Zastosowanie algorytmu genetycznego w dwu-etapowej metodzie rozwiązywania zagadnienia szeregowania zadań na maszynach równoległych z niezerowymi kosztami przezbrojeń i ograniczeniami zasobowymi

Ewa Figielska

Instytut Automatyki i Informatyki Stosowanej, Politechnika Warszawska,
ul. Nowowiejska 15/19, 00-665 Warszawa.

Streszczenie: Prezentowana praca dotyczy zagadnienia szeregowania zadań w gnieździe produkcyjnym stanowiącym zespół równoległych jednostek wytwórczych powiązanych wspólnymi zasobami dostępnymi w ograniczonych ilościach przy uwzględnieniu niezerowych kosztów przezbrojeń. Celem jest skonstruowanie harmonogramu produkcji, który minimalizuje łączny koszt wytwarzania wyrobów i przezbrajania urządzeń.

W pracy proponowana jest dwu-etapowa metoda stanowiąca połączenie dwu technik optymalizacji: generacji kolumn i algorytmów genetycznych. Przedstawione zostały wyniki eksperymentów numerycznych, na podstawie których określono wartości parametrów algorytmu genetycznego pozwalające na uzyskanie jak najmniejszych całkowitych kosztów przezbrojeń i dokonano analizy efektywności dwu-etapowej metody. Ogólnie można stwierdzić, że proponowane podejście pozwala przy niewielkim nakładzie obliczeń uzyskać rozwiązania zachowujące możliwie niskie koszty produkcji i przezbrojeń oraz spełniające dowolne ograniczenia strukturalne i zasobowe.

1. WSTĘP

W prezentowanej pracy rozważane jest zagadnienie krótkoterminowego harmonogramowania produkcji w gnieździe produkcyjnym stanowiącym zespół równolegle pracujących jednostek wytwórczych (np. maszyn, linii montażowych itp.). W gnieździe tym należy wyprodukować zadane ilości różnych wyrobów. Określona liczba identycznych wyrobów stanowi zadanie. Podczas wykonywania każdego zadania zużywana jest w jednostce czasu określona liczba jednostek pewnych typów zasobów. Zasoby te są dostępne w każdej chwili w ograniczonych ilościach. Przykładem tego typu zasobów, nazywanych *zasobami odnawialnymi*, jest paliwo, moc lub półprodukty dostarczane z gniazda poprzedzającego. Wykonywanie każdego zadania może być w dowolnej chwili przerwane i dokończone później na tej samej lub innej maszynie. Przestawienie (przezbrojenie) maszyny z produkcji jednego zadania na inne, jak również początkowe przygotowanie maszyny, obarczone jest kosztem. Celem jest znalezienie harmonogramu produkcji minimalizującego łączne koszty będące sumą kosztów wytwarzania i kosztów przeprowadzanych przezbrojeń.

Problemy szeregowania zadań na maszynach wymagających przezbrojeń należą do najtrudniejszych zagadnień optymalizacji dyskretnej o dużym znaczeniu praktycznym. Są one często spotykane w rzeczywistych systemach, np. w przemyśle farmaceutycznym, chemicznym lub spożywczym. Zagadnienie szeregowania zadań podzielnych na maszynach

równoległych przy uwzględnieniu dowolnych ograniczeń zasobowych należy do klasy problemów NP-trudnych, nawet przy pominięciu kosztów przezbrojeń. W tym przypadku zagadnienie to może być optymalnie rozwiązane przy użyciu technik generacji kolumn [1, 3, 7, 8]. Zostały również opracowane pewne heurystyki dla tego typu zgadnień [9, 2]. Wprowadzenie niezerowych kosztów przezbrojeń sprawia, że problem staje się tak skomplikowany, iż niemożliwe jest nawet jego matematyczne sformułowanie [5].

Dlatego też, w celu rozwiązania zagadnienia szeregowania zadań podzielnych na maszynach równoległych z uwzględnieniem przezbrojeń i ograniczeń zasobowych, konieczne staje się stosowanie metod aproksymacyjnych. W pracy [5] zostało zasugerowane trzy stopniowe podejście do rozwiązywania problemów z niezerowymi czasami przezbrojeń. W naszej pracy prezentujemy dwu-etapowe podejście do minimalizacji całkowitych kosztów, na które składają sie koszty produkcji i koszty przezbrojeń. W pierwszym etapie, stosując algorytm generacji kolumn, znajdowane jest rozwiązanie optymalne problemu z pominięciem kosztów przezbrojeń, a następnie plany elementarne otrzymane w pierwszym etapie są szeregowane przy użyciu algorytmu genetycznego w celu otrzymania końcowego harmonogramu o minimalnym całkowitym koszcie przezbrojeń. W pracy przedstawione zostały wyniki eksperymentów obliczeniowych, na podstawie których określono wartości parametrów algorytmu genetycznego (AG) dające największą redukcję kosztów przezbrojeń i dokonano analizy efektywności dwu-etapowej metody. W celu zilustrowania działania metody podany został również przykład obliczeniowy, umożliwiający porównanie postaci i kosztów harmonogramu utworzonego przez algorytm generacji kolumn z harmonogramem otrzymanym po redukcji kosztów przezbrojeń za pomocą AG.

2. SFORMUŁOWANIE PROBLEMU

Rozpatrywany problem można sformułować następująco. Danych jest n podzielnych zadań, które należy uszeregować na m równoległych maszynach. Oznaczmy przez M_j podzbiór maszyn, które mogą wykonywać zadanie j, przez N_i podzbiór zadań, które mogą być wykonywane na maszynie i. Każda maszyna może wykonywać tylko jedno zadanie w danej chwili, ale dane zadanie może być wykonywane na kilku maszynach równolegle (sytuacja taka występuje np. w fabryce szkła, gdzie wykonanie zadania, polegającego na wyprodukowaniu partii jednakowych butelek, jest rozdzielane między maszyny o różnej wydajności). Wprowadzamy następujące oznaczenia: r_{ij} — szybkość wykonywania zadania j na maszynie i, d_j — liczba zadań j (liczba partii identycznych wyrobów), które należy wyprodukować, α_{ijr} — szybkość zużywania zasobu r podczas wykonywania zadania j na maszynie i, W_r — szybkość dostarczania zasobu r do gniazda, σ_{ij} — koszt przygotowania maszyny i do produkcji zadania j (koszt przezbrojenia), c_0 — koszt jednostki czasu, c_{ij} — koszt bezpośredni wykonywania zadania j na maszynie i, J_r — podzbiór zadań, których wykonanie wymaga dostarczenia zasobu r.

Celