ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1988
Seria:AUTOMATYKA z.94 Nr kol.970

Jacek Błażewicz
Instytut Automatyki
Politechniki Poznańskiej

Sławomir Osiak
Pozn. Spółdzielnia Informatyki

Bolesław Soniewicki
Centralny Ośrodek Oświaty
i Postępu w Rolnictwie
Oddział w Poznaniu

Rafał Walkowiak
Instytut Automatyki
Politechniki Poznańskiej

SYSTEM WSPOMAGANIA PODEJMOWANIA DECYZJI W ZAGADNIENIACH ROZKROJU

Streszczenie. W pracy przedstawiono system wspomagania podejmowania decyzji w zagadnieniu rozkroju polegającym na znajdowaniu takiego upakowania elementów w pojemniku, aby straty materiału były jak najmniejsze. Opracowany system wybiera najlepsze z trzech generowanych rozwiązań, a użytkownik ma możliwość interakcyjnej komunikacji z systemem celem ulepszenia poszukiwanego rozwiązania.

1. Wstęp.

Problem rozkroju dwuwymiarowego materiału na elementy o dowolnym kształcie w taki sposób, aby zminimalizować powstające przy tym straty materiału, pojawia się w wielu sytuacjach praktycznych. Przykładem może być rozkrój blach stalowych przy produkcji kadłubów statków, samochodów itp., rozkrój materiału przy produkcji odzieży, rozkrój płyt drewnianych przy produkcji mebli itp. Ponieważ problem ten jest problemem bardzo złożonym, równolegle z próbami automatyzacji generowania rozwiązań dla powyższego problemu podjęto badanie jego prostszych wersji, zwanych w literaturze regularnymi problemami pakowania. Problemy pakowania są równoważne problemowi rozkroju, gdyż zamiast rozcinania materiału na elementy wymagają upakowania w pojemniku odpowiadającym materiałowi odpowiednich elementów. W literaturze sformułowano jednowymiarowy i dwuwymiarowy regularny problem pakowania. W przypadku jednowymiarowym polega on na upakowaniu do minimalnej liczby pojemników o jednostkowej szerokości i jednakowej wysokości odpowiedniego zbioru elementów także o jednostkowej szerokości oraz o wysokości nie przekraczającej wysokości pojemników.

W przypadku dwuwymiarowym wyróżnia się kilka wersji problemu regularnego. Pierwsza z nich polega na upakowaniu prostokątnych elementów do prostokątnego pojemnika o zadanej szerokości i nieograniczonej wysokości w taki sposób, aby zminimalizować wysokość jego części zajętej przez elementy. Druga wersja problemu polega na upakowywaniu prostokątnych elementów do szeregu prostokątnych pojemników o ograniczonych rozmiarach w taki sposób, aby zminimalizować liczbę pojemników potrzebnych do zapakowania wszystkich elementów. W ogólności problem pakowania jest już NP-zupełny nawet w przypadku jednowymiarowym. Zatem w celu rozwiązania problemu opracowano szereg algorytmów aproksymacyjnych i zbadano ich zachowanie się w najgorszym przypadku.

W bardziej złożonych sytuacjach, gdy dwuwymiarowe elementy mają nieregularne kształty (tzn. są dowolnymi figurami płaskimi), algorytmów o gwarantowanej dokładności nie udało się opracować. Pozostaje zatem jedynie konstrukcja algorytmów heurystycznych i eksperymentalna ocena ich dokładności.

