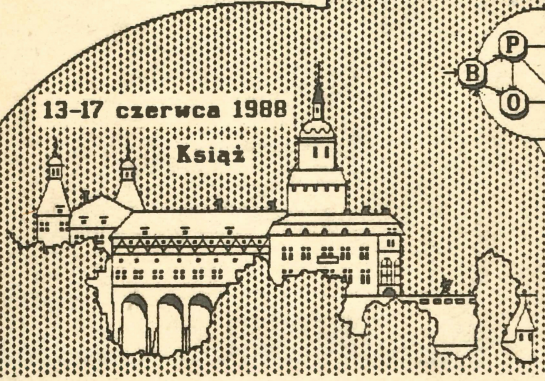
Redaktorzy:
A.Straszak
Z.Nahorski
J.Sikorski




# 1. Krajowa Konferencja Badań Operacyjnych i Systemowych

Tom 1

BOS'88

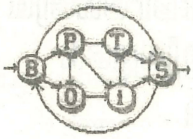
POLSKIE TOWARZYSTWO BADAŃ OPERACYJNYCH I SYSTEMOWYCH

INSTYTUT BADAŃ SYSTEMOWYCH POLSKIEJ AKADEMII NAUK

POLSKIE TOWARZYSTWO BADAŃ OPERACYJNYCH I SYSTEMOWYCH

Tom 1

# OPTYMALIZACJA METODY I ZASTOSOWANIA



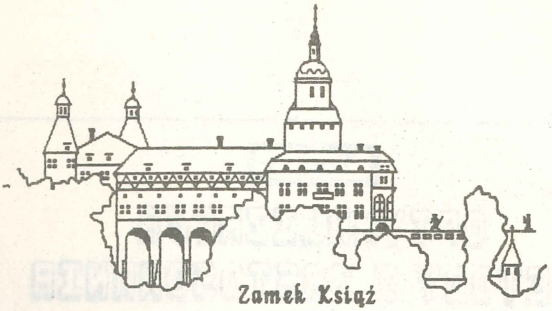

## I KRAJOWA KONFERENCJA BADAŃ OPERACYJNYCH i SYSTEMOWYCH

Książ, 13 - 17 czerwca 1988

BOS'88

INSTYTUT BADAŃ SYSTEMOWYCH POLSKIEJ AKADEMII NAUK

1989
WARSZAWA



Zamek Książ

# I Krajowa Konferencja Badań Operacyjnych i Systemowych

**Organizator konferencji**

Polskie Towarzystwo Badań Operacyjnych i Systemowych
przy współpracy
Instytutu Badań Systemowych PAN



**Komitet naukowy konferencji**

Jerzy Hołubiec, Andrzej Kałuszko, Jerzy Kisielnicki, Henryk Kowalowski, Roman Kulikowski, Franciszek Marecki, Zbigniew Nahorski, Stanisław Piasecki, Jarosław Sikorski, Jan Stachowicz, Jan Stasierski, Andrzej Straszak, Maciej Sysło, Władysław Świtalski

**Redaktorzy naukowi materiałów**

Andrzej Straszak, Zbigniew Nahorski, Jarosław Sikorski

# 3. Optymalizacja w transporcie

3.3

I Krajowa Konferencja
Badań Operacyjnych i Systemowych
Kozłę, 13 - 17 czerwca 1988r.

# PAKIET XTSP1 ROZWIĄZUJĄCY PROBLEM KOMIWOJAŻERA NA PŁASZCZYŹNIE[1]

Stanisław Berka
Instytut Badań Systemowych PAN
ul. Newelska 6
01-447 Warszawa

Streszczenie

W pracy przedstawiono możliwości pakietu służącego do znajdowania przybliżonych rozwiązań zadania komiwojażera na płaszczyźnie i do wygodnego testowania heurystyk dla tego zadania. Jest on wyposażony w bibliotekę najbardziej znanych heurystyk tworzenia drogi komiwojażera, jak też algorytmów poprawiania istniejącej drogi oraz w algorytm znajdowania dolnego oszacowania dla zadania. W pracy zamieszczono opis wszystkich heurystyk bibliotecznych. Dla zadania z 160-ma punktami pakiet pozwala znaleźć rozwiązanie o dokładności 4-5% w czasie rzędu kilkudziesięciu sekund na komputerze IBM PC/AT.

