

ZASTOSOWANIE ALGORYTMU SYMULOWANEGO WYŻARZANIA DO ROZWIĄZANIA PROBLEMU DOSTAW Z OKNAMI CZASOWYMI

Streszczenie

W artykule przedstawiono metodę znalezienia najlepszych rozwiązań problemu dostawy z oknami czasowymi. Zaproponowany algorytm jest implementacją sekwencyjnego symulowanego wyżarzania. Zastosowano nową postać funkcji przejścia, dzięki czemu bardzo szybko otrzymuje się rozwiązania bliskie optymalnym.

Słowa kluczowe: symulowane wyżarzanie, problem dostaw z oknami czasowymi

1. WPROWADZENIE

Transport dóbr materialnych jest bardzo ważnym elementem działalności człowieka. Prawidłowe zarządzanie siecią dystrybucyjną oraz parkiem pojazdów pozwala na znaczne obniżenie kosztów dostaw.

Obecnie na rynku systemów ERP można zaobserwować zwiększone zapotrzebowanie na aplikacje wspomagające planowanie tras [6]. Jest to związane z faktem, że nawet niewielka oszczędność na pojedynczych wysyłkach pozwala na znaczne redukcje kosztów w dłuższym okresie. Prawidłowe zarządzanie wysyłką pozwala na lepsze spełnianie potrzeb klientów, co sprzyja poprawie wizerunku firmy.

2. PROBLEM DOSTAW Z OKNAMI CZASOWYMI

Problem dostaw (Vehicle Routing Problem – VRP) jest rozwinięciem problemu komiwojażera. W problemie dostaw pomiędzy zbiorem klientów poruszają się pojazdy o określonej ładowności. Każdy z klientów ma określone zapotrzebowanie na towar, zaś kolejność ich odwiedzania nie ma znaczenia. Każda trasa zaczyna się oraz kończy w tym samym miejscu, w centralnym magazynie. Rozwiązanie problemu dostaw oznacza znalezienie najkrótszej łącznej długości tras wszystkich pojazdów, przy czym ładowność każdego z nich nie może zostać przekroczona.

W problemie dostaw z oknami czasowymi (VRPTW – Vehicle Routing Problem with Time Windows), oprócz założeń przyjętych w klasycznym problemie dostaw VRP, każdy klient ma zdefiniowany najwcześniejszy możliwy czas rozpoczęcia e_i obsługi dostawy oraz najpóźniejszy możliwy czas rozpoczęcia f_i . Wartości te wyznaczają tzw. okno czasowe klienta. Każdy klient posiada określony czas trwania obsługi s_i , czyli okres, w jakim u klienta przebywa pojazd.

Problem dostaw został formalnie opisany przez Salomona [8, 9] w 1987 roku. Przygotował on zbiór zagadnień, które są używane do testowania algorytmów oraz do porównywania wyników badań z aktualnie istniejącymi najlepszymi rozwiązaniami. W niniejszym artykule

* Doktorant Wydziału Zarządzania Akademii Górniczo-Hutniczej

** Akademia Górniczo-Hutnicza, Wydział Zarządzania

także użyto testów Salomona do sprawdzenia skuteczności proponowanego programu. Problem dostaw z oknami czasowymi rozwiązany został przy pomocy metody symulowanego wyżarzania.

3. SYMULOWANE WYŻARZANIE

Algorytm symulowanego wyżarzania jest rozwinięciem metod przeszukiwania lokalnego, które polegają na ulepszeniu istniejącego rozwiązania do momentu, gdy nie udaje się go dalej poprawić.

Podstawy symulowanego wyżarzania zostały po raz pierwszy opisane w algorytmie Metropolis symulującym chłodzenie ciał stałych. Przy powolnym stygnięciu, cząsteczki ciała oddając energię rozkładają się w sposób systematyczny tworząc równomierne struktury. Natomiast, jeśli chłodzenie będzie zbyt szybkie, to cząsteczki rozłożą się bardziej chaotycznie. Algorytm Metropolis symuluje zmianę energii systemu podczas procesu wyżarzania. Jeżeli energia spada, to system przenosi się do nowego stanu, jeżeli energia wzrasta, to nowy stan jest akceptowany zgodnie z prawdopodobieństwem danym wzorem (1).

$$p = e^{\frac{-\Delta E}{kT}} \quad (1)$$

gdzie:

p – prawdopodobieństwo akceptacji,

E – energia systemu,

K – stała Boltzmana,

T – temperatura.

