ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1999
Seria: INFORMATYKA z. 37 Nr kol. 1421

Adam SKÓRCZYŃSKI
Politechnika Śląska, Instytut Informatyki

ROZWIĄZYWANIE PROBLEMU DOSTAWY ZA POMOCĄ ALGORYTMÓW MRÓWKOWYCH

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. Podjęto próbę zaadoptowania algorytmów mrówkowych do rozwiązywania problemu dostawy. W niniejszej pracy przedstawiono wyniki działania wybranych modyfikacji algorytmów mrówkowych do rozwiązywania problemu dostawy dla zbiorów danych wejściowych o różnych rozmiarach.

SOLVING THE DELIVERY PROBLEM USING ANT ALGORITHMS

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

1. Wprowadzenie

Problem rozbicia zbioru, który wraz z pokrewnymi problemami pokrycia i upakowania tworzy grupę zagadnień pokrycia, należy do klasy tzw. problemów NP-zupełnych, tj. takich, dla których nie jest znany deterministyczny algorytm o złożoności wielomianowej znajdujący rozwiązanie optymalne. W niniejszym artykule zajmiemy się szczególnym przypadkiem problemu rozbicia zbioru, jakim jest problem dostawy. Dla problemu dostawy istnieje kilka metod poszukiwania rozwiązania optymalnego, jednak charakteryzują się one wykładniczą

złożonością czasową, w związku z czym metody te tracą zastosowanie dla zbiorów danych wejściowych o dużej mocy.

W praktycznych zastosowaniach często nie jest konieczne znalezienie rozwiązania optymalnego i wystarczające jest znalezienie rozwiązania do niego zbliżonego, co jest charakterystyczne dla rozwiązań uzyskiwanych przez algorytmy heurystyczne. Spośród algorytmów heurystycznych wybrano algorytmy mrówkowe, które z powodzeniem były stosowane do rozwiązywania różnych 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:

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

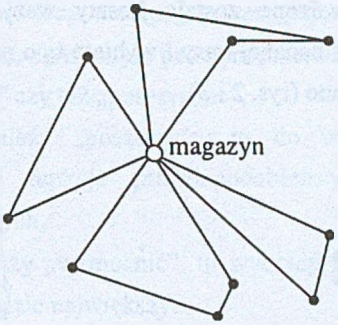
Rozbicie zbioru N można zdefiniować jako podzbiór Q rodziny P, której wszystkie elementy są rozłączne i pokrywają wszystkie elementy zbioru N.

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

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

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. 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 oznaczona k 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. Algorytmy mrówkowe i ich modyfikacje

Pierwszym algorytmem należącym do rodziny algorytmów mrówkowych był Ant Cycle, przedstawiony w 1992 r. przez Marco Dorigo, który badał implementację dla problemu komiwojażera. Uzyskane bardzo dobre wyniki zaowocowały kontynuacją badań i powstaniem kilku różnych odmian tego algorytmu, tworzących rodzinę algorytmów mrówkowych. W dalszej części przedstawiono cztery implementacje algorytmów mrówkowych dla problemu dostawy. Omówione algorytmy są modyfikacjami odpowiednich algorytmów dla problemu komiwojażera [4].

3.1. Algorytm Ant Cycle

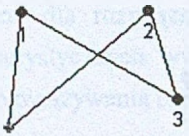
Adaptacja algorytmu Ant Cycle dla problemu komiwojażera do założeń problemu dostawy wymaga modyfikacji sposobu wyboru przez „mrówkę" kolejnego miasta. Należy uwzględnić dodatkowe warunki, które muszą być spełnione dla problemu dostawy:

- w ramach jednej trasy „mrówka" może odwiedzić maksymalnie 3 klientów,
- „mrówka" oprócz możliwości wyboru spośród klientów nie odwiedzonych powinna mieć możliwość wyboru bazy-magazynu.

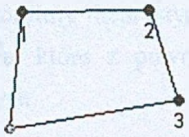
Spełnienie obu warunków pociąga za sobą konieczność modyfikacji funkcji określającej prawdopodobieństwo wyboru drogi [6],[4].

W celu poprawienia uzyskiwanych przez algorytm wyników wprowadzono dodatkową modyfikację – optymalizację tras. Wszystkie trasy wchodzące w skład rozwiązania uzyskanego przez każdą z „mrówek", po zakończeniu każdego cyklu poddaje się działaniu

dodatkowej procedury. Sprawdzone zostają koszty wszystkich możliwych permutacji klientów wchodzących w skład badanej trasy i wybiera tę o najmniejszym koszcie. Wybrana trasa zastępuje trasę, którą badano (rys. 2 i 3).



