STUDIA INFORMATICA 2005
Volume 26 Number 1 (62)

Bożena WIECZOREK
Górnośląska Wyższa Szkoła Handlowa, Katedra Informatyki

RÓWNOLEGŁE ALGORYTMY SYMULOWANEGO WYŻARZANIA DLA PROBLEMU TRASOWANIA POJAZDÓW Z OGRANICZENIAMI CZASOWYMI

Streszczenie. W pracy przedstawiono trzy równoległe algorytmy symulowanego wyżarzania dla problemu trasowania pojazdów z ograniczeniami czasowymi. Zbadane zostały sposoby współpracy procesów w algorytmach równoległych w celu uzyskania rozwiązań o jak najwyższej jakości. Przedstawiono wyniki badań eksperymentalnych dla wybranych danych testowych Solomona.

Słowa kluczowe: symulowane wyżarzanie, trasowanie pojazdów z ograniczeniami czasowymi, algorytmy równoległe

PARALLEL SIMULATED ANNEALING ALGORITHMS FOR THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Summary. The paper describes three parallel simulated annealing algorithms to solve the vehicle routing problem with time windows. The ways of co-operation between processes to achieve a better accuracy of solution to the problem is investigated. The experimental results carried out on the test set by Solomon are presented.

Keywords: simulated annealing, vehicle routing problem with time windows, parallel algorithms

1. Wstęp

Problem trasowania pojazdów z ograniczeniami czasowymi jest jednym z wielu problemów optymalizacji kombinatorycznej o charakterze logistycznym. Ogólnie polega on na znalezieniu tras dla pojazdów realizujących dostarczanie towarów do klientów. Dostarczane towary są składowane w centralnym magazynie. Pojazdy mają określoną ładowność i muszą dotrzeć do każdego klienta na swojej trasie w odpowiednim czasie, w tzw. oknie czasowym. Rozwiązanie problemu polega na zminimalizowaniu liczby pojazdów dostawczych oraz sumarycznej drogi, jaką pokonują pojazdy realizując dostawę towarów.

Problem trasowania pojazdów z ograniczeniami czasowymi należy do klasy problemów *NP-trudnych*. Z racji swoich zastosowań praktycznych, takich jak: planowanie dostawy towarów do sieci sklepów, trasowanie autobusów szkolnych, dystrybucja poczty, gazet i in., podejmowanych jest wiele prób zastosowania różnych metod metaheurystycznych [10] do znalezienia rozwiązań bliskich optymalnym.

W niniejszej pracy przedstawiono trzy równoległe algorytmy symulowanego wyżarzania dla problemu trasowania pojazdów z ograniczeniami czasowymi. Zbadane zostały sposoby współpracy procesów w algorytmach równoległych w celu uzyskania rozwiązań o jak najwyższej jakości. Przedstawiono wyniki badań eksperymentalnych dla wybranych danych testowych Solomona.

Praca składa się z siedmiu rozdziałów. W rozdziale 2 dokonano sformułowania rozwiązywanego problemu. Rozdział 3 zawiera opis sekwencyjnego algorytmu symulowanego wyżarzania. W rozdziale 4 przedstawiono algorytm symulowanego wyżarzania dla problemu trasowania pojazdów z ograniczeniami czasowymi. Rozdział 5 poświęcony jest równoległym algorytmom symulowanego wyżarzania. Wyniki badań eksperymentalnych przedstawiono w rozdziale 6. Rozdział 7 zawiera podsumowanie pracy.

