Polskie Towarzystwo Badań Operacyjnych i Systemowych
Instytut Badań Systemowych Polskiej Akademii Nauk
Wojskowa Akademia Techniczna

Redaktorzy:
Zbigniew Nahorski
Marian Chudy
Andrzej Straszak

OPTYMALIZACJA

Warszawa 1991

POLSKIE TOWARZYSTWO
BADAŃ OPERACYJNYCH I SYSTEMOWYCH

INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

WOJSKOWA AKADEMIA TECHNICZNA

OPTYMALIZACJA

ZADANIA, METODY, ALGORYTMY

Redaktorzy

Zbigniew Nahorski, Marian Chudy, Andrzej Straszak

WARSZAWA 1991

OPTYMALIZACJA:
Zadania, metody, algorytmy
PTBOiS, Warszawa 1991.

XTSP3 - pakiet edukacyjno-badawczy do rozwiązywania symetrycznego zadania komiwojażera

S. Berka, S. Kryński, M. Libura
Instytut Badań Systemowych PAN
Warszawa

Streszczenie

Pakiet XTSP3 jest przeznaczony do znajdowania przybliżonych rozwiązań symetrycznego zadania komiwojażera o rozmiarze do 800 punktów, za pomocą komputerów klasy IBM PC XT/AT. Zawiera bibliotekę kilkunastu algorytmów, które można zestawiać ze sobą tworząc złożone heurystyki. Pakiet wyposażony jest ponadto w procedury obliczające dolne oszacowania wartości optymalnej. W przypadku zadań definiowanych na płaszczyźnie możliwe jest śledzenie na ekranie procesu tworzenia i poprawiania rozwiązań.

W niniejszym artykule przedstawiono ogólny opis pakietu i jego najważniejszych właściwości. Ponadto omówiono wyniki eksperymentów obliczeniowych przeprowadzonych za jego pomocą.

1. Zadanie komiwojażera

Zadanie komiwojażera jest klasycznym problemem optymalizacji kombinatorycznej mającym bardzo bogatą literaturę i liczne zastosowania (patrz np. [7]). Jest definiowane w sposób następujący. Rozważany jest graf nieskierowany $G=(V,E)$. Liczba $n=|V|$ jest rozmiarem zadania. Dla każdej krawędzi $e=(i,j)\in E$ zadana jest jej długość (waga) $c(i,j)\in R$. Parom wierzchołków $i,j\in V$ nie połączonych krawędzią przyporządkowuje się wagę $c(i,j)=\infty$. Dane zadania są w pełni reprezentowane przez macierz $n\times n$-wymiarową $C=(c(i,j))$. Drogę zamkniętą przechodzącą przez każdy wierzchołek $i\in V$ dokładnie jeden raz nazywamy *obwodem Hamiltona*. Symetryczne zadanie komiwojażera polega na znalezieniu najkrótszego obwodu Hamiltona (drogi komiwojażera).

Szczególnym przypadkiem symetrycznego zadania komiwojażera jest *euklidesowe zadanie komiwojażera*, w którym wierzchołkom grafu G przyporządkowane są punkty przestrzeni R^m, natomiast długości krawędzi są odległościami euklidesowymi między tymi punktami. Dla m=2 mamy do czynienia z *euklidesowym zadaniem komiwojażera na płaszczyźnie*, a danymi zadania są współrzędne *n* punktów płaszczyzny.

Nawet w tym ostatnim przypadku zadanie komiwojażera jest problemem NP-trudnym (patrz [7]). Ogranicza to obecne praktyczne możliwości d o k ł a d n e g o rozwiązania tego zadania do kilkuset punktów w przypadku użycia bardzo szybkich komputerów i odpowiednio mniej w przypadku mikrokomputerów. Istnieje natomiast możliwość uzyskania dobrych rozwiązań p r z y b l i ż o n y c h nawet dla problemów znacznie większych.

W trakcie badań nad problemem komiwojażera zaproponowano znaczną liczbę różnorodnych algorytmów heurystycznych. Kilka z nich zaimplementowano w pakiecie XTSP3.

2. Ogólny opis pakietu

Przy pomocy pakietu XTSP3 można rozwiązywać zadanie komiwojażera w grafie nieskierowanym określone za pomocą współrzędnych punktów na płaszczyźnie lub macierzy odległości między nimi. Dane zadania można wprowadzać ze zbioru lub generować losowo za pomocą generatora zadań wbudowanego w pakiet.

Biblioteka heurystyk pakietu XTSP3 zawiera kilkanaście procedur tworzących suboptymalne rozwiązanie zadania komiwojażera (algorytmy najbliższego sąsiada, najdalszego wstawiania, oszczędności i inne) oraz heurystyki poprawy (algorytmy 2-Opt, 3-Opt, Or-Opt, Lina-Kernighana). Pakiet umożliwia łatwe łączenie dowolnej heurystyki z pierwszej grupy z kilkoma heurystykami poprawy. Taki ciąg heurystyk nazywany jest strategią.

Przy pomocy pakietu możliwe jest rozwiązywanie zadań w zasadzie o dowolnym rozmiarze chociaż pojemność pamięci mikrokomputera oraz praktycznie dopuszczalny czas rozwiązywania ograniczają go do ok. jednego tysiąca punktów. Aktualna wersja pakietu przeznaczona jest dla zadań o rozmiarze do 800 punktów.

Możliwe jest też obliczanie dolnego oszacowania wartości optymalnej zadania. Dostępne są trzy poziomy dokładności oszacowania.

W trakcie obliczeń na ekranie wypisywane są wyniki takie jak długość drogi i ostatnie znalezione skrócenie drogi oraz rysowana jest aktualna droga (dla zadań euklidesowych). Możliwe jest też wyświetlenie tabeli zawierającej wszystkie uzyskane do danego momentu wyniki, a także narysowanie najlepszej drogi. Wybrane opcje i strategie mogą być modyfikowane w razie potrzeby w trakcie obliczeń.

Pakiet może działać w trybie szybkim (tj. bez zatrzymywania się) lub w trybie krokowym, w którym zatrzymuje się po każdym kroku, tzn. na przykład przed rozpoczęciem nowej strategii, po znalezieniu poprawy drogi, po zakończeniu heurystyki itp. Trzeci tryb pracy, "wsadowy", służy do przeprowadzania eksperymentów. W trybie tym pakiet nie wymaga ingerencji użytkownika w trakcie obliczeń. Umożliwia to m.in. inicjowanie rozwiązywania wielu zadań lub serii zadań z jednego zbioru komend systemowych.

Pakiet został napisany w języku Turbo Pascal 5.5 z wykorzystaniem biblioteki graficznej Turbo Graphics Toolbox. Wykorzystuje bibliotekę obsługi menu ekranowych MENU.PAS opisaną w [2]. Jest przeznaczony na mikrokomputery zgodne z IBM PC XT/AT.

3. Strategie rozwiązywania zadania w pakiecie XTSP3

3.1. Heurystyki

Pakiet zawiera dwie grupy algorytmów do rozwiązywania symetrycznego zadania komiwojażera: (i) algorytmy tworzące drogę startową nazywane *preprocesorami* oraz (ii) algorytmy poprawiające drogę startową (*poprocesory*).

W obecnej wersji pakiet zawiera nastepujace preprocesory:

1) Algorytm RND (Random tour)

Algorytm ten generuje drogę losową tworzoną przy użyciu generatora liczb pseudolosowych. Algorytm jest bardzo szybki, ale generowana droga jest kilkakrotnie dłuższa od drogi optymalnej.

2) Algorytm NEARN (Nearest Neighbour)

Algorytm ten tworzy drogę startową zaczynając od dowolnego punktu i przedłużając drogę częściową przez dołączenie najkrótszej krawędzi wychodzącej z ostatnio dołączonego punktu i nie tworzącej cyklu o liczbie krawędzi mniejszej niż n. Jest to również szybka heurystyka, której implementacja wymaga $O(n^2)$ operacji. Wiadomo (patrz [9]), że jeśli długości krawędzi grafu spełniają warunek trójkąta, to droga utworzona przez tę heurystykę nie jest dłuższa niż $0.5\lceil \log n\rceil+1$ długości drogi optymalnej.

3) Algorytm SAVNG (Savings)

Jest to heurystyka porządkująca krawędzie grafu według tzw. zasady oszczędności [4]. Jej implementacja wymaga $O(n^2 \log n)$ operacji. Daje dobre drogi startowe. Wiadomo [5], że przy spełnionym warunku trójkąta długość drogi utworzonej przez tę heurystykę nie może być większa niż $\lceil \log n\rceil+1$ długości drogi optymalnej.