Proces wyżarzania jest powtarzany, aż do schłodzenia materiału.

Ogólny algorytm symulowanego wyżarzania dla minimalizacji funkcji F , gdzie N oznacza sąsiedztwo ma postać:

Algorytm 1.

Krok 1. Stwórz rozwiązanie początkowe x_0

Krok 2. Określ temperaturę początkową T_0

Krok 3. Dopóki $i <$ liczba iteracji

Krok 3.1. licznik=0

Krok 3.2. Dopóki licznik $<$ nrep //długość epoki

Krok 3.2.1. Wybierz rozwiązanie x z sąsiedztwa N

Krok 3.2.2. $\delta = F(x) - F(x_0)$

Krok 3.2.3. Jeżeli $\delta < 0$

$x_0 = x$

Krok 3.2.4. W przeciwnym wypadku

Krok 3.2.4.1. Wylosuj liczbę b z przedziału $[0,1]$

Krok 3.2.4.2. Jeżeli $b < \exp(-\delta/T)$

$x_0 = x$; //przejście „w górę”

Krok 3.2.5. licznik = licznik + 1;

Krok 3.3. $T = A(T)$ //zmiana temperatury

Krok 4. Zapamiętaj znalezione rozwiązanie

Istotną cechą symulowanego wyżarzania jest możliwość omijania minimów lokalnych i głębokie przeszukiwanie przestrzeni rozwiązań dopuszczalnych. Osiągnięte to zostało dzięki dopuszczeniu możliwości wyboru gorszego – od już znalezionych – rozwiązania. Wybór taki jest dokonywany z prawdopodobieństwem wyrażonym wzorem (1). Dzięki temu algorytm symulowanego wyżarzania może w określonych warunkach wyjść ze znalezionego optimum lokalnego i dalej podążać w kierunku rozwiązania optymalnego.

Zmiana stanu, czyli przejście z jednego potencjalnego rozwiązania do drugiego jest realizowane za pomocą tzw. funkcji przejścia. Proces ten, specyficzny dla konkretnego problemu, polega na znalezieniu rozwiązania sąsiedniego.

Wybór temperatury początkowej oraz funkcji zmiany temperatury bezpośrednio wpływa na zachowanie się algorytmu symulowanego wyżarzania. Prawdopodobieństwo wyboru gorszego rozwiązania jest zależne od temperatury. Jeżeli końcowe rozwiązanie ma być jak najlepsze, temperatura musi być na tyle wysoka, aby umożliwić dokładne przeszukanie sąsiedztwa. Zbyt wysoka temperatura początkowa może spowolnić pracę algorytmu, zbyt duża liczba potencjalnych rozwiązań jest sprawdzana.

W początkowym etapie działania algorytmu, gdy temperatura jest wysoka, prawdopodobieństwo wyboru gorszego rozwiązania jest także wysokie. Gdy temperatura maleje, prawdopodobieństwo również. Ten etap polega na ulepszaniu istniejącego rozwiązania. To właśnie znajduje odzwierciedlenie w drugim ważnym aspekcie algorytmu symulowanego wyżarzania, czyli w powolnym ochładzaniu.

4. DOSTOSOWANIE SYMULOWANEGO WYŻARZANIA DO PROBLEMU DOSTAW

Parametry algorytmu symulowanego wyżarzania można podzielić na dwie grupy. Pierwsza grupa, tzw. parametrów ogólnych polega na inicjalizacji temperatury początkowej, określeniu warunków stopu oraz planu wyżarzania. Druga grupa parametrów zależy od konkretnego problemu i są to: funkcja kosztu, funkcja przejścia i sposób tworzenia rozwiązania początkowego [10].