2. Sformułowanie problemu

Problem trasowania pojazdów z ograniczeniami czasowymi formułowany jest następująco. Dane są centralny magazyn towarów oraz zbiór n klientów oznaczony jako C. Dane są lokalizacje magazynu oraz klientów w postaci grafu nieskierowanego $G = (N, A)$. Graf składa się z $|C|+2$ wierzchołków, gdzie wierzchołki 1, 2, ..., n oznaczają klientów, magazyn zaś reprezentowany jest przez wierzchołek 0 (punkt startowy) oraz $n+1$ (punkt powrotu). Zbiór wierzchołków, tj. 0, 1, 2, ..., $n+1$, oznaczony jest jako N. Zbiór krawędzi, oznaczony jako A, reprezentuje połączenia pomiędzy magazynem a klientami oraz pomiędzy każdą parą klientów. Dla każdej krawędzi (i, j), gdzie $i \neq j$, określony jest koszt $c_{i,j}$ przejazdu od klienta i do klienta j oraz czas tego przejazdu $t_{i,j}$, który zawiera również czas obsługi klienta i. Zadaniem zbioru pojazdów dostawczych V jest dostarczenie towarów (lub ich odebranie) do każdego klienta i zgodnie z jego zapotrzebowaniem d_i. Każdy pojazd obsługuje podzbiór klientów w ramach trasy, która zaczyna się i kończy w magazynie. Pojazdy mają określoną ładowność q. Całkowita suma zapotrzebowań klientów obsługiwanych przez pojazd na danej trasie nie może przekroczyć q. Dla każdego klienta zdefiniowane jest okno czasowe obsługi $[a_i, b_i]$. Wartości a_i oraz b_i określają, odpowiednio, najwcześniejszy oraz najpóźniejszy czas rozpoczęcia obsługi klienta. Pojazd może przybyć do klienta przed jego oknem czasowym, lecz wtedy musi czekać do momentu a_i, w którym obsługa może się rozpocząć. Magazyn ma również zdefiniowane okno czasowe $[a_0, b_0]$ równe $[a_{n+1}, b_{n+1}]$, które oznacza całkowity, możliwy zakres obsługi klientów. Żaden pojazd nie może opuścić magazynu przed czasem a_0 oraz wrócić z powrotem po czasie b_{n+1}.

Przyjmijmy, że a_i jest nieujemną liczbą rzeczywistą, a b_i, $c_{i,j}$, d_i, $t_{i,j}$ i q są dodatnimi liczbami rzeczywistymi. Model matematyczny problemu zawiera zmienną decyzyjną x. Dla każdej krawędzi (i, j), gdzie $i \neq j$, $i \neq n+1$, $j \neq 0$, oraz dla każdego pojazdu k definiujemy zmienną $x_{i,j,k}$ jako:

$$x_{i,j,k} = \begin{matrix} 0, \text{ jeśli pojazd } k \text{ nie jedzie od wierzchołka } i \text{ do wierzchołka } j, \\ 1, \text{ jeśli pojazd } k \text{ jedzie od wierzchołka } i \text{ do wierzchołka } j. \end{matrix}$$

Zmienna $s_{i,k}$, zdefiniowana dla każdego wierzchołka i oraz każdego pojazdu k, oznacza czas, w którym pojazd k rozpoczyna obsługę klienta i. Jest to maksimum z czasu przybycia pojazdu k do klienta i oraz czasu otwarcia okna czasowego tego klienta. W przypadku gdy pojazd k nie obsługuje klienta i, zmienna $s_{i,k}$ nie ma określonej wartości. Rozwiązanie problemu polega na znalezieniu zbioru tras gwarantujących dostawę towarów do wszystkich klientów, które spełniają ograniczenia czasowe oraz ograniczenia na ładowność pojazdów. Ponadto rozmiar zbioru tras równy liczbie niezbędnych pojazdów powinien być minimalny (cel podstawowy), a także minimalna powinna być sumaryczna długość tras, jaką przebywają pojazdy realizując dostawę towarów (cel dodatkowy). Zakładając, że każda trasa zaczyna się w wierzchołku 0 i kończy w wierzchołku n + 1, problem trasowania pojazdów z ograniczeniami czasowymi można sformułować następująco:

(1)

$= 1 \qquad \forall i \in C$ (2)

$\leq q \qquad \forall k \in V$

(3)