4) Algorytmy: DFI (Diameter + Furthest Insertion)
CHFI (Convex Hull + Furthest Insertion)

Algorytmy te wykorzystują ideę tzw. wstawiania najdalszego punktu. W algorytmie DFI budowa drogi rozpoczyna się od pary najbardziej odległych punktów, natomiast w algorytmie CHFI od drogi częściowej zawierającej wierzchołki powłoki wypukłej punktów definiujących zadanie. Obie heurystyki są stosunkowo szybkie. Ich implementacja wymaga $O(n^2)$ operacji i wiadomo, że przy spełnionym warunku trójkąta długość drogi wyznaczonej przez algorytm DFI jest nie większa niż 2 długości drogi optymalnej (patrz [9]).

5) Algorytmy: SFC (Space Filling Curve)
SFCLS (Space Filling Curve + Local Search)

Heurystyki te wykorzystują konstrukcję tzw. krzywej wypełniającej kwadrat [1]. W przypadku algorytmu SFCLS droga utworzona przez algorytm SFC jest poprawiana metodą lokalnego przeglądu (patrz [6]). Wiadomo, że w najgorszym przypadku długość drogi utworzonej przez algorytm SFC jest nie większa niż $O(\log n)$ długości drogi optymalnej. Implementacja algorytmu SFC wymaga $O(n\log n)$ operacji.

Jedną z możliwości utworzenia startowej drogi komiwojażera jest wprowadzenie jej ze zbioru przygotowanego przez użytkownika. Umożliwia to m.in. poprawianie znanych rozwiązań dla danego zadania.

Podane wyżej oceny jakości preprocesorów dotyczą najgorszego przypadku i są bardzo pesymistyczne. Średnie błędy rozwiązań są znacznie mniejsze, chociaż dowodzi się, że istnieją szczególne klasy zadań, dla których te oszacowania są ścisłe.

Obwody Hamiltona generowane przez preprocesory mogą być traktowane jako zgrubne rozwiązania zadania komiwojażera. W zależności od użytego preprocesora błędy rozwiązań mogą się wahać od kilku do kilkudziesięciu procent. Rozwiązania te mogą być znacznie ulepszone przez zastosowanie jednego lub kilku algorytmów poprawy rozwiązania (poprocesorów).

W obecnej wersji w pakiecie zaimplementowane są cztery algorytmy poprawy rozwiązania startowego. Są to:

1) Algorytm 2-OPT

Jest to klasyczny algorytm poprawy drogi komiwojażera poprzez wymianę par krawędzi. Jest stosunkowo szybki i w przypadku euklidesowego zadania komiwojażera na płaszczyźnie daje drogi bez skrzyżowań krawędzi.

2) Algorytm 3-OPT

Algorytm ten poprawia istniejącą drogę przez wymianę trójek krawędzi. Daje bardzo dobre drogi, ale wymaga znacznego nakładu obliczeń. Bardzo istotne przyspieszenie obliczeń bez zauważalnego pogorszenia jakości rozwiązań uzyskuje się w wyniku opisanego niżej rozrzedzania grafu.

3) Algorytm OROPT

Algorytm ten jest ograniczoną wersją algorytmu 3-OPT dającą dobre rozwiązania w znacznie krótszym niż w przypadku 3-OPT czasie obliczeń.

4) Algorytm LK (Lina i Kernighana)

Algorytm ten jest implementacją heurystyki Lina i Kernighana (patrz [8]). Polega on również na wymianie podzbiorów krawędzi, przy czym liczba wymienianych krawędzi nie jest ustalona z góry. Daje zwykle bardzo dobre rozwiązania w stosunkowo niedługim czasie.

3.2. Sąsiedztwa

Niemal wszystkie opisane wyżej algorytmy mają czas obliczeń istotnie zależny od liczby krawędzi grafu. Z drugiej strony, istnieje stosunkowo niewiele krawędzi, które występują w kilku kolejnych najlepszych rozwiązaniach zadania. Obserwacja ta (patrz np. [10]) leży u podstaw idei rozrzedzania grafu definiującego zadanie komiwojażera. Daje to bardzo istotne zmniejszenie czasu działania algorytmu przy niezauważalnym zwykle pogorszeniu jakości rozwiązania. Wszystkie poprocesory zaimplementowane w pakiecie XTSP3 mogą wykorzystywać rozrzedzenie grafu. O sposobie rozrzedzania decyduje użytkownik wybierając odpowiednią pozycję w menu.

