WYKORZYSTANIE ALGORYTMÓW GENETYCZNYCH I MRÓWKOWYCH W PROBLEMACH TRANSPORTOWYCH

Justyna Zduńczuk, Wojciech Przystupa
Katedra Zastosowań Matematyki, Uniwersytet Przyrodniczy w Lublinie

Streszczenie. W pracy przedstawiono możliwości zastosowania metaheurystyk w transporcie. Przy użyciu algorytmu genetycznego i mrówkowego dokonano optymalizacji długości trasy przejazdu, a rezultaty porównano ze znanymi wynikami. Przedstawiono również próbę optymalizacji tras ze względu na czas trwania przejazdu.

Słowa kluczowe: transport**,** algorytm genetyczny, algorytm mrówkowy, metaheurystyka

Wstęp

Głównym celem transportu w przedsiębiorstwie rolnym jest organizacja i synchronizacja systemu fizycznego przepływu surowców i materiałów od producentów lub hurtowników do konsumentów, poprzez wszystkie fazy procesu produkcyjnego, zgodnie z zasadami zarządzania logistycznego [Marczuk 2002]. Wydatki na transport stanowią niemałą część ogólnych kosztów, które tworzą różnicę między kosztem wyprodukowania towaru, a ceną płaconą przez konsumenta. Ponadto, niektóre towary powinny być możliwie szybko przetransportowane ze względu na ograniczony okres ważności. Powody te skłaniają do zwrócenia uwagi na problem optymalizacji transportu. Jednocześnie ciągły postęp technologiczny i informacyjny gwarantuje pojawianie się nowych metod, które z powodzeniem mogą być używane w rozwiązywaniu problemów transportowych. Do takich metod z pewnością należą algorytmy sztucznej inteligencji przedstawione w niniejszej pracy.

Celem pracy było przedstawienie możliwości zastosowania metaheurystyk w transporcie. Przy użyciu algorytmu genetycznego i mrówkowego dokonano optymalizacji długości trasy przejazdu, a rezultaty porównano ze znanymi wynikami. Przedstawiono również próbę optymalizacji tras ze względu na czas trwania przejazdu.

Materiały i metody

Metody heurystyczne nie zapewniają znalezienia optymalnego rozwiązania problemu, ale z dużym prawdopodobieństwem wykrywają rozwiązania bliskie optimum. Do takich metod należą właśnie algorytmy genetyczne i algorytmy mrówkowe. Są to metody, których zasady działania zostały zaczerpnięte z obserwacji natury i z dużym powodzeniem są one wdrażane do rzeczywistych problemów optymalizacyjnych [Trojanowski 2005].

Algorytmy genetyczne powstały w latach 60-tych, ich twórcą był John Holland. Metoda jest inspirowana procesami ewolucji i genetyki. Operuje na populacji osobników będących potencjalnymi rozwiązaniami danego problemu. Osobniki podlegają procesowi reprodukcji, który można porównać do rozmnażania w naturalnym środowisku. Potomstwo powstaje poprzez zmieszanie ze sobą materiału genetycznego rodziców. Sporadycznie zdarzają się również losowe zmiany w genach, czyli mutacje. Przeżywają tylko jednostki najsilniejsze, czyli najlepiej dostosowane do otoczenia i to one się rozmnażają.

Twórcą algorytmów mrówkowych był Marco Dorigo. Metoda ta powstała w latach 90-tych i wzięła inspirację z zachowania prawdziwych mrówek. W drodze od źródła pożywienia do mrowiska mrówki porozumiewają się za pomocą tzw. śladu feromonowego. Każda z mrówek pozostawia na swojej trasie taki ślad. Im ścieżka jest krótsza, tym większe stężenie feromonu, ponieważ więcej mrówek zdążyło nią przejść. W momencie wyboru, mrówka wybiera przeważnie tą ścieżkę, na której ślad feromonowy jest silniejszy. Algorytm operuje na populacji „agentów” tzw. sztucznych mrówek. Każda mrówka samodzielnie tworzy rozwiązanie. Opiera się jednak o ilość feromonu pozostawionego przez inne mrówki. Istotną kwestią jest równomierne odparowywanie feromonu, tak jak to się dzieje w naturze. Dzięki temu algorytm jest w stanie „zapomnieć” znalezione wcześniej optimum lokalne.