Rys. 2. Badana trasa (b,2,3,1,b), kolejność miast nie jest optymalna
Fig. 2. Original route (b,2,3,1,b)



Rys. 3. Wybrana trasa (b,1,2,3,b), ze zmienioną kolejnością miast
Fig. 3. Chosen route (b,1,2,3,b)

Ilość feromonu znajdującego się na drodze łączącej wszystkie pary klientów stanowi globalną „pamięć" algorytmów mrówkowych. Wybór kolejnego klienta dokonywany jest na podstawie ilości feromonu. Po zakończeniu cyklu następuje uaktualnienie ilości feromonu znajdującego się na drodze łączącej wszystkie pary klientów (tzw. uaktualnienie globalne). Ilość feromonu na każdej drodze zostaje zwiększona w zależności od jakości wygenerowanego rozwiązania (o tzw. ilość uaktualnienia feromonu).

3.2. Algorytm Ant Quantity i Ant Density

Algorytmy Ant Quantity i Ant Density odróżnia od algorytmu Ant Cycle sposób uaktualniania ilości feromonu na drogach pomiędzy klientami. Uaktualnienie to następuje zawsze zaraz po przejściu mrówki od jednego klienta do drugiego. Tego rodzaju uaktualnienie nazywamy uaktualnieniem lokalnym. Zastosowanie uaktualnienia lokalnego oznacza konieczność zmiany optymalizacji tras, która następuje zaraz po wygenerowaniu trasy, a nie dopiero po wygenerowaniu pełnego cyklu.

Algorytmy Ant Quantity i Ant Density są do siebie bardzo podobne. Odróżnia je jedynie ilość feromonu uaktualnienia. W algorytmie Ant Quantity ilość uaktualnienia feromonu uzależniona jest od długości trasy, w przeciwieństwie do algorytmu Ant Density.

3.3. Algorytm Ant Colony

Algorytm Ant Colony odróżnia od pozostałych algorytmów mrówkowych sposób wyboru drogi „mrówki". Dotychczasowe algorytmy wyznaczały kolejnego klienta zgodnie z prawdopodobieństwem określonym przez pewną funkcję.

W algorytmie Ant Colony wybór drogi następuje w następujący sposób:

- z prawdopodobieństwem określonym pewnym parametrem ustala się, czy rozwiązanie zostanie „wzmocnione" czy też „poszerzone",
- jeżeli rozwiązanie należy „poszerzyć", to do wyznaczenia kolejnego klienta wykorzystana zostaje funkcja prawdopodobieństwa wykorzystana w innych algorytmach mrówkowych,
- jeżeli rozwiązanie należy „wzmocnić", to wybrany zostanie ten klient, dla którego określony wskaźnik będzie największy.

W algorytmie Ant Colony zastosowano dwie metody uaktualniania ilości feromonu: uaktualnienie lokalne i globalne.

3.4. Modyfikacja Max-Min

Dowolny algorytm mrówkowy można poddać pewnym ogólnym modyfikacjom, poprawiającym jego efektywność.

Wadą rodziny algorytmów mrówkowych jest występowanie fazy stagnacji, w której generowane są wciąż identyczne rozwiązania. Dzieje się tak wtedy, gdy na drogach należących do generowanego rozwiązania odłoży się znacznie więcej feromonu, niż na pozostałych drogach.

Celem wprowadzenia modyfikacji Max-Min jest powstrzymanie występowania fazy stagnacji. Zawężeniu ulega dziedzina ilość feromonu, zostaje ona ograniczona do pewnego przedziału, określającego jej minimalną i maksymalną dozwoloną wartość.

3.5. Zastosowanie lokalnej optymalizacji

Algorytmy rozwiązujące problem dostawy można podzielić na dwie grupy: algorytmy „budujące" rozwiązanie i algorytmy „poprawiające" rozwiązanie, zwane również algorytmami optymalizacji lokalnej. Algorytmy mrówkowe zaliczamy do pierwszej grupy, gdyż optymalizacja polega na tworzeniu nowych rozwiązań, które porównywane są z najlepszym dotychczas znalezionym. Algorytmy drugiej grupy, tj. algorytmy „poprawiające" rozwiązanie, zaczynają od pewnego arbitralnie przyjętego rozwiązania i starają się je poprawić przy zastosowaniu pewnych reguł.

Po zbudowaniu rozwiązania przez „mrówkę" można poddać to rozwiązanie lokalnej optymalizacji, aby poprawić wygenerowane rozwiązanie przed porównaniem z dotychczas najlepszym. W badanych algorytmach mrówkowych z lokalną optymalizacją wykorzystano dobrze znany algorytm 2-optymalny [5]. Za jego wykorzystaniem przemawia niska złożoność

obliczeniowa, $O(n^2)$ oraz łatwość implementacji. Lokalną optymalizację zastosowano w odniesieniu do wszystkich wygenerowanych rozwiązań w danym cyklu, co poprawia jakość informacji, którą wykorzystuje „kolonia mrówek" podczas optymalizacji.