$= 1 \qquad \forall k \in V$

(4)

$$= 0 \qquad \forall\, h \in C,\ \forall k \in V \qquad (5)$$

$$= 1 \qquad \forall k \in V \qquad (6)$$

$$\leq s_{j,k} \qquad \forall i, j \in N,\ \forall k \in V \qquad (7)$$

$$\forall i \in N,\ \forall k \in V \qquad (8)$$

$$\forall i, j \in N,\ \forall k \in V \qquad (9)$$

Warunek (2) mówi, że każdy klient może zostać obsłużony tylko raz, a warunek (3) oznacza, że żaden pojazd nie może przewozić ładunku większego niż wynosi jego ładowność. Równania (4), (5) i (6) zapewniają, że każdy z pojazdów wyjeżdża z magazynu 0, a następnie porusza się od klienta do klienta, kończąc swoją trasę w magazynie $n + 1$. Nierówność (7) mówi, że pojazd k jadąc od klienta i do klienta j nie może przyjechać do klienta j przed czasem $s_{i,k} + t_{i,j}$. K jest dużą stałą dodatnią, zapewniającą spełnienie warunku dla $x_{i,j,k} = 0$. Warunek (8) zapewnia przestrzeganie okien czasowych, a warunek (9) określa wartości zmiennej $x_{i,j,k}$.

3. Sekwencyjny algorytm symulowanego wyżarzania

Algorytm symulowanego wyżarzania został zaproponowany przez Metropolisa i in. [8] w 1953 r., a następnie użyty do rozwiązywania problemów optymalizacyjnych przez Kirkpatricka, Gellata i Vecchiego [6] oraz przez Cerny'ego [2]. Swoją nazwę zawdzięcza analogii do pewnych zjawisk fizycznych. W procesach chłodzenia (zestalania) cieczy zauważono, że powolne obniżanie temperatury pozwala atomom na znalezienie optymalnego miejsca w czasie krystalizowania się cieczy. Dzięki temu struktura zestalonego materiału jest silniejsza i stabilniejsza. Jeżeli spadek temperatury jest zbyt szybki, to cząsteczki rozłożą się chaotycznie, nie znajdując optymalnego położenia.

Algorytm symulowanego wyżarzania [1] jest wariantem lokalnego poszukiwania, w którym iteracyjnie ulepszamy istniejące rozwiązanie, stosując tzw. funkcję przejścia. Funkcja przejścia, zależna od rozwiązywanego problemu, znajduje rozwiązanie sąsiednie w stosunku do bieżącego przez dokonanie zmian w rozwiązaniu bieżącym. Jeżeli znalezione rozwiązanie jest lepsze, to poprawiamy istniejące rozwiązanie i kontynuujemy poszukiwania. Jeżeli żadne z sąsiednich rozwiązań nie przynosi poprawy istniejącego rozwiązania, to kończymy poszukiwania. Taka strategia poszukiwań z reguły prowadzi do znalezienia rozwiązania nieoptymalnego, stanowiącego jedynie minimum lokalne optymalizowanej funkcji. Brak możliwości wyjścia z tego minimum, tak by kontynuować poszukiwania w kierunki globalnego optimum, jest główną wadą metody lokalnego poszukiwania.

W metodzie symulowanego wyżarzania lokalne poszukiwanie jest realizowane przez próbkowanie w sposób losowy sąsiedztwa bieżącego rozwiązania. Aby uniknąć przedwczesnego zatrzymania poszukiwań w lokalnym optimum, akceptowane są od czasu do czasu gorsze rozwiązania. Prawdopodobieństwo przyjęcia gorszego rozwiązania jest równe:

, (10)

gdzie δ oznacza wzrost kosztu rozwiązania, a T_i, $i = 0, 1, ..., i_{max}$ jest parametrem algorytmu zwanym temperaturą wyżarzania. Temperatura ta jest obniżana od wartości początkowej T_0 zgodnie ze wzorem $T_{i+1} = rT_i$, gdzie r jest pewną stałą mniejszą niż jeden. Wzór (10) wskazuje, że im wyższa jest temperatura, tym prawdopodobieństwo zaakceptowania gorszego rozwiązania jest większe. Wraz z postępem procesu optymalizacyjnego temperatura spada, by w końcu zbliżyć się do zera, a wtedy większość rozwiązań o wyższym koszcie jest odrzucana. Obliczenia algorytmu symulowanego wyżarzania są zatrzymywane w momencie, gdy spełniony zostaje warunek stopu. Może on być zdefiniowany jako wykonanie określonej liczby epok schładzania, czyli obniżeń temperatury, lub jako osiągnięcie równowagi termicznej, w której podczas przyjętej liczby etapów schładzania najlepsze znalezione rozwiązanie problemu nie zmienia się.