Przedstawione metody zastosowano do wyboru optymalnej trasy pomiędzy określonymi punktami, które mogą być traktowane jako miejscowości (problem komiwojażera). Aplikacja napisana została w języku C# z wykorzystaniem narzędzia Microsoft Visual Studio 2005. Trasa była optymalizowana nie tylko pod względem jej długości, ale również czasu trwania przejazdu. Możliwe to było dzięki wykorzystaniu danych obrazujących średnie prędkości pomiędzy poszczególnymi punktami.

Algorytm genetyczny

W algorytmie wykorzystano reprezentację ścieżkową, gdzie kolejne numery odpowiadają kolejno odwiedzanym punktom. Pojedynczy osobnik przechowuje zarówno wybraną trasę, jak i jej długość, oraz czas trwania przejazdu. Algorytm pracuje na populacji 50 osobników, czyli potencjalnych rozwiązań problemu. Pozostałymi parametrami algorytmu jest prawdopodobieństwo krzyżowania równe 0,5 oraz prawdopodobieństwo mutacji równe 0,1. Liczba iteracji algorytmu (warunek zakończenia) jest wybierana przez użytkownika. Schemat algorytmu jest pokazany na rys 1.

```
procedure gen
Wybierz populację początkową
do
{
   Oceń populację
   Wybierz osobniki do nowej populacji
   Przeprowadź krzyżowanie
   Poddaj osobniki mutacji
}
while (warunek końca algorytmu)
```

Rys. 1. Schemat algorytmu
Fig. 1. Algorithm diagram

Populacja początkowa nie jest wybierana całkowicie losowo, a generowana na podstawie tablicy najbliższych sąsiadów. Jeżeli najbliższy danemu punktowi sąsiad nie został jeszcze odwiedzony, to zostaje dodany do trasy, w przeciwnym wypadku jest wybierany następny. Jeżeli wszyscy najbliżsi sąsiedzi znajdują się już na trasie, to wybierany jest losowy sąsiad spoza listy.

Ocena populacji jest przeprowadzana za pomocą procedury, która każdemu rozwiązaniu przyporządkowuje prawdopodobieństwo wyboru do nowej populacji na podstawie wartości wybranej funkcji (długości trasy, czasu). Osobniki do nowej populacji są wybierane metodą selekcji ruletkowej, przy czym do populacji dodatkowo wchodzi zawsze najlepszy osobnik dotąd znaleziony.

Wymiana materiału genetycznego pomiędzy osobnikami następuje poprzez krzyżowanie OX [Michalewicz 2003].

Mutacja polega na zamianie miejscami dwóch sąsiadujących ze sobą punktów. Losowo jest wybierana liczba, która reprezentuje pozycję punktu do zamiany. Jeżeli wybrany punkt znajduje się po lewej stronie od środka łańcucha, to jest zamieniany z punktem leżącym po prawej stronie, w przeciwnym przypadku zamienia się ze swoim lewym sąsiadem.