W niniejszej pracy rozpatrzono powyższy problem. Dla regularnego przypadku jedno- i dwuwymiarowego przedstawiono najważniejsze istniejące algorytmy aproksymacyjne wraz z oszacowaniami. Natomiast dla przypadku dwuwymiarowego nieregularnego opisano skonstruowany system wspomagania podejmowania decyzji, wykorzystujący najbardziej adekwatne metody dla zadanych parametrów rozważanego zagadnienia rozkroju. W pracy rozdział 2 zawiera sformułowanie problemu, rozdział 3 omawia złożoność obliczeniową algorytmów, rozdział 4 obejmuje algorytmy przybliżone o gwarantowanej dokładności. W rozdziale 5 omówione zostały podstawy algorytmów przystosowanych do rozwiązywania nieregularnego zagadnienia rozkroju, spotykane w literaturze. Rozdział 6 obejmuje opis opracowanego na podstawie powyższych algorytmów systemu wspomagania podejmowania decyzji oraz wyniki badań testowych.

2. Sformułowanie problemu

Klasyczna postać jednowymiarowego problemu pakowania jest następująca. Dana jest lista elementów

$$L = (a_1, a_2, \dots , a_n).$$

Każdy z elementów posiada określoną wysokość $0<R(a_i)\le 1$, $i=1,\dots,n$. Dana jest także nieograniczona liczba pojemników o jednostkowej wysokości. Należy wszystkie elementy upakować do pojemników w taki sposób, aby liczba użytych pojemników była minimalna.

Powyższy problem łatwo przetransformować w problem rozkroju "prętów" o jednostkowej długości na elementy.

W dwuwymiarowym problemie pakowania dane są prostokąty o rozmiarach AxB. Wyróżnia się zadania pakowania elementów regularnych i nieregularnych.

W pierwszym przypadku elementy $a_1, a_2, \ldots, a_n$ są prostokątami o zadanych wymiarach $R(a_i) \times S(a_i)$, $0 < R(a_i) \le \min(A,B)$, $0 < S(a_i) \le \min(A,B)$. W drugim przypadku do prostokątnych pojemników pakuje się elementy o dowolnych kształtach (dowolne figury płaskie). Jedynym ograniczeniem jest tutaj często brak możliwości uwzględnienia ubytków w materiale elementu, takich jak wszelkiego rodzaju szczeliny i otwory. W literaturze spotyka się także sformułowania narzucające wypukłość pakowanych elementów, co często ułatwia konstruowanie algorytmów pakowania. W większości przypadków dąży się do znalezienia takiego upakowania, które minimalizuje liczbę zajętych pojemników. (Czasami sprawdza się, czy dany zbiór elementów można zapakować do określonego pojemnika.)

3. Złożoność obliczeniowa algorytmów pakowania

Podstawowymi pojęciami w teorii złożoności obliczeniowej algorytmów są definicje algorytmów efektywnych (wielomianowych) i algorytmów nieefektywnych (wykładniczych). W celu zapoznania się z podstawami tej teorii odsyłamy do literatury (np. [10]).

Poszukiwanie efektywnych algorytmów umożliwiających rozwiązywanie wielu problemów doprowadziło do wyodrębnienia klasy problemów NP-trudnych. Przyjmuje się, że dla problemów należących do tej klasy nie istnieją wielomianowe algorytmy optymalne. Dlatego dla problemów z tych klas uzasadnione jest zastosowanie wielomianowych algorytmów przybliżonych. W pracy [10] pokazano, że już jednowymiarowy problem pakowania należy do klasy problemów NP-trudnych, co oznacza, że do klasy tej należą także wszystkie problemy dwuwymiarowe. Dlatego w dalszej części pracy skoncentrujemy się na zastosowaniu do rozwiązywania problemów pakowania wielomianowych algorytmów przybliżonych.

Oprócz złożoności obliczeniowej podstawowym parametrem charakteryzującym algorytmy przybliżone jest oszacowanie ich dokładności. Wyraża ono stosunek wartości rozwiązania heurystycznego, uzyskanego w najgorszym przypadku do rozwiązania optymalnego. Wyróżnia się oszacowania :

- bezwzględne

$$A(L) \le \alpha' OPT(L)$$

gdzie $A(L)$ – rozwiązanie otrzymane za pomocą algorytmu przybliżonego dla danej listy L elementów,
$OPT(L)$ – rozwiązanie optymalne dla tej samej listy L,
α' – oszacowanie bezwzględne,

- asymptotyczne

$$A(L) \le \alpha'' OPT(L) + \beta$$

gdzie α'' – oszacowanie asymptotyczne,
β – stała.