## 1. Wprowadzenie

Problem komiwojażera polega na znalezieniu najkrótszej drogi zamkniętej (tj. zaczynającej się i kończącej w tym samym punkcie) przechodzącej co najmniej raz przez każdy punkt danego zbioru. Problem ten pojawia się w wielu dziedzinach zastosowań praktycznych, jak też jako podproblem powstający przy relaksacji trudniejszych problemów optymalizacji dyskretnej. Obecnie nie udaje się wyznaczyć w

[1] - Praca wykonana w ramach tematu badawczego CPBP 02.15.

sensownym czasie optymalnego rozwiązania zadania komiwojażera przy pomocy metody podziału i oszacowań dla zadań z liczbą punktów rzędu stu czy więcej na mikrokomputerze klasy IBM PC (na większych komputerach rozwiązuje się dokładnie zadania rzędu kilkuset punktów), a zachodzi nieraz potrzeba rozwiązywania praktycznych zadań z kilkoma lub nawet kilkudziesięcioma tysiącami punktów.

Powstało wiele heurystyk znajdujących rozwiązanie suboptymalne dla zadania komiwojażera. Część z nich to algorytmy geometryczne często wykorzystujące specyficzne właściwości zadania płaskiego. Należą do nich między innymi: algorytm oparty na konstrukcji krzywej wypełniającej kwadrat Bartholdiego i Platzmana (1982) testowany w pracy Kryński i Libura (1988), algorytmy wykorzystujące powłokę wypukłą jako poddrogę (droga zamknięta nie obejmująca wszystkich punktów) startową - na przykład algorytm Norback i Love (1977), czy najprostszy algorytm włączający punkty wewnętrzne (tj. nie leżące na powłoce wypukłej) w kolejności wyznaczonej przez współrzędne biegunowe. Istnieje rodzina algorytmów włączających kolejne punkty zbioru do tworzonej w ten sposób stopniowo poddrogi opisanych także w książce Lawler i in. (1985). Część z nich może być wykorzystywana do rozwiązywania zadań w przestrzeni wielowymiarowej. Przykładem algorytmu z tej grupy jest *heurystyka najdalszego wstawiania* (ang. furthest insertion).

Inną grupę stanowią algorytmy poprawy istniejącej drogi. Należy do nich algorytm wymiany par krawędzi 2-Opt, wymiany trójek krawędzi 3-Opt, algorytm Lina-Kernighana (1971) znajdujący rozwiązania o dokładności pojedyńczych procentów, dobierający liczbę wymienianych krawędzi w sposób dynamiczny, czy algorytm Or-Opt polegający na przemieszczaniu s kolejnych punktów drogi z s równym po kolei 3,2,1 opisany we wspomnianej książce. Wśród algorytmów poprawy drogi należy wymienić obecnie intensywnie rozwijane *algorytmy genetyczne* opisane w Holland (1975). Dwa takie algorytmy zostały opisane w Fruhwirth (1987) i w Oliver i in. (1987). Drugi z tych algorytmów w ogólnym zarysie polega

na tym, że w kilku znanych drogach wyszukiwane są fragmenty drogi obejmujące te same podzbiory punktów i z tych fragmentów tworzona jest nowa droga. Czas działania wielu heurystyk można bardzo istotnie skrócić przez odrzucenie a priori niektórych krawędzi (np. poprzez rozpatrywanie jedynie n/2 najkrótszych krawędzi wychodzących z każdego punktu, jeśli n jest liczbą punktów), które z dużym prawdopodobieństwem nie należą do drogi optymalnej. Opisy tak zmodyfikowanych algorytmów i wyniki eksperymentów prowadzonych z tymi algorytmami przedstawiono w Fruhwirth (1987), Stewart (1987). Pakiet jest wyposażony w bibliotekę heurystyk, w której znajdują się niektóre spośród wyżej wymienionych. W drugiej części pracy przedstawiono ogólny opis pakietu, a w trzeciej opisano każdą z heurystyk składających się na bibliotekę pakietu. Część czwarta zawiera krótki opis sposobu korzystania z pakietu. W piątej części przedstawiono wyniki testowania pakietu i wyniki eksperymentów obliczeniowych wykonanych przy pomocy pakietu.