Zastosowano dwie wersje rozrzedzania poprzez definiowanie tzw. *sąsiedztw* wierzchołków grafu. Poprzez sąsiedztwo dla danego wierzchołka jest rozumiany podzbiór krawędzi incydentnych z tym wierzchołkiem i znajdujących się w grafie rozrzedzonym. Pierwsza metoda definiowania sąsiedztw polega na pozostawieniu w grafie rozrzedzonym zadanej przez użytkownika liczby k najkrótszych krawędzi incydentnych z danym wierzchołkiem. Druga metoda polega na wybraniu zadanej przez użytkownika liczby k kolejnych najkrótszych dendrytów (drzew rozpinających), czyli podgrafów spójnych grafu G nie zawierających obwodów. Z badań empirycznych wynika, że najbardziej odpowiednim zakresem dla wyboru parametru k jest przedział [10,20] w pierwszej metodzie oraz [6,12] w drugiej.

3.3. Liczenie dolnych oszacowań

Wyznaczanie dolnego oszacowania długości najkrótszej drogi komiwojażera jest potrzebne do oceny dokładności znalezionych przybliżonych rozwiązań. W pakiecie XTSP3 wykorzystano algorytm znajdowania oszacowania dolnego polegający na rozwiązywaniu zadania dualnego do zadania komiwojażera, opartego na konstrukcji dendrytów uzupełnionych. Przez *dendryt uzupełniony* grafu G jest rozumiany taki podgraf, który ma n krawędzi i zawiera dokładnie jeden obwód. Najkrótszy dendryt uzupełniony w G może być wyznaczony przez nieznacznie zmodyfikowany algorytm znajdowania dendrytu w grafie. Przy rozwiązywaniu zadania dualnego wykorzystuje się fakt, że jego funkcja celu jest kawałkami liniową wklęsłą funkcją na R^n. W pakiecie do jej maksymalizacji zastosowano algorytm subgradientowy. Szczegółowy opis tego podejścia zawiera praca [3].

4. Eksperyment obliczeniowy

4.1. Opis eksperymentu

Jednym z głównych celów podjęcia prac nad pakietem było stworzenie narzędzia do testowania i porównywania przybliżonych algorytmów dla zadania komiwojażera. Poniżej przedstawiamy wyniki pierwszego z zaplanowanych eksperymentów. Poza wynikającymi z nich informacjami dotyczącymi użytych heurystyk ilustrują one również możliwości i praktyczne własności pakietu XTSP3.

Rozwiązywano generowane losowo zadania komiwojażera dwóch zasadniczych typów uwzględnianych w pakiecie: (a) zadania euklidesowe o punktach losowanych z rozkładem równomiernym w kwadracie $[0,10000]\times[0,10000]$ oraz (b) zadania definiowane za pomocą symetrycznej macierzy odległości o elementach wybieranych losowo, równomiernie w przedziale $[0,10000]$. Dla obu typów wygenerowano 4 zestawy po 10 zadań w każdym, różniące się rozmiarem zadań: ustalono rozmiary n = 50, 100, 200, 400. Otrzymane zestawy zadań nazwano SQn, DIn, odpowiednio dla typów (a) i (b), gdzie n przyjmuje podane wyżej wartości.

Do obliczeń wybrano 12 strategii. Pierwszych 6 strategii ma następującą postać (zauważmy, że wszystkie zaczynają się tym samym preprocesorem DFI):

St1:	DFI + OPT2 + OPT3 + LK
St2:	DFI + OPT2 + OROPT + LK + OPT3
St3:	DFI + OPT2 + LK + OPT3
St4:	DFI + OPT3 + LK
St5:	DFI + OROPT + LK + OPT3
St6:	DFI + LK + OPT3

Pozostałe strategie St7 , ... , St12 różnią się od powyższych jedynie preprocesorem: zamiast DFI zastosowano NEARN.

Dokładniej, poprocesory oraz ich kolejność w strategii St*j* są identyczne jak w strategii St[*j-6*], dla *j*=7,...,12.

