STUDIA INFORMATICA 2001
Volume 22 Number 4 (46)

Adam SKÓRCZYŃSKI
Politechnika Śląska, Instytut Informatyki

MODYFIKACJA WIDOCZNOŚCI W ALGORYTMACH MRÓWKOWYCH ROZWIĄZUJĄCYCH PROBLEM DOSTAWY

Streszczenie. Problem dostawy jest przykładem złożonej optymalizacji kombinatorycznej i należy do grupy problemów NP-zupełnych. Rodzina algorytmów mrówkowych doskonale radzi sobie ze złożonymi problemami tej grupy. W niniejszej pracy zaproponowano modyfikację algorytmów mrówkowych rozwiązujących problem dostawy. W pracy przedstawiono wpływ zastosowanej modyfikacji na działanie algorytmów mrówkowych oraz jakość uzyskanych rozwiązań dla zbiorów danych wejściowych o różnych rozmiarach.

VISIBILITY MODIFICATION APPLIED TO ANT ALGORITHMS SOLVING DELIVERY PROBLEM

Summary. The delivery problem is an example of a complex combinatorial optimization, which belongs to the NP-complete group. The family of algorithms is suitable for solving the problems of this group. In this paper we present a new modification of algorithms solving the delivery problem. We present the effect of applied modification to the results of some algorithms for the sets of various sizes.

1. Wprowadzenie

Algorytmy mrówkowe zaliczane są do grupy algorytmów heurystycznych. Pierwszy algorytm mrówkowy Ant Cycle pojawił się w roku 1992. Algorytmy mrówkowe należą do grupy algorytmów, których powstanie zainspirowały zjawiska zaobserwowane w przyrodzie. Do tej grupy możemy zaliczyć dobrze znane algorytmy genetyczne czy też sieci neuronowe. Algorytmy mrówkowe wykorzystują te same mechanizmy, które umożliwiają kolonii mrówek odnalezienie najkrótszej drogi pomiędzy mrowiskiem a pożywieniem. Algorytmy mrówkowe stanowią symulację uproszczonego modelu kolonii mrówek, poruszających się w warunkach określonych przez założenia rozwiązywanego problemu.

Badania algorytmów mrówkowych trwające od 1992 roku nie ukazały, jak dotąd, wszystkich właściwości tych algorytmów. W chwili obecnej nie dysponujemy teorią wyjaśniającą wszystkie aspekty pracy algorytmów mrówkowych, opisującą mechanizmy zachodzące w trakcie ich pracy. Głównym celem obecnie prowadzonych badań jest poprawa efektywności algorytmów mrówkowych. Wielu badaczy przedstawia modyfikacje dotyczące wybranego algorytmu, zapewniające poprawę uzyskiwanych wyników. Przykładem proponowanych modyfikacji jest zastosowanie lokalnej optymalizacji oraz algorytm Max-Min Ant System.

Niniejsza praca zawiera propozycję kolejnej modyfikacji algorytmów mrówkowych. Modyfikacji poddano algorytm mrówkowy Ant Colony rozwiązujący problem dostawy, należący do klasy problemów NP-zupełnych.

2. Sformułowanie problemu dostawy

Niech dany będzie zbiór bazowy $N = \{1,2,...,n\}$ oraz rodzina podzbiorów tego zbioru $P = \{P_j\}_{j \in M}$, gdzie $M = \{1,2,...,m\}$. Zdefiniujmy funkcję kosztu F przyjmującą wartości nieujemne dla każdego podzbioru zbioru bazowego N:

$$\underset{P_j \in P}{\forall} F(P_j) = c_j \text{, gdzie } c_j \geq 0. \tag{1}$$

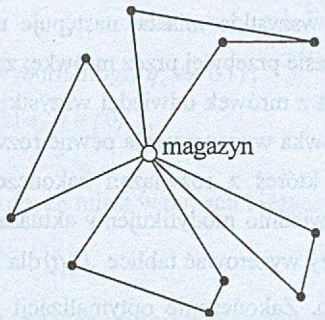
Rozbicie zbioru N można zdefiniować jako podzbiór Q rodziny P, której wszystkie elementy są rozłączne i pokrywają wszystkie elementy zbioru N.

Funkcja kosztu dla problemu rozbicia jest zdefiniowana jako suma wartości funkcji kosztów dla podzbiorów wchodzących w skład rozbicia:

$$F(Q) = \sum_{i=1}^{s} F(Q_i) \text{, gdzie } Q_i \in P. \tag{2}$$

Problem rozbicia zbioru polega na znalezieniu rozbicia Q o minimalnym koszcie.