Wszystkie zaprezentowane w niniejszym rozdziale parametry są ustalone na podstawie doświadczeń przeprowadzonych za pomocą opisywanego algorytmu. Najlepsze wyniki uzyskiwane były dla zaproponowanych parametrów:

– temperatura początkowa wynosi 100 i zredukowana jest wg wzoru (2):

$$A(T) = aT \quad (1)$$

gdzie:

a jest jak to ustalono $i - i$ doświadczalnie wybranym współczynnikiem i i wynosi 0,965.

– długość epoki (Algorytm 1) $nrep = 200$,

– liczba iteracji $= 200$.

– funkcja kosztu konfiguracji zwana również funkcją celu ma postać:

$$f(s) = \alpha R + \beta D; \quad \alpha \gg \beta \quad (2)$$

gdzie:

R – liczba tras,

D – łączna długość wszystkich tras,

α, β – współczynniki wag.

W pracy przyjęto, że współczynnik β wynosi 1, współczynnik $\alpha = 10000$. Dzięki tak przyjętym parametrom funkcji, minimalizowana jest przede wszystkim liczba tras.

5. OPIS ALGORYTMU

Procedura tworzenia rozwiązania inicjalnego jest wykorzystywana na początku algorytmu symulowanego wyżarzania. W niniejszej implementacji wykorzystano własną procedurę. Jej ogólny algorytm jest następujący:

Algorytm 2.

Krok 1. Posortuj klientów rosnąco wg okna czasowego

Krok 2. Dla każdego klienta

 Krok 2.1. Dla $i = 1$ do MT_CLIENT

 Krok 2.1.1. Dla każdej trasy

 Krok 2.1.1.1. Znajdź najlepsze miejsce na trasie dla klienta

 Krok 2.1.1.2. Jeżeli nowa długość trasy < poprzednia długość

 Zapamiętaj tę trasę i miejsce

 Krok 2.1.2. Jeżeli znaleziono miejsce wstawienia

 Wstaw klienta do tej trasy w określonym miejscu

 Krok 2.1.2. W przeciwnym wypadku

 Wstaw klienta na początku nowo utworzonej trasy

 Krok 2.1.3. Usuń klienta z pierwotnej listy

 Krok 2.2. Dla $i = Liczba_klientow_z_pierwotnej_listy$ w dół do LT_CLIENT

 Krok 2.2.1. Dla każdej trasy

 Krok 2.2.1.1. Znajdź najlepsze miejsce na trasie dla klienta

 Krok 2.2.1.2. Jeżeli nowa długość trasy < poprzednia długość

 Zapamiętaj tę trasę i miejsce

 Krok 2.2.2. Jeżeli znaleziono miejsce wstawienia

 Wstaw klienta do tej trasy w określonym miejscu

 Krok 2.2.3. W przeciwnym wypadku

 Wstaw klienta na początku nowo utworzonej trasy

 Krok 2.2.4. Usuń klienta z pierwotnej listy

Krok 3. Zapamiętaj znalezione rozwiązanie

Sortowanie wg rosnącego okna czasowego w kroku 1 pozwala na umieszczenie klientów z najmniejszych oknem czasowym na początku listy. Ci najbardziej „kłopotliwi” klienci są dzięki temu przetwarzani w pierwszej kolejności. Okno czasowe jest obliczane wg formuły:

najpóźniejszy możliwy czas zakończenia obsługi – najwcześniejszy możliwy czas rozpoczęcia obsługi – czas obsługi.

Według algorytmu generowania rozwiązania początkowego najpierw bierze się pod uwagę kilku najbardziej kłopotliwych klientów – ich liczba zdefiniowana jest za pomocą parametru MT_CLIENT, a następnie przeszukuje się kilku najmniej kłopotliwych klientów – ich liczba definiowana jest za pomocą parametru LT_CLIENT. W toku doświadczeń i badań okazało się, że najbardziej efektywnymi wartościami parametrów MT_CLIENT oraz LT_CLIENT są odpowiednio 5 i 2. Dopasowywanie klientów z dużym oknem czasowym do klientów z małym oknem czasowym pozwala już na etapie tworzenia rozwiązania początkowego zredukować liczbę tras. Krok ten jest powtarzany aż do momentu, w którym wszyscy klienci są rozmieszczeni na trasach.