## 2. Ogólny opis pakietu

Pakiet XTSP1 służy do znajdowania suboptymalnego rozwiązania zadania komiwojażera na płaszczyźnie z geometrią euklidesową. Wyposażony jest w bibliotekę kilku wybranych, najbardziej efektywnych heurystyk, które można łączyć po kilka w tzw. strategie w taki sposób, że pierwsza z nich tworzy drogę (nazywaną drogą startową), a następne służą do poprawiania znalezionej drogi. Możliwe jest zdefiniowanie kilku strategii i łatwe testowanie ich na zadaniach generowanych losowo lub wprowadzanych ze zbioru. Algorytm znajdujący dolne oszacowanie rozwiązywanego problemu pozwala ocenić dokładność otrzymanego rozwiązania.

Pakiet jest wyposażony w grafikę, dzięki której możliwe jest śledzenie działania heurystyk. Wyniki obliczeń wyprowadzane są do zbioru dyskowego w formacie umożliwiającym ich łatwą dalszą obróbkę. Wybór opcji pakietu oraz definiowanie strategii są dokonywane przy pomocy systemu menu ekranowych. Opcje mogą być przechowywane w

zbiorze dyskowym. W czwartej części zamieszczono opis korzystania z pakietu.

## 3. Opis stosowanych heurystyk

Opisywane heurystyki dzielą się na algorytmy służące do znajdowania drogi startowej (tzw. *preprocesory*) oraz algorytmy poprawiające istniejącą drogę (tzw. *postprocesory*). Poniżej opisany zostanie także pokrótce algorytm znajdujący dolne oszacowanie rozwiązywanego problemu. Opisy służą jedynie przedstawieniu idei algorytmu; w miarę możliwości podawane są dodatkowe źródła informacji na temat opisywanych algorytmów. W nawiasie podano symbol, którym heurystyki są oznaczane w dalszej części pracy.

### Preprocesory

1) Powłoka wypukła i algorytm najdalszego wstawiania (CHFI)
Najpierw znajduje się powłokę wypukłą stosując efektywny algorytm Kirkaptricka i Seidela (1986). Powłoka tworzy poddrogę (zamknięta droga nie obejmująca wszystkich punktów), w którą włącza się pozostałe punkty stosując heurystykę najdalszego wstawiania (ang. farthest insertion, patrz TSP (1985) rozdz. 7 i Sysło i in. (1983) ), którą można opisać następująco:

1° Dla każdego punktu y nie włączonego jeszcze do drogi znajdź najbliższy punkt należący już do drogi. Odległość między oboma punktami nazwij $d_y$.

2° Spośród punktów nie włączonych do drogi wybierz punkt y o minimalnej wartości $d_y$.

3° Wstaw wybrany punkt y między dwa sąsiednie punkty x,z będące w drodze, takie że wartość wyrażenia d(x,y) + d(y,z) - d(x,z) jest minimalna.

2) Najdłuższa para i algorytm najdalszego wstawiania (DFI)
Algorytm ten tym tylko różni się od powyższego algorytmu, że początkową poddrogę tworzy nie powłoka wypukła a dwie krawędzie łączące dwa najdalej położone od siebie punkty. Algorytm ten jest zwykle nieco gorszy zarówno pod względem

długości znalezionej drogi jak i czasu obliczeń od powyższego algorytmu.

3) Krzywa wypełniająca kwadrat (SFC)
Konstruuje się krzywą Sierpińskiego wypełniającą najmniejszy kwadrat obejmujący wszystkie punkty zbioru. Następnie tworzona jest droga, która obchodzi punkty zbioru w tej samej kolejności, w której obchodzone są one przez krzywą Sierpińskiego. Sposób konstruowania krzywej Sierpińskiego opisano w Bartholdi i Platzman (1982). Algorytm tworzy drogę o szacunkowej dokładności rzędu 25%, za to wymaga bardzo niewielkiego nakładu obliczeń.

4) Krzywa Sierpińskiego z przeglądem (SFCLS)
Najpierw znajduje się drogę według powyższej heurystyki. Następnie dokonuje się lokalnego przeglądu tej drogi według algorytmu opracowanego przez Kryńskiego i Liburę (1988). Przegląd ten polega na wykorzystaniu niejednoznaczności w konstruowaniu krzywej Sierpińskiego. Dzięki temu można wygenerować szereg różniących się niektórymi fragmentami dróg. Spośród powstałych w ten sposób wariantów wybiera się najlepszy. Przegląd daje zwykle kilkuprocentową poprawę dokładności dla małych zadań i kilkunastoprocentową poprawę (10-13%) dla większych zadań przy kilkakrotnym wzroście czasu obliczeń (Kryński i Libura (1988) ).