Szczególnym przypadkiem problemu rozbicia zbioru jest zagadnienie dostawy, gdzie zbiór bazowy N reprezentuje zbiór klientów (miast, węzłów), którzy mają być obsłużeni przez bazę-magazyn [6]. Zakłada się, że w magazynie dysponuje się nieograniczoną liczbą samochodów dostawczych, służących do dostarczania towarów do klientów. Ograniczona jest jedynie liczba klientów, którzy mogą być obsłużeni w ramach jednej trasy. Zakładamy, że liczba ta wynosi 3. Każdej możliwej trasie obejmującej jednego, dwóch lub trzech klientów przyporządkowany jest koszt będący długością trasy. Rozwiązanie problemu dostawy polega na znalezieniu takiej kombinacji tras, której koszt będzie minimalny, a każdy element zbioru bazowego (węzeł) pokryty będzie jednokrotnie. Rysunek 1 przedstawia przykład rozwiązania problemu dostawy dla $n = 10$.



Rys. 1. Przykładowe rozwiązanie ($n = 10$)
Fig. 1. Sample solution ($n = 10$)

3. Algorytm Ant Colony dla problemu dostawy

W trakcie badań algorytmów mrówkowych rozwiązujących problem dostawy najlepsze wyniki uzyskano dla algorytmu Ant Colony, co zadecydowało o wyborze tego algorytmu do dalszych badań. Opis wcześniej przeprowadzonych badań znajduje się w pracy [5] i [6].

Optymalizację wykonuje zespół agentów, nazywanych dalej mrówkami. Niech m określa liczbę mrówek. W chwili $t = 0$ mrówki zostają rozmieszczone w pewien sposób we wszystkich miastach, najczęściej stosowane jest rozmieszczenie losowe. Niech $b_i(t), i \in \{1,2,3,...,n\}$, określa liczbę mrówek w mieście i w chwili t, gdzie t oznacza aktualny czas.

Każdą mrówkę można scharakteryzować w następujący sposób:

- po upływie pewnego czasu mrówka wybiera miasto, do którego przejdzie, tzn. jeżeli w chwili t_c znajduje się w mieście i, to musi wybrać miasto j lub bazę magazyn b, w którym znajdzie się w chwili $t_{c+1} = t_c + \Delta t$ (dalej przyjmiemy $\Delta t = 1$); kolejne miasto zostaje wybrane zgodnie z zależnością nazywaną funkcją wyboru drogi (12); jeżeli mrówka znajdzie się w trzecim z rzędu mieście, to należy wymusić jej powrót do bazy-magazynu b,
- po przejściu do kolejnego miasta lub bazy-magazynu następuje uaktualnienie lokalne ilości feromonu zgodnie z zależnością (14),

- każda mrówka posiada pamięć w postaci listy miast odwiedzonych $J_k(t)$ (dla k-tej mrówki), wykorzystywaną podczas wyboru kolejnego miasta, tak aby nie odwiedzić tego samego miasta dwa razy;
- kiedy mrówka odwiedzi wszystkie miasta, następuje uaktualnienie globalne ilości feromonów leżących na trasie przebytej przez mrówkę; zależność (17).

Okres czasu, w którym każda z mrówek odwiedzi wszystkie miasta, nazywamy cyklem. Po zakończeniu cyklu każda mrówka wygenerowała pewne rozwiązanie, które porównujemy z aktualnie najlepszym. Jeżeli któreś z rozwiązań zakończonego cyklu jest lepsze od dotychczas najlepszego, to odpowiednio modyfikujemy aktualnie najlepsze rozwiązanie. Po zakończeniu każdego cyklu należy wyzerować tablice $J_k(t)$ dla każdej mrówki, aby możliwe było wykonanie kolejnego cyklu. Zakończenie optymalizacji następuje wtedy, gdy liczba cykli osiągnie wartość NC_{max} (maksymalna liczba cykli). W szczególnych przypadkach optymalizacja może zakończyć się wcześniej, np. gdy tymczasowe rozwiązanie jest wystarczająco dobre. Jeżeli w kolejnych cyklach wszystkie mrówki generują wciąż te same rozwiązanie (wybierają tę samą trasę), oznacza to, iż optymalizacja znalazła się w fazie stagnacji. W takim przypadku należy przerwać wykonanie algorytmu, gdyż rozwiązanie nie zostanie już poprawione.