4. Algorytm symulowanego wyżarzania dla problemu trasowania pojazdów z ograniczeniami czasowymi

Dla problemu trasowania pojazdów została zdefiniowana następująca funkcja kosztu:

$$f(s) = \alpha R + \beta D, \qquad \alpha \gg \beta,$$

gdzie R jest liczbą tras, zaś D łączną długością wszystkich tras. W badaniach eksperymentalnych przyjęto $\alpha = 40000$ oraz $\beta = 1$. W celu wygenerowania rozwiązania początkowego klienci sortowani są według długości okna czasowego i wstawiani do tras tak, by wzrost kosztu rozwiązania był jak najmniejszy. Jeżeli nie jest możliwe wstawienie klienta do żadnej z tras, wówczas zostaje on umieszczony na początku nowej trasy. W pracy wykorzystano funkcję przejścia skonstruowaną przez Czarnasa [3]. Pozwala ona na dokonywanie stosunkowo dużych zmian w rozwiązaniu bieżącym i opiera się na przemieszczeniu jednego bądź większej liczby klientów pomiędzy trasami tego rozwiązania. Zarówno przemieszczani klienci, jak i trasy są wybierane losowo. Temperatura początkowa T_0 ustalana jest w taki sposób, by w pierwszej iteracji symulowanego wyżarzania gorsze rozwiązanie zostało przyjęte z prawdopodobieństwem $p_0 = 0.01$, zaś współczynnik schładzania r równy jest 0.98. Długość epoki, czyli liczba iteracji symulowanego wyżarzania dla ustalonej temperatury T_i, jest wielokrotnością liczby klientów w rozwiązywanym problemie trasowania pojazdów i wynosi 100000.

5. Równoległe algorytmy symulowanego wyżarzania

Załóżmy, że możliwa jest równoległa realizacja p procesów symulowanego wyżarzania. Procesy te zostaną użyte do otrzymania rozwiązań o lepszej jakości, czyli rozwiązań bliższych rozwiązaniu optymalnemu. W pracy badano trzy równoległe algorytmy: niezależnie pracujących procesów (NP), procesów komunikujących się po każdej epoce wyżarzania (KP) i procesów często komunikujących się w czasie każdej epoki (CKP). Pierwszy algorytm (NP) polega na wykonaniu p niezależnych procesów symulowanego wyżarzania i wyznaczeniu najlepszego rozwiązania spośród rozwiązań uzyskanych przez poszczególne procesy. W pozostałych dwóch algorytmach (KP i CKP) procesy P_1, P_2, ..., P_p komunikują się ze sobą przekazując swoje najlepsze znalezione rozwiązania. Proces P_1 przekazuje w czasie komunikacji swoje najlepsze rozwiązanie procesowi P_2, ten porównuje otrzymane rozwiązanie ze swoim i kontynuuje obliczenia od rozwiązania lepszego. Następnie przesyła je procesowi P_3, który także porównuje je z dotychczas znalezionym przez siebie najlepszym rozwiązaniem i aktualizuje je, jeżeli jest ono gorsze od otrzymanego. I tak najlepsze rozwiązanie znalezione przez proces l jest propagowane do dalszego ulepszania przez procesy o numerach większych niż l. We wszystkich algorytmach (NP, KP, CKP) procesy rozpoczynają poszukiwania od tego samego rozwiązania początkowego.

