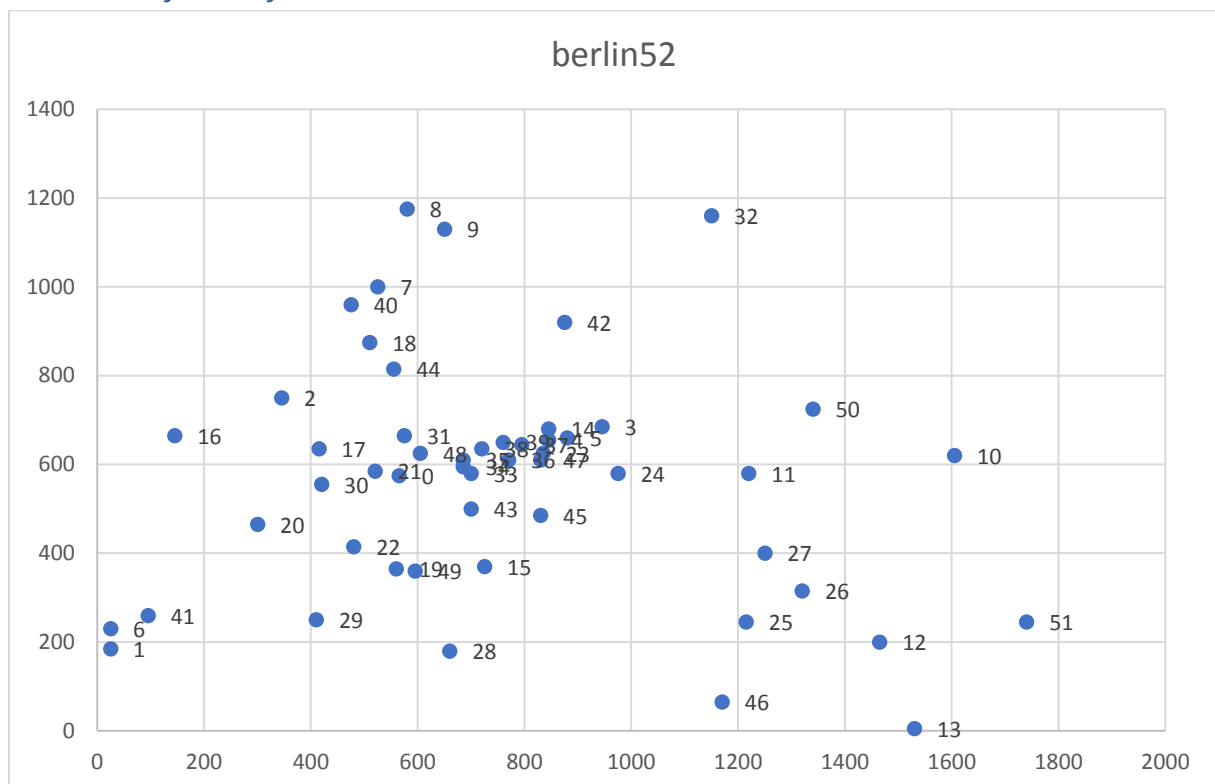


Problem Komiwojażera (algorytm genetyczny)

1. Inicjalizacja



2. Opis algorytmu

Algorytm interpretuje wczytaną instancję początkową jako tablice sąsiedztwa (zamiast stosować prawda-falsz jako jest krawędź-nie ma krawędzi, używamy tutaj wartości liczbowych określających odległość między wierzchołkami). Na tej podstawie tworzy heurystyczną trasę wybierając za każdym razem nieodwiedzony wierzchołek do którego jest najbliżej. Pierwsza (najkrótsza) trasa jest zapisywana. Następnie tworzona jest populacja, pierwszym jej elementem jest heurystyczna trasa, wszystkie pozostałe to zmutowana losową ilość razy pierwsza trasa (mutacja polega na zamienieniu 2 wierzchołków nie startowych miejscami (np. [0,1,2,3,4,5,0] po pojedynczej mutacji może wyglądać tak: [0,1,5,3,4,2,0])). Populacja jest sortowana według odległości jaką trzeba przebyć żeby ją „przejechać” w podanej kolejności. Część najlepszych tras trafia do następnej populacji i tworzy potomstwo. Potomstwo jest tworzone w następujący sposób: dwa losowo dobrane wyniki z poprzedniej populacji, który był na tyle dobry, żeby zostać przeniesiony tworzą parę, para tworzy potomka jak w przykładzie: Rodzic 1 -> 0123450, Rodzic 2 -> 0543210, usuwamy 0 z rodziców (punkt startowy do którego wracamy na końcu), tworzymy potomka zaczynającego się od 0, losowo wybieramy rodzica, z którego pobierzemy pierwszą od lewej wartość i wpisujemy ją do potomka, na