Przyjmijmy następujące oznaczenia:

$w = n$	liczba kroków w cyklu, równa liczbie miast,	(3)
d_{ij}	odległość miasta j od miasta i,	(4)
$\eta_{ij} = \dfrac{1}{d_{ij}}$	widoczność miasta j z miasta i,	(5)
$\tau_{ij}(t)$	ilość feromonu na trasie łączącej miasta i, j w czasie t,	(6)
$\Delta\tau_{ij}^{k}(t)$	przyrost feromonu na trasie i-j w czasie t dla k-tej mrówki,	(7)
$L_k(t)$	długość trasy dla k-tej mrówki w czasie t,	(8)
$p_{ij}^{k}(t)$	prawdopodobieństwo przejścia z miasta i do j dla k-tej mrówki w czasie t,	(9)
$J_k(t)$	zbiór dozwolonych miast dla k-tej mrówki w czasie t,	(10)
$T_k(t) = N - J_k(t)$	zbiór odwiedzonych miast dla k-tej mrówki po czasie t.	(11)

Miasto j, do którego przejdzie k-ta mrówka, wybierane jest zgodnie z zależnością:

$$j_k(t)=\begin{cases} b & \text{dla } lm_k = l \\ \underset{l\in J_k(t)\cup\{b\}}{\arg\max}\{[\tau_{il}(t)]*[\eta_{il}(t)]^\beta\} & \text{dla } q \le q_o, \text{gdy } lm_k < l \\ S(t) & \text{dla } q > q_o, \text{gdy } lm_k < l \end{cases} \tag{12}$$

q_o - współczynnik wyboru drogi, $q_o \in (0,1)$,

q - wylosowana liczba, $q \in (0,1)$,

b - baza-magazyn,

lm_k - licznik odwiedzonych miast w ramach bieżącej trasy dla k-tej mrówki.

$S(t)$ - miasto wybrane na podstawie funkcji prawdopodobieństwa:

$$p_{ij}^{\,k}(t)=\begin{cases} \dfrac{[\tau_{ij}(t)]*[\eta_{ij}(t)]^\beta}{\sum\limits_{l\in J_k(t)\cup\{b\}}[\tau_{il}(t)]*[\eta_{il}(t)]^\beta} & \text{dla } j\in J_k(t)\cup\{b\} \\ 0 & \text{w innym przypadku} \end{cases} \tag{13}$$

β - współczynnik istotności widoczności,

Zależności lokalnego uaktualnienia feromonu przyjmują postać:

$$\tau_{i,j}(t+h+1) = (1-\vartheta) * \tau_{i,j}(t) + \vartheta * \Delta\tau_{i,j}(t+h+1) \tag{14}$$

$$\Delta\tau_{i,j}(t) = \sum_{k=1}^{m} \Delta\tau_{i,j}^{\,k}(t) \tag{15}$$

$$\Delta\tau_{i,j}^{\,k}(t) = \begin{cases} Q & \text{jeżeli } k\text{-ta mrówka przeszła drogę } i\text{-}j \text{ w czasie } (t\text{-}h\text{-}1,t) \\ 0 & \text{w innym przypadku} \end{cases} \tag{16}$$

Q - parametr określający ilość wydzielanego feromonu przez jedną mrówkę podczas przejścia do kolejnego miasta,

ϑ - parametr określający lokalne parowanie feromonu,

$h = lm_k$ - liczba miast-klientów wchodzących w skład danej trasy.

Zależności globalnego uaktualnienia feromonu przyjmują postać:

$$\tau_{i,j}(t+w) = (1-\alpha) * \tau_{i,j}(t) + \alpha * \Delta\tau_{i,j}(t+w) \tag{17}$$

$$\Delta\tau_{i,j}(t) = \begin{cases} \dfrac{1}{L_{gb}} & \text{jeżeli droga } i\text{-}j \text{ należy do najlepszej trasy w czasie } (t\text{-}w,t) \\ 0 & \text{w innym przypadku} \end{cases} \tag{18}$$

L_{gb} - aktualnie najlepszy wynik – długość najlepszej globalnie trasy,

α - parametr określający globalne parowanie feromonu.

4. Modyfikacja widoczności

Eksperymenty przeprowadzone dla różnych wartości współczynnika istotności widoczności wykazały, że bardzo ważna dla przebiegu optymalizacji jest informacja o wzajemnym położeniu miast. W szczególnym przypadku, gdy $\beta = 0$, informacja o widoczności nie była wykorzystywana w trakcie wyboru kolejnego miasta. Mrówki niepotrzebnie sprawdzały rozwiązania, które były dalekie optimum. Wykorzystanie widoczności pozwala lepiej ukierunkować proces poszukiwania rozwiązania poprzez ograniczenie przestrzeni poszukiwań.