6. Wyniki eksperymentów

Równoległe algorytmy symulowanego wyżarzania zostały zaimplementowane w języku C. Ich równoległa praca została zasymulowana w programie sekwencyjnym tak, by można było wykonać obliczenia na jednym procesorze (w ośrodkach obliczeniowych, w których wykonywano obliczenia, stosuje się ograniczenia liczby procesorów jednocześnie używanych przez użytkownika). Badania przeprowadzone zostały dla różnej liczby procesów (p = 5, 10, 15, 20) oraz danych testowych R101, R108, R112 i RC208 opracowanych przez Solomona [9]. Zbiór Solomona składa się z 56 zestawów danych testowych. Każdy zestaw zawiera współrzędne 100 klientów oraz magazynu w układzie kartezjańskim. Współrzędne te określone są liczbami całkowitymi z przedziału 0..100. Przyjęto, że czasy przejazdu $t_{i,j}$ są równe odpowiadającym im odległościom euklidesowym $d_{i,j}$ pomiędzy klientami i oraz j. W zbiorze danych testowych można wyróżnić testy sześciu typów. W zbiorach R1 (dane testowe R101..R112) oraz R2 (dane R201..R211) położenia klientów zostały wygenerowane losowo w obszarze 100 × 100 zgodnie z rozkładem równomiernym. Geograficzna dystrybucja klientów w zbiorach C1 (dane C101..C109) oraz C2 (dane C201..C208) ma charakter klastrowy, zaś w zbiorach RC1 (dane RC101..RC108) oraz RC2 (dane RC201..RC208) - „częściowo klastrowy", tj. z częścią klientów, których lokalizacje zostały określone losowo, oraz częścią klientów grupowanych w klastrach. Zbiory R1, C1 oraz RC1 mają wąskie okna czasowe, co pozwala na grupowanie mniejszej liczby klientów (pomiędzy 5 a 10) w ramach pojedynczych tras. Okna czasowe w zbiorach R2, C2 i RC2 są szersze, co umożliwia obsługę większej liczby klientów (więcej niż 30) w poszczególnych trasach.

Dla każdego zestawu danych testowych przeprowadzono 12000 eksperymentów (120 serii po 100 eksperymentów, czyli po 10 serii dla każdego algorytmu przy ustalonej liczbie procesów). Poza porównaniem średnich sumarycznych długości tras otrzymanych z użyciem algorytmów NP, KP, CKP dla liczby procesów p = 5, 10, 15, 20, badano również wartość odchylenia standardowego sumarycznej długości tras (s), liczbę trafień w uzyskane najlepsze rozwiązanie (H), liczbę rozwiązań gorszych co najwyżej o 0.1% od uzyskanego rozwiązania najlepszego (B) oraz otrzymane liczby różnych rozwiązań problemu (R). Wyniki zostały przedstawione w tabelach 1, 2, 3 i 4. Wartości D_T, s, H, B i R są średnimi z trzech serii eksperymentów, których średnie były najbliższe średniej z dziesięciu serii (odrzucamy siedem skrajnych serii z dziesięciu serii).

Tabela 1 zawiera wyniki uzyskane dla testu R101 (p – liczba procesów, L_T – liczba tras, D_T – sumaryczna długość tras, s – wartość odchylenia standardowego sumarycznej liczby tras, H – liczba trafień w najlepsze uzyskane rozwiązanie, B – liczba rozwiązań gorszych co najwyżej o 0.1% od rozwiązania najlepszego, R – liczba uzyskanych różnych rozwiązań). Ten zestaw danych okazał się trudniejszy do rozwiązania niż pozostałe i jedynie dla niego nie udało się uzyskać najlepszego znanego rozwiązania. Można zauważyć, że w algorytmie CKP wraz ze wzrostem liczby procesów maleje liczba różnych rozwiązań R i sumaryczna długość tras D_T. Oznacza to, że im więcej współpracujących procesów, tym poszukiwania koncentrują się na mniejszej liczbie lepszych rozwiązań. Dla algorytmu CKP uzyskano największą liczbę trafień w rozwiązanie najlepsze H oraz liczbę najlepszych rozwiązań B w stosunku do pozostałych algorytmów.

Tabela 2 zawiera wyniki uzyskane przez algorytmy NP, KP, CKP dla testu R108. Można zauważyć, że wartość D_T, czyli sumaryczna długość tras, jest najlepsza dla algorytmu KP. Oznacza to, że w przypadku tego testu częsta komunikacja, a tym samym większe ograniczenie przeszukiwanej przestrzeni rozwiązań w algorytmie CKP, wpływa niekorzystnie na jakość otrzymywanych rozwiązań. Dla algorytmu CKP największa jest jednak liczba trafień w rozwiązanie najlepsze H oraz liczba uzyskanych najlepszych rozwiązań B.