4. Algorytmy przybliżone stosowane do rozwiązywania regularnych problemów pakowania.

Podstawowymi algorytmami pakowania dla przypadku jednowymiarowego są algorytmy FF (First Fit) i FFD (First Fit Decreasing) [10] (algorytmy BF (Best Fit) i BFD (Best Fit Decreasing) posiadają takie same oszacowania dokładności jak algorytmy FF i FFD). Algorytm FF pakuje kolejno elementy z listy L do pierwszego pojemnika, w którym się on zmieści. Oszacowanie dokładności tego algorytmu jest następujące [10] :

$$FF(L) \le \frac{17}{10} OPT(L) + 1.$$

Algorytm FFD(L) jest wersją algorytmu FF dla listy L elementów uporządkowanych w kolejności ich nie rosnących wag. Jego oszacowanie jest następujące [10] :

$$FFD(L) \le \frac{11}{9} OPT(L) + 3$$

Dla regularnego problemu dwuwymiarowego opracowano algorytmy będące uogólnieniem powyższych algorytmów. Dla problemu pakowania prostokątów do pojemnika o nieograniczonej wysokości opracowano algorytmy zorientowane na tworzenie poziomów pakowania. Wymagają one wstępnego uporządkowania elementów listy L zgodnie z nie rosnącą wartością wysokości elementów. Są to następujące algorytmy :

1. Algorytm NFDH (Next Fit Decreasing Height).

Algorytm ten [5] pakuje elementy z listy L na aktualnym poziomie rozpoczynając od skrajnie lewej wolnej pozycji. Jeśli na aktualnym poziomie brak miejsca, to definiuje się nowy poziom. Oszacowanie dokładności tego algorytmu jest następujące :

$$NFDH(L) \le 2\ OPT(L) + 1.$$

2. Algorytm FFDH (First Fit Decreasing Height)

Algorytm ten [5] umieszcza kolejny element z listy L na najniższym poziomie, w którym się on mieści. Jeśli brak miejsca na poziomach istniejących, to definiuje się nowy poziom. Uzyskano dla tego algorytmu następujące oszacowanie dokładności:

$$FFDH(L) \le \frac{17}{10} OPT(L) + 7.3$$

3. Algorytm UD (Up-Down).

Powstał on [4] w wyniku analizy algorytmów NFDH i FFDH. Oszacowanie dokładności tych algorytmów jest złe, gdy szerokość pakowanych elementów jest duża w stosunku do szerokości pojemnika. Elementy o szerokości nie przekraczającej $\frac{1}{5}$ traktuje on tak samo jak algorytm NFDH. Dla pozostałych elementów stosuje kilka strategii pakowania w zależności od szerokości elementów. Jest to algorytm o najlepszym znanym oszacowaniu dla powyższego problemu. Oszacowanie to częściowo zależne jest od maksymalnej wysokości pakowanych elementów H :

$$UD(L) \le \frac{5}{4}\, OPT(L) + \frac{53}{8}\, H.$$

Dla problemu pakowania prostokątnych elementów do szeregu prostokątnych pojemników opracowano algorytm hybrydowy HFF [8]. Składa się on z zastosowanych kolejno algorytmów FFDH i FFD.

