

# Problem rozmieszczenia hurtowni - algorytmy genetyczne

Daniel Olszowski, Paweł Sikora

16 czerwca 2014

## Streszczenie

Projekt semestralny na przedmiot Stochastyczne Algorytmy Obliczeniowe w roku akademickim 2013/2014. AGH Kraków. Zaproponowaliśmy dwa rozwiązania genetyczne dla problemu przydzielenia klientów do hurtowni. Jedno to rozwiązanie opierające się na klasycznym algorytmie ewolucyjnym  $\mu, \lambda$  a drugie to rozwiązanie hierariczne.

## 1 Opis problemu

Capacitated Warehouse Location to problem polegający na próbie rozmieszczenia hurtowni (magazynu) oraz przyporządkowania do nich klientów.

W problemie występują 2 główne dane:

- Pierwszą z nich jest spis możliwych lokalizacji hurtowni. Każda z lokalizacji posiada określony koszt zbudowania na niej hurtowni, a także maksymalną pojemność jaką dana hurtownia jest w stanie posiadać. Hurtownie przechowują jeden typ towaru.
- Drugą daną jest lista lokalizacji klientów. Do niej dołączona jest mapa przechowująca informacje o koszcie dostarczenia towaru z danej lokalizacji hurtowni do danego klienta. Mapa ta przechowuje informacje o wszystkich możliwych połączeniach (klient, lokalizacja hurtowni). Innymi słowy jest to funkcja, która dla argumentów (klient, lokalizacja hurtowni) podaje koszt takiego połączenia.

Wprowadzane są dwie funkcje kosztu:

- Pierwszą z nich jest suma kosztów wybudowania hurtowni w danych lokalizacjach.
- Drugą funkcją jest koszt pobierania towarów z hurtowni. Jej wartość to suma po wszystkich parach (klient, lokalizacja hurtowni), dla których hurtownia jest zbudowana a klient jest do niej przyporządkowany.

Rozwiązanie problemu polega takim rozmieszczeniu hurtowni i takim przydzieleniu klientów do hurtowni, aby:

- zminimalizować sumę dwóch wyżej wymienionych funkcji kosztu

- hurtownia nie była zmuszona dostarczać większej ilości towaru niż jest w stanie pomieścić.

Opis matematyczny

- założenia: każdy klient ma posiadać hurtownię, obciążenie każdej wybudowanej hurtowni nie może być większe niż jej maksymalna pojemność
- $h_i$  to lokalizacja i-tej hurtowni. Jest to zmienna, która przyjmuje wartość prawda jeżeli w tej lokalizacji jest wybudowana hurtownia oraz fałsz w przeciwnym przypadku. Funkcja  $H(i)$  podaje dla i-tej lokalizacji koszt wybudowania hurtowni. Funkcja  $Hp(i)$  podaje maksymalną pojemność hurtowni, jeśli została ona wybudowana w i-tej lokalizacji.
- $k_j$  to lokalizacja j-tego klienta. Jest to zmienna, której wartością jest liczba i reprezentująca lokalizację hurtowni, z której dany klient pobiera towar. Klient pobiera towar tylko z jednej hurtowni, hurtownia ta musi być wybudowana ( $h_i$  musi być prawdą). Funkcja  $K(j)$  podaje zapotrzebowanie j-tego klienta. Funkcja  $HK(i, j)$  podaje koszt dostarczenia towaru z i-tej lokalizacji hurtowni do j-tego klienta.
- $SH()$  to funkcja wyliczająca łączny koszt wybudowania wszystkich hurtowni. Jest to suma po wszystkich wartościach funkcji  $H(i)$ , dla których  $h_i$  posiada wartość prawda.
- $SHK()$  to funkcja wyliczająca łączny koszt powiązań hurtownia - klient. Jest to suma po wszystkich wartościach funkcji  $HK(i, j)$ , dla których  $h_i$  posiada wartość prawda a  $k_j$  ma wartość i.

Rozwiązanie problemu to wybranie takiej ilości hurtowni i ich lokalizacji oraz takie przydzielenie hurtowni klientom, aby suma funkcji  $SH() + SHK()$  miała jak najmniejszą wartość.

## 2 Nasze rozwiązanie

### 2.1 Algorytm genetyczny

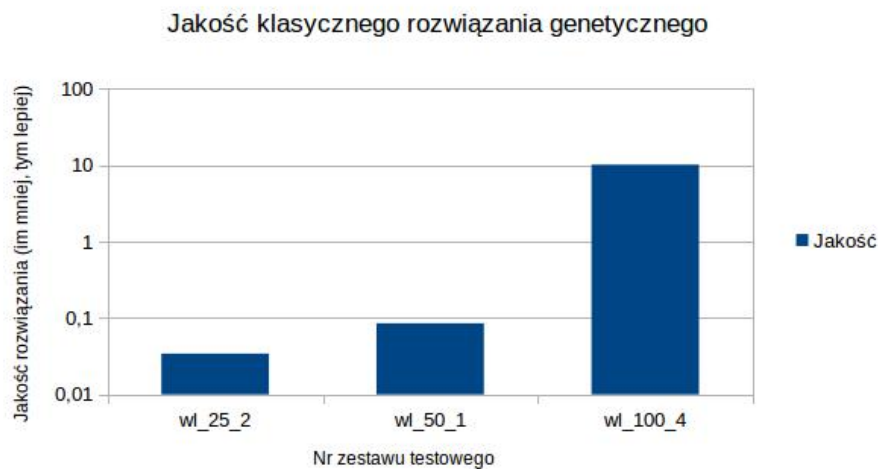
Pierwszym rozwiązaniem jest algorytm ewolucyjny  $(\mu, \lambda)$  o następujących właściwościach:

**Genotyp** Genotyp to lista hurtowni. Nazwijmy ją  $L$ . Jeżeli  $L[i] = w_j$  to znaczy, że klient  $c_i$  jest przypisany do hurtowni  $w_j$ .

**Selekcja** Wykorzystaliśmy selekcję turniejową. Wielkość turnieju to 15 procent wielkości populacji.

**Mutacja** Mutacja polega na wylosowaniu pewnej liczby klientów i ustawienia dla każdego losowej hurtowni.

**Krzyżowanie** Krzyżujemy dwóch osobników ind1 i ind2. Wybieramy pewną hurtownię, do której jest przypisany co najmniej jeden klient w osobniku ind1. Wybieramy indeksy wszystkich klientów, którzy są przypisani do tej hurtowni w osobniku ind1. Wybranych klientów kopiujemy do ind2. W ten sposób otrzymaliśmy pierwszego potomka. Następnie czynność powtarzamy ale tym razem wybieramy klientów z ind2 do ind1.



Rysunek 1: Rozwiązanie klasyczne

#### Wyniki pomiarów

Wykonywanych było 200 iteracji, populacja 100 osobników. Dla każdego zestawu testowego przeprowadziliśmy 3 próby, po czym uśrednialiśmy wyniki. Każde wygenerowane rozwiązanie było poprawne.

1. wl\_25\_2  
25 hurtowni, 50 klientów  
Dla tego zestawu testowego znalezione rozwiązanie jest stosunkowo bliskie rozwiązaniu optymalnemu, przy bardzo krótkim czasie wykonywania.
2. wl\_50\_1  
50 hurtowni, 50 klientów  
Dla tego zestawu testowego rozwiązanie również jest bliskie rozwiązaniu optymalnemu
3. wl\_100\_4  
100 hurtowni, 1000 klientów  
Dla dużego zestawu testowego algorytm genetyczny niestety daje wyniki odległe od optymalnych. Prawdopodobnie związane jest to ze zbyt krótką liczbą iteracji (200), która jest wystarczająca dla małych danych.

