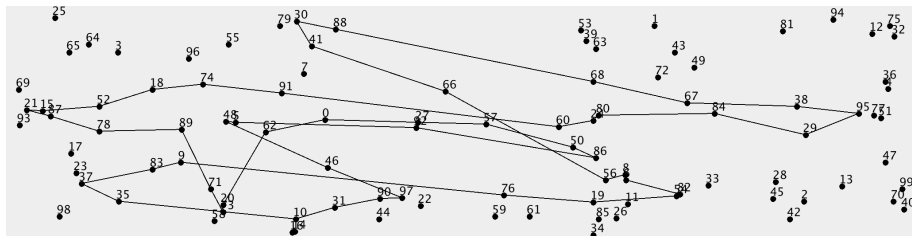
1.4 Wizualizacje najlepszych rozwiązań

1.4.1 Zamiana wierzchołków w wersji stromej dla kroA100



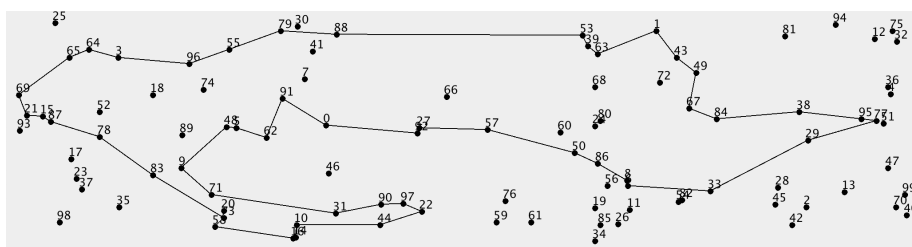
Rysunek 1: Zamiana wierzchołków w wersji stromej dla kroA100

1.4.2 Zamiana wierzchołków w wersji zachłannej dla kroA100



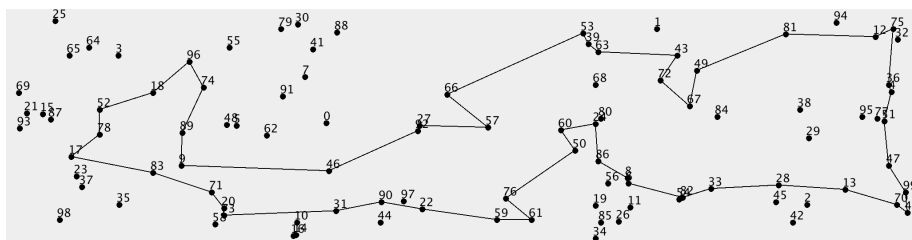
Rysunek 2: Zamiana wierzchołków w wersji zachłannej dla kroA100

1.4.3 Zamiana krawędzi w wersji stromej dla kroA100



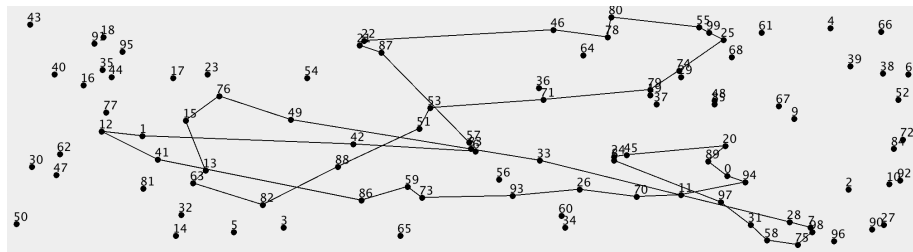
Rysunek 3: Zamiana krawędzi w wersji stromej dla kroA100

1.4.4 Zamiana krawędzi w wersji zachłannej dla kroA100



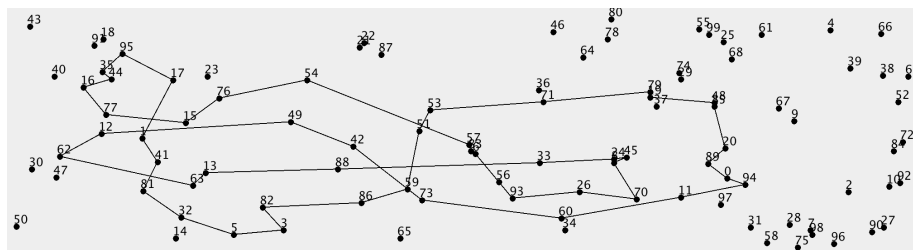
Rysunek 4: Zamiana krawędzi w wersji zachłannej dla kroA100

