ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2002
Seria: AUTOMATYKA z. 136 Nr kol. 1556

Luiza DOLATA, Paweł KOMINEK
Politechnika Poznańska

WYKORZYSTANIE INFORMACJI DODATKOWYCH W ALGORYTMIE EWOLUCYJNYM DLA 2-WYMIAROWEGO NIEREGULARNEGO PROBLEMU ROZKROJU

Streszczenie. W pracy rozważany jest problem 2-wymiarowego nieregularnego rozkroju (2-DCP) minimalizujący długość taśmy. Ze względu na dużą złożoność obliczeniową do jego rozwiązania proponuje się zastosowanie algorytmu ewolucyjnego z lokalnym przeszukiwaniem (GLS) bazującym na istotnych cechach wskazanych przez eksperta. Wszystkie cechy sugerowane przez eksperta przed uwzględnieniem ich w algorytmie są zweryfikowane eksperymentalnie za pomocą testu korelacji. W pracy przedstawiono adaptację algorytmu GLS, a w szczególności dostosowanie algorytmu lokalnego przeszukiwania oraz operatora rekombinacji do rozważanego 2-DCP. Rozdział 5 zawiera eksperyment, w którym porównuje się działania standardowego algorytmu ewolucyjnego z algorytmem bazującym na proponowanych przez eksperta cechach.

THE USE OF ADDITIONAL INFORMATION IN EVOLUTIONARY ALGORITHM FOR SOLVING TWO DIMENTION IRREGULAR CUTTING PROMBLEM

Summary. The paper considers 2-dimention cutting problem (2-DCP) of minimizing the stripe length. Because of high computational complexity of the problem the use of evolutionary algorithm with local search (GLS) based on significant features is proposed. The paper presents adaptation of local search algorithm and recombination operator to the considered 2-DCP. The operator, which takes into account significant features proposed by expert is experimentally verified. Section 5 includes computational experiment, which compares the standard evolutionary algorithm EA with the GLS based on significant features.

1. Wprowadzenie

Różnorodność praktycznych problemów wywołuje wielkie zainteresowanie rozpoznawaniem natury tych problemów w celu opracowania efektywnych metod służących do ich rozwiązania. Większość praktycznych problemów optymalizacji kombinatorycznej zaliczana jest do klasy tzw. problemów NP-trudnych (Garey, Johnson 1979), dla których praktyczne zastosowanie dokładnych metod optymalizacji jest ograniczone jedynie do problemów o niewielkich rozmiarach. Wykazanie NP-trudności problemu optymalizacyjnego nie pociąga za sobą konieczności rezygnacji z poszukiwań prostego (wielomianowego) algorytmu rozwiązującego ten problem, tzn. znajdującego rozwiązanie o ekstremalnej wartości funkcji celu. Pozostają do wyboru dwie drogi poszukiwania rozwiązania. Pierwsza z nich polega na zastosowaniu algorytmu o złożoności wykładniczej, np. algorytm podziału i ograniczeń lub algorytmu programowania dynamicznego (Janiak 1999). Druga polega na znalezieniu wielomianowego algorytmu aproksymacyjnego, znajdującego rozwiązanie przybliżone (suboptymalne). Metody przybliżone, do których można zaliczyć m.in.: heurystyki, metaheurystyki i sieci neuronowe nie gwarantują uzyskania rozwiązania optymalnego, ale pozwalają na znalezienie „dobrego" rozwiązania w czasie akceptowalnym dla użytkownika. Dobrym przykładem algorytmów z zakresu rozwiązywania problemów kombinatorycznych mogą być algorytmy hybrydowe, które powstają wskutek połączenia algorytmów metaheurystycznych z algorytmami lokalnego przeszukiwania. Na przykład Johnson (1990) zastosował algorytm lokalnego przeszukiwania z symulowanym wyżarzaniem, a Freisleben, Merz (1996) połączyli algorytm lokalnego przeszukiwania z algorytmem genetycznym. Zadaniem algorytmu lokalnego przeszukiwania (ang. local search) jest znalezienie optimum lokalnego w obrębie sąsiedztwa, natomiast zadaniem algorytmu metaheurystycznego wytyczenie kierunku przeszukiwań.