W pierwszej kolejności algorytm ten rozwiązuje problem pakowania elementów do jednego pojemnika o nieograniczonej wysokości. Następnie każdy z utworzonych poziomów traktowany jest jako element w problemie pakowania jednowymiarowego. Do rozwiązania tego problemu stosuje się algorytm FFD. Uzyskane oszacowanie dla algorytmu hybrydowego jest następujące :

$$HFF(L) \le \frac{17}{8}\, OPT(L) + 5.$$

5. Pakowanie dwuwymiarowych elementów o dowolnych kształtach

Ze względu na wielkie znaczenie praktyczne tego problemu, już w latach sześćdziesiątych rozpoczęto prace nad budową algorytmów umożliwiających automatyzację prac związanych z rozkrojem prostokątnego materiału na elementy o nieregularnych kształtach zapewniających możliwie duże oszczędności rozcinanego materiału. Pierwszy algorytm zaproponował R.C. Art. [3]. Algorytm ten umożliwia pakowanie elementów wypukłych o obwodzie aproksymowanym odcinkami, których współrzędne końców stanowią dane problemu. Nie ma tutaj możliwości uwzględnienia obrotów elementów. Algorytm ten za pomocą prostych reguł ustala kolejne położenia wszystkich elementów na materiale (w pojemniku).

Kolejne dwie propozycje algorytmów pakowania nieregularnych elementów pojawiły się w roku 1969. Jednym z nich jest algorytm Adamowicza [1], drugim natomiast algorytm Gurela [11].

Podejście zaproponowane przez M. Adamowicza jest bardzo ogólne - konstruuje rozwiązania optymalne z dowolną dokładnością. Zadaniem algorytmu jest rozwiązanie problemu, gdy do jego parametrów, obok współrzędnych obszaru i elementów oraz ograniczeń na nienakładanie się elementów, dołącza się dodatkowe ograniczenia liniowe, logiczne i geometryczne. Metoda rozwiązując problem wykorzystuje programowanie liniowe. Wyniki uzyskuje się w drodze iteracyjnego zastosowania procedury składającej się z dwóch faz. Pierwsza część ma za zadanie znaleźć rozwiązanie (będące zbiorem elementów) które spełnia liniowe zależności i niesprzeczne ograniczenia, minimalizujące liniową funkcję celu. Faza druga (geometryczna) umożliwia wykazanie, czy zbiór elementów uzyskany w pierwszej fazie może być rozmieszczony w pojemniku bez przekroczenia ograniczeń. Jeśli zbiór nie może być rozmieszczony, to powtarza się pierwszą fazę, z dołączeniem dodatkowych ograniczeń.

Algorytm będący propozycją Gurela wykorzystuje spostrzeżenie, że najczęściej tak można dobrać pakowane elementy, że utworzą one stosy elementów o podobnej długości. Wyróżnia się tutaj dwa stosy brzegowe, pakowane przy krawędziach materiału. Pozostałe stosy nazywane są wewnętrznymi. Tworzy się je w ten sposób, aby straty materiału między elementami były możliwie małe. Po utworzeniu każdego stosu rozmieszcza się go na materiale w taki sposób, aby także minimalizować powstające przy tym straty. Algorytm umożliwia także uwzględnienie obrotów pakowanych elementów o dowolny kąt, co jednak pociąga za sobą zwiększenie czasu wykonywania obliczeń. W celu uproszczenia algorytmu zaproponowano, aby elementy o małej powierzchni wyłączyć z procedury automatycznego pakowania i dołączać je do upakowania w sposób interakcyjny, wykonywany przez operatora, ponieważ najczęściej, w trakcie pakowania dużych elementów powstaje dostatecznie dużo miejsca wystarczającego do upakowania elementów mniejszych.