Tabela 1

Wyniki uzyskane przez algorytmy NP, KP i CKP dla testu R101

Alg.	p	L_T	D_T	s	H	B	R
NP	5	19	1671.0812	9.3884	0	0	300
	10	19	1666.3819	4.8637	0	0	300
	15	19	1659.0104	2.1003	0	0	300
	20	19	1656.7441	1.6672	0	6	300
KP	5	19	1671.5510	9.9942	0	0	300
	10	19	1664.6649	4.4547	0	0	300
	15	19	1656.7299	2.3244	0	9	300
	20	19	1654.4854	1.7435	0	43	297
CKP	5	19	1665.8676	9.0917	7	11	294
	10	19	1659.8588	8.1106	46	67	245
	15	19	1655.2577	6.0047	102	146	163
	20	19	1653.3532	4.0550	129	186	107

Tabela 2

Wyniki uzyskane przez algorytmy NP, KP i CKP dla testu R108

Alg.	p	L_T	D_T	s	H	B	R
NP	5	9	977.0570	7.0188	0	0	160
	10	9	974.6354	6.2511	0	0	150
	15	9	972.7801	7.5177	0	1	154
	20	9	972.2796	5.6866	0	0	157
KP	5	9	974.6365	6.5556	0	3	170
	10	9	972.2314	6.7362	0	3	148
	15	9	969.7821	6.0789	3	8	139
	20	9	969.3186	5.2936	3	7	132
CKP	5	9	977.6927	9.8338	3	3	145
	10	9	974.2366	8.1952	3	6	132
	15	9	971.8504	7.6760	5	9	117
	20	9	971.7676	7.5525	6	9	123

Tabela 3

Wyniki uzyskane przez algorytmy NP, KP i CKP
dla testu R112

Alg.	p	L_T	D_T	s	H	B	R
NP	5	9	1006.5857	15.1602	0	0	14
	10	9	1006.0618	12.4916	0	0	17
	15	9	1005.2574	13.2834	0	0	15
	20	9	1004. 5593	9.6086	0	0	15
KP	5	9	1005.6822	11.5951	0	0	13
	10	9	1002.4252	10.3340	0	0	15
	15	9	1004.1200	14.4451	0	0	13
	20	9	1001.6683	9.4684	0	0	17
CKP	5	9	1010.9892	9.5242	0	0	16
	10	9	1010.2633	14.4649	0	0	14
	15	9	1004.7814	15.8403	0	0	18
	20	9	1006.5007	12.0616	0	0	15

Tabela 4

Wyniki uzyskane przez algorytmy NP, KP i CKP
dla testu RC208

Alg.	p	L_T	D_T	s	H	B	R
NP	5	3	830.5892	2.9074	1	97	65
	10	3	829.3078	1.3060	0	159	36
	15	3	828.9557	0.6650	3	209	29
	20	3	828.8232	0.2951	3	241	23
KP	5	3	830.4425	3.1175	4	101	46
	10	3	829.2165	1.3120	3	151	23
	15	3	829.0624	1.0027	9	159	18
	20	3	828.9081	0.5328	13	170	15
CKP	5	3	830.6323	3.2200	3	107	49
	10	3	829.4776	1.5329	4	128	27
	15	3	829.1692	1.0279	6	139	22
	20	3	829.1701	1.0156	5	151	20

W tabeli 3 pokazano wyniki uzyskane dla testu R112. To drugi najtrudniejszy zestaw danych spośród testów, które zostały przebadane. Najlepsze znane rozwiązanie zostało znalezione jedynie trzy razy w trakcie 12000 eksperymentów. Podobnie jak dla testu R108 całkowita długość tras D_T jest najlepsza dla algorytmu KP.

Tabela 4 zawiera wyniki uzyskane dla testu RC208, który okazał się najłatwiejszy do rozwiązania. Można zauważyć, że we wszystkich algorytmach wraz ze wzrostem liczby procesów maleje wartość D_T, czyli znajdowane są lepsze rozwiązania. Liczba trafień w najlepsze rozwiązanie H jest największa dla algorytmu KP, ale algorytm NP jest lepszy od pozostałych pod względem liczby uzyskanych najlepszych rozwiązań B.