potrzeby przykładu wybierzmy 1 (potomek obecnie: 01), usuwamy pobraną wartość z obu rodziców (rodzic 1 -> 42345, rodzic 2 – 54324), uzupełniamy tak potomka i dopisujemy mu 0 na końcu. Uzupełniamy nową populację potomkami i powtarzamy proces wyboru najlepszych osobników w populacji. Jeżeli któryś z osobników jest najlepszy zbyt długo, tworzymy nową populację dodając najlepszych osobników do nowej populacji i tworząc nowych poprzez mutacje trasy wygenerowanej przez algorytm heurystyczny.

3. Pseudokod

funkcja losuj (start, koniec):

- zwróć losową liczbę z przedziału od start do end

funkcja mutuj(genotyp, V):

- powtarzaj:

 - losuj dwa geny

- jeżeli są różne zamień je miejscami i zwróć nowy genotyp

funkcja potomek(par1, par2):

- utwórz pusty genotyp dziecka

- dopóki rodzic1 i rodzic2 nie są puste:

 - losowo wybierz gen z rodzica 1 lub rodzica 2

 - dodaj wybrany gen do genomu dziecka

 - usuń wybrany gen z rodzica 1 i rodzica 2

- zwróć genom dziecka

funkcja dystans(wierzchołek 1, wierzchołek 2):

- zwróć odległość między wierzchołkiem 1 a wierzchołkiem 2

funkcja heurystyka(ilość wierzchołków, macierz sąsiedztwa):

- utwórz pustą trasę

- ustaw aktualny wierzchołek na 0

- oznacz aktualny wierzchołek jako odwiedzony

- dodaj aktualny wierzchołek do trasy

- dla i od 1 do ilość wierzchołków-1:

znajdź następny nieodwiedzony wierzchołek z najmniejszym dystansem od aktualnego wierzchołka

dodaj następny wierzchołek do trasy i oznacz go jako odwiedzony

ustaw aktualny wierzchołek na następny wierzchołek

dodaj startowy wierzchołek, aby ukończyć trasę

oblicz całkowity dystans dla trasy

utwórz osobnika z genotypem jako trasa i fitness jako obliczony dystans

zwróć tego osobnika

funkcja oblicz fitness(genotyp):

oblicz fitness jako całkowitą odległość trasy reprezentowanej przez genotyp

zwróć fitness

funkcja algorytm genetyczny (map, V):

utwórz pustą populację

wygeneruj pierwszego osobnika używając funkcji heurystyka

dodaj go do populacji

wygeneruj resztę populacji mutując pierwszego osobnika

powtarzaj aż nie upłyną 3 minuty

posortuj populację według fitnessu

zapisz pierwszego osobnika jako najkrótszy

utwórz nową populację

jeśli najlepszy osobnik nie dominuje zbyt długo:

wybierz procent najlepszych osobników którzy przejdą do następnej populacji

i dodaj do nowej populacji

dla pozostałych osobników:

wybierz dwoje rodziców losowo spośród elitarnych osobników

wygeneruj potomka za pomocą funkcji potomek

dodaj potomka do nowej populacji

w przeciwnym razie:

wybierz procent najlepszych osobników i dodaj ich do nowej populacji

dla pozostałych osobników:

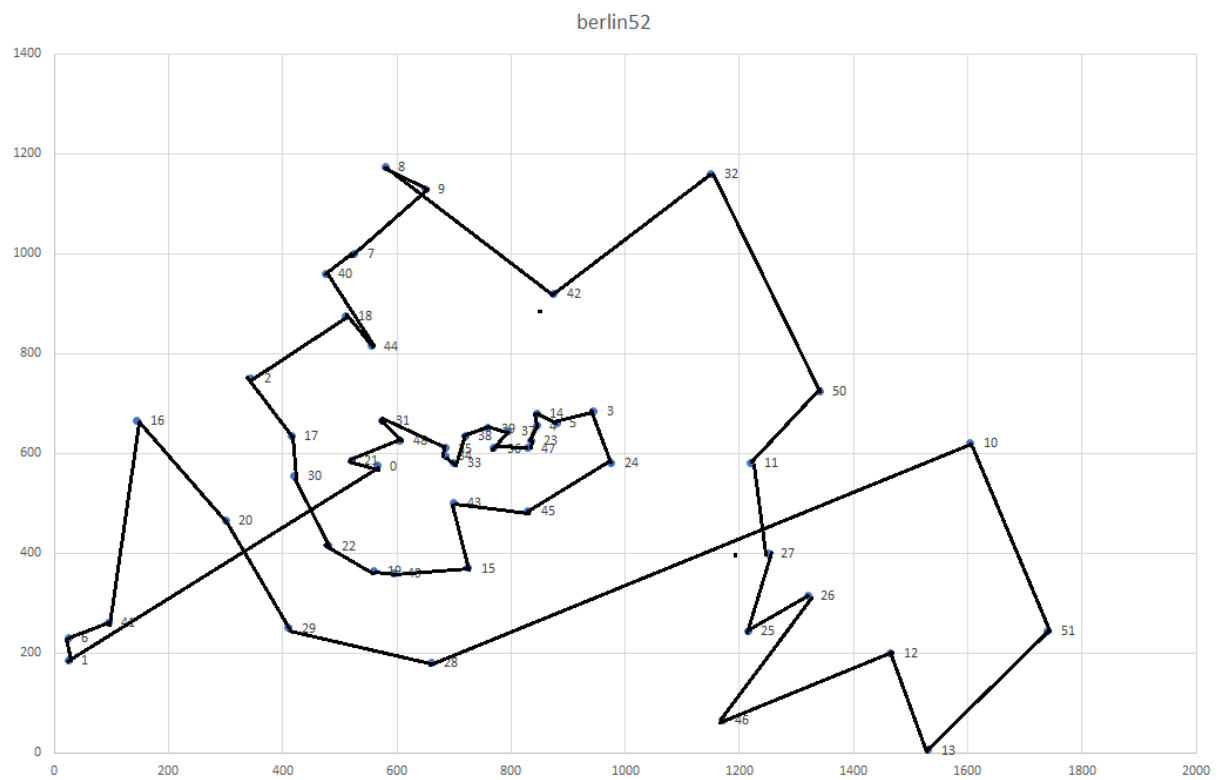
generuj nowych osobników mutując losową ilość razy trasę wybraną przez funkcję heurystyka

ustaw nową populację jako aktualną populację

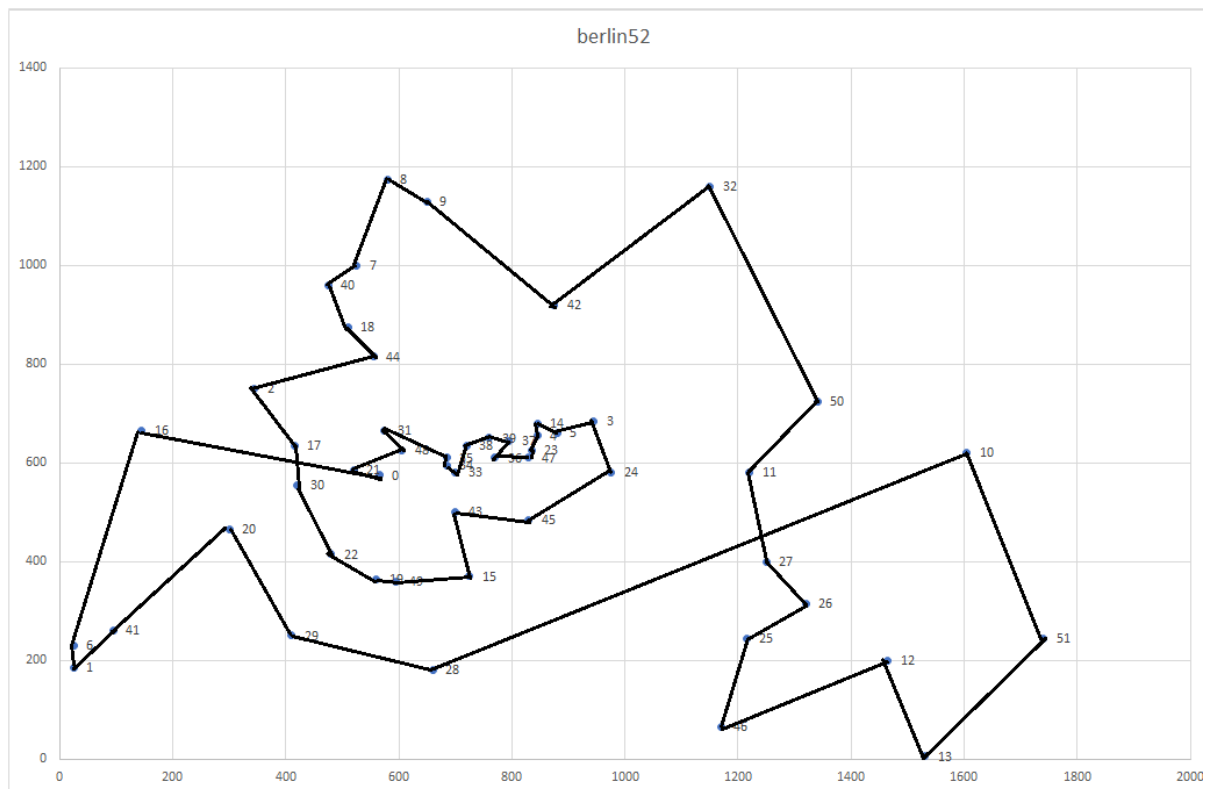
gdy pętla się zakończy, wypisz najlepszą trasę i jej długość.