W wyniku badań przeprowadzonych nad metaheurystykami zauważono, że wiele spośród trudnych obliczeniowo problemów cechuje tzw. zjawisko „globalnej wypukłości" (ang. big valley) (Boese 1994). Oznacza to, że bardzo dobre rozwiązania – optima lokalne – w większości są położone stosunkowo blisko siebie. Bliskość tę można zinterpretować jako nieprzypadkowe podobieństwo tych rozwiązań między sobą, a elementy wspólne rozwiązań za kluczowe w określeniu kierunku przeszukiwań.

Przykładem występowania pewnych wspólnych cech, charakteryzujących korzystne położenie w zbiorze rozwiązań, które można traktować jako zbiór reguł zapewniających dobre rozwiązania, może być praca Jaszkiewicz i in. (1999) czy też Jaszkiewicz, Kominek (2000). W pracach autorzy wymieniają pojedyncze łuki jak również fragmenty ścieżek (cykli) jako cechy, które powinny możliwie często występować w kolejnych rozwiązaniach. Informacje o istotnych cechach dobrych rozwiązań stanowią cenne repozytorium wiedzy, które powinno być wykorzystane do ukierunkowania poszukiwań, a zatem przyspieszenia procesu wyznaczania najlepszego rozwiązania. Wspomniana wiedza może pochodzić od eksperta (wiedza dziedzinowa) i opierać się np. na statystycznie istotnych cechach dobrych rozwiązań bądź też może być pozyskiwana automatycznie (uczenie maszynowe, ang. machine learning) na podstawie analizy historii przebiegu algorytmu (Kominek 2001). Wydaje się zatem, że efektywność wymienionych metod metaheurystycznych można poprawić, „wzmacniając" je wiedzą o problemie pozyskaną od eksperta i/lub wyindukowaną z historii przeszukiwania przestrzeni rozwiązań.

W pracy przedstawimy sposób rozwiązywania problemu 2-wymiarowego rozkroju (2-DCP) za pomocą algorytmu ewolucyjnego z lokalnym przeszukiwaniem (GLS), wykorzystującego wiedzą pochodzącą od eksperta i zweryfikowaną eksperymentalnie. Sformułowanie problemu zamieszczone jest w rozdziale 2. Rozdział 3 przedstawia cechy rozwiązań proponowane przez eksperta oraz weryfikację eksperymentalną tych cech. Opis algorytmu GLS oraz adaptację operatora rekombinacji do rozpatrywanego problemu przedstawia rozdział 4. Eksperyment obliczeniowy zawarty jest w rozdziale 5. Rozdział 6 zawiera podsumowanie.

2. Sformułowanie problemu

Rozważana jest wersja 2-DCP, w której obszar taśmy, z którego mają być wycinane figury o nieregularnych kształtach, jest prostokątem o ustalonej szerokości i nieograniczonej długości. W ogólności wycinane figury są dowolnymi wypukło-wklęsłymi wielokątami, które mogą również zawierać otwory. Kryterium stanowi długość wykorzystanej taśmy a tym samym powierzchnia użytej taśmy, która jest minimalizowana.

W celu rozwiązania problemu zastosowana została procedura Bottom Up-Left Justified, w której pierwszy element umieszczany jest w lewym górnym narożniku taśmy. Umieszczanie kolejnych elementów jest związane z ich przesuwaniem maksymalnie do góry i w lewo tak, aby elementy nie nachodziły na siebie. Z uwagi na format posiadanych danych konieczne było obliczanie pola powierzchni układanych elementów. Do obliczenia pola powierzchni zastosowaliśmy następujący wzór:

$$P = \tfrac{1}{2}\sum_i x_i \Delta y_i\,, \tag{1}$$

gdzie:

P – pole powierzchni figury,

i – kolejna para wierzchołków,

x_i – współrzędna x środka boku pomiędzy parą wierzchołków i,

Δy_i – przyrost względem współrzędnej y pomiędzy parą wierzchołków i.

Wzór można wykorzystać również dla figur posiadających otwór. W tym przypadku obliczane jest pole figury, a następnie pole otworu według tego samego wzoru.