Na podstawie średnich sumarycznych długości tras D_T oraz wartości odchylenia standardowego tych długości s z tabel 1, 2, 3 i 4 zweryfikowano następujące hipotezy statystyczne [7]: $H_0 : \mu_{NP} \leq \mu_{KP}$ i $H_0^{/} : \mu_{CKP} \leq \mu_{KP}$ wobec hipotez alternatywnych $H_a : \mu_{NP} > \mu_{KP}$ i $H_a^{/} : \mu_{CKP} > \mu_{KP}$ (μ oznacza średnią wartość populacji długości tras). Stosując test oparty na następującej statystyce:

i , (11)

odrzucone zostają hipotezy H_0 i $H_0^{/}$ na poziomie istotności $\alpha = 0.01$, jeżeli $z > z_{0.01} = 2.33$, co oznacza, że algorytmy, odpowiednio, NP i CKP dają gorsze rozwiązania w porównaniu z algorytmem KP.

Tabela 5

Wartości statystyki Z dla porównania algorytmów NP i KP ($n_1 = n_2 = 300$)

	Zestaw danych testowych			
p	R101	R108	R112	RC208
5	-0.59	4.37	0.82	0.60
10	4.51	4.53	3.89	0.85
15	12.61	5.37	1.00	-1.54
20	16.22	6.60	3.71	-2.41

Tabela 6

Wartości statystyki $Z^{/}$ dla porównania algorytmów CKP i KP ($n_1 = n_2 = 300$)

	Zestaw danych testowych			
p	R101	R108	R112	RC208
5	-7.29	4.48	6.13	0.73
10	-9.00	3.27	7.64	2.24
15	-3.96	3.66	0.53	1.29
20	-4.44	4.60	5.46	3.96

Wartości statystyk dla danych testowych R101, R108, R112 oraz RC208 przedstawiono w tabelach 5 i 6. Przypadki, w których hipotezy H_0 i $H_0^/$ zostały odrzucone, zaznaczono szarym kolorem. Można zauważyć, że algorytm KP daje lepsze rozwiązania w porównaniu z algorytmem NP dla danych testowych R101, R108 i częściowo dla R112 (tabela 5). W porównaniu z algorytmem CKP algorytm KP jest lepszy dla danych R108 oraz R112 (tabela 6). Tabele 7 i 8 zawierają wartości statystyk dla hipotez H_0 : $\mu_{NP} \le \mu_{CKP}$ i $H_0^/$: $\mu_{KP} \le \mu_{CKP}$ wobec hipotez alternatywnych H_a : $\mu_{NP} > \mu_{CKP}$ i $H_a^/$: $\mu_{KP} > \mu_{CKP}$. Jak widać, algorytm CKP daje lepsze rozwiązania, porównując z algorytmem NP i KP, dla testu R101.

Tabela 7

Wartości statystyki Z dla porównania algorytmów NP i CKP
($n_1 = n_2 = 300$)

p	Zestaw danych testowych			
	R101	R108	R112	RC208
5	6.91	-0.91	-4.26	-0.17
10	11.95	0.67	-3.81	-1.46
15	10.22	1.50	0.40	-3.02
20	13.40	0.94	-2.18	-5.68

Tabela 8

Wartości statystyki $Z^/$ dla porównania algorytmów KP i CKP
($n_1 = n_2 = 300$)

p	Zestaw danych testowych			
	R101	R108	R112	RC208
5	7.29	-4.48	-6.13	-0.73
10	9.00	-3.27	-7.64	-2.24
15	3.96	-3.66	-0.53	-1.29
20	4.44	-4.60	-5.46	-3.96

7. Podsumowanie

W niniejszej pracy przedstawiono trzy równoległe algorytmy symulowanego wyżarzania dla problemu trasowania pojazdów z ograniczeniami czasowymi. Zbadano eksperymentalnie wpływ komunikacji równoległych procesów na jakość otrzymanych rozwiązań. Wyniki eksperymentów dla wybranych danych testowych Solomona pokazują, że współpraca procesów w równoległych algorytmach symulowanego wyżarzania przyczynia się do uzyskania lepszych rozwiązań w porównaniu z procesami realizującymi poszukiwania niezależnie.