1.4.5 Zamiana wierzchołków w wersji stromej dla kroB100



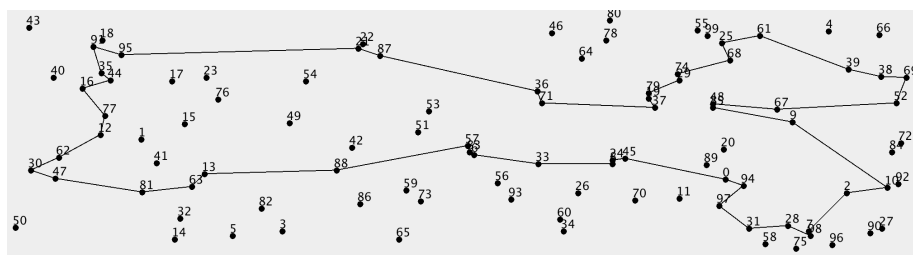
Rysunek 5: Zamiana wierzchołków w wersji stromej dla kroB100

1.4.6 Zamiana wierzchołków w wersji zachłannej dla kroB100



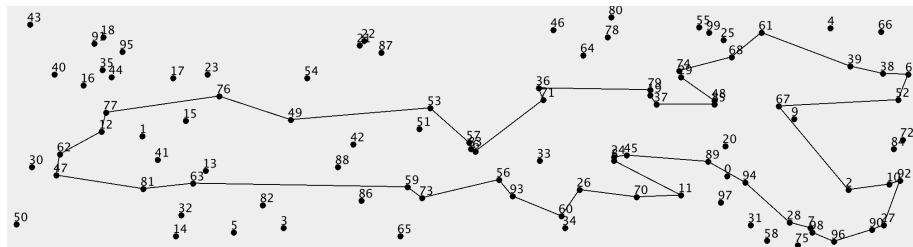
Rysunek 6: Zamiana wierzchołków w wersji zachłannej dla kroB100

1.4.7 Zamiana krawędzi w wersji stromej dla kroB100



Rysunek 7: Zamiana krawędzi w wersji stromej dla kroB100

1.4.8 Zamiana krawędzi w wersji zachłannej dla kroB100



Rysunek 8: Zamiana krawędzi w wersji zachłannej dla kroB100

1.5 Wnioski

Z wymienionych wyżej pomiarów można wywnioskować, że dla podanych warunków problemu zamiana kolejności wierzchołków jest lepsza od wymiany wierzchołków w rozwiązaniu na wierzchołki poza rozwiązaniem. Można również zaobserwować, że wersja zachłanna dla zamiany wierzchołków na wierzchołki poza rozwiązaniem daje średnio lepsze wyniki od wersji stromej. Dla zamiany kolejności wierzchołków rozwiązania wyniki są odwrotne - wersja stroma daje średnio lepsze rezultaty od wersji zachłannej.

2 Zadanie 1

2.1 Opis zadania

Rozważany problem to zmodyfikowany problem komiwojażera. Dany jest zbiór wierzchołków i macierz odległości pomiędzy każdą parą wierzchołków. Celem zadania jest znalezienie najkrótszej ścieżki zamkniętej przechodzącej przez 50% wszystkich wierzchołków (w przypadku nieparzystej liczby wierzchołków liczba jest zaokrąglana w górę).

2.2 Opis zaimplementowanych algorytmów

2.2.1 Algorytm zachłanny greedy cycle

```
1 Wybierz pierwszy punkt.
2 Wybierz drugi punkt leżący najbliżej pierwszego.
3 Jeżeli nie dodałeś wszystkich punktów:
4   Dla pozostałych wolnych punktów:
5     Dla każdej krawędzi w aktualnym rozwiązaniu:
6       Oblicz koszt dodania punktu do rozwiązania w danej
        krawędzi.
7       Sprawdź czy to jest najlepsze rozwiązanie w danym
        momencie.
8   Dodaj znaleziony najlepszy punkt w wybranej krawędzi do
    cyklu.
```

2.2.2 Algorytm z żalem oparty o 1-żal

```
1 Wybierz pierwszy punkt.
2 Wybierz drugi punkt leżący najbliżej pierwszego.
3 Jeżeli nie dodałeś wszystkich punktów:
4   Dla pozostałych wolnych punktów:
5     Dla każdej krawędzi w aktualnym rozwiązaniu:
6       Oblicz koszt dodania punktu do rozwiązania w danej
        krawędzi.
7       Dodaj punkt do listy potencjalnych rozwiązań wraz z
        kosztem dodania.
8     Oblicz żal dla danego punktu
9   W liście potencjalnych rozwiązań znajdź rozwiązanie z
    największym żalem.
10 Dodaj znalezione rozwiązanie z największym żalem do cyklu.
```

2.3 Wyniki pomiarów

2.3.1 Algorytm zachłanny dla problemu kroA100

Pomiar	Wynik
Wartość średnia	12898.45
Wartość minimalna	11325.00
Wartość maksymalna	14067.00

2.3.2 Algorytm oparty o żal dla problemu kroA100

Pomiar	Wynik
Wartość średnia	16879.11
Wartość minimalna	14456.00
Wartość maksymalna	17899.00

2.3.3 Algorytm zachłanny dla problemu kroB100

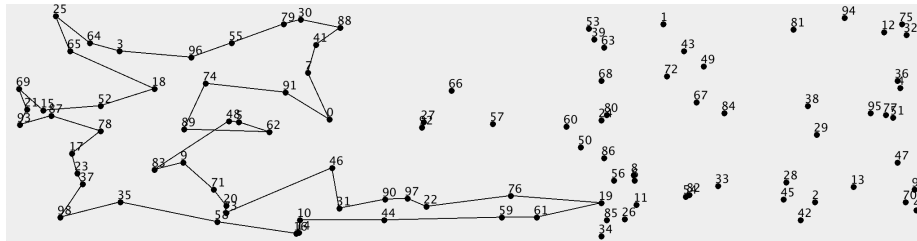
Pomiar	Wynik
Wartość średnia	12710.59
Wartość minimalna	10240.00
Wartość maksymalna	11320.00

2.3.4 Algorytm oparty o żal dla problemu kroB100

Pomiar	Wynik
Wartość średnia	17245.51
Wartość minimalna	15547.00
Wartość maksymalna	16965.00

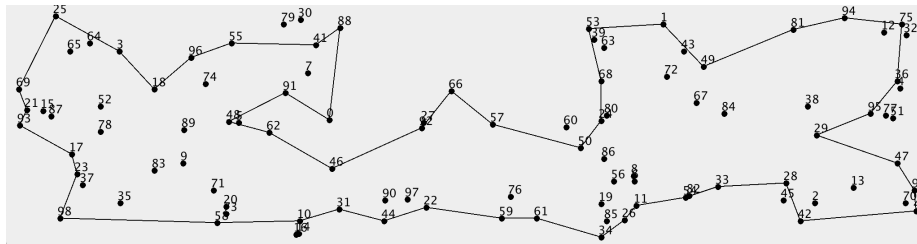
2.4 Wizualizacje najlepszych rozwiązań

2.4.1 Algorytm zachłanny dla problemu kroA100



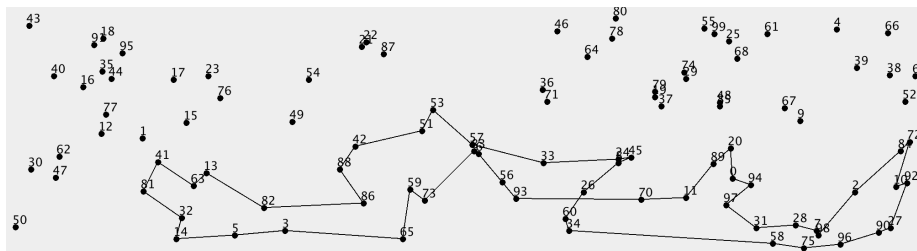
Rysunek 9: Algorytm zachłanny dla problemu kroA100

2.4.2 Algorytm oparty o żal dla problemu kroA100



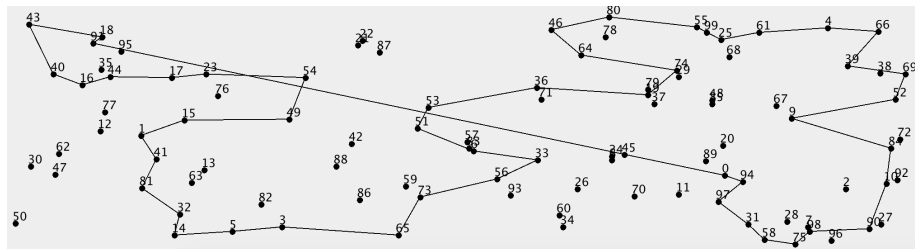
Rysunek 10: Algorytm oparty o żal dla problemu kroA100

2.4.3 Algorytm zachłanny dla problemu kroB100



Rysunek 11: Algorytm zachłanny dla problemu kroB100

2.4.4 Algorytm oparty o żal dla problemu kroB100



Rysunek 12: Algorytm oparty o żal dla problemu kroB100

2.5 Wnioski

Z wymienionych wyżej pomiarów można wywnioskować, że dla podanych warunków problemu (odwiedzanie połowy punktów), algorytm zachłanny radzi sobie lepiej od algorytmu opartego o żal (cykl, który generuje ma mniejszą długość). Przeprowadzono również testy dla przypadku, gdy oba te algorytmy uruchomione zostaną dla wszystkich punktów. Wtedy wyniki są odmienne, algorytm z żalem okazuje się lepszy od algorytmu zachłannego. Jest to spowodowane tym, że dla warunków zadania z odwiedzeniem połowy punktów algorytm z żalem czasami dodaje punkty, które mają duży żal, a w ogóle nie powinny zostać dodane do cyklu z powodu dużego kosztu ich dodania. Gdy odwiedzone mają być wszystkie punkty, koszt dodania punktu nie ma takiego znaczenia, ponieważ prędzej lub później i tak każdy punkt będzie musiał zostać dodany.

3 Kod programu

Repozytorium z kodem programu dostępne jest pod adresem: <https://github.com/adrianstepienfsw/AEM1>