3. Analiza rozwiązań

Z doświadczenia wiemy, że dla wielu problemów kombinatorycznych, np.: 0/1 problemu plecakowego (Jaszkiewicz 2000), problemu komiwojażera (Freisleben, Merz 1996), problemu szeregowania zadań (Józefowska i in. 1998, Józefowska, Węglarz 1999, Janiak 1999) algorytmy metaheurystyczne działają lepiej niż algorytmy losowego przeglądu. Ponadto wśród badaczy algorytmów metaheurystycznych (bazujących na algorytmie lokalnego przeszukiwania) panuje przekonanie, że dobra reprezentacja oraz dobra struktura sąsiedztwa gwarantuje dobre działanie tych algorytmów.

Przyjmuje się zatem hipotezę, że musi istnieć jakaś szczególna reguła bądź cecha rządząca ułożeniem rozwiązań, na której to w niejawny sposób opierają swoje działanie popularne metaheurystyki.

W przypadku, kiedy zachodzi potrzeba badania, czy pomiędzy badanymi zmiennymi zachodzą jakieś zależności, często stosowanym testem jest test korelacji. Zależność korelacyjna dwóch zmiennych polega na określeniu zmiany średniej wartości jednej zmiennej w przypadku zmiany drugiej zmiennej. Zależność ta jest szczególnym przypadkiem zależności stochastycznej, która występuje wtedy, gdy wraz ze zmianą jednej zmiennej zmienia się rozkład prawdopodobieństwa drugiej zmiennej.

Jeśli między badanymi zmiennymi nie ma związku stochastycznego, to nie ma również związku korelacyjnego. W celu określenia stopnia zależności między badanymi zmiennymi można posłużyć się współczynnikiem korelacji. Współczynnik ten mówi o sile zależności między badanymi zmiennymi.

W rozdziale zamieszczono wyniki badań zależności pomiędzy wartością funkcji celu (funkcji dopasowania) rozwiązania $\mathbf{x}$ a średnim podobieństwem $\bar{s}(\mathbf{x})$ do innych ze zbioru C, ale nie gorszych rozwiązań, wyznaczanym na podstawie następującego wzoru:

$$\hat{s}(x) = \frac{\sum_{y \in C | f(y) \le f(x)} s(x,y)}{|C|}, \tag{2}$$

gdzie $s(x,y)$ - jest podobieństwem x i y.

Badania przeprowadzono na różnych instancjach 2-DCP o różnych liczbach układanych figur. Dla każdej instancji wygenerowano 50 lokalnych minimów. Tablica 1 przedstawia średnie wartości korelacji dla tych instancji.

Tablica 1

Rezultaty testów korelacyjnych dla 50 przykładowo wygenerowanych optimów lokalnych problemu 2-DCP

	Cecha podobieństwa			
	średnia wielkość pól figur ułożona od początku taśmy	średnia wielkość pól figur ułożona od końcu taśmy	średnia liczba par figur tak samo ułożonych na taśmie	średnia liczba par figur powyżej średniej tak samo ułożonych na taśmie
Korelacja z funkcją dopasowania	-0.23	-0.22	-0.30	-0.41

Z przedstawionych wyników można zauważyć, że najwyższą korelację otrzymano dla cechy, jaką jest liczba par figur powyżej średniej tak samo ułożonych na taśmie. Cecha ta wynika bezpośrednio z liczby par figur tak samo ułożonych na taśmie. W związku z tym kolejnym krokiem jest wykorzystanie tej cechy do rekombinacji rozwiązań. Przykładowy algorytm operatora rekombinacji zachowującego pary figur zdefiniowano w rozdziale 4.3.