## 2.2 Hierarchiczny algorytm genetyczny

Drugie rozwiązanie to algorytm hierarchiczny. Algorytm ten działa tak samo jak algorytm genetyczny w rozwiązaniu pierwszym z wyjątkiem kilku rzeczy:

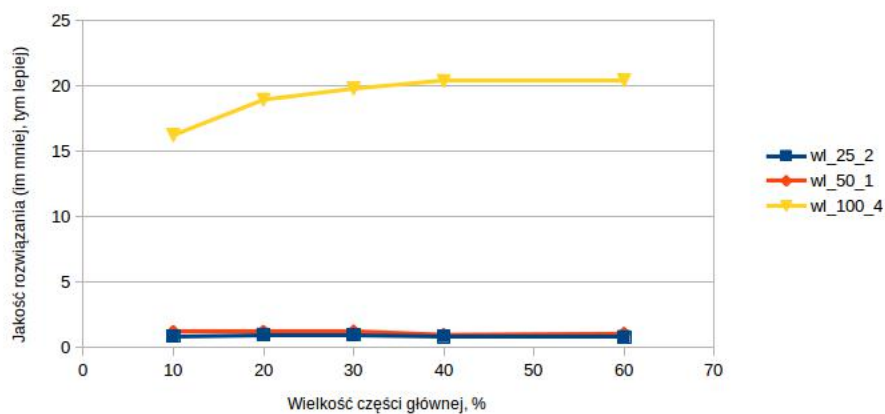
- Dzielimy klientów na dwie części - główną i podrzędną.
- Uruchamiamy lekko zmieniony algorytm z rozwiązania pierwszego tylko dla głównych klientów - tak jakby klienci podrzędni nie istnieli.
- W algorytmie z rozwiązania pierwszego nie wykorzystujemy klasycznej funkcji kosztu. Zamiast tego w funkcji kosztu uruchamiamy rekurencyjnie algorytm z rozwiązania pierwszego dla klientów podrzędnych. Przy tym wywołaniu uwzględniamy wykorzystanie części zasobów hurtowni przez klientów głównych.
- Przy obliczaniu kolejnych generacji populacji klientów podrzędnych kontrolujemy zmianę przystosowania populacji. Jeżeli przystosowanie populacji nie polepszyło się, o co najmniej  $d$  w ciągu  $k$  generacji to algorytm jest przerywany i kolejne generacje nie są już obliczane.

Wyniki pomiarów

Wykonywanych było 200 iteracji, populacja 100 osobników. Dla każdego zestawu testowego przeprowadziliśmy 3 próby, po czym uśrednialiśmy wyniki. Każde wygenerowane rozwiązanie było poprawne. Każde rozwiązanie jest gorsze niż w przypadku standardowego algorytmu genetycznego.

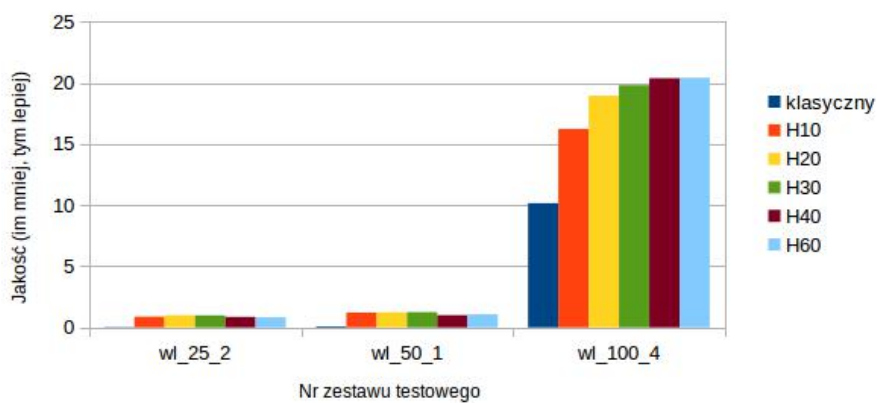
1. wl\_25\_2  
25 hurtowni, 50 klientów  
Dla tego zestawu testowego znalezione rozwiązanie jest stosunkowo bliskie rozwiązaniu optymalnemu.
2. wl\_50\_1  
50 hurtowni, 50 klientów  
Dla tego zestawu testowego rozwiązanie również jest bliskie rozwiązaniu optymalnemu.
3. wl\_100\_4  
100 hurtowni, 1000 klientów  
Dla dużego zestawu testowego algorytm genetyczny niestety daje wyniki odległe od optymalnych. Prawdopodobnie związane jest to ze zbyt krótką liczbą iteracji (200), która jest wystarczająca dla małych danych.

