



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

**ROZWÓJ I ZASTOSOWANIA
TECHNOLOGII I SYSTEMÓW
INFORMATYCZNYCH**

pod redakcją:

Jana Studzińskiego

Ludostawa Drelichowskiego

Olgierda Hryniewicza



**ROZWÓJ I ZASTOSOWANIA TECHNOLOGII
I SYSTEMÓW INFORMATYCZNYCH**

Polska Akademia Nauk • Instytut Badań Systemowych

Seria: BADANIA SYSTEMOWE
tom 28

Redaktor naukowy:

Prof. dr hab. Jakub Gutenbaum

Warszawa 2001

ROZWÓJ I ZASTOSOWANIA TECHNOLOGII I SYSTEMÓW INFORMATYCZNYCH

pod redakcją

Jana Studzińskiego, Ludosława Drelichowskiego
i Olgierda Hryniewicza

Wydano z wykorzystaniem dotacji KOMITETU BADAŃ NAUKOWYCH

Książka zawiera wybór artykułów poświęconych omówieniu aktualnego stanu badań w kraju w zakresie rozwoju technologii, modeli i systemów informatycznych oraz ich zastosowań w różnych dziedzinach gospodarki narodowej. Wyodrębnioną grupę stanowią artykuły aplikacyjne omawiające wyniki projektów badawczych i celowych KBN.

Recenzenci artykułów:

Dr hab. inż. Ryszard Budziński, prof. US

Prof. dr hab. inż. Janusz Kacprzyk

Dr hab. Adam Kopiński, prof. AE we Wrocławiu

Doc dr hab. inż. Marek Libura

Prof. dr hab. inż. Andrzej Straszak

© Instytut Badań Systemowych PAN, Warszawa 2001

ISBN 83-85847-59-6

ISSN 0208-8028

Rozdział 3

**Metody i algorytmy obliczeniowe
w systemach informatycznych**

ALGORYTMY EWOLUCYJNE W ROZWIĄZYWANIU PROBLEMÓW DOBORU TRASY POJAZDU

Radostaw Sakierski

Wydział Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej

W artykule zawarto propozycję algorytmów: ewolucyjnego oraz hybrydowego zaimplementowanych w środowisku Matlab 5.3, które pozwalają na określenie rozwiązania zbliżonego do optymalnego dla dowolnego problemu doboru trasy pojazdu (ang. vehicle routing problem) o dużym stopniu skomplikowania. Ponadto zamieszczona została analiza porównawcza wyników zaproponowanych przez powyższe algorytmy oraz algorytm oszczędności Clarka i Wrighta dla rzeczywistego problemu optymalizacji trasy pojazdu definiującego 37 punktów zapotrzebowania oraz 3 warunki ograniczające.

1. Wstęp

Współcześnie główny nurt badań dotyczący problematyki rozwiązywania problemów kombinatorycznych skupił się na próbach opracowania algorytmów pozwalających na poszukiwanie rozwiązań zbliżonych do rozwiązań optymalnych przy wykorzystaniu między innymi metod heurystycznych oraz metod inteligencji obliczeniowej, które to metody roją największe nadzieje na opracowanie efektywnego algorytmu (Matthews, 2000, TSPBIB, 1999, TSPLIB, 1998)

Optymalny dobór trasy, a w szczególności problem komiwojażera, jest niezwykle szeroko omawiany w literaturze poświęconej problemom optymalizacji. Znanych jest wiele metod rozwiązywania tych zadań, jednakże z uwagi na fakt, że problemy doboru optymalnej trasy należą do klasy problemów kombinatorycznych, rozwiązanie optymalne w większości przypadków niemożliwe jest do osiągnięcia metodami klasycznymi w rozsądnym czasie.

Problemy doboru trasy (ang. routing problems) w ogólności dzielą się na dwie klasy: pokrywania węzłów sieci (ang. node-covering) oraz pokrywania krawędzi sieci (ang. edge-covering) (Larson, Odoni, 1999). Do pierwszej klasy problemów należy między innymi problem komiwojażera oraz problem doboru trasy pojazdu (ang. vehicle routing problem).

Problem doboru trasy pojazdu formułowany jest w następujący sposób: Danych jest n punktów zapotrzebowania na danym obszarze, z których każdy zgłasza zapotrzebowanie na pewną ilość towaru wyrażoną w jednostkach wagowych Q_i , gdzie $i = 1, \dots, n$. Towary przechowywane są w magazynie D , gdzie stacjonują także

wszystkie pojazdy, które mają identyczne właściwości wyrażone w maksymalnej ładowności pojazdu lub maksymalnej trasie jaką pojazd może pokonać. Wszystkie pojazdy rozpoczynają i kończą swoją trasę w magazynie D . Problem polega na wyznaczeniu zbioru tras zaopatrzeniowych od magazynu D do wszystkich punktów zapotrzebowania, w taki sposób, aby zminimalizować całkowitą długość tras. Ponadto zakłada się, że waga towaru Q_i dla żadnego z punktów zapotrzebowania nie może przewyższać maksymalnej ładowności pojazdu oraz całkowite zapotrzebowanie każdego z punktów zaspokajane jest przez dokładnie jeden pojazd.

Problem doboru trasy pojazdu jest znacznie bardziej skomplikowany od tradycyjnego problemu komiwojażera. Zgodnie z (Larson, Odoni, 1999), dotychczas nie udało się uzyskać rozwiązań optymalnych dla problemu definiującego więcej niż 30 punktów zapotrzebowania.

2. Metody rozwiązywania problemów doboru trasy pojazdu

Algorytm oszczędności Clarka i Wrighta. Ideą tego algorytmu heurystycznego jest wykorzystanie prostej metody stopniowego dołączania kolejnych węzłów do tworzonego rozwiązania (Larson, Odoni, 1999). Początkowe rozwiązanie bazuje na wykorzystaniu n pojazdów, z których każdy zaopatruje jeden z n punktów zapotrzebowania. Całkowita droga pojazdów w takim przypadku jest równa dwukrotności sumy odległości każdego z punktów od magazynu D . Modyfikacja rozwiązania początkowego polegająca na obsłużeniu dwóch punktów i oraz j przez jeden pojazd pozwala na wprowadzenie oszczędności $s(i, j)$ w całkowitej drodze pojazdów o wartości:

$$\begin{aligned} s(i, j) &= 2d(D, i) + 2d(D, j) - [d(D, i) + d(i, j) + d(D, j)] \\ &= d(D, i) + d(D, j) - d(i, j) \end{aligned} \quad (1)$$

Algorytm działa według następujących czterech kroków:

1. Wyliczenie wartości $s(i, j)$ dla wszystkich par (i, j) punktów zapotrzebowania.
2. Posortowanie punktów zapotrzebowania malejąco według wartości $s(i, j)$.
3. Włączenie krawędzi (i, j) do danej trasy przy przeglądaniu posortowanej listy, jeżeli poprzez to włączenie nie zostanie naruszony żaden z warunków zadania i jeżeli:
 - żaden z punktów i oraz j nie został jeszcze włączony do żadnej trasy, tworzona jest nowa trasa postaci $\{D, i, j, D\}$;
 - jeden z punktów został włączony do dowolnej trasy i nie należy on do wnętrza tej trasy, gdzie przez wnętrze trasy rozumie się podzbiór tych punktów tej trasy, które nie mają bezpośredniego połączenia z magazynem, drugi z punktów włączany jest do trasy pomiędzy magazynem, a pierwszym punktem;

- oba punkty zostały już włączone do dwóch różnych tras i żaden z tych punktów nie należy do wnętrza tych tras, wtedy obie trasy są łączone.
- 4. Jeżeli lista posortowanych punktów zapotrzebowania nie została przeglądnięta do końca powrót do punktu 3.

Powyżej przedstawiony algorytm jest niezwykle prosty w implementacji programowej oraz bardzo efektywny. Jego niepowtarzalną zaletą jest możliwość budowania dowolnie skomplikowanych ograniczeń zadania, które sprawdzane są przed każdym dołączeniem punktu do rozwiązania. Jednakże należy zauważyć, że algorytm ten może generować wyniki znacznie odbiegające od rozwiązań optymalnych. Doświadczenia eksperymentalne dowiodły, że istnieje grupa zadań, dla których algorytm zwraca daleko nie zadawalające rozwiązania.

Algorytm zamiatania. Działanie algorytmu polega na przeglądaniu punktów zapotrzebowania z wykorzystaniem zwiększającego się kąta przeszukiwania promienia zamiatającego (Larson, Odoni, 1999). Promień zamiatający inicjowany jest w układzie współrzędnych o początku w magazynie D oraz obrocie dla którego oś argumentów przechodzi przez jeden dowolnie obrany punkt początkowy. Zwiększanie kąta nachylenia promienia zamiatającego do dodatniej osi argumentów pozwala na tworzenie tras z punktów zapotrzebowania zamiecionych przez promień, jeżeli tylko nie zostały przekroczone ograniczenia zadania. Po przekroczeniu ograniczeń układ współrzędnych jest inicjalizowany na nowo, przy czym tym razem oś argumentów przechodzi przez punkt, który spowodował przekroczenie ograniczenia. Wszystkie zamiecione punkty tworzą nową trasę, do czasu ponownego naruszenia warunków zadania. Algorytm działa do momentu włączenia wszystkich punktów zapotrzebowania do którejś ze stworzonych tras. Otrzymane rozwiązanie można znacznie przybliżyć do rozwiązania optymalnego przez wykorzystanie metod rozwiązywania problemu komiwojażera dla każdej z wyodrębnionych tras niezależnie, co pozwoli na optymalne ustalenie kolejności odwiedzanych punktów.

Reprezentacja dla problemu doboru trasy pojazdu. W (Matthews, 2000) zaproponowany został model reprezentacji, w której geny chromosomów są odzwierciedleniem krawędzi sieci. Każdy genotyp składa się z tylu chromosomów, ile pojazdów zdefiniowanych jest w problemie. Funkcja przystosowania jest sumą kwadratów długości tras każdego z pojazdów. Operator krzyżowania podobnie jak i operator mutacji generuje rozwiązania niedopuszczalne i wymaga mechanizmów naprawy opisanych szerzej w (Matthews, 2000). Idea działania operatora mutacji polega na zmianie przypisania krawędzi do pojazdu, natomiast operator krzyżowania tworzy dla każdego pojazdu zbiór będący iloczynem zbiorów krawędzi obu rodziców, a wszystkie pozostałe krawędzie przydziela losowo do jednego z potomków zachowując przypisanie do pojazdu.

3. Implementacja algorytmu ewolucyjnego

Z tego względu, iż w dostępnych opracowaniach dotyczących algorytmów ewolucyjnych nie przedstawiono reprezentacji dla problemu doboru trasy pojazdu

pozwalającego na jego prostą implementację programową, konieczne stało się opracowanie algorytmu ewolucyjnego od podstaw.

Logiczna reprezentacja populacji opiera się na trójwymiarowej macierzy, której niezerowy element e_{ijk} , wskazuje na k -tą miejscowość j -tej trasy i -tego osobnika. Fizyczna reprezentacja korzysta z macierzy dwuwymiarowej, dzięki czemu pozwala na wielokrotne skrócenie czasu przetwarzania dla bardziej złożonych problemów i wnosi dodatkowe korzyści wynikające z możliwości posłużenia się operatorami macierzowymi jakie oferuje Matlab 5.3. Opis algorytmu zawarty w dalszej części opracowania bazuje na reprezentacji logicznej.

Główna pętla algorytmu zaprojektowana jest zgodnie ze wzorcowym pseudokodem typowego algorytmu ewolucyjnego i wykonywana jest do czasu przekroczenia jednego z dwóch warunków ograniczających: maksymalnej liczby iteracji lub oczekiwanej wartości oceny. Zgodnie ze schematem algorytmu genetycznego działanie algorytmu rozpoczyna się od inicjalizacji populacji składającej się z osobników, z których każdy zawiera losowo wygenerowany ciąg tras. Ocena osobników korzysta z funkcji kary. Można zauważyć, iż w wyniku zastosowania operatorów genetycznych często generowane są rozwiązania niedopuszczalne, czyli takie które przekraczają jedno lub więcej z trzech nałożonych ograniczeń dotyczących ładowności pojazdu, długości trasy i czasu pracy. Aby zmniejszyć prawdopodobieństwo przejścia przez sito selekcji osobnika reprezentującego rozwiązanie niedopuszczalne wskazane okazało się pogorszenie jego wyniku o zsumowane wartości przekroczenia ograniczeń dla każdej z tras pomnożone przez odpowiednie mnożniki kary. Ocena każdego z osobników zrealizowana została według poniższych wzorów:

$$ocena = suma_dlugosci_tras + kara_czasu + kara_dlugosci + kara_ladownosc \quad (2)$$

$$kara_czasu = mnoznik_czasu * suma_kar_czasu_tras \quad (3)$$

$$kara_dlugosci = mnoznik_dlugosci * suma_kar_dlugosci_tras \quad (4)$$

$$kara_ladownosci = mnoznik_ladownosci * suma_kar_ladownosci_tras \quad (5)$$

Należy zauważyć, że osobnik ogólnie bardzo dobry, który jednak przekroczył nieznacznie jeden z warunków ograniczających może być oceniony w sposób zbliżony do osobnika gorzej przystosowanego, który nie przekroczył ograniczeń. Zastosowany mechanizm oceny z funkcją kary okazał się w takich przypadkach niezwykle efektywny umożliwiając osobnikom dobrze przystosowanym i równocześnie reprezentującym rozwiązanie niedopuszczalne przetrwanie selekcji oraz branie udziału w krzyżowaniu i mutacji.

Selekcja osobników wykorzystuje mechanizm skalowania funkcji oceny, strategię elitarną oraz selekcję metodą ruletki. Oceny zostały przeskalowane w taki sposób, aby przyjmowały wartości z przedziału $[0; 1]$, gdzie ocenę równą 1 otrzymuje najlepiej przystosowany osobnik w populacji. Tak więc mechanizm skalowania wykorzystany w algorytmie nieznacznie różni się swoim działaniem od skalowania liniowego i opiera się na wykorzystaniu wzoru:

$$ocena_sk = \frac{ocena - \max\{oceny\}}{\min\{oceny\} - \max\{oceny\}} \quad (6)$$

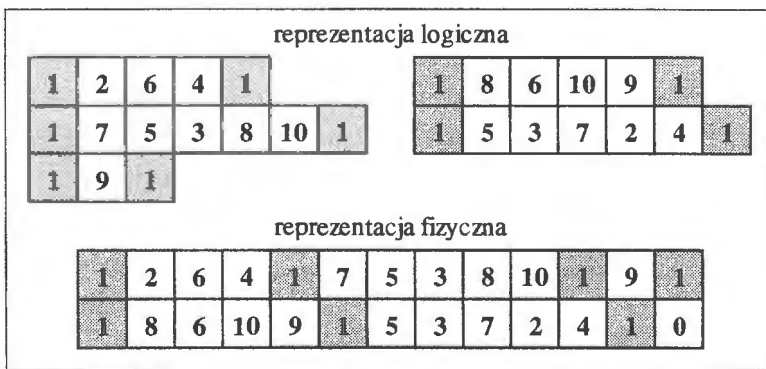
Algorytm oferuje dwa rozwiązania implementujące strategię elitarną:

- kopiowanie bez zmian pewnej liczby osobników najlepiej przystosowanych;
- kopiowanie bez zmian pewnej liczby różnych osobników najlepiej przystosowanych.