Funkcja przejścia jest jednym z podstawowych elementów, który warunkuje sprawność algorytmu symulowanego wyżarzania. To od niej zależy zdolność algorytmu do wychodzenia z minimów lokalnych, co jest szczególnie istotne przy niskich wartościach temperatury. W niniejszej pracy postanowiono zaproponować własną funkcję przejścia.

Algorytm 3.

Krok 1. Wylosuj maks. liczbę klientów określoną parametrem CNUMBER w tym celu:

Krok 1.1. Wylosuj trasę

Krok 1.2. Wylosuj klienta C_i na trasie

Krok 1.3. Zastosuj Metodę Zwiększających się Promieni do wylosowania sąsiedztwa danego klienta C_i

Krok 1.4. Usuń wylosowanych klientów z tras

Krok 1.5. Jeśli usunięto tylko jednego klienta wylosuj drugą trasę

Krok 1.6. Wstaw wylosowanego klienta do drugiej trasy jeśli to jest możliwe

Krok 1.7. Znajdź zawadzającego klienta na drugiej trasie. Jeśli udało się znaleźć zaznacz go do usunięcia.

Krok 1.8. Dla zawadzającego klienta z drugiej trasy wykonaj kroki jak w kroku 1.3.

Krok 1.9. Jeśli liczba klientów < CNUMBER wróć do kroku 1.1.

Krok 1.10. Jeśli liczba klientów > CNUMBER zakończ działanie funkcji i wróć do rozwiązania wcześniejszego

Krok 2. Posortuj trasy wg dystansów rosnąco

Krok 3. Dla każdego wylosowanego klienta

Krok 3.1. Dla każdej trasy

Krok 3.1.1. Znajdź najlepsze miejsce dla bież. klienta na trasie

Krok 3.1.2. Jeżeli nowa długość trasy < poprzednia długość

Zapamiętaj tę trasę i miejsce

Krok 3.2. Jeżeli znaleziono miejsce wstawienia

Wstaw klienta do tej trasy w określonym miejscu

Krok 3.3. W przeciwnym wypadku

Wstaw klienta na początku nowo utworzonej trasy

- Krok 4. Dla każdej trasy
 - Krok 4.1. Ulepsz pojedynczą trasę
- Krok 5. Posortuj trasy wg liczby klientów rosnąco
- Krok 6. Ulepsz wszystkie trasy

Zaletą zaprezentowanego algorytmu jest szybkość oraz dokładne przeszukiwanie przestrzeni rozwiązań. Elementem tej procedury jest zaproponowany przez autorów sposób grupowania klientów w zależności od odległości od bieżącego klienta.

Metoda ta polega na zwiększaniu promienia odległości od bieżącego klienta i sprawdzanie czy wewnątrz okręgu stworzonego przez ten promień znajdują się jacyś klienci. Jeśli są to tacy klienci zostają dodani do losowanych klientów wg określonego prawdopodobieństwa. Prawdopodobieństwo to zależy od promienia. Im mniejszy promień, tym prawdopodobieństwo wylosowania klienta jest większe. Metoda ta została nazwana *metodą zwiększających się promieni*.

Ten algorytm preferuje bliskich klientów dla bieżącego i ma szczególne znaczenie przy niskich temperaturach, gdzie uzyskane rozwiązanie jest ulepszane. Wtedy przenoszone są całe grupy klientów pomiędzy trasami.

W toku badań stwierdzono, że najlepszą wartością dla parametru wyrażającego liczbę usuniętych klientów w pojedynczej iteracji CNUMBER jest 10-15 procent liczby wszystkich klientów. CNUMBER określa maksymalną liczbę klientów usuniętych z tras w pojedynczej iteracji. W procesie ich ponownego wstawiania do tras bardzo duże znaczenie ma kolejność, w jakiej klienci zostali wylosowani.

W kroku 4 wykonywana jest funkcja ulepszania pojedynczej trasy. Polega ona na znajdowaniu na trasie najlepszego położenia dla każdego klienta. W tym celu bieżący klient jest z tej trasy usuwany, a następnie znajduwane jest jego najlepsze położenie w tej marszrucie. Jeżeli najlepszym miejscem dla danego klienta jest jego położenie początkowe, to trasa nie zmienia się.

W kroku 5 trasy są sortowane wg liczby klientów na trasie, a w przypadku, gdy liczba klientów dla kilku tras jest taka sama, to porównywane są sumy żądań dostawy dla klientów z danej trasy. Sortowanie umożliwia ustawienie tras z najmniejszą liczbą klientów na początku, co jest bardzo przydatne w kroku 6, czyli ulepszaniu tras.

Algorytm ulepszania tras jest własnym algorytmem polegającym na usuwaniu po kolei każdego klienta z trasy i wstawianiu go w inne miejsce. Ogólny schemat algorytmu jest następujący [10]:

Algorytm 4.

- Krok 1. Dla każdej trasy
 - Krok 1.1. Dla każdego klienta na trasie
 - Krok 1.1.1. Usuń klienta z trasy
 - Krok 1.1.1.1. Dla każdej trasy za wyjątkiem trasy bieżącej
 - Krok 1.1.1.1.1. Znajdź najlepsze położenie klienta
 - Krok 1.1.1.1.2. Jeżeli nowy koszt rozwiązania < poprzedni koszt
 - Zapamiętaj tę trasę i miejsce
 - Krok 1.1.1.1.3. Jeżeli znaleziono miejsce wstawienia
 - Wstaw klienta do tej trasy w określonym miejscu
 - Krok 1.2. Jeżeli podstawowa trasa jest pusta
 - Usuń trasę
 - Krok 1.3. Jeżeli nie znaleziono miejsca wstawienia
 - Wstaw klienta do trasy, z której go usunięto

Krok 2. Zapamiętaj znalezione rozwiązanie

Dzięki posortowaniu tras wg rosnącej liczby klientów na trasie jest możliwe usunięcie krótkich tras, które powstały we wcześniejszych krokach.

6. WYNIKI BADAŃ

Do sprawdzenia niniejszego algorytmu użyta została grupa testów RC1xx oraz R2xx zaproponowana przez M. Solomona [8]. W zbiorze tym znajdują się klienci, których położenie jest losowe oraz tacy, których lokalizacja jest częściowo uporządkowana, a częściowo wybrana losowo.

Wartości parametrów konfiguracyjnych były następujące: liczba iteracji algorytmu – 200, temperatura początkowa – 100, długość epoki – 200.

Uzyskane wyniki zostały porównane z najlepszymi aktualnie znanymi wynikami światowymi. Lista tych wyników jest dostępna w Internecie na stronie <http://www.top.sintef.no/VRP/bknown.html>

Tabela 1. Wyniki testów

Nazwa testu	Wyniki światowe		Uzyskane wyniki	
	Liczba tras	Dług. Tras	Liczba tras	Dług. Tras
RC101	14	1696,94	14	1696,94
RC102	12	1554,75	12	1554,75
RC103	11	1261,67	11	1261,67
RC104	10	1135,48	10	1135,48
RC105	13	1629,44	13	1629,44
RC106	11	1424,73	11	1424,73
RC107	11	1230,48	11	1230,95
RC108	10	1139,82	10	1139,82
R201	4	1252,37	4	1252,37
R202	3	1191,70	3	1191,70
R203	3	939,54	3	939,50
R204	2	825,52	2	825,52
R205	3	994,42	3	994,42
R206	3	906,14	3	906,14
R207	2	893,33	2	905,28
R208	2	726,75	2	728,86
R209	3	909,16	3	909,16
R210	3	939,34	3	939,37
R211	2	892,71	2	885,71

Źródło: opracowanie własne

Wyniki osiągnięte w niniejszych testach są zbliżone do wyników światowych. Dla 13 testów z 19 wyniki uzyskane przez opisany algorytm są identyczne jak najlepsze wyniki światowe. W dwóch przypadkach, czyli dla testów R203 oraz R211 zostały pobite aktualne rekordy światowe utrzymywane odpowiednio od 2005 oraz 2001 roku. Nowy wynik został zatwierdzony przez komisję międzynarodową z siedzibą w Oslo w Norwegii. W żadnym wypadku otrzymany wynik nie różnił się od wyników światowych liczbą tras. Pod względem liczby kilometrów maksymalna różnica wynosi 1,34%.