Funkcja określająca widoczność zdefiniowana jest jako odwrotność odległości (5). Taka definicja jest zbyt restrykcyjna, ograniczenie przestrzeni poszukiwań zbyt duże. Następuje nadmierne faworyzowanie najbliższych miast i w rezultacie proces poszukiwania rozwiązania zostaje zawężony do miast najbliżej leżących.

Proponowana przez nas modyfikacja polega na określeniu widoczności jako funkcji liniowej odległości, co można zapisać w następujący sposób:

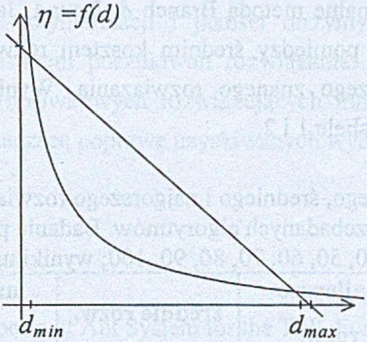
$$\eta_{ij} = (d_{\max} - d_{ij} + d_{\min}) \tag{19}$$

$$d_{\max} = \arg\max_{i,j \in N}\{d_{ij}\}$$

$$d_{\min} = \frac{d_{\max}}{100}$$

Tak określona funkcja widoczności przyjmuje wartości dodatnie z zakresu $(d_{\min}, d_{\min} + d_{\max})$.

Określenie widoczności jako funkcji liniowej odległości miast powoduje selektywne rozszerzenie przestrzeni poszukiwań rozwiązania. Rysunek 2 przedstawia oryginalną i zmodyfikowaną funkcję widoczności.



Rys. 2. Funkcja określająca widoczność
Fig. 2. Visibility function

5. Wyniki przeprowadzonych eksperymentów

Badano Algorytm Ant Colony z modyfikacją widoczności. Uzyskane wyniki porównano z wynikami uzyskanymi w trakcie wcześniejszych prac, opisanych w [5], w trakcie których badano cztery podstawowe algorytmy mrówkowe: Ant Cycle, Ant Density, Ant Quantity, Ant Colony oraz dwie znane modyfikacje algorytmu Ant Colony: Max-Min i zastosowanie lokalnej optymalizacji. Wartości parametrów badanych algorytmów podano w [5].

Dane wejściowe stanowiły specjalnie przygotowane zbiory współrzędnych kartezjańskich miast-klientów. Współrzędne miast były liczbami całkowitymi i zostały wylosowane z przedziału <-30,30>. Baza-magazyn znajdował się w punkcie (0,0). W ramach jednego zbioru nie mogą występować klienci o tych samych współrzędnych, ponadto nie mogą występować klienci, których współrzędne pokrywają się ze współrzędnymi bazy-magazynu.

Dla jednego zbioru miast można przeprowadzić badania z różnymi liczbami klientów, np. dla wygenerowanego zbioru 100 klientów można przeprowadzić badania dla pierwszych 30 klientów, pierwszych 50 lub wszystkich klientów. Liczbę klientów, których pobierano ze zbioru klientów, nazwano rozmiarem danych zbioru. Dla jednego zbioru klientów można przeprowadzić kilka eksperymentów dla różnych rozmiarów danych.

Pierwsze badanie polegało na 1000-krotnym wykonaniu każdego algorytmu dla zbioru klientów DANE100 dla rozmiarów 30, 40, 50, 60, 70, 80, 90, 100. Dla każdego wykonania odnotowano koszt uzyskanego rozwiązania i czas wykonania. Drugie badanie polegało na 20-krotnym wykonaniu każdego algorytmu dla 20 zbiorów miast, każdy o rozmiarach 30, 40, 50, 60, 70, 80, 90, 100.

Rozwiązania porównywano z najlepszym rozwiązaniem znalezionym podczas prowadzonych badań (opisanych w pracach [2], [5] i [9]). Jedynie dla zbioru o n = 30 znaleziono rozwiązanie optymalne metodą Branch & Bound. Jako miarę jakości algorytmu przyjęto różnicę procentową pomiędzy średnim kosztem rozwiązania znalezionego przez algorytm a kosztem najlepszego znanego rozwiązania. Wyniki otrzymane w kolejnych etapach badań przedstawiają tabele 1 i 2.