Postprocesory
5) Algorytm 2-Opt (OPT2)
W algorytmie wymiany par krawędzi, nazywanym w skrócie algorytmem 2-Opt (Sysło i in. (1983) ), dokonuje się prób wymiany każdej pary krawędzi w utworzonej przy pomocy innego algorytmu drodze. Pierwsza wymiana, która powoduje skrócenie drogi, jest dokonywana i algorytm kontynuje sprawdzanie. Algorytm kończy po sprawdzeniu wszystkich możliwych wymian.

6) Algorytm 3-Opt (OPT3)

W algorytmie wymiany trójek krawędzi, nazywanym w skrócie 3-Opt (Sysło i in. (1983) ), dokonuje się prób wymiany wszystkich możliwych trójek krawędzi. Każda wymiana, która powoduje skrócenie drogi, jest wykonywana i algorytm kontynuuje sprawdzanie. Algorytm kończy działanie po sprawdzeniu wszystkich możliwych wymian. Algorytm wymaga dużego nakładu obliczeń i to ogranicza jego stosowanie do kilkudziecięciu (np. do 80-ciu) punktów.

7) Algorytm Or-Opt (OROPT)

Algorytm ten, opracowany w Or (1976), jest ograniczonym wariantem algorytmu 3-Opt. Sprawdzana jest tylko część wymian trójek krawędzi. Algorytm sprawdza czy można przenieść s kolejnych miast na inną pozycję w drodze. Jeżeli taka modyfikacja powoduje skrócenie drogi, to jest ona wykonywana i sprawdzanie jest kontynuowane aż do momentu, gdy nie daje się poprawić drogi. Powyższy test przeprowadza się kolejno dla s=3,2,1. Algorytm pozwala na znalezienie drogi nieco tylko gorszej niż algorytm 3-Opt przy znacznie mniejszym czasie obliczeń.

8) Algorytm NOr-Opt (NOR)

Jest to zmodyfikowany algorytm Or-Opt, w którym nie są rozpatrywane wymiany związane z dodaniem krawędzi nie należących do pewnego zadanego zbioru krawędzi. Zbiór ten w aktualnej wersji algorytmu jest tworzony w ten sposób, że dla każdego wierzchołka do zbioru wprowadzamy p-procent (parametr) najkrótszych krawędzi. Wyniki eksperymentów prowadzonych z tym algorytmem można znaleźć w Fruhwirth (1987). Przy uwzględnianiu tylko 10-20% najkrótszych krawędzi algorytm znajduje drogę nieznacznie dłuższą niż algorytm Or-Opt, przy istotnym zmniejszeniu czasu obliczeń.

We wszystkich algorytmach wymiany krawędzi (algorytmy 5-8) po znalezieniu wymiany dającej poprawę i wykonaniu jej, sprawdzanie kontynuowane jest od tego miejsca, w którym przerwano sprawdzanie.

9) Algorytm znajdujący dolne oszacowanie (LBND)
Algorytm tworzy ciąg minimalnych drzew uzupełnionych. Drzewo uzupełnione otrzymuje się z drzewa rozpinającego grafu z usuniętym jednym wierzchołkiem przez dołączenie do niego dwóch najkrótszych krawędzi incydentnych z wierzchołkiem usuniętym. Jeżeli drzewo uzupełnione jest obwodem, to jest on rozwiązaniem optymalnym. W przeciwnym przypadku suma długości krawędzi drzewa jest dolnym oszacowaniem długości drogi optymalnej. Na każdy wierzchołek o stopniu różnym od 2 nakłada się karę proporcjonalną do różnicy stopnia wierzchołka i liczby 2, a następnie długości wszystkich krawędzi modyfikuje się poprzez dodanie do nich kar nadanych obu wierzchołkom incydentnym z daną krawędzią. Jest to ogólnie znany algorytm stosowany w metodach podziału i oszacowań - patrz Lawler i in. (1985). W pakiecie zaimplementowano wersję opisanego algorytmu opracowaną przez Volgenanta i Jonkera (1982). Algorytm i wyniki jego testowania zamieszczono w Berka i Libura (1988). Znaczny nakład obliczeń związanych z algorytmem powoduje ograniczenie stosowania go do zadań liczbą punktów rzędu 100.