Algorytm mrówkowy

W programie wykorzystano algorytm *MAX-MIN*, w którym ilość feromonu na ścieżkach jest limitowana i tylko najlepsza dotąd znaleziona mrówka pozostawia ślad feromonowy. Do parametrów algorytmu należą współczynniki α i β, określające wpływ pozostawionego feromonu na prawdopodobieństwo wyboru punktu, ustalone odpowiednio na 1 i 2 [Dorigo, Stützle 2004]. Liczbę mrówek ustalono na 10. Liczba iteracji algorytmu jest wybierana przez użytkownika. Schemat algorytmu jest pokazany na rys. 2.

```
procedure MAX-MIN
Stwórz struktury danych
Nałóż początkową wartość feromonu na
  krawędzie
  i oblicz prawdopodobieństwa
do
{
  Stwórz ścieżkę dla każdej mrówki
  Oceń trasy
  Odparuj feromon
  Nanieś nowy feromon na krawędzie
}
while (warunek końca algorytmu)
```

Rys. 2. Schemat algorytmu MAX-MIN
Fig. 2. MAX-MIN algorithm diagram

Ilość feromonu pozostawionego na krawędziach jest przechowywana w odpowiedniej tablicy. Pojedyncza mrówka przechowuje zarówno kolejność odwiedzonych punktów, jak również długość całej przemierzonej trasy, czas jej trwania oraz wartość funkcji wybranej do optymalizacji. Ponadto, w skład struktury pojedynczej mrówki wchodzi tablica, w której wartości boolowskie informują, czy dany punkt znajduje się już na trasie danej mrówki. Algorytm przechowuje także informacje o najlepszej dotychczas znalezionej trasie i jej długości.

Jak wspomniano wcześniej, ilość feromonu na krawędziach jest limitowana. Jeżeli ilość feromonu na danej krawędzi ma przekroczyć górny limit τ_{max}, to jest ona ustalana na wartość tego limitu. Podobnie, najniższa wartość feromonu nie może być mniejsza niż τ_{min} [Dorigo, Stützle 2004].

Prawdopodobieństwo przejścia pojedynczej mrówki od punktu i do punktu j jest określone wzorem:

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i^k} \left(\left[\tau_{il}(t)\right]^{\alpha} \cdot \left[\eta_{il}\right]^{\beta}\right)} \; dla \; j \in N_i^k \tag{1}$$

gdzie:

$\tau_{ij}(t)$ – natężenie feromonu na krawędzi łączącej punkty i oraz j w bieżącej iteracji t,
$\eta_{ij} = 1/d_{ij}$ – odwrotność jakości krawędzi d_{ij} (długości lub czasu),
α i β – parametry określające wpływ pozostawionego feromonu na wyliczone prawdopodobieństwo,
N^k_i – dopuszczalne sąsiedztwo dla aktualnego położenia mrówki k, tj. zbiór jeszcze nie odwiedzonych punktów, do których istnieje połączenie z punktu i.

Wartości prawdopodobieństwa są przechowywane w tablicy i uaktualniane po każdorazowym naniesieniu nowych wartości feromonu.

Każda mrówka buduje swoją drogę niezależnie, opierając się jedynie o informacje pozostawione przez inne mrówki w postaci ilości feromonu na krawędziach. Rozpoczyna swoja wędrówkę od losowego miasta i w pierwszej kolejności sprawdza swoich najbliższych sąsiadów. Jeżeli dla danego punktu wszyscy najbliżsi sąsiedzi znajdują się już na trasie, to wybierany jest punkt, któremu odpowiada największe prawdopodobieństwo przejścia.

Po wygenerowaniu tras przez wszystkie mrówki wyliczane są jakości tras. Następnie feromon zostaje odparowany, poprzez zmniejszenie wartości w tablicy feromonu i najlepsza mrówka z całego przebiegu algorytmu pozostawia ślad na swojej ścieżce. Ilość pozostawionego przez nią feromonu zależy od jakości przebytej przez nią trasy.

Wyniki i analiza

Program przetestowano na ogólnodostępnej bibliotece TSPLIB [TSPLIB]. Biblioteka zawiera kilkadziesiąt przykładów danych dla problemu komiwojażera, wraz z najlepszymi znalezionymi rozwiązaniami tych przykładów. Wyniki testowania programu pokazane są w tabeli 1. Ilość iteracji algorytmów była określana w zależności od wielkości problemu i dla 1000 punktów wynosiła 1 000 000.