W roku 1980 zaproponowano jeszcze jedno podejście do problemu rozkroju materiału na elementy o nieregularnych kształtach. Jest to algorytm poszukiwania heurystycznego oparty na metodzie przeszukiwania przestrzeni rozwiązań wywodzącej się z dziedziny sztucznej inteligencji. Rozwiązywanie problemu może być w tym przypadku rozpatrywane jako proces poszukiwania ścieżki w grafie, którego węzły symbolizują stany, natomiast łuki symbolizują operatory, prowadzącej ze stanu (wierzchołka) początkowego do stanu (wierzchołka) końcowego. Ze względu jednak na liczbę obliczeń i ich koszt najczęściej wykonywany jest tylko przegląd mocno ograniczony. Przegląd ten sterowany jest za pomocą funkcji kosztu przejścia z jednego stanu do drugiego. Funkcja ta składa się z dwóch elementów : kosztu przejścia ze stanu początkowego do stanu aktualnego i kosztu przejścia ze stanu aktualnego do stanu końcowego. Pierwszy ze składników funkcji kosztu można wyznaczyć, natomiast drugi ze składników może być tylko zgrubnie szacowany. Przyjęto w algorytmie, że kosztem przejścia z jednego stanu do innego, co oznacza dołączenie nowego elementu do zbioru elementów już upakowanych, jest całkowita suma strat materiału, którą generuje dołączony element. Straty oblicza się tutaj jako różnicę pola zajętej części materiału po dołączeniu elementu i pola zajętej części materiału przed dołączeniem elementu. Drugi ze składników funkcji kosztu przyjęto szacować jako odpowiedni procent sumy pól niezaplanowanych elementów.

6. System wspomagania podejmowania decyzji w zagadnieniach rozkroju.

Idee zawarte w trzech ostatnich podejściach były podstawą opracowania i oprogramowania systemu wspomagania podejmowania decyzji w zagadnieniu rozkroju materiału na figury o nieregularnych kształtach. Przyjęto wszystkie trzy metody, ponieważ mogą być one stosowane dla różnych sformułowań zagadnienia rozkroju, i ponieważ przeprowadzone badania nie umożliwiły wybrania jednej z nich generującej zdecydowanie lepsze rozwiązania od innych dla takich zadań testowych, które mogą być rozwiązywane przez wszystkie algorytmy.

Program oparty na algorytmie Adamowicza umożliwia znalezienie odpowiedzi na pytanie, czy dany zestaw elementów można wykroić z materiału o ustalonych rozmiarach. Rozwiązuje on więc zadanie decyzyjne. Jeśli odpowiedź jest pozytywna, to możliwy jest wydruk wyznaczonego upakowania. W przypadku odpowiedzi negatywnej, możliwa jest zmiana parametrów i ponowne wykonywanie obliczeń.

Program oparty na idei zamieszczonej w pracy Gurela umożliwia rozmieszczenie elementów na materiale o nieograniczonej długości i ustalonej szerokości. Dla zadanego zbioru elementów minimalizuje on zatem długość materiału, na którym można upakować wszystkie elementy. Jego zastosowanie jest wskazane w przypadku, gdy w zbiorze elementów można wyróżnić podzbiory o podobnej długości.

Program oparty na idei zawartej w pracy Albano-Sapuppo umożliwia także minimalizację długości tej części materiału, na której można upakować wszystkie zadane elementy, przy ustalonej jego szerokości. Ponieważ każdy element rozpatrywany jest pod kątem możliwości jego upakowania niezależnie od innych elementów, program jest stosowany w przypadku pakowania elementów o stosunkowo dużym zróżnicowaniu wymiarów upakowywanych elementów.

Ze względu na implementację wszystkich trzech algorytmów na mikrokomputerze typu IBM PC/XT/AT w języku Turbo Pascal musiały one zostać zmodyfikowane w stosunku do pierwotnych wersji. W implementacji algorytmu Adamowicza wykorzystano struktury dynamiczne języka Pascal oraz zastosowano procedurę umożliwiającą usuwanie wklęsłości wprowadzonych elementów. Jednak ze względu na dużą zajętość pamięci tej implementacji umożliwia ona pakowanie tylko takiej liczby elementów, dla której iloczyn tej wartości oraz liczby możliwych kątów obrotu nie przekracza 15. Natomiast zwiększenie krotności pakowanych elementów o danym typie powoduje tylko nieznaczny wzrost zajętości pamięci.