4. Sposób korzystania z pakietu

Szczegółową instrukcję korzystania z pakietu można znaleźć w Berka i in. (1988). W pakiecie został wykorzystany system menu ekranowych opisany w Berka (1988). Poniżej zostały podane tylko podstawowe informacje. Istnieją dwa tryby pracy z pakietem XTSP1. W pierwszym z nich wywołujemy pakiet bez podawania parametrów w linii komend. Na ekranie ukazuje się główne menu, które umożliwia:

(i) przejście do wyboru heurystyk i definiowania strategii (tj sekwencji: preprocesor + postprocesory);
(ii) przejście do wyboru opcji;
(iii) zapis zdefiniowanych strategii oraz wybranych opcji do zbioru parametrów na dysku;
(iv) wczytanie parametrów pakietu z dyskowego zbioru

parametrów;

(v) rozpoczęcie obliczeń;

(vi) zakończenie pracy z pakietem i powrót do sytemu operacyjnego.

Definowanie stategii dokonuje się za pomocą menu przez wybór heurystyki tworzącej drogę spośród widocznych w menu, a następnie przez wybór kolejno do czterech heurystyk poprawy. W przypadku gdy nie zostanie wybrana heurystyka tworząca drogę, jako droga początkowa dla heurystyk poprawy będzie traktowana droga wczytana ze zbioru lub wygenerowana losowo. Można zdefiniować do dziesięciu strategii. W dolnej części ekranu widoczna jest reprezentacja graficzna aktualnie definiowanej strategii. Menu wyboru opcji pozwala m. in. określić, czy:

(i) ma zostać obliczone dolne oszcowanie zadania;

(ii) odległosci miedzy punktami mają być obliczone i zapamiętane w tablicy;

(iii) każda znaleziona droga ma być zapisywana do zbioru dyskowego;

(iv) grafika ma być włączona czy wyłączona;

(v) zadanie ma zostać wygenerowane losowo, czy wczytane ze zbioru i o jakiej nazwie;

(vi) ile najkrótszych krawędzi ma być uwzględnianych w algorytmie Nor-Opt (o ile został wybrany).

Po ustawieniu opcji lub wczytaniu ich ze zbioru należy wybrać w głównym menu pozycję rozpoczęcia obliczeń. Natępuje wczytanie zadania ze zbioru lub wygenerowanie losowego zadania, po czym zadanie to jest rozwiązywane według pierwszej ze zdefiniowanych strategii. Po wykonaniu każdej heurystyki następuje zatrzymanie. Naciśnięcie dowolnego klawisza powoduje przejście do dalszej części obliczeń. Po wykonaniu wszystkich elementów tej strategii zadanie rozwiązywane jest według kolejnych strategii. Na koniec obliczane jest dolne oszacowanie zadanie i następuje powrót do głównego menu. Możliwe jest polecenie wygenerowania i rozwiązania dowolnie licznej serii zadań losowych.

Czas działania każdej wykonanej heurystyki i długość każdej znalezionej drogi jest zapisywana do zbioru dyskowego wraz z numerem strategii, kolejnym numerem heurystyki w strategii oraz numerem zadania w serii zadań generowanych losowo.

Drugi tryb pracy pakietu polega na tym, że po jego wywołaniu pakiet przechodzi bezpośrednio do wykonywania obliczeń bez zatrzymywania się po wykonaniu każdej heurystyki, a po wykonaniu wszystkich obliczeń wraca do systemu operacyjnego. Aby pakiet działał w drugim trybie należy wywołując pakiet podać w linii komend nazwę zbioru parametrów, według których wykonywane mają być obliczenia.

## 5. Wyniki eksperymentów obliczeniowych

Dla celów testowania i porównywania heurystyk zostały wygenerowane dwie serie po 10 zadań każda: pierwsza złożona z zadań z 80-ma punktami, druga - ze 160-ma punktami. Współrzędne punktów tych zadań były generowane z przedziału [1,1000] przy pomocy generatora o rozkładzie równomiernym na tym przedziale. Czasy obliczeń zamieszczone poniżej podawane są w sekundach. Obliczenia były przeprowadzane na mikrokomputerze IBM PC/AT. Zarówno czasy jak i długości dróg podane w tabelach oznaczają wartości średnie dla serii 10 zadań. Heurystyki oznaczane są w tabelach symbolami podanymi w części 3.