Tabela 1

Średnia jakość najlepszego, średniego i najgorszego rozwiązania oraz odchylenie standardowe dla każdego z przebadanych algorytmów. Badanie przeprowadzono dla zestawu DANE100 o rozmiarach 30, 40, 50, 60, 70, 80, 90, 100; wyniki uśredniono dla 1000 wykonań

algorytm	**najlepsze rozw.**	**średnie rozw.**	**najgorsze rozw.**	**odchylenie std.**
AntColony VIS	**100,061%**	**100,639%**	**102,481%**	**4,98**
AntColony	100,336%	101,429%	103,988%	6,01
AntColony LO	100,064%	101,121%	103,678%	7,01
AntColony MM	100,259%	101,792%	105,516%	9,18
AntDensity	100,524%	102,078%	103,607%	6,32
AntQuantity	100,483%	101,963%	103,382%	6,09
AntSystem	100,836%	102,819%	104,693%	8,08

Tabela 2

Średnia jakość najlepszego, średniego i najgorszego rozwiązania oraz odchylenie standardowe dla każdego z przebadanych algorytmów. Badanie przeprowadzono dla 20 zestawów danych o rozmiarach 30, 40, 50, 60, 70, 80, 90, 100; wyniki uśredniono dla 20 wykonań

algorytm	**najlepsze rozw.**	**średnie rozw.**	**najgorsze rozw.**	**odchylenie std.**
AntColony VIS	**100,289%**	**100,874%**	**101,783%**	**5,25**
AntColony	100,937%	101,961%	103,252%	7,54
AntColony LO	100,369%	101,364%	102,789%	8,18
AntColony MM	100,907%	101,991%	103,299%	7,56
AntDensity	101,294%	102,296%	103,167%	6,06
AntQuantity	101,289%	102,196%	103,029%	5,83
AntSystem	101,813%	103,088%	104,172%	7,76

6. Wnioski

Zastosowanie modyfikacji widoczności przyniosło znaczącą poprawę jakości otrzymanych rozwiązań. W stosunku do algorytmu Ant Colony uzyskano poprawę średnio o ok. 1%, w stosunku do algorytmu Ant Colony z lokalną optymalizacją poprawa wyników wyniosła ponad 0,5%. Przytoczone wyniki przeprowadzonych badań wskazują na możliwość poprawy przebiegu procesu optymalizacji i jakości otrzymywanych rozwiązań, poprzez selektywne poszerzenie przestrzeni poszukiwań rozwiązania. Zastosowanie proponowanej modyfikacji w algorytmach mrówkowych rozwiązujących inne problemy optymalizacyjne powinno przynieść równie znaczącą poprawę uzyskiwanych wyników.

LITERATURA

1. Boryczka M.: Some Aspects of Ant System for the TSP, Fundamenta Informaticae 35 (1998) 197-209.
2. Deorowicz S.: Heurystyczne i genetyczne algorytmy optymalizacji kombinatorycznej, Praca dyplomowa magisterska, Politechnika Śląska, Instytut Informatyki, Gliwice 1998.
3. Dorigo M., Gambardella L. M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, TR Iridia, Belgium (1996).
4. Dorigo M., Maniezzo V., Colorni A.: The Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics B, 26, 1 (1996).
5. Skórczyński A.: Algorytmy mrówkowe, Praca dyplomowa magisterska, Politechnika Śląska, Instytut Informatyki, Gliwice 1998.
6. Skórczyński A.: Rozwiązywanie problemu dostawy za pomocą algorytmów mrówkowych, Zeszyty Naukowe Politechniki Śląskiej, Gliwice 1999.
7. Stützle T., Hoos H.: Max-Min Ant System and Local Search for the Traveling Salesman Problem, In Proc. IEE Int. Conf. Evolut. Comp. (ICEC'97).
8. Stützle T., Hoos H.: Improvements on the Ant-System: Introducing the Max-Min Ant System, In Proc. Int. Conf. Artificial Neural Networks and Genetic Algorithms, Springer Verlag, Wien 1997.
9. Szołtysek M.: Algorytmy heurystyczne dla rozwiązywania problemu dostawy, Praca dyplomowa magisterska, Politechnika Śląska, Instytut Informatyki, Gliwice 1998.

Recenzent: Dr inż. Mariusz Boryczka

Wpłynęło do Redakcji 22 października 2001 r.

Abstract

The Set Partitioning Problem (SPP) is a difficult combinatorial optimization problem, which belongs to the NP-complete problems. The Delivery Problem (DP) is a special case of the SPP. In this paper we present the new Ant Colony modification called "visibility modification".

We applied visibility modification to the Ant Colony algorithm solving the delivery problem. Prepared algorithm (Ant Colony VIS) and six other Ant algorithms were examined in two stages. The stages differ in the number of cities in input data and the number of test runs. The costs of the best solution, the worst solution, and the average of the costs of all solutions were found. The difference between the best-known cost and the average cost for each set was taken for comparison of qualities of the algorithms. Obtained results are presented in table 1 and 2.

The tests show that the visibility modification gives significant improvements. Ant Colony VIS gives the best results.