W implementacji algorytmu Gurela zastosowano możliwość podjęcia decyzji przez użytkownika co do wielkości tzw. kroku. Im mniejsza jego wartość, tym dłuższy czas wykonywania obliczeń, lecz także większa możliwość uzyskania dobrego upakowania. Program składa się z trzech kolejno uruchamianych podprogramów.

W trzeciej implementacji ze względu na małą moc obliczeniową sprzętu zrezygnowano z możliwości przeglądania kilku ścieżek równoległych w trakcie przeglądania drzewa rozwiązań. Na każdym poziomie generowania drzewa przeglądane są wszystkie możliwe stany następujące (to jest powstające ze stanu poprzednio rozważanego przez dołożenie jednego elementu) i do dalszego rozwijania wybierany jest tylko jeden wybrany stan.

System wspomagania podejmowania decyzji automatycznie wybiera daną metodę w zależności od danych problemu, to znaczy: czy decydent poszukuje odpowiedzi na pytanie decyzyjne o możliwości upakowania w zadanym prostokącie wszystkich elementów (metoda 1), czy też poszukiwany jest minimalny prostokąt, w którym można upakować wszystkie elementy (metody 2 i 3). Metodę 2 stosuje się w przypadku możliwości dobrania wielu elementów w ciągi o podobnej długości, natomiast metodę 3 w pozostałych przypadkach.

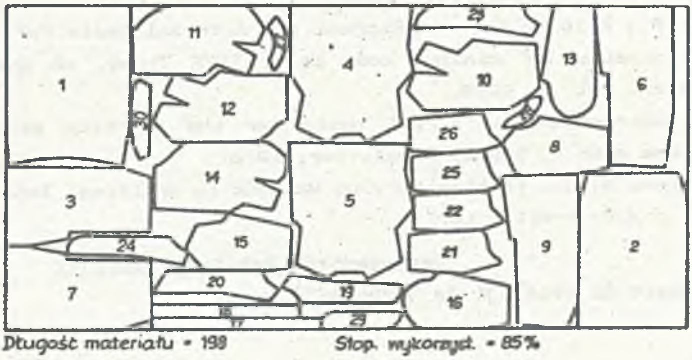
W celu porównania przedstawionych metod wykonano eksperyment obliczeniowy rozwiązując problem pakowania dla znanego z literatury [3] zbioru elementów, z których wygenerowano 15 problemów testowych (por. rys. 1). Wyniki przeprowadzonych badań umieszczono w tablicy 1. Analizując jej zawartość można stwierdzić, że dla powyższych zadań testowych jakość rozwiązań generowanych przez wszystkie trzy algorytmy nie odbiega w zasadniczy sposób, a więc dla celów praktycznych można stosować wszystkie, z uwzględnieniem obszarów zastosowania przedstawionych wyżej.

7. Zakończenie

W pracy omówiono zagadnienie rozkroju. Dla przypadku jedno- i dwuwymiarowego przedstawiono podstawowe algorytmy heurystyczne wraz z ich oszacowaniami dokładności. Dla nieregularnego problemu rozkroju opisano ideę systemu wspomagania decyzji, opartego na trzech znanych w literaturze algorytmach. Przedstawione zostały modyfikacje tych algorytmów w związku z ich implementowaniem na sprzęcie mikrokomputerowym. Przedstawiono także sugestie co do obszarów praktycznego zastosowania poszczególnych algorytmów. Na końcu zaprezentowano wyniki badań porównawczych algorytmów.