Wszystkie zestawy zadań rozwiązywano zdefiniowanymi strategiami wielokrotnie, dla różnych sposobów rozrzedzania grafu. Uwzględniono obie metody rozrzedzania opisane w punkcie 3.2, przy czym każdą stosowano z parametrem *k* przyjmującym wartości *k* = 4, 6, 8, 10, 12, 16, 20.

Dla wszystkich zadań obliczono ponadto dolne oszacowania wartości optymalnej.

Poniżej przedstawiamy w tabelach uśrednione wyniki obliczeń uzyskane dla zestawu zadań SQ200 rozwiązywanego strategiami St1 , ... , St6. W kolumnach ukazane są wyniki otrzymane dla zastosowanych rozrzedzeń: S*k* odpowiada rozrzedzeniu określonemu za pomocą *k* kolejnych najkrótszych dendrytów, natomiast N*k* - rozrzedzeniu określonemu przez *k* najbliższych sąsiadów dla każdego wierzchołka grafu (*k* = 4, 6, 8, 10, 12, 16, 20).

W pierwszej tabeli zamieszczamy średnie czasy τ (w sekundach) rozwiązywania zadań z zestawu SQ200 za pomocą strategii St1 ,..., St6 przy użyciu rozrzedzeń typu S*k*. W drugiej tabeli podajemy średnie procentowe odstępy Δ pomiędzy uzyskanymi przez dane strategie, przy tych samych rozrzedzeniach, wartościami przybliżonych rozwiązań, a dolnym oszacowaniem wartości optymalnej obliczonym przez pakiet. Dokładniej, uśrednione są liczby postaci

$$\Delta_i = \frac{L_i - B_i}{B_i} \cdot 100$$

gdzie L_i jest wartością rozwiązania otrzymaną przez daną strategię przy danym rozrzedzeniu, B_i jest oszacowaniem od dołu wartości optymalnej, a i jest numerem zadania w zestawie. Zauważmy, że liczba Δ_i jest oszacowaniem od góry względnego błędu wartości L_i rozwiązania uzyskanego daną strategią.

Pozostałe dwie tabele zawierają analogiczne wyniki (tj. uśrednione czasy w sekundach i procentowe odstępy) dla tego samego zestawu zadań i tych samych strategii, otrzymane przy zastosowaniu rozrzedzeń typu N*k*.

τ	S4	S6	S8	S10	S12	S16	S20
St1	13.3	18.4	25.0	36.1	42.2	55.3	61.1
St2	12.0	15.9	20.5	25.1	31.6	35.9	37.5
St3	11.5	14.8	19.0	23.8	31.6	37.2	38.0
St4	12.1	18.8	29.2	35.4	47.1	57.0	65.2
St5	11.1	14.5	17.4	22.2	28.0	31.7	33.4
St6	10.5	12.9	16.7	20.1	26.3	33.6	33.0

Tabela 1. Średni czas τ [s] dla rozrzedzeń typu *Sk*

Δ	S4	S6	S8	S10	S12	S16	S20
St1	4.04	3.77	3.68	3.67	3.82	3.71	3.76
St2	3.98	3.82	3.87	3.93	3.86	3.85	3.88
St3	4.14	3.97	3.96	3.92	3.92	3.92	3.92
St4	3.72	3.62	3.59	3.75	3.64	3.57	3.52
St5	3.86	3.74	3.82	3.86	3.83	3.81	3.83
St6	4.11	3.96	3.77	3.80	3.80	3.80	3.80

Tabela 2. Średni odstęp Δ [%] dla rozrzedzeń typu *Sk*

τ	N4	N6	N8	N10	N12	N16	N20
St1	9.4	11.0	13.4	16.2	19.5	24.1	38.6
St2	9.6	10.7	12.4	13.9	15.5	19.2	25.3
St3	9.4	10.9	11.5	13.2	14.9	20.0	25.5
St4	9.4	10.8	13.6	16.0	18.5	24.6	37.3
St5	9.1	10.0	11.6	12.5	14.3	17.3	22.2
St6	8.8	10.0	10.7	11.9	13.3	16.9	19.7