Tabela 1. Wyniki testowania programu na problemach z TSPLIB
Table 1. Results of the program testing for problems with TSPLIB

Problem	Ilość miast	Rozwiązanie optymalne*	Algorytm mrówkowy	Algorytm genetyczny
bays29	29	2020	2038	2032
eil51	51	426	452	438
kroa100	100	21282	22943	24659
linhp318	318	41345	47306	45497
pr1002	1002	259045	305052	416104

Jako rozwiązanie optymalne rozumie się najlepsze dotychczas znalezione rozwiązanie danego problemu, znajdujące się w bibliotece TSPLIB

Źródło: obliczenia własne autora

Otrzymane wyniki wydają się być satysfakcjonujące i w miarę zbliżone do rozwiązań optymalnych, za jakie uznano najlepsze dotychczas znalezione rozwiązania. Niestety, w bibliotece TSPLIB, oprócz najlepszych rozwiązań nie są podane metody jakimi te rozwiązania otrzymano. Algorytmów nie testowano na problemach większych niż przedstawione w powyższej tabelce, gdyż w takim przypadku korzystne by było dynamiczne zarządzanie parametrami algorytmów, dla szybszego i bardziej efektywnego poszukiwania wyniku. Być może, zaistniałaby także potrzeba przeorganizowania struktur danych.

Przeprowadzono również optymalizację trzech rodzajów parametrów: długości trasy, czasu przejazdu i kosztów przejazdu. Każdy z problemów był optymalizowany osobno dla każdego parametru. Dla każdej optymalizacji wykonano 10 prób i wybrano najlepsze wyniki, które zostały przedstawione w tabeli 2. Poniższe wyniki różnią się od tych przedstawionych w tabeli 1, ze względu na zastosowaną mniejszą ilość iteracji algorytmów, równą 10 000.

Tabela 2. Wyniki optymalizacji długości trasy, czasu przejazdu i kosztów
Table 2. Results of route length, drive duration and costs optimisation

Ilość miast	Rodzaj optymalizacji	Algorytm mrówkowy			Algorytm genetyczny		
		długość	czas	koszt	długość	czas	koszt
29	długość	2033	1884	2162	2033	1934	2187
	czas	2048	1872	2165	2113	1825	2180
	koszt	2033	1884	2162	2081	1829	2163
51	długość	455	601	574	438	537	531
	czas	481	533	555	497	514	555
	koszt	468	528	545	470	512	538
100	długość	22943	29422	28477	27457	31682	32315
	czas	23400	28453	28266	28676	31099	32755
	koszt	23319	29108	28545	28025	31729	32680

Źródło: obliczenia własne autora

Wspomniane koszty przejazdu nie są rzeczywistymi kosztami transportu, a jedynie funkcją pozostałych dwóch parametrów – długości trasy oraz czasu trwania przejazdu. Optymalizację kosztów możemy potraktować jako jednoczesną optymalizację pozostałych dwóch parametrów (długości oraz czasu) z odpowiednimi współczynnikami wagowymi. Przedstawione w tabeli 2 wyniki wskazują możliwość zastosowania opisanych algorytmów do optymalizacji problemu z uwzględnieniem kilku optymalizowanych parametrów.

Przedstawione wyniki wskazują na możliwość zastosowania opisanych metod w optymalizacji transportu. Algorytmów nie stosowano z innymi parametrami, zwłaszcza prawdopodobieństwo mutacji może się wydawać za duże w świetle znanej literatury. Sam operator mutacji, zaproponowany przez autora, również niekoniecznie wpływa pozytywnie na algorytm. W świetle tych stwierdzeń, należy założyć, że zwłaszcza algorytm genetyczny może w innych warunkach zwracać dużo lepsze wyniki. Algorytmy mrówkowe natomiast nie są tak wrażliwe na dobór parametrów, dlatego też wyniki otrzymane tą metodą wydają się być lepsze. Jednak celem autora było przedstawienie możliwości zastosowania powyższych metod w optymalizacji transportu i dobór parametrów uważał w tym przypadku za sprawę drugorzędną.