Przeprowadzone badania nie dają jednoznacznej odpowiedzi na pytanie, jak często procesy powinny komunikować się w czasie symulowanego wyżarzania. Najlepsze rozwiązania uzyskano dla algorytmu KP, w którym procesy komunikują się po każdej epoce wyżarzania (wyniki eksperymentów dla testów R101, R108 i R112). Dla danych testowych R101 korzystna okazała się częsta komunikacja w czasie każdej epoki. Konieczne jest więc przeprowadzenie dalszych badań eksperymentalnych, które dadzą odpowiedź na to pytanie.

8. Podziękowania

Dziękujemy Piotrowi Czarnasowi i Przemysławowi Gocyle za ich wkład w niniejszą pracę. Dziękujemy również następującym ośrodkom obliczeniowym, w których wykonano badania eksperymentalne: Interdyscyplinarnemu Centrum Modelowania Matematycznego i Komputerowego Uniwersytetu Warszawskiego (grant G-27-9), Centrum Informatycznemu TASK w Gdańsku, Akademickiemu Centrum Komputerowemu CYFRONET AGH w Krakowie (granty 027 i 069/2004), Wrocławskiemu Centrum Sieciowo-Superkomputerowemu oraz Poznańskiemu Centrum Superkomputerowo-Sieciowemu.

LITERATURA

1. Aarts E. H. L., and van Laarhoven P. J. M.: Simulated annealing: Theory and applications. Wiley, New York, 1987.
2. Cerny V.: A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. Journ. of Optimization Theory and Applic. 45, (1985), 41-55.
3. Czarnas P.: Algorytm symulowanego wyżarzania. Praca magisterska, Wrocław (2001).
4. Czech Z. J. , Czarnas P., Gocy³a P.: Parallel simulated annealing for bicriterion optimization problems. Proc. of the 5th International Conference on Parallel Processing and Applied Mathematics (PPAM'03), (2003), Częstochowa, Poland, 233-240.
5. Czech Z. J., Wieczorek B.: Parallel simulated annealing algorithms. Złożone do publikacji.
6. Kirkpatrick S., Gellat C. D., Vecchi M. P.: Optimization by simulated annealing. Science 220, (1983), 671-680.
7. Maliński M.: Weryfikacja hipotez statystycznych wspomagana komputerowo. Wydawnictwo Politechniki Śląskiej, Gliwice (2004).
8. Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Teller E.: Equation of state calculation by fast computing machines. Journ. of Chem. Phys. 21, (1953), 1087-1091.

9. Solomon M. M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35, (1987), 254-265.
10. Tan K. C., Lee L. H., Zhu Q. L., Ou K.: Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering 15, (2001), 281-295.

Recenzent: Dr inż. Mariusz Boryczka

Wpłynęło do Redakcji 28 lutego 2005 r.

Abstract

The parallel simulated annealing algorithms, i.e. the independent, communicating and frequently communicating processes, are investigated. The algorithms solve the vehicle routing problem with time windows (VRPTW) in which both the number of vehicles and the total distance traveled by the vehicles should be minimized. The goal is to improve the quality of solutions to the problem through co-operation of parallel processes. The quality of solutions is measured by their proximity to the optimum solution. The parallel simulated annealing algorithms were serialized and implemented using C language. The computational experiments for p = 5, 10, 15 and 20 parallel processes were carried out on the R101, R108, R112 and RC208 test instances of the test set by Solomon (results in table 1, 2, 3, and 4). The empirical evidence supported by the statistical analysis (table 5, 6, 7 and 8) indicate that co-operation of processes in parallel simulated annealing yields solutions to the VRPTW of better quality as compared to the case when the processes run independently.

Adres

Bożena WIECZOREK: Górnośląska Wyższa Szkoła Handlowa, Katedra Informatyki, ul. Harcerzy Września 3, 40-659 Katowice, Polska, Bożena.Wieczorek@polsl.pl.