Tablica 1

L. p.	Liczba form.	Opis form.	Szer. mater.	Alg. Adamowicza Dł. m.	Alg. Gurela		Alb.-Sap.
					Dł. mat. wygener. automat.	Dł. mat. wygener. po popr.	Dł. mat. wygener. automat.
1	30	wszystkie	90	250	209	198	210.4
2	15	1,3,...29	90	120	113	107	115.8
3	15	2,4,...30	90	130	134	100	110
4	10	1,4,...28	90	100	91	81	86.5
5	10	2,5,...29	90	95	83	72	78
6	10	3,6,...30	90	85	74	62	67.25
7	6	1 - 6	90	120	125	100	113
8	6	7 - 12	90	60	82	59	70.3
9	6	13 - 18	90	40	62	38	34
10	6	19 - 24	90	30	36	22	27
11	6	1 - 6	45	240	191	191	221
12	6	7 - 12	45	130	123	119	134
13	6	13 - 18	45	70	108	69	64.8
14	6	19 - 24	45	53	71	43	46.8
15	6	25 - 30	45	34	50	23	27



Rys.1. Przykładowe rozmieszczenie zbioru elementów
Fig.1. An example packing of a set of elements

LITERATURA

[1] Adamowicz A., Cutting irregular shapes by mathematical programming, Ph D. thesis, MIT 1969.

[2] Albano A., Sapuppo G., Optimal allocation of two-dimensional irregular shapes using heuristic search methods, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-10, No.5, 1980.

[3] Art R.C., An approach to the two-dimensional irregular cutting stock problem, IBM Cambridge Scientific Center, Cambridge, MA, Rep. No. 320-2006, 1966.

[4] Baker B.S., Brown D.S., Katseff H.P., A 5/4 algorithm for two-dimensional packing, J. of Alg. vol 2, 1981.

[5] Baker B.S., Coffman E.G.,jr, Rivest R.L., Orthogonal packings in two dimensions, SIAM J. Comp., vol 9, 1980.

[6] Błażewicz J., (red.), Analiza złożoności obliczeniowej algorytmów automatycznego rozplanowywania rozkrojów, Opracowanie niepublikowane, PTI, Poznań, 1986.

[7] Błażewicz J., (red), Moduł automatycznego rozkroju materiału na elementy o dowolnym kształcie, Opracowanie niepublikowane, PTI, Poznań 1987.

[8] Chung F.R.K., Garey M., Johnson D.S., On packing two-dimensional bins, SIAM J. Alg. Disc. Meth., vol 3, No 1, 1982.

[9] Coffman E.G.,jr, Garey M., Johnson D.S., Tarjan R.E., Performance bounds for level-oriented two-dimensional packing algorithms, SIAM J. Comp., vol 9, 1980.

[10] Garey M.R., Johnson D.S., Computers and Intractability: A Guide to Theory of NP-Completness, Freeman W.H., San Francisco 1979.

[11] Gurel O. Circular graph of marker layout, New York Scientific Center, Report No. 320-2965, Feb. 1969.

[12] Hart P., Nilsson N., and Raphael B., A formal basis for the heuristic determination of minimum cost path, IEEE Trans. on Syst. Man, Cybern., vol. 4, 1968.

[13] --, Correction to "Formal basis for the heuristc determination of minimum path ", SIGART Newsletter, 1972.

[14] Nilsson, N. J., Problem Solving Methods in Artifical Intelligence, New York : McGraw-Hill, 1971.

Recenzent: Dr hab.inż.F.Marecki

Wpłynęło do Redakcji do 1988-04-30.

СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ РАЗРЕЗА

Р е з ю м е

В работе представлена система поддержки принятия решений в задачах разреза. Задачей оптимизации является такая упаковка элементов в контейнере, чтобы затраты материала были минимальные. Разработанная система выбирает лучшее из трёх генерированых этой системой решений и пользователь имеет возможность улучшения искомого решения путём интерактивной связи с системой.

DECISION SUPPORT SYSTEM FOR STOCK CUTTING PROBLEMS

S u m m a r y

In the paper, decision support system for stock cutting problem, is presented. The elements are packed into a bin so that its unused portion is minimized. The presented system chooses the best solution /from among three generated/ and the user can improve the solution via an interactive procedure.