7. WNIOSKI

Aplikacja testująca algorytm rozwiązujący problem dostaw z oknami czasowymi została napisana w języku C. Podsumowując niniejsze eksperymenty, dla 13 testów osiągnięte wyniki były równe najlepszym wynikom światowym, dwa światowe rekordy utrzymywane od 2001 oraz 2005 roku zostały pobite, pozostałe testy różnią się liczbą kilometrów, w żadnym teście wyniki nie różnią się liczbą tras.

Dużą zaletą zaproponowanego rozwiązania jest wspomniana już łatwość jego implementacji oraz krótki czas działania programu – do ok. 40 minut. Z tego powodu wydaje się być słuszne kontynuowanie badań nad zaproponowaną postacią funkcji przejścia.

LITERATURA

- [1] Cerny V.: *A thermodynamical approach to traveling salesman problem: an efficient simulation algorithm*. Journal of Optimization Theory and Applic., 1985, 45, s. 41-45.
- [2] Czarnas P.: *Problem komiwojażerów z oknami czasowymi. Rozwiązanie metodą symulowanego wyżarzania*. Uniwersytet Wrocławski, 2001.
- [3] Czech Z. J., Czarnas P.: *A Parallel Simulated Annealing for the Vehicle Routing Problem with Time Windows*. Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, 2002, Canary Islands, Spain, s. 376-383.
- [4] Gluhovsky I.: *Evolutionary Simulated Annealing with Application to Image Restoration*. Journal of Computational & Graphical Statistics. 2007.
- [4] Kirkpatrick S., Gellat C.D., Vecchi M.P.: *Optimization by simulated annealing*. 1983, Science, 220, s. 671-680.
- [6] Michłowicz E.: *Algorytm ewolucyjny dla problem komiwojażera z wieloma centrami dystrybucyjnymi*. Logistyka, 2009, 4, s. 83 (1-10).
- [7] Parks G.T., Suppaitnarm A., Seffen K.A., Clarkson P.J., Liu J.S.: *Design by multi-objective optimization using simulated annealing*. 12th International Conference in Engineering Design, 1999, 3, s. 1395-1400.
- [8] Solomon M.: *Algorithms for the Vehicle Routing and Scheduling Problem with Time Windows Constraints*. Oper. Res., 1987, 35, s. 254-265.
- [9] Solomon M., Desrosiers J.: *Time windows constrained routing and scheduling problems*. Transp. Sci., 1988, 22, s. 1-13.
- [10] Watson J., Boswell S., Sexton J.: *Simulated annealing - a method in finding efficient loading solutions*. Land Warfare Conference, 22-24., pp. 445-453, Brisbane, Australia. 2006.
- [11] Woch M.: *Rozwiązanie problem dostaw z oknami czasowymi za pomocą symulowanego wyżarzania*. Studia Informatica 2(58), 2004.
- [12] Woch M., Łebkowski P.: *Rozwiązanie problemu dostaw z oknami czasowymi w systemie Microsoft Dynamics NAV 5.0*. Innowacyjno-efektywnościowe problemy teorii i praktyki zarządzania pod redakcją Piotra Łebkowskiego, AGH Uczelniane Wydawnictwa Naukowo-Dydaktyczne. s. 73-80, Kraków, 2009.

SIMULATED ANNEALING FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Abstract

In this article a method to solve the Vehicle Routing Problem with Time Windows is presented. The proposed algorithm is an implementation of sequential simulated annealing. New transfer function is presented which guarantees feasible optimal solutions computed in reasonable time.

Keywords: simulated annealing, vehicle routing problem with time windows