4. Adaptacja algorytmu ewolucyjnego do rozwiązania problemu rozkroju

Do znalezienia suboptymalnego rozwiązania proponuje się algorytm ewolucyjny z lokalnym przeszukiwaniem (GLS). Zadaniem algorytmu lokalnego przeszukiwania jest zoptymalizowanie osobników po przeprowadzonej operacji rekombinacji (standardowo w algorytmie genetycznym operacji krzyżowania i mutacji). Zatem populacja rozwiązań, na której operuje algorytm ewolucyjny, składa się zawsze z lokalnych optimów. Procedurę algorytmu GLS wykorzystywaną w pracy można przedstawić następująco:

```
Procedure 1 ALGORYTM_GLS;
  begin
    k :=0
    for i:=1 to N do
    begin
      UTWÓRZ_NOWE_LOSOWE_ROZWIĄZANIE z
      ZNAJDŹ_LOKALNE_OPTIMUM y' ∈ Z_z^z(I)
      DODAJ y' do P
    end;
    repeat
      WYBIERZ_LOSOWO_DWA_ROZWIĄZANIA z1 i z2 z P
      REKOMBINUJ z1 i z2 W CELU_OTRZYMANIA z3 ;
      ZNAJDŹ_LOKALNE_OPTIMUM y' ∈ Z_z^z3(I)
      IF y' JEST_LEPSZY_NIŻ_NAJGORSZY_Z P I_INNY THEN;
            WYMIEŃ_NAJGORSZEGO_OSOBNIKA;
       k := k+1
    UNTIL WARUNEK_STOPU ( k > k_max )
  end;
```

gdzie,

P – oznacza populację,

N – rozmiar populacji,

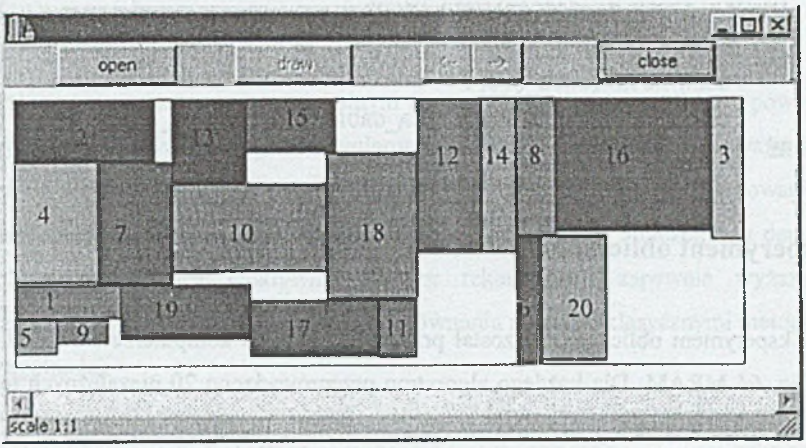
k_{max} – ustalona liczba iteracji.

$Z_z^z(I)$ – sąsiedztwo rozwiązania z dla instancji I

4.1. Sposób kodowania

Sposób kodowania rozwiązania nie powinien być uzależniony jedynie od intuicji czy też naturalnego podejścia. Kodowanie powinno być przede wszystkim efektywne z punktu widzenia procedury lokalnego przeszukiwania i operatora rekombinacji.

Rozwiązanie dla 2-DCP jest zdefiniowane jako lista priorytetowa, na której znajdują się figury w kolejności ich układania na taśmie. Przykład takiego rozwiązania składającego się z 20 figur i odpowiadającej listy priorytetowej przedstawia rys.1.



LISTA PRIORYTETOWA [2- 4-1-5-9-7- 9-13-10-15-17-18-11-12-14-8-6-20-16-3]

Rys.1. Przykładowe rozwiązanie

Fig.1. Solution example

4.2. Lokalna optymalizacja

Algorytm lokalnego przeszukiwania wykorzystywany dla tego problemu bazuje na algorytmie zachłannego lokalnego przeszukiwania (ang. Gready Local Search). Polega na znalezieniu rozwiązania lokalnie optymalnego w obrębie sąsiedztwa. Rozwiązanie sąsiednie generowane jest poprzez losowy wybór dwóch figur i zamianie ich miejscami na liście priorytetowej.