Tabela 3. Średni czas τ [s] dla rozrzedzeń typu *Nk*

Δ	N4	N6	N8	N10	N12	N16	N20
St1	4.96	4.02	3.97	3.85	4.02	3.83	3.67
St2	4.90	4.25	4.07	3.86	3.85	3.79	3.88
St3	4.87	3.90	4.10	4.05	4.00	3.92	3.99
St4	4.67	3.93	4.22	3.51	3.88	3.64	3.39
St5	4.73	4.09	3.95	3.84	3.87	3.74	3.86
St6	4.64	3.89	3.93	4.02	3.88	3.80	3.80

Tabela 4. Średni odstęp Δ [%] dla rozrzedzeń typu *Nk*

4.2. Wnioski

Na podstawie wyników otrzymanych w opisywanym eksperymencie, a których część ukazano w tabelach, można sformułować szereg wniosków. Zebraliśmy je w następujących punktach. (Weryfikacja wniosków (5), (7), (8) wymaga wglądu do pełnych, nie zamieszczonych tutaj wyników).

(1) Nie obserwuje się wyraźnego związku odstępu Δ z mocą sąsiedztw (zależną liniowo od parametru k) dla S4 ÷ S20 oraz N6 ÷ N20. Jedynie dla N4 występują większe błędy.

(2) Odstępy Δ otrzymywane przy różnych typach sąsiedztw (S lub N) nie różnią się między sobą w istotny sposób.

(3) Żadna z rozpatrywanych strategii nie dominuje nad pozostałymi dokładnością znajdowanych rozwiązań.

(4) Czasy obliczeń dla strategii St1 i St4 są wyraźnie dłuższe niż dla pozostałych. Jest to spowodowane dużym nakładem obliczeń algorytmu OPT3, jeśli umieszczony on jest przed algorytmem LK.

(5) Dla zadań typu SQ*n* preprocesor NEARN jest istotnie lepszy od DFI zarówno pod względem odstępów Δ, jak i średnich czasów obliczeń τ (rejestrowanych dla pełnych strategii). Dla zadań typu DI*n* oba preprocesory dają zbliżone wyniki.

(6) Zależność średniego czasu τ od mocy sąsiedztw (dokładniej: od parametru k) może być estymowana jako zależność liniowa. Czasy uzyskane dla rozrzedzeń typów S*k* oraz N*l* są dla $l=2k$ zbliżone.

(7) Średni czas obliczeń τ dla wszystkich strategii rośnie nieco wolniej niż kwadrat rozmiaru zadania n.

(8) Błąd rozwiązania szacowany za pomocą odstępu Δ jest istotnie większy dla zadań typu DI*n* niż SQ*n*.

Literatura

[1] Bartholdi J.J., Platzman L.K.: An O(NlogN) planar travelling salesman heuristic based on spacefilling curves. Oper. Res. Lett. 1 (1982) 121-125.

[2] Berka S.: MENU.PAS 2.0 Biblioteka procedur obsługi menu ekranowych dla IBM PC w języku Turbo Pascal 5.5 - opis użytkowy. Raport ZPM-2/A1529/90 IBS PAN, Warszawa, maj 1990.

[3] Berka S., Libura M.: Oszacowania dolne długości najkrótszego obwodu Hamiltona w grafie nieskierowanym. Raport ZPM-42/A1529/88, Warszawa, wrzesień 1988.

[4] Clarke G., Wright J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. 12 (1964) 568-581.

[5] Frieze A.M.: Worst case analysis of algorithms for travelling salesman problems. Operations Research Verfahren 32 (1978) 94-112.

[6] Kryński S., Libura M.: Algorytmy przybliżone dla zadania komiwojażera na płaszczyźnie oparte na konstrukcji krzywej wypełniającej kwadrat. Zesz. Nauk. Politechniki Śląskiej (Automatyka) z. 94 (1988) 177-187.

[7] Lawler E.L. et al.: The Traveling Salesman Problem. Wiley, 1985.

[8] Lin S., Kernighan B.W.: An effective heuristic algorithm for the traveling-salesman problem. Oper. Res. 21 (1973) 498-516.

[9] Rosenkrantz D., Stearns R., Lewis P.: An analysis of several heuristics for the traveling salesman problem. SIAM Journal of Computing 6 (1977) 563-581.

[10] Stewart W.R., Jr.: Accelerated branch exchange heuristics for symmetric traveling salesman problem. Networks, 17 (1987) 423-437.

ISBN 83-900412-1-9