4. Wyniki przeprowadzonych badań

Cztery podstawowe algorytmy mrówkowe oraz dwa algorytmy stanowiące modyfikacje algorytmu Ant Colony poddano badaniom w celu sprawdzenia wyników osiąganych dla problemu dostawy. Badano algorytmy Ant Cycle, Ant Density, Ant Quantity, Ant Colony oraz dwie modyfikacje: algorytm Ant Colony Max-Min i Ant Colony z lokalną optymalizacją.

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 100-krotnym wykonaniu każdego algorytmu dla zbioru DANE500 dla rozmiarów 150, 200, 250, 300, 350, 400, 450, 500. Trzecie badanie polegało na 20-krotnym wykonaniu każdego algorytmu dla 20 zbiorów miast, każdy dla rozmiarów 30, 40, 50, 60, 70, 80, 90, 100.

Rozwiązania porównywano z najlepszym rozwiązaniem znalezionym podczas prowadzonych badań (opisanych w pracach [2], [6] 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 znalezionego rozwiązania. Wyniki otrzymane w kolejnych etapach badań przedstawiają tabele 1, 2 i 3.

Tabela 1

Średnia jakość najlepszego, średniego i najgorszego rozwiązania, oraz odchylenie standardowe uzyskane w trakcie pierwszego badania

algorytm	najlepsze	średnie	najgorsze	odch. std.
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
AntCycle	100,836%	102,819%	104,693%	8,08

Tabela 2

Średnia jakość najlepszego, średniego i najgorszego rozwiązania, oraz odchylenie standardowe uzyskane w trakcie drugiego badania

Algorytm	najlepsze	średnie	najgorsze	odch. std.
AntColony	101.193%	101.748%	102.218%	10.14
AntColony LO	100,475%	100,989%	101,581%	13,39
AntColony MM	101,158%	101,735%	102,276%	10,07
AntDensity	102,212%	102,814%	103,239%	11,23
AntQuantity	102,016%	102,636%	103,036%	12,07
AntCycle	103,880%	104,785%	105,344%	16,38

Tabela 3

Średnia jakość najlepszego, średniego i najgorszego rozwiązania, oraz odchylenie standardowe uzyskane w trakcie trzeciego badania

algorytm	najlepsze	średnie	najgorsze	odch. std.
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
AntCycle	101,813%	103,088%	104,172%	7,76

Opis metody wyznaczenia wartości parametrów badanych algorytmów oraz wartości tychże parametrów można znaleźć w pracy [6].

5. Wnioski

Najlepszym algorytmem spośród badanych, ze względu na jakość otrzymanych rozwiązań, okazał się algorytm Ant Colony z lokalną optymalizacją. Najgorsze wyniki uzyskano dla algorytmu Ant Cycle. Badania potwierdziły oczekiwaną poprawę wyników otrzymywanych dla algorytmu mrówkowego z lokalną optymalizacją. Lepsze wyniki

uzyskano kosztem większego czasu wykonania algorytmu. Badany algorytm Ant Colony z lokalną optymalizacją uzyskiwał lepsze rozwiązania od algorytmu Ant Colony średnio o 0,5%. Modyfikacja algorytmu Max-Min nie przyniosła oczekiwanej poprawy jakości otrzymywanych wyników. Jakość wyników otrzymywanych przez najlepszy spośród algorytmów mrówkowych Ant Colony z lokalną optymalizacją oscylowała wokół 101%. Jest to znacznie więcej niż jakość wyników generowanych przez inne algorytmy heurystyczne, takie jak algorytmy k-optymalne, czy przeszukiwanie tabu. Niewątpliwą zaletą algorytmów mrówkowych jest ich niska złożoność obliczeniowa, która jest $O(n^2)$.

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. Lin S.: Computer solutions of the traveling salesman problem. Bell Systems Journal, vol.44, pp. 2245-2269, 1965
6. Skórczyński A.: Algorytmy mrówkowe. Praca dyplomowa magisterska, Politechnika Śląska, Instytut Informatyki, Gliwice 1998.
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 10 marca 1999 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 the adaptations of some Ant algorithms are presented.

We discuss four basic variants of the Ant algorithms. Each of them was examined in three 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. For each algorithm and all stages of examination the average of differences for all sets was computed. The results are presented in table 1,2 and 3.

The tests show that the Ant Colony algorithm with the local optimization is the best. The obtained results were better when the local optimization was used.. The Max-Min modification gives no improvement of results.