Poniżej zamieszczono wyniki porównania preprocesorów.

| | | CHFI | DFI | SFC | SFCLS |
|---|---|---|---|---|---|
| n=80 | t [s] | 2.89 | 3.46 | 0.71 | 3.26 |
| | l | 7540.01 | 7600.63 | 8662.77 | 8065.35 |
| n=160 | t [s] | 10.71 | 13.76 | 1.40 | 7.51 |
| | l | 10408.07 | 10478.62 | 12036.39 | 11027.84 |

Pokazują one, że oba algorytmy oparte na heurystyce najdalszego wstawiania dla tych rozmiarów zadań są efektywniejsze od pozostałych heurystyk. Natomiast czas działania obu tych algorytmów rośnie znacznie szybciej przy wzroście rozmiaru zadania, niż czas działania algorytmów

wykorzystujących krzywą Sierpińskiego. Dla bardzo dużych zadań będą więc efektywniejsze algorytmy wykorzystujące krzywą Sierpińskiego.

Wyniki porównania postprocesorów zostały zestawione w poniższej tabeli.

| | CHFI | +OPT2 | +OROPT | +NOR | +OPT3 |
|---|---|---|---|---|---|
| n=80 | t [s] | 8.40 | 22.80 | 14.03 | 813.70 |
| | l | 7403.56 | 7282.05 | 7268.89 | 7213.05 |
| n=160 | t [s] | 37.17 | 117.13 | 62.65 | |
| | l | 10354.59 | 9989.89 | 9989.69 | |

Brak wyników dla heurystyki 3-Opt i dla zadań ze 160-ma punktami jest spowodowany zbyt dużym nakładem obliczeń potrzebnym do wykonania przeglądu wszystkich możliwych wymian trójek krawędzi. Przedstawione wyniki pokazują, że najbardziej efektywnym postprocesorem (w przypadku stosowania powłoki wypukłej i najdalszego wstawiania jako preprocesora) jest algorytm Or-Opt z przeglądem ograniczonym do najbliższego sąsiedztwa. Dlatego dla tego algorytmu przeprowadzono test mający na celu ustalenie najlepszej wartości parametru p określającego wielkość tego sąsiedztwa. Wyniki testu przedstawiono poniżej.

| | p | =9 | =18 | =27 | =36 |
|---|---|---|---|---|---|
| n=160 | t [s] | 42.61 | 49.22 | 56.39 | 62.68 |
| | l | 10052.68 | 9979.18 | 9989.69 | 9989.69 |

Wyniki zebrane w tabeli 3. wskazują na, że przegląd w zbyt dużym sąsiedztwie może także prowadzić do otrzymywania gorszych rozwiązań. Najlepsza wartość parametru p dla zadań ze 160-ma punktami wynosi 18, co stanowi ok. $0.1 \cdot n$.

Poniżej zestawiono dane dotyczące porównania strategii z różnymi drogami startowymi poprawianymi przy pomocy tego samego algorytmu poprawy. Wykorzystano algorytm Or-Opt z przeglądem w sąsiedztwie, z parametrem p=9 dla zadań z 80-ma punktami i z parametrem p=18 dla zadań ze 160-ma punktami.

| Preprocesor | | CHFI | DFI | SFC | SFCLS |
|---|---|---|---|---|---|
| n=80 | t [s] | 11.81 | 12.14 | 10.74 | 12.89 |
| | l | 7287.43 | 7384.78 | 7373.57 | 7368.56 |
| n=160 | t [s] | 49.24 | 52.87 | 45.37 | 48.40 |
| | l | 9979.18 | 10093.51 | 10192.71 | 10157.22 |

Wyniki pokazują, że algorytmy, które dają lepsze drogi startowe w przypadku tej klasy postprocesorów pozwalają znaleźć lepsze rozwiązania końcowe.

Korzystając z algorytmu wbudowanego w pakiet znaleziono dolne oszacowania dla zadań testowych. Odstęp średni od najlepszego znanego rozwiązania wynosił dla zadań z 80-ma punktami 2.55%, a dla zadań ze 160-ma punktami 4.27% przy średnich czasach obliczeń odpowiednio ok. 16 minut i ok. 2.5 godzin.

Bibliografia

Bartholdi J.J., Platzman L.K. (1982) An O(n log n) Planar Travelling Salesman Heuristic Based on Spacefilling Curves, OR Lett., Vol.4.