Jakość rozwiązania w zależności od wielkości części głównej algorytmu hierarchicznego

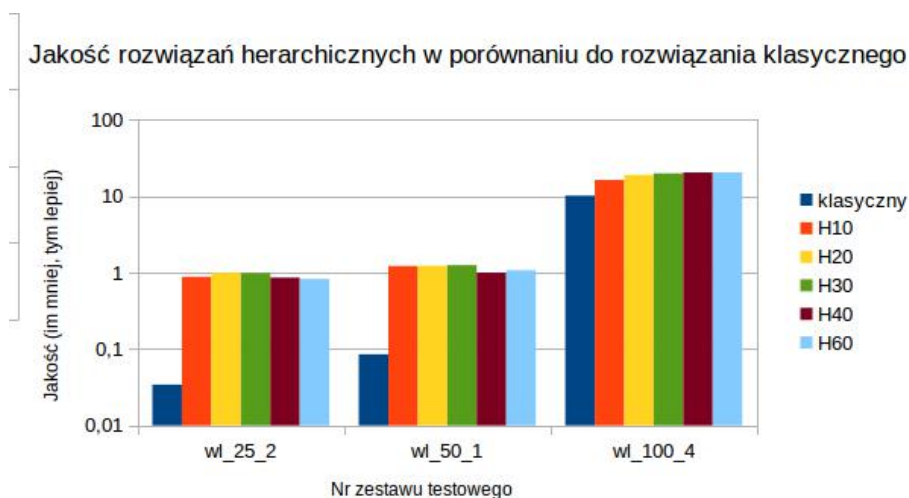


Rysunek 2: Rozwiązanie hierarchiczne. Numer serii oznacza numer zestawu testowego.

Jakość rozwiązań herarchicznych w porównaniu do rozwiązania klasycznego



Rysunek 3: Porównanie rozwiązań klasycznego i hierarchicznego. H10 oznacza rozwiązanie hierarchiczne z 10% wielkością części głównej.



Rysunek 4: Porównanie rozwiązań klasycznego i hierarchicznego. H10 oznacza rozwiązanie hierarchiczne z 10% wielkością części głównej. Skala Y logarytmiczna.

### 3 Podsumowanie

Klasyczny algorytm genetyczny daje wyniki stosunkowo bliskie rozwiązaniom optymalnym w bardzo dobrym czasie działania. Osiągnięcie minimum globalnego sprawia mu jednak kłopoty i rzadko się udaje. Algorytm hierarchiczny dał jednak znacznie gorsze rezultaty. Obliczenie pojedynczej generacji trwa dłużej i wyniki znacznie odbiegają od rozwiązania optymalnego. Wielkość części głównej dla małych zestawów danych nie ma większego znaczenia. Dla dużych zestawów danych jednak algorytm hierarchiczny osiągał lepsze wyniki, gdy wielkość części głównej była mała. Oznacza to, że im bardziej staramy się wyróżnić hierarchię wśród klientów, tym gorsze daje to wyniki. Stąd wniosek, że zależności między ustawieniem fragmentów genotypu są zbyt duże i nie można go podzielić na część główną i podrzędną.