

# Rozproszone rozwiązanie problemu komiwojażera. Algorytm kolonii mrówek

Marcin Fabrykowski, Jan Kleszczyński

10 maja 2012

# 1 Opis problemu

Problem komiwojażera 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 Schemat blokowy

Schemat blokowy przedstawiony jest na Rys. 1

# 3 Informacje o danych we/wyjściowych

Plik map.ini zawierający następujące dane:

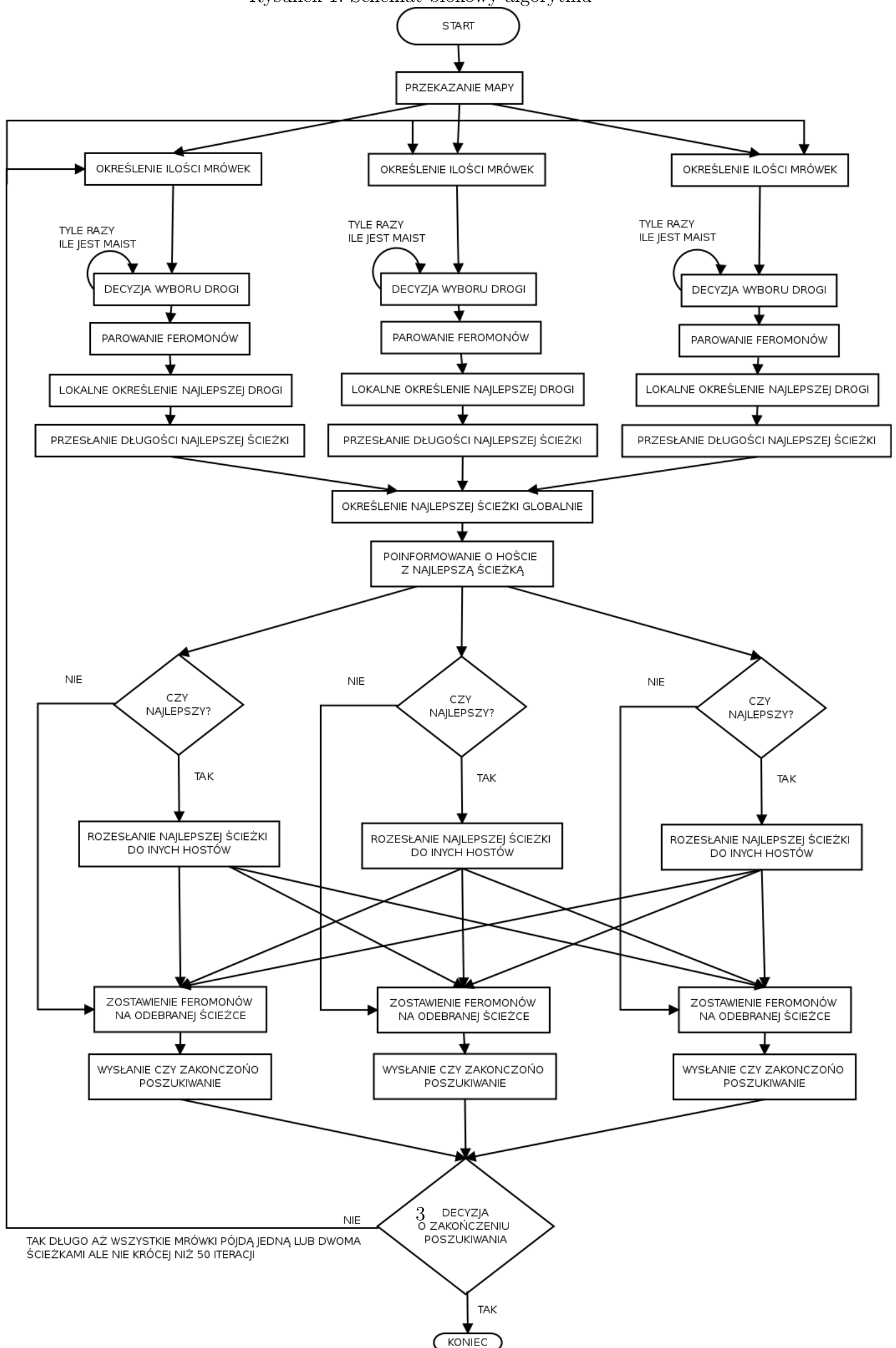
$c < liczba : num >$  – liczba miast

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

# – komentarz

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

Rysunek 1: Schemat blokowy algorytmu



## 4 Uruchomienie

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