

WYDZIAŁ PODSTAWOWYCH PROBLEMÓW TECHNIKI
POLITECHNIKA WROCŁAWSKA

EWOLUCYJNY ALGORYTM DLA NIELINIOWEGO ZADANIA TRANSPORTOWEGO

PIOTR BEREZOWSKI
NR INDEKSU: 236749

Praca inżynierska napisana
pod kierunkiem
dr hab. Pawła Zielińskiego



Politechnika
Wrocławska

WROCŁAW 2019

Spis treści

1	Wstęp	1
2	Analiza problemu	3
2.1	Zadanie transportowe	3
2.1.1	Wersja liniowa	4
2.1.2	Wersja nieliniowa	4
2.2	Klasyczne rozwiązania	4
2.3	Algorytmy metaheurystyczne	4
2.3.1	Algorytmy ewolucyjne	4
3	Równoległa implementacja algorytmu ewolucyjnego	5
3.1	Użyte technologie	5
3.2	Reprezentacja chromosomu	5
3.3	Operator krzyżowania	5
3.4	Operator mutacji	5
3.5	Funkcje oceny	5
3.6	Metoda selekcji	5
3.7	Wersja równoległa	5
4	Wyniki eksperymentalne	7
4.1	Model klasyczny algorytmu genetycznego	7
4.2	Model wyspowy algorytmu genetycznego	7
5	Podsumowanie	9
	Bibliografia	11
A	Zawartość płyty CD	13

Wstep



Analiza problemu

2.1 Zadanie transportowe

Zadanie transportowe należy do grupy zadań optymalizacyjnych z ograniczeniami. Rozwiązując je staramy się przy użyciu n punktów nadania zaspokoić zapotrzebowanie m punktów odbioru w taki sposób, aby całkowity koszt transportu był minimalny. Zadanie wymaga określenia ilości towaru znajdującej się w każdym z punktów nadania, oraz zapotrzebowania na towar w każdym z punktów odbioru. Dodatkowo musimy określić koszt transportu pomiędzy każdym punktem nadania i każdym punktem odbioru. Klasyczne zadanie transportowe ogranicza się do transportu tylko jednego towaru, dzięki czemu punkty odbioru mogą być zaopatrywane przez jeden lub więcej punktów nadania.

Zadanie transportowe nazywamy zbilansowanym, jeśli całkowita podaż towaru jest równa całkowitemu popytowi. W przeciwnym wypadku zadanie jest niezbilansowane. Rozwiązywanie zadania niezbilansowanego polega na sprowadzeniu go do zadania zbilansowanego, poprzez dodanie fikcyjnego dostawcy (w przypadku większego popytu), lub fikcyjnego odbiorcy (w przypadku większej podaży). Wartość kosztu dostawy między fikcyjnym dostawcą a odbiorcami, lub między dostawcami a fikcyjnym odbiorcą najczęściej ustalany jest jako zerowy.

Założmy, że mamy n punktów nadania i m punktów odbioru. Początkowa ilość towaru w i -tym punkcie nadania jest równa $supply(i)$, a początkowe zapotrzebowanie w j -tym punkcie odbioru jest równe $demand(j)$. Jeśli x_{ij} jest ilością towaru dostarczanego przez i -ty punkt nadania do j -tego punktu odbioru, to zbilansowane zadanie transportowe możemy zdefiniować w następujący sposób:

$$\min \sum_{i=1}^n \sum_{j=1}^m f_{ij}(x_{ij})$$

Przy spełnionych ograniczeniach:

$$\sum_{j=1}^m x_{ij} = supply(i), \text{ dla } i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = demand(j), \text{ dla } j = 1, 2, \dots, m$$

$$x_{ij} \geq 0, \text{ dla } i = 1, 2, \dots, n \text{ i } j = 1, 2, \dots, m$$

Pierwszy zestaw ograniczeń mówi o tym, że całkowita ilość towaru transportowana z pojedynczego punktu nadania musi być równa jego początkowej ilości znajdującej się w tym punkcie. Z kolei drugi zestaw mówi o tym, że całkowita ilość towaru transportowana do pojedynczego punktu odbioru musi być równa jego początkowemu zapotrzebowaniu. W przypadku zadania niezbilansowanego równości w dwóch pierwszych zestawach ograniczeń należy zmienić na odpowiednie nierówności. Zadanie jest liniowe, jeśli koszt transportu między punktami nadania i odbioru jest wprost proporcjonalny do ilości transportowanego towaru, tzn. jeśli $f_{ij}(x_{ij}) = cost_{ij}x_{ij}$, gdzie $cost_{ij}$ jest jednostkowym kosztem transportu między i -tym punktem nadania, a j -tym punktem odbioru.



2.1.1 Wersja liniowa

Zadanie transportowe w wersji liniowej należy do problemów programowania liniowego, a więc można rozwiązać je przy pomocy algorytmu sympleks.

[...*TODO*...]

2.1.2 Wersja nieliniowa

O ile liniowa wersja zadania jest stosunkowo łatwa w rozwiązaniu, o tyle dla wersji nieliniowej nie ma ogólnej metody rozwiązywania. Należy ono do problemów z kategorii NP-trudnych [TODO - znaleźć publikację]. Do jego rozwiązania używa się różnych algorytmów wyznaczających rozwiązania przybliżone, takich jak algorytmy metaheurystyczne.

2.2 Klasyczne rozwiązania

2.3 Algorytmy metaheurystyczne

2.3.1 Algorytmy ewolucyjne

Algorytmy ewolucyjne stanowią podzbiór algorytmów metaheurystycznych. Sposób ich działania

Równoległą implementacja algorytmu ewolucyjnego

3.1 Użyte technologie

3.2 Reprezentacja chromosomu

3.3 Operator krzyżowania

3.4 Operator mutacji

3.5 Funkcje oceny

3.6 Metoda selekcji

3.7 Wersja równoległa



Wyniki eksperymentalne

4.1 Model klasyczny algorytmu genetycznego

4.2 Model wyspowy algorytmu genetycznego



Podsumowanie

W tym rozdziale znajdzie się podsumowanie pracy.



Bibliografia



Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.