Zastosowanie pierwszego z powyższych rozwiązań wiąże się z niedogodnością w postaci powolnej zbieżności algorytmu, jednakże pozwala na zachowanie bardziej różnorodnego materiału genetycznego reprezentującego najlepsze przystosowania. Druga metoda, zwana dyskryminującą, w wyniku swego działania w początkowych iteracjach zapewnia szybszą zbieżność, jednakże w przypadku, gdy wynik bliski jest rozwiązaniu optymalnego, powoduje lawinowy wzrost liczby kopii najlepszego osobnika w populacji, zwiększając udział jego materiału genetycznego z każdą iteracją. W skrajnym przypadku większość osobników w kolejnej generacji jest wynikiem krzyżowania i mutacji jednego superosobnika.

Selekcja metodą ruletki dokonuje losowego wyboru osobnika z prawdopodobieństwem proporcjonalnym do jego przystosowania spośród osobników poprzedniej populacji.

Krzyżowanie osobników realizowane jest z wykorzystaniem operatora krzyżowania VRPC. Powstali w wyniku krzyżowania potomkowie wpisywani są na miejsce rodziców. Krzyżowane mogą być potencjalnie wszystkie osobniki z wyłączeniem genotypów kopiowanych według strategii elitarnej.

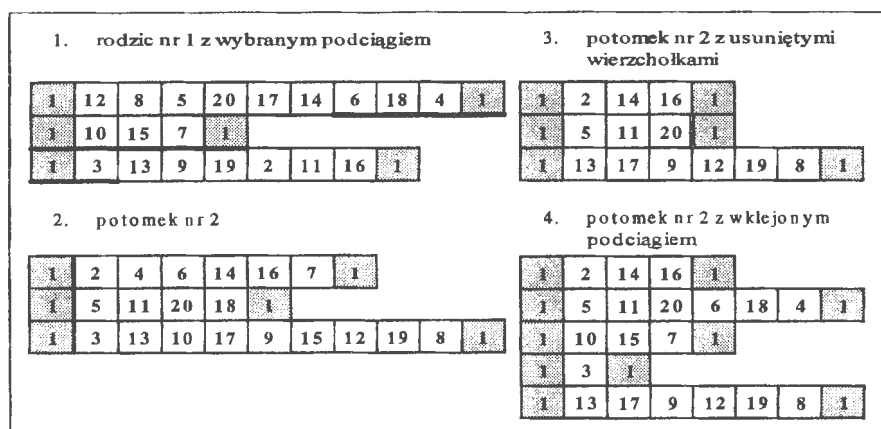


Rys. 1 Przykładowa populacja składająca się z dwóch osobników dla problemu doboru trasy dla 10 miejscowości w formie macierzy trójwymiarowej (logicznej) i macierzy dwuwymiarowej (fizycznej).

Autorski operator krzyżowania VRPC (ang. VRP crossover) zapewnia wymianę informacji dotyczącej przyporządkowania wierzchołków do tras występującego u rodziców oraz uporządkowania wierzchołków w każdej z tras. Działanie operatora rozpoczyna się od losowego wyboru podciągu wierzchołków pierwszego z rodziców, który to podciąg może rozpoczynać się i kończyć na dowolnym wierzchołku w dowolnej trasie. W przypadku szczególnym wybrany zostaje podciąg

wierzchołków jednej trasy. Następnie po skopiowaniu genotypu rodzica drugiego do potomka drugiego, usuwane są z potomka drugiego wierzchołki zawarte w kopionym podciągu rodzica pierwszego oraz ewentualne puste trasy postaci {1, 1}, które mogły powstać w procesie usuwania wierzchołków. Wklejanie do potomka drugiego rozpoczyna się od losowo wybranego miejsca wklejania. Cała operacja powtarza się analogicznie dla pierwszego potomka.

Algorytm implementuje dwie metody mutacji: VRPM1 (ang. VRP mutation 1) i VRPM2 (ang. VRP mutation 2), obie bardzo zbliżone do tradycyjnie wykorzystywanych metod dla problemów doboru trasy, zmodyfikowane jednak odpowiednio, w celu dopasowania ich do wykorzystywanej reprezentacji macierzowej tras.



Rys. 2 Przykład zastosowania operatora krzyżowania VRPC.