Berka S. (1988) Opis użytkowy pakietu obsługi menu ekranowych dla IBM PC, Oprac wewn. 23-A1529/88, IBS PAN, Warszawa.

Berka S., Kryński S., Libura M. (1988) Pakiet XTSP1 dla euklidesowych zadań komiwojażera dla IBM PC - sposób korzystania, Oprac. wewn 22-A1529/88, IBS PAN, Warszawa.

Berka S., Libura M. (1988) Oszacowanie dolne długości najkrótszego obwodu Hamiltona w grafie nieskierowanym, Oprac. wewn. 42-A1529/88, IBS PAN, Warszawa.

Fruhwirth B. (1987) Untersuchung über genetisch motivierte Heuristiken für das euklidischen Rundreiseproblem, Techn. Bericht 87-103, Inst. für Mathem., Techn. Universität Graz.

Holland J.H. (1975) Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, MI.

Kirkpatrick D.G., Seidel R. (1986), The Ultimate Planar Convex Hull Algorithm ?, SIAM J.Comput., Vol. 15, No. 1.

Kryński L.S., Libura M. (1988) Algorytmy przybliżone dla zadania komiwojażera na płaszczyźnie oparte na konstrukcji krzywej wypełniającej kwadrat, Zeszyty Naukowe Polit. Śląskiej, z. 94, ss. 177-187.

Lawler E.L. i in. (1985) The Travelling Salesman Problem, ed. E.L. Lawler et al., John Wiley & Sons Ltd.

Lin S., Kernighan B.W. (1971) An Effective Heuristic Algorithm for the Travelling Salesman Problem, Oper. Res., Vol. 21, pp. 498-516.

Norback J.P., Love R.F. (1977) Geometric Approach to Solving the Travelling Salesman Problem, Management Science, Vol. 23, No. 11.

Oliver I.M., Smith D.J., Holland J.R.C. A Study of Permutation Crossover Operators on the Travelling Salesman Problem, Proc. of the Second Int. Conf. on Genetic Alg., Cambridge, MA.

Or I. (1976) Travelling Salesman-Type Combinatorial Problem and their Relation to the Logistics of Regional Blood Banking, Ph.D. Thesis, Northwestern University, Evanston, IL.

Stewart W.R.Jr (1987) Accelerated Branch Exchange Heuristics for Symmetric Travelling Salesman Problem, Networks, Vol. 17, pp. 423-487.

Sysło M.M., Deo N., Kowalik J.S. (1983) Discrete Optimization Algorithms with Pascal Programs, Prentice Hall.

Volgenant T., Jonker R. (1982) A branch and bound algorithm for the symmetric travelling salesman problem based on 1-tree relaxation, European J. Oper. Res. 9, pp 83-89.

# Zarząd

## Polskiego Towarzystwa Badań Operacyjnych i Systemowych

### Prezes

prof.dr hab.inż. Andrzej Straszak

Instytut Badań Systemowych PAN

### Wiceprezes

prof.dr hab.inż. Jan Stasierski

Wojskowa Akademia Techniczna

### Wiceprezes

prof.dr hab.inż. Stanisław Piasecki

Instytut Badań Systemowych PAN

### Sekretarz generalny

dr inż. Zbigniew Nahorski

Instytut Badań Systemowych PAN

### Sekretarz

dr inż. Jarosław Sikorski

Instytut Badań Systemowych PAN

### Skarbnik

dr inż. Andrzej Kałuszko

Instytut Badań Systemowych PAN

### Członkowie

prof.dr hab. Jerzy Kisielnicki

Wydział Zarządzania UW

doc.dr hab.inż. Bohdan Korzan

Wojskowa Akademia Techniczna

doc.dr hab.inż. Jan Stachowicz

Zakład Nauk Zarządzania PAN

doc.dr hab.inż. Maciej Sysło

Instytut Informatyki UWr.

### Komisja rewizyjna

PRZEWODNICZĄCY

dr Władysław Świtalski

Katedra Cybernetyki i Badań Operacyjnych UW

CZŁONKOWIE

dr inż. Janusz Kacprzyk

Instytut Badań Systemowych PAN

dr inż. Marek Malarski

Instytut Transportu PW

doc.dr hab.Henryk Sroka

Akademia Ekonomiczna w Katowicach

dr inż. Leon Słomiński

Instytut Badań Systemowych PAN

# PION III