LISTA PRIORYTETOWA [2- 4-1-5-9-7- 9-13-10-15-17-18-11-12-14-8-6-20-16-3]

LISTA PRIORYTETOWA PO ZAMIANIE [2- 4-1-5-9-7- 9-13-10-12-17-18-11-15-14-8-6-20-16-3]

4.3. Rekombinacja rozwiązań

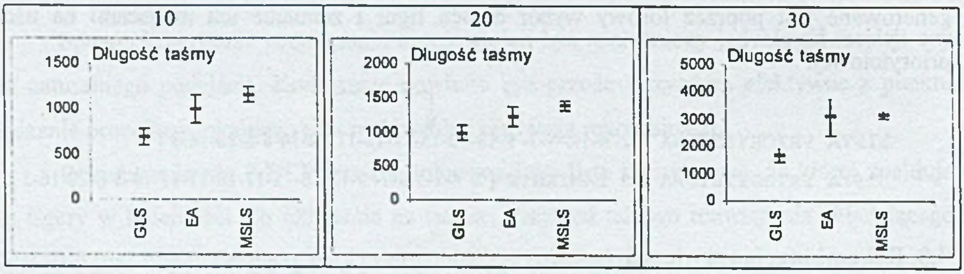
Wszystkie miary podobieństwa proponowane przez eksperta i testowane w rozdziale 3 są bardziej lub mniej skorelowane z jakością rozwiązania. Operator rekombinacji zdefiniowany w tej pracy zachowuje w generowanym potomku pary figur wspólne dla obu wcześniej wybranych rodziców i jest sformułowany w następujący sposób:

```
Procedure OPERATOR_REKOMBINACJI
 begin
   UTWÓRZ_GRUPY_Z_FIGUR_KTÓRE_WYSTĘPUJĄ_
           U_RODZICÓW_W_TEJ_SAMEJ_KOLEJNOŚCI
```

```
      UTWÓRZ_GRUPY_Z_ POZOSTAŁYCH_FIGUR
    for KAŻDEJ_GRUPY do
         OCEŃ_WSTAWIENIE_GRUPY
         WSTAW_DO_POTOMKA_NAJLEPSZĄ_GRUPĘ W_NAJLEPSZE_MIEJSCE
    end
```

5. Eksperyment obliczeniowy

Eksperyment obliczeniowy został przeprowadzony na komputerze PC klasy Pentium 150 MHz, 64 MRAM. Dla każdego algorytmu przeprowadzono 20 niezależnych testów dla każdej instancji. Rysunek 2 przedstawia rezultaty eksperymentu dla algorytmu EA, GLS oraz MSLS. EA oznacza standardowy algorytm ewolucyjny, w którym operator rekombinacji polegał na losowym wyborze miejsca wymiany figur między rozwiązaniami (miejsca krzyżowania). GLS jest proponowanym algorytmem GLS z operatorem rekombinacji opisanym w rozdziale 4.3 i algorytmem lokalnego przeszukiwania. Dodatkowo, rezultaty EA i GLS zostały porównane z algorytmem wielokrotnego lokalnego przeszukiwania (ang. multiple start local search – MSLS). Testowane instancje zawierały figury czworokątne bez otworów. Wielkość instancji wynosiła 10, 20 i 30. Czas obliczeń w każdym przypadku wynosił 120 sekund.



Rys.2. Graficzne porównanie rezultatów. Każdy wykres przedstawia inną instancję problemu i zawiera wyniki porównań algorytmów (od lewej) GLS, EA, MSLS w funkcji dlugości taśmy. Z uwagi na maly rozrzut wyników niektóre slupki błędów mogą być niewidoczne

Fig. 2. Graphical comparison of the results. Each chart corresponds to a different instance. Each chart contains three box plots representing the distribution of stripe length for (from left to right) GLS, EA, and MSLS. Note that the box plots are in many cases practically invisible because of the very low dispersion of the results

6. Podsumowanie

Algorytm genetyczny z lokalnym przeszukiwaniem został z powodzeniem wykorzystany do 2-wymiarowego nieregularnego problemu rozkroju. Zastosowano podejście bazujące na technice sugerowania i weryfikacji cech rozwiązań w celu zastosowania ich do tworzenia operatora rekombinacji. Przeprowadzone eksperymenty obliczeniowe demonstrują, że GLS wykorzystujący efektywny operator rekombinacji zapewnia wyższą jakość rozwiązania w relatywnie krótkim czasie w porównaniu z innymi klasycznymi metodami.