4. Przykład obrazujący działanie

Heurystyka:



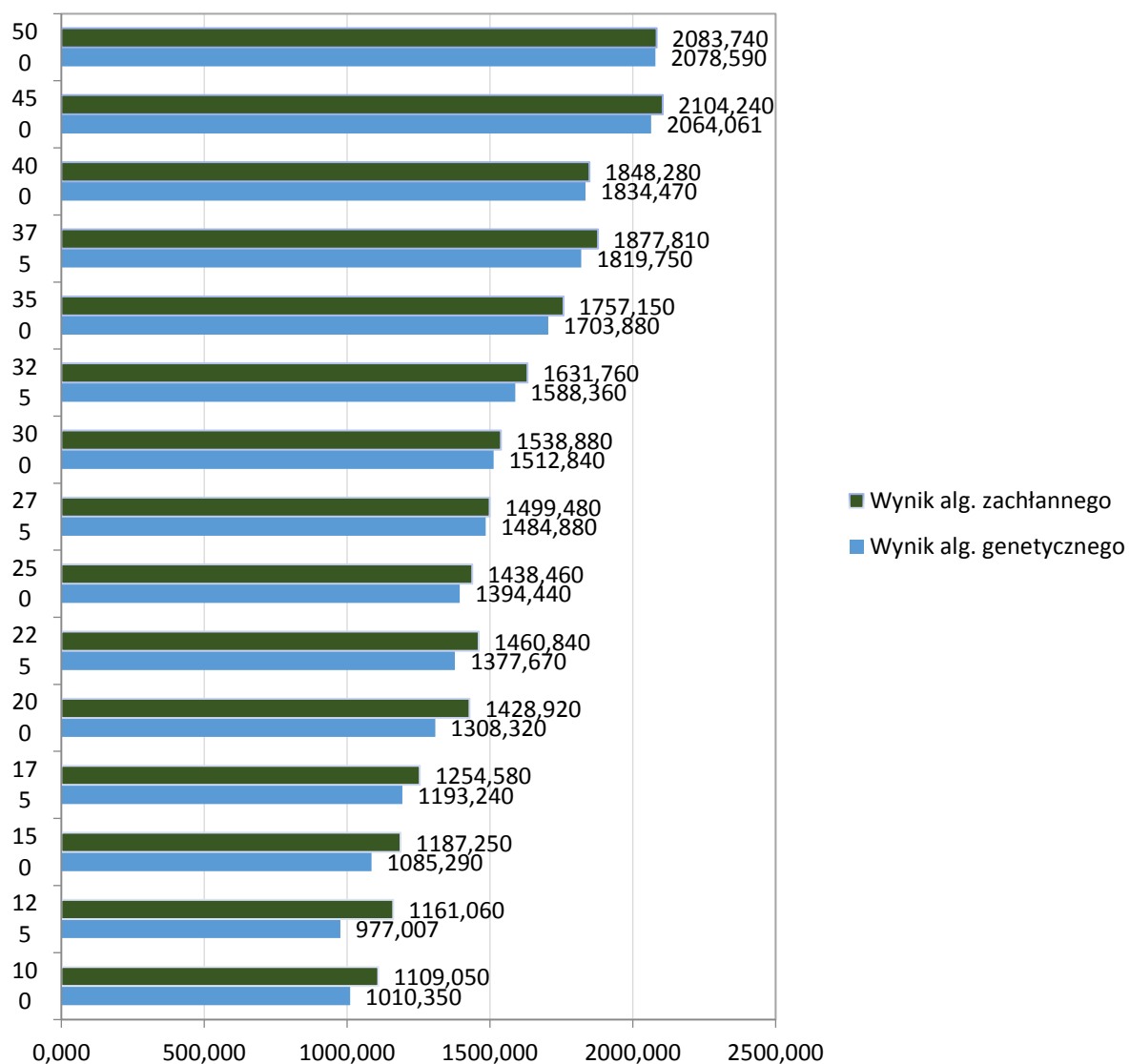
Algorytm genetyczny:



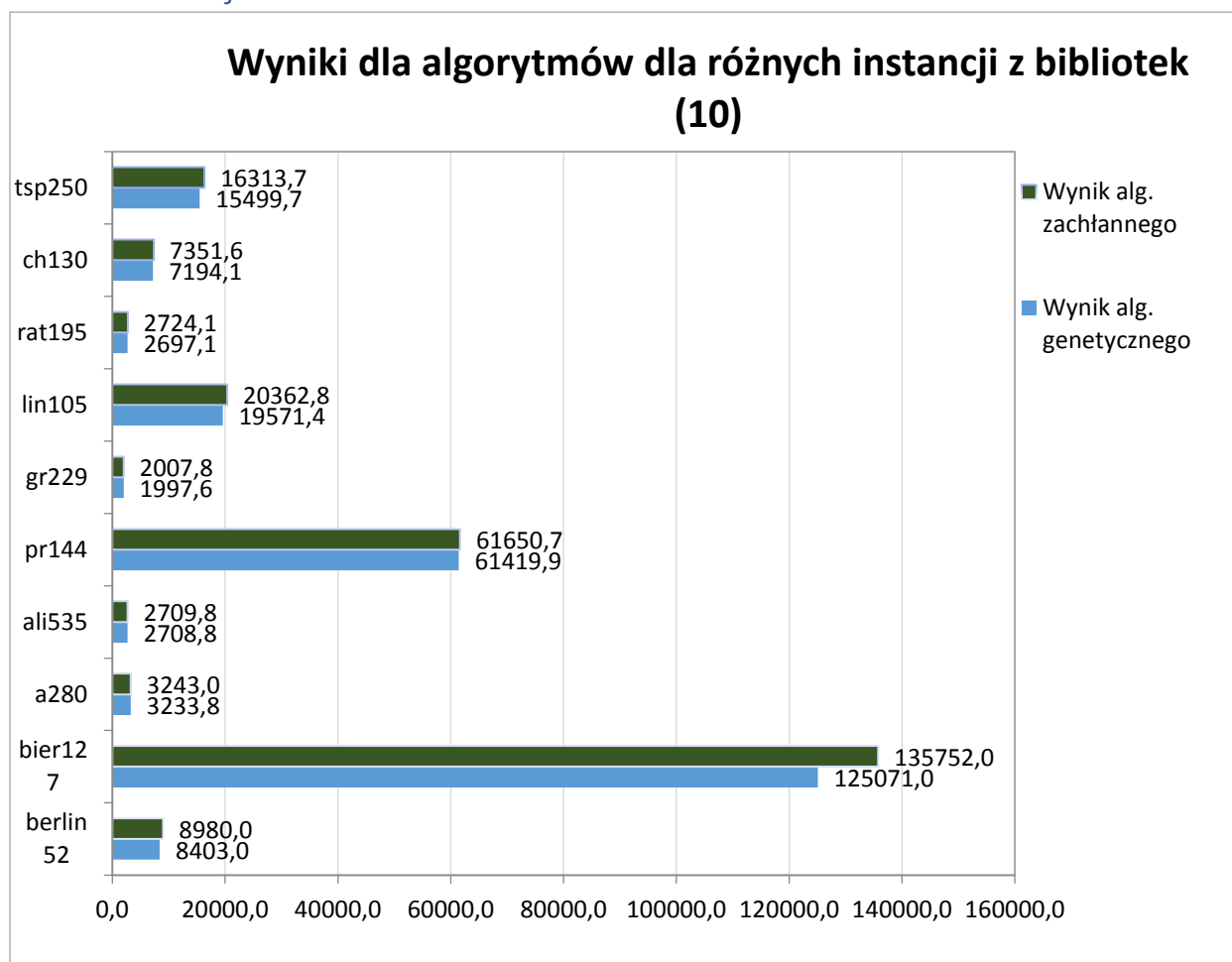
Problem Komiwożera (wykresy)

1. Porównanie optymalizowanych wartości z alg. zachłannym

Wykres wyników algorytmów dla danej liczby wierzchołków



2. Wyniki algorytmu generycznego i zachłannego dla danych instancji z bibliotek



3. Instancje:

| | |
|-----------------------------|----------|
| Berlin52 (8980,9, 7544,37) | 8403,0 |
| Bier127 (135751,8, 118772) | 125071,0 |
| tsp250 (16313,7, 13013,2) | 15499,7 |
| Tsp500 (97874,5, 88005,1) | 97874,5 |
| tsp1000 (29197,7 25238,3) | 29197,7 |