Operator mutacji VRPM1 z danym prawdopodobieństwem zmienia pozycję mutowanego wierzchołka. Działanie tego operatora jest zbliżone do działania operatora PBM (ang. position based mutation). Operator mutacji VRPM2 wybiera wierzchołek z danym prawdopodobieństwem i zamienia go miejscem z innym losowo wybranym wierzchołkiem. Jest to odmiana mutacji zbliżona do tradycyjnego operatora OBM (ang. order based mutation).

4. Zastosowanie algorytmu

Problem doboru trasy dla przedstawiciela handlowego Zakładu Konfekcji Odzieżowej „HALSAK” sformułować można następująco:

Dla danych 37 punktów zapotrzebowania zgłaszających zapotrzebowanie na produkty zakładu wyrażone w ilości sztuk Q_i , gdzie $i = 1, \dots, 37$ znaleźć zbiór tras zaopatrzeniowych od magazynu zlokalizowanego w Grudziądzu, w taki sposób aby zminimalizować całkowitą długość tras, nie przekraczając następujących warunków:

- maksymalnej ładowności pojazdu wynoszącej 150 sztuk wyrobu;

- b) maksymalnej trasy jaką jednego dnia może pokonać przedstawiciel handlowy wynoszącej 450 kilometrów;
- c) maksymalnego czasu pracy przedstawiciela handlowego wynoszącego 8 godzin dziennie, równoważnego z maksymalnym czasem otwarcia sklepów; przy zachowaniu następujących założeń:
 - przedstawiciel handlowy rozpoczyna i kończy trasę w mieście Grudziądz;
 - macierz odległości dla poszczególnych punktów zapotrzebowania jest symetryczna;
 - zapotrzebowanie żadnego z punktów zapotrzebowania nie przekracza maksymalnej ładowności pojazdu;
 - całkowite zapotrzebowanie każdego z punktów zapotrzebowania zaspokajane jest jednorazowo.

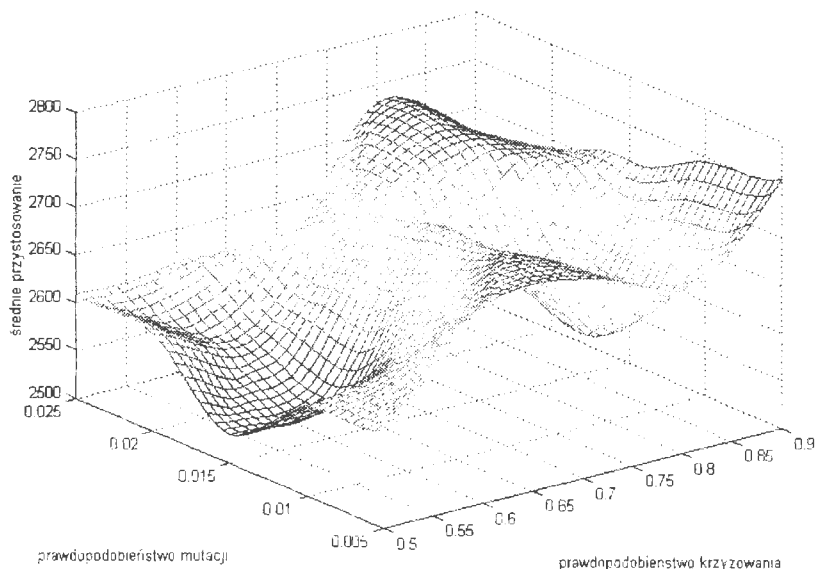
W powyższym sformułowaniu punkt zapotrzebowania rozumiany jest jako jedna z 37 miejscowości. W przypadku, gdy zakład kooperuje z większą niż jeden liczbą odbiorców w danej miejscowości, zapotrzebowanie zgłaszane przez każdego z odbiorców jest sumowane. Ponadto rozróżnieniu nie podlegają wyroby zakładu, które ze względu na swą specyfikę oraz sposób transportowania zajmują zbliżoną objętość w przestrzeni ładunkowej pojazdu.

5. Wyniki

Algorytm oszczędności jest algorytmem deterministycznym, który dla ściśle określonych danych wejściowych oraz wartości parametrów algorytmu w każdej iteracji zachowuje się jednoznacznie i zawsze generuje ten sam wynik. W przeciwieństwie do algorytmu oszczędności, algorytm ewolucyjny nie jest deterministyczny. Wielokrotne wykonywanie algorytmu dla identycznych danych wejściowych i wartości parametrów, pozwala na wyznaczenie pewnej wartości średniej, wokół której oscylują rozwiązania. Proces wyznaczenia optymalnego zestawu parametrów, dla którego otrzymywana wartość średnia jest najbardziej zbliżona do oczekiwanej, określany jest mianem strojenia algorytmu.

Problem dopuszczalności generowanych rozwiązań przez algorytm ewolucyjny jest ściśle związany z zagadnieniem doboru właściwych wartości mnożników funkcji kary. Generalnie zbyt duże wartości skutecznie obniżają jakość proponowanych przez algorytm rozwiązań, natomiast zbyt małe wartości prowadzą do sytuacji, gdy większość rozwiązań przekracza dowolne z narzuconych ograniczeń. Efektywna metoda doboru właściwych współczynników opiera się na początkowym ustaleniu ich wartości na poziomie minimalnym i stopniowemu ich zwiększaniu w przypadku częstego naruszania danego warunku ograniczającego przez proponowane rozwiązania.

Rys. 3 Średnie wartości przystosowań rozwiązań dla różnych wartości prawdopodobieństw krzyżowania i mutacji w postaci graficznej.



Funkcja kary pozwala także na badanie zasadności przyjętych wartości warunków ograniczających. Wykonanie obliczeń z wartościami mnożników nieznacznie mniejszymi od wartości idealnych pozwala często na znaczne poprawienie jakości rozwiązania przy równoczesnym minimalnym przekroczeniu ograniczeń. Proces doboru właściwych wartości mnożników przeprowadzany jest jednorazowo dla danych ograniczeń.

Na podstawie przeprowadzonych eksperymentów wykazano, iż najlepsze wyniki algorytm ewolucyjny proponuje dla prawdopodobieństwa mutacji równego 0,015 oraz prawdopodobieństwa krzyżowania równego 0,5. Ponadto eksperymenty przeprowadzono dla wartości warunku zakończenia równej 1, aby algorytm wykonał pełne 2500 iteracji dla każdego z przypadków. Wielkość populacji ustalona na 50 osobników jest kompromisem pomiędzy szybkością obliczeń, a jakością proponowanych rozwiązań. Zwiększenie liczby osobników w populacji do 100 osobników nieznacznie poprawiało otrzymywane wyniki równocześnie dwukrotnie wydłużając czas ich otrzymywania.

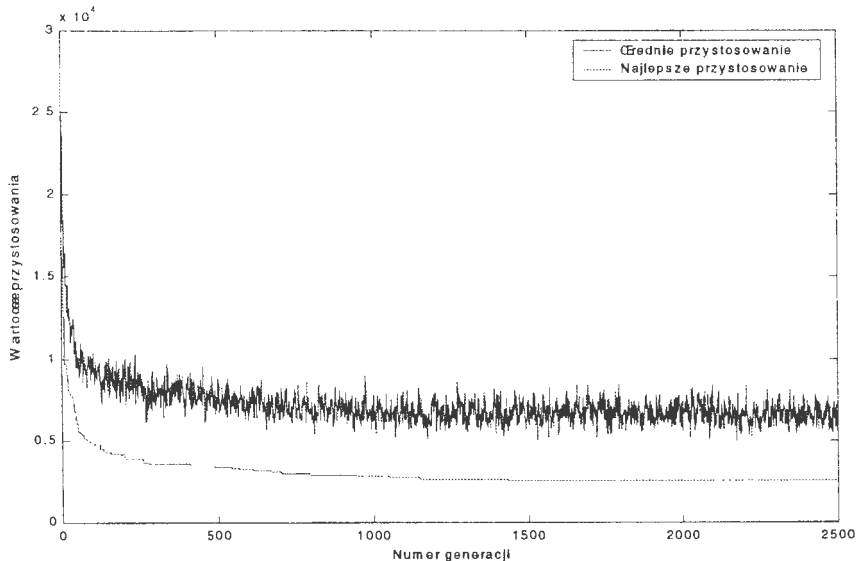
Interesujące rezultaty można uzyskać używając algorytmu oszczędności Clarka i Wrighta jako generatora populacji początkowej w algorytmie hybrydowym. Wadą takiego rozwiązania jest rozpoczęcie przetwarzania od lokalnego minimum wskazanego przez algorytm oszczędności, które nie koniecznie jest minimum globalnym. Wskazane w takim przypadku zwiększenie prawdopodobieństwa mutacji do wartości równej 0,05 pozwoliło w większości przypadków testowych na dalsze poprawienie jakości rozwiązania średnio w granicach nie przekraczających 3%.

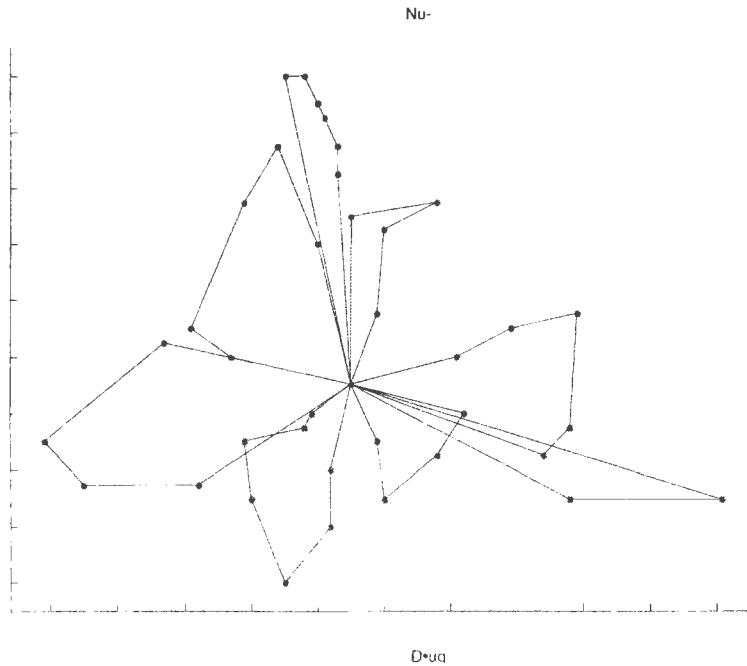
Tab. 1 Porównanie metod rozwiązywania problemów doboru trasy pojazdu.