W szczególności:

- dokonano zweryfikowania zależności pomiędzy proponowanymi cechami podobieństwa rozwiązań i wartością funkcji celu,
- zdefiniowano miarę podobieństwa pomiędzy rozwiązaniami,
- przedstawiono ideę uwzględnienia istotnych cech rozwiązań w operatorach tworzenia nowego rozwiązania, w szczególności:
  - zaproponowano procedurę algorytmu GLS, w której każde nowe rozwiązanie jest optymalizowane algorytmem lokalnego przeszukiwania,
  - zaproponowano procedurę rekombinacji rozwiązań, w której nowe rozwiązanie zawiera wspólne cechy rodziców, zapewniające tworzenie relatywnie dobrego rozwiązania.

Podziękowania

Praca ta została wykonana i finansowana w ramach projektu badawczego KBN nr 8 T11F 006 19 oraz w ramach subsydium dla uczonych nr 4/2001, finansowanego przez Fundację Nauki Polskiej.

LITERATURA

1. Boese K., Kahng A., Muddu S.: A new adaptive multistart technique for combinatorial global optimization. Operations Research Letters, vol. 16, pp. 101-113, 1994.
2. Dolata L.: Przetwarzanie wektorowe dla problemu rozkroju. Praca magisterska, Politechnika Poznańska, Poznań 2000.

3. Freisleben B., Merz P.:, A genetic local search algorithm for travelling salesman problem. In H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (eds.), Proceedings of the 4th Conference on Parallel Problem Solving fram Nature- PPSN IV, pp. 890-900, 1996.

4. Garey M., Johnson D.: Computers and intractability: A guide to the theory of NP-completeness, Freeman, San Francisco, Calif, 1979.

5. Janiak A.: Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów. Akademicka oficyna wydawnicza PLJ, Warszawa 1999.

6. Jaszkiewicz A., Kominek P.: Developing efficient genetic local search type heuristics on the basis of global convexity tests - vehicle routing example, Research Report RA-001/2000, Institute of Computing Science, Poznań 2000.

7. Johnson D.S.: Local Optimization and the Traveling Salesman Problem. Annual Int. Colloqium on Automata, Languages and Programming, s.116-161, 1990.

8. Józefowska J., Mika M., Różycki R., Waligóra G., Węglarz J.: Minimalizacja maksymalnego opóźnienia w dyskretno-ciągłych problemach szeregowania – algorytmy heurystyczne, Zeszyty Naukowe Polit. Śl., ser.: Automatyka, z. 123, s.221-232, Gliwice 1998.

9. Józefowska J., Węglarz J.: Approximation algorithms for some discrete-continuous scheduling problems, Bulettion of the Polish Academy of Sceince, Technical Sciences, 41, 391-399, 1993.

10. Kominek P.: Zastosowanie algorytmów metaheurystycznych sterowanych wiedzą do rozwiązywania złożonych problemów optymalizacji kombinatorycznej. Rozprawa doktorska, Politechnika Poznańska, Poznań 2001.

11. Kominek P.: Operatory genetyczne + uczenie maszynowe = efektywne algorytmy ewolucyjne, Zeszyty Naukowe Polit. Śl., ser. Automatyka, z. 125, s.165-175, Gliwice 1998.

12. Walkowiak R.: Sekwencyjne i współbieżne algorytmy dla problemu rozkroju, Rozprawa doktorska, Politechnika Poznańska, Poznań 1996.

Recenzent: Prof. dr hab. inż. Józef Grabowski

Abstract

In this paper we study the 2-dimential irregular cutting problem. As the problem is NP-hard there exists no exact algorithm to minimize the stripe length in polynomial time unless P=NP. Although the metaheuristic procedures do not guarantee the optimal solution, they give good suboptimal solutions in relatively short time. In the paper an outline of evolutionary algorithm with local search (GLS) is presented with a description of local search algorithm and recombination operator. GLS algorithms define only a general scheme of the

calculations. This general scheme has to be customized for a given combinatorial problem. The customization consists in defining the way new solutions are obtained.

The new solution in GLS algorithm is obtained by recombination operator, which takes into account significant features. All features of the solution proposed by expert are experimentally verified. Section 5 includes computational experiment, which compares the standard evolutionary algorithm EA with the GLS based on significant features.

Some relevant conclusions are included in the last part of the paper.