Podsumowanie

Metody sztucznej inteligencji, do jakich zaliczyć możemy wykorzystane algorytmy, znajdują coraz szersze zastosowania, również w kręgu problemów inżynierii rolniczej. Dzięki prostym założeniom i znacznej efektywności, mogą być stosowane do wielu problemów. Najważniejszą kwestią jest zastosowanie odpowiedniej funkcji oceny, dzięki której znalezione rozwiązania mogą być właściwie sklasyfikowane. W przypadku algorytmów mrówkowych istotną kwestią jest także przedstawienie wyników w postaci zbliżonej do trasy rzeczywistej mrówki. Algorytmy genetyczne natomiast potrzebują odpowiednich operatorów krzyżowania i mutacji, właściwych dla przyjętej reprezentacji rozwiązań i tworzących zawsze osobniki dopuszczalne. W bardziej złożonych problemach, kiedy trudno jest dobrać odpowiednie operatory tworzące osobniki dopuszczalne, można skorzystać ze strategii postępowania z osobnikami niedopuszczalnymi [Michalewicz 2003]. Istotne jest także dobranie odpowiednich parametrów algorytmu, takich jak wielkość populacji, prawdopodobieństwo krzyżowania i mutacji oraz oczywiście warunku zakończenia algorytmu.

Pomimo, że w pracy przedstawiono prosty i znany problem, intencją autora było zwrócenie uwagi na tzw. „inteligentne techniki obliczeniowe" i możliwość ich zastosowania w problemach transportu rolniczego. Opisane techniki można z powodzeniem stosować do bardziej złożonych problemów, takich jak np. organizacja transportu z użyciem okien czasowych. Metody sztucznej inteligencji są również skutecznie wdrażane w dziedzinie ekonomii, gdzie optymalizacja często opiera się na dużej liczbie parametrów [Gwiazda 1998].

Bibliografia

Dorigo M., Stützle T. 2004. Ant Colony Optimization. MIT Press. ISBN 0-262-04219-3.

Gwiazda T. 1998. Algorytmy genetyczne. Zastosowania w finansach. Wydawnictwo Wyższej Szkoły Przedsiębiorczości i Zarządzania im. L. Koźmińskiego. ISBN 83-86846-19-4.

Marczuk A. 2002. Logistyczne zarządzanie transportem truskawek. Acta Scientiarum Polonorum. Seria Technica Agraria. Nr 1(2). s. 5-12.

Michalewicz Z. 2003. Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT. ISBN 83-204-2881-5.
Trojanowski K. 2005. Metaheurystyki praktycznie. Wydawnictwo Wyższej Szkoły Informatyki Stosowanej i Zarządzania (WIT). ISBN 83-88311-78-6.
TSPLIB The TSPLIB Symmetric Traveling Salesman Problem Instances [online] Heidelberg [dostęp 20.03.2007] Dostępny w internecie: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html

USING GENETIC AND ANT ALGORITHMS TO SOLVE TRANSPORT PROBLEMS

Abstract. The paper presents possibilities to employ metaheuristics in transport. The research involved using genetic and ant algorithm to optimise drive/ride route length, and obtained results were compared to known results. Moreover, the paper presents an effort to optimise routes with regard to drive duration.

Key words: transport**,** genetic algorithm, ant algorithm, metaheuristics

Adres do korespondencji:
Justyna Zduńczuk; e-mail: justyna.zdunczuk@ar.lublin.pl
Katedra Zastosowań Matematyki
Uniwersytet Przyrodniczy w Lublinie
ul. Akademicka 13
20-950 Lublin