Algorytm	Wynik [km]	Uwagi
oszczędności	2 447	błąd 0,08%
ewolucyjny	2 579	błąd 5,48%
hybrydowy	2 445	prawdopodobna wartość optymalna

Analiza jakości generowanych wyników dla testowych problemów doboru trasy o zmiennej liczbie miejscowości w granicach od 10 do 200, pozwala wysnuć wnioski o wyższości heurystycznego algorytmu oszczędności Clarka i Wrighta nad algorytmem ewolucyjnym. Algorytm heurystyczny generuje wyniki jakościowo lepsze średnio o 5% i ponadto pozwala na uzyskanie ich w wielokrotnie krótszym czasie. Informacje, zawarte w literaturze, jakoby algorytm oszczędności proponował daleko nie zadawalające wyniki dla nietypowych problemów, nie potwierdziły się. Algorytm doskonale dawał sobie radę z doбором trasy pojazdu z uwzględnieniem wielu ograniczeń, co jest jego niepowtarzalną zaletą. Jednakże do wad można zaliczyć brak możliwości badania zasadności przyjętych ograniczeń dla danego problemu, których często minimalne naruszenie pozwala na znaczne poprawienie jakości wyniku. Ponadto algorytm Clarka i Wrighta ma stosunkowo niekorzystną czasową złożoność obliczeniową rzędu $\Theta(n^3)$. Wynika to z faktu gwałtownego wzrostu rozmiaru tablicy oszczędności w miarę wzrostu liczby miejscowości. Dla większych i bardziej złożonych problemów lawinowy wzrost czasu przetwarzania może przyćmić zalety tego algorytmu.

Rys. 4 .Wartości najlepszego przystosowania i przystosowania średniego populacji dla rozwiązania o wartości przystosowania zbliżonej do wartości średniej przystosowań rozwiązań zaproponowanych przez algorytm ewolucyjny w postaci graficznej.



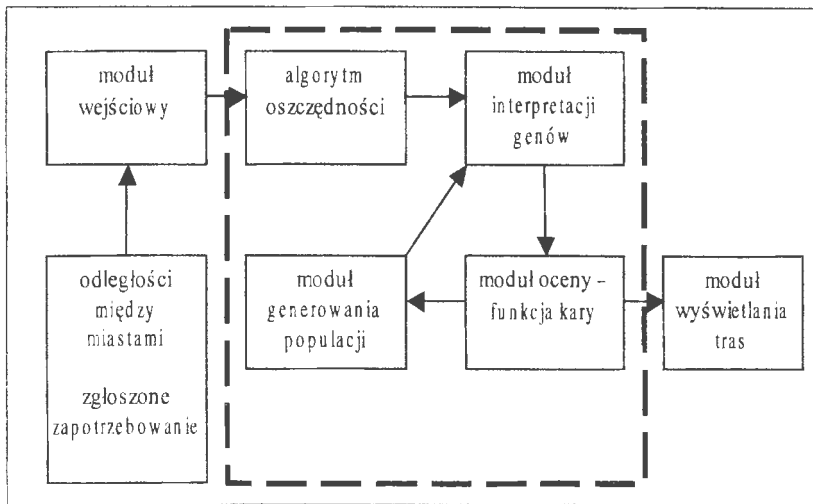


Rys. 5 Rozwiązanie zaproponowane przez algorytm hybrydowy.

Algorytm ewolucyjny dla tak skomplikowanego problemu jakim jest dobór trasy pojazdu generował wyniki znacznie odbiegające od oczekiwanych. Spowodowane jest to bezpośrednio dużą złożonością problematyki grupowania danych miejscowości w konkretnej trasie z uwzględnieniem zagadnień kolejnościowych i warunków ograniczających. Ponadto wygenerowanie rozwiązania prowadzącego do minimum lokalnego powoduje, iż w kolejnych iteracjach algorytm podąża w jego kierunku i po osiągnięciu minimum nie potrafi się z niego wydostać.

Algorytm ewolucyjny, jak zostało wyżej wspomniane, wymaga długiego czasu obliczeń już dla prostych problemów rzeczywistych. Jednakże jego czasowa złożoność obliczeniowa rzędu $\Theta(n)$ stanowi pewną przewagę nad algorytmem oszczędności. Można oczekiwać, iż dla bardzo złożonych problemów, będzie on proponował rozwiązanie w czasie krótszym od algorytmu Clarka i Wrighta.

