

Równoległe rozwiązanie problemu komiwojażera. Algorytm kolonii mrówek

Marcin Fabrykowski, Jan Kleszczyński

17 maja 2012

1 Opis problemu

Problem komiwożera jest podstawowym zagadnieniem logistycznym. Polega on na znalezieniu najtańszej drogi między miastami, w sposób, który pozwala odwiedzić każde miasto tylko jeden raz. Zauważono, że mrówki, które wyruszają z mrowiska na poszukiwanie pożywienia, po znalezieniu jego źródła zaczynają poruszać się najkrótszą możliwą ścieżką. Dalsze obserwacje zdradzają nam następujący algorytm:

1. Każda mrówka wyrusza na poszukiwania i obchodzi raz każde ze źródeł pożywienia, po czym wraca do mrowiska.
2. Przechodząc po ścieżce, jeżeli jest ona krótsza od poprzedniej, pozostawia feromony.
3. Feromony każdej z mrówek sumują się, lecz co pewien czas parują.
4. Przy następnym wyjściu z mrowiska, mrówka decydując o wyborze drogi pokieruje się do miasta, do którego droga zawiera najwięcej feromonów. Jednakże każda mrówka ma swoją ciekawość, która również zostanie uwzględniona.
5. Po pewnym czasie mrówki zaczną wybierać zawsze tą samą trasę, która jest trasą najkrótszą.

W celu zrównoleglenia tego problemu zastosowaliśmy algorytm widoczny poniżej na schemacie. Mrówek jest tyle ile miast i są równomiernie rozłożone pomiędzy procesy. Gdy liczba miast jest nie podzielna przez ilość procesów reszta z dzielenia jest dodawana do procesu o ranku równym 0.

2 Opis algorytmu

Algorytm zastosowany w programie przedstawia się w następujący sposób:

1. Wczytanie mapy.
2. Przekazanie jej do wszystkich hostów. (pierwsze zrównoleglenie)
3. Określenie ilości mrówek.
4. Wyruszenie przez mrówki na wędrówkę
 - Wybór ścieżki (powtarzany aż do przejścia przez każde miasto)

5. W tym miejscu wszystkie mrówki przeszły przez wszystkie miasta i znane są ich trasy oraz długości drogi.
6. Parowanie feromonów
7. Określenie która ze ścieżek jest najkrótsza na danym hoście.
8. Przesłanie długości najlepszej ścieżki (pierwsza redukcja)
9. Wybór najlepszej ścieżki z nadesłanych.
10. Rozesłanie informacji o identyfikatorze hosta z najlepszą ścieżką (drugie zrównoleglenie).
11. Sprawdzenie, czy identyfikator hosta zgadza się z identyfikatorem rozesłanym przez proces zarządzający. Jeżeli nie to skok do punktu 13.
12. Rozesłanie najlepszej ścieżki do wszystkich hostów.
13. Rozłożenie feromonów na otrzymanej ścieżce (najlepszej ze wszystkich innych).
14. Przesłanie decyzji o zakończeniu (druga redukcja).
15. Jeżeli wszystkie mrówki przeszły nie więcej niż dwoma trasami i wykonano co najmniej 50 iteracji, aplikacja kończy działanie, drukuje długość znalezionej trasy i zapisuje szczegółowe informacje o trasie do pliku. W przeciwnym razie następuje skok do punktu 3.

Schemat blokowy przedstawiony jest na Rys. 1

3 Informacje o danych we/wyjściowych

Plik map.ini zabierający następujące dane:

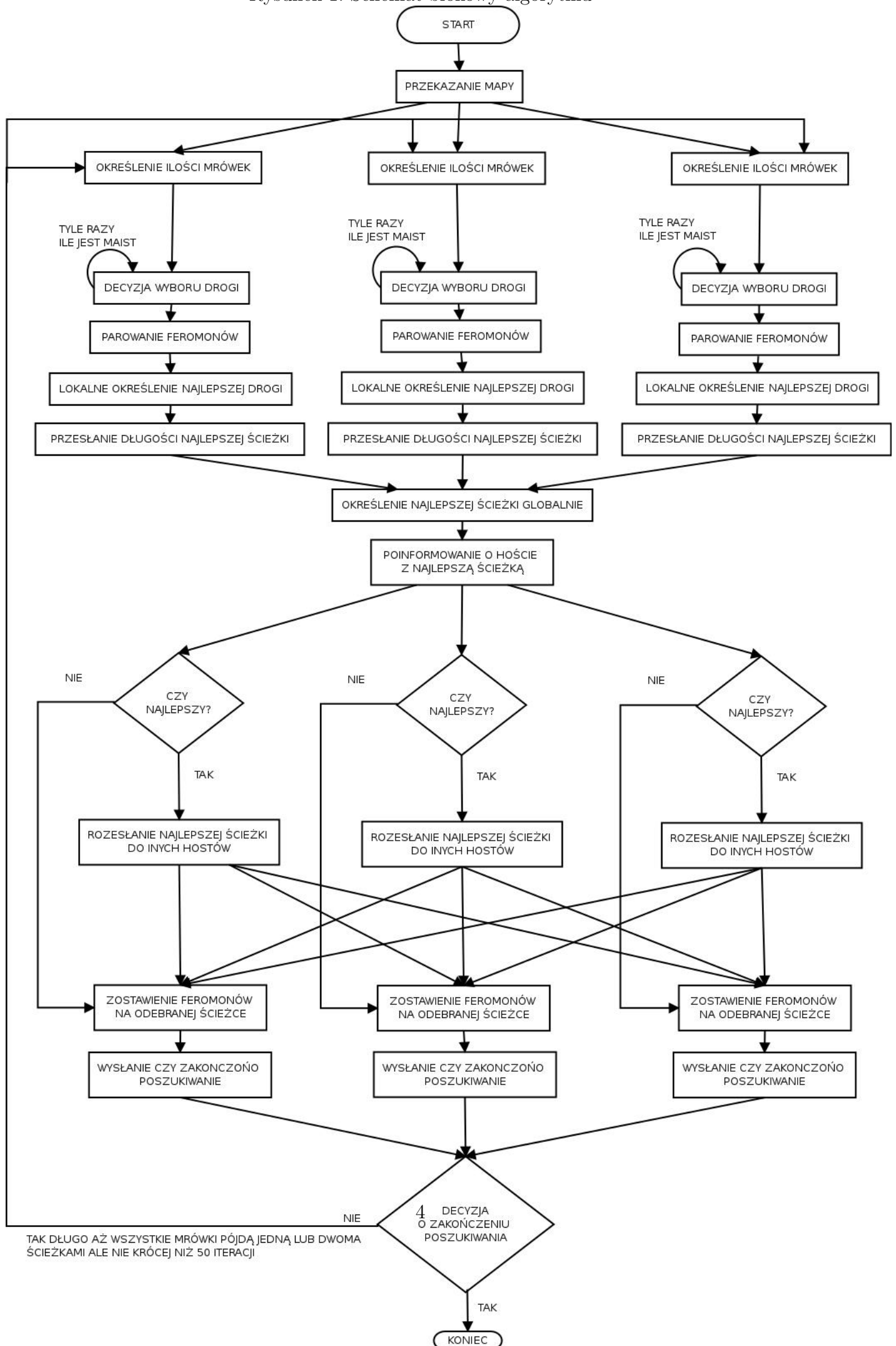
c < *liczba : num* > – liczba miast

r < *src : num* > < *dst : num* > < *len : num* > – definicja drogi

– komentarz

W wynikowym pliku wynik.txt znajduje się długość trasy oraz kolejność przechodzenia przez miasta.

Rysunek 1: Schemat blokowy algorytmu



4 Uruchomienie

W celu uruchomienia aplikacji należy wykonać następujące komendy:

```
./configure  
make  
make run
```

Spowoduje to sprawdzeniem systemu pod kątem zgodności, skompilowaniem aplikacji oraz jej uruchomieniem.