Spośród rozwiązań zamieszczonych w opracowaniu jakościowo najlepsze wyniki dla problemu doboru trasy prezentuje algorytm hybrydowy oparty na działaniu algorytmu ewolucyjnego, dla którego populacja początkowa generowana jest przez algorytm oszczędności. Takie wykorzystanie algorytmu ewolucyjnego jest najbardziej uzasadnione. Ponadto algorytm ewolucyjny pozwala na kontrolowanie wyników prezentowanych przez algorytm oszczędności, który dla przypadków szczególnych może generować rozwiązania dalece odbiegające od optymalnych. Rozwiązanie zaproponowane przez algorytm ewolucyjny może w takim przypadku prezentować jakościowo lepszy rezultat działania.



Rys. 6 Architektura hybrydowego systemu doboru trasy pojazdu.

6. Podsumowanie

Problematyka sterowania i planowania produkcji jest niezwykle szerokim pojęciem mającym decydujące znaczenie dla przedsiębiorstw produkcyjnych pragnących uzyskać przewagę konkurencyjną na rynku. Ostatnie lata przemian gospodarczych wymusiły na firmach racjonalizację wykorzystania zasobów ludzkich i maszyn, co wprowadziło konieczność opracowania narzędzi pozwalających na optymalizację problemów sterowania i planowania produkcji o rzeczywistej złożoności.

Algorytmy oparte na technologii sztucznej inteligencji mimo wyników nieznacznie różniących się od rezultatów prezentowanych przez algorytmy heurystyczne są niewątpliwie niezwykle obiecującym narzędziem rozwiązywania problemów kombinatorycznych o wykładniczej złożoności obliczeniowej. Należy tu zaznaczyć, iż obecnie barierą dla efektywnego wykorzystania tej klasy metod pozostaje nadal niewystarczająca moc obliczeniowa współczesnych komputerów. W dobie niemal wykładniczego wzrostu możliwości systemów komputerowych, należy oczekiwać, iż w niedalekiej przyszłości algorytmy ewolucyjne oraz sztuczne sieci neuronowe pozwolą na stworzenie narzędzi doskonale radzących sobie z problemami NP-trudnymi.

Literatura

Dubey A., Crabbe R. (2000) *Answers to Questions about Artificial Intelligence*, <http://www.faqs.org/faqs/ai-faq/>.

- Goldberg D. (1998) *Algorytmy genetyczne i ich zastosowania*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Heitkoetter J., Beasley D. (2000) *The Hitch-Hiker's Guide to Evolutionary Computation*, <http://www.faqs.org/faqs/ai-faq/>.
- Korbicz J., Obuchowicz A., Uciński D. (1994) *Sztuczne sieci neuronowe. Podstawy i zastosowania*, Akademicka Oficyna Wydawnicza PLJ, Warszawa.
- Larson R., Odoni A. (1999) *Urban Operations Research*, Massachusetts Institute of Technology, http://web.mit.edu/urban_or_book/www.
- Lasecki J. (1994) *Zastosowanie algorytmów genetycznych i sieci neuronowych w rozwiązywaniu problemów optymalizacyjnych*, Praca magisterska, Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Gdańsk.
- Marshall D. (2000) *Artificial Intelligence II*, <http://www.cs.cf.ac.uk/Dave/AI2/>.
- Matthews J. (2000) *Experiments in VRP using satisfiability and genetic algorithms*, <http://www.owl.net.rice.edu/~jacobm/540/>.
- Michalewicz Z. (1999) *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Osowski S. (1996) *Sieci neuronowe w ujęciu algorytmicznym*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Rutkowska D., Piliński M., Rutkowski L. (1997) *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, Wydawnictwo Naukowe PWN, Warszawa.
- Sarle W. (2000) comp.ai.neural-nets Frequently Asked Questions, <http://www.faqs.org/faqs/ai-faq/>.
- Sarnatowicz T. (1998) *Rozwiązywanie kombinatorycznych zagadnień optymalizacyjnych przy użyciu metod uczenia i samoorganizacji*, Praca magisterska, Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki, Gdańsk.
- Syśło M., Deo N., Kowalik J. (1999) *Algorytmy optymalizacji dyskretnej*, Wydawnictwo Naukowe PWN, Warszawa.
- Tadeusiewicz R. (1993) *Sieci neuronowe*. Warszawa, Akademicka Oficyna Wydawnicza RM.
- Tadeusiewicz R. (1998) *Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami*, Akademicka Oficyna Wydawnicza PLJ, Warszawa.
- TSPBIB (1999) http://www.densis.fee.unicamp.br/~moscato/TSPBIB_home.html.
- TSPLIB (1998) <http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/>.
- Zawadzka L. (1994) *Metody ilościowe w organizacji i zarządzaniu. Część I, Część II* (1996), Wydawnictwo Politechniki Gdańskiej, Gdańsk.
- Żurada J., Barski M., Jędruch W. *Sztuczne sieci neuronowe*, Wydawnictwo Naukowe PWN, Warszawa.

ISSN 0208-8028
ISBN 83-85847-59-6

**W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy
prosimy o kontakt z Instytutem Badań Systemowych PAN
ul. Newelska 6, 01-447 Warszawa
tel. 837-35-78 w. 241 e-mail: bibliote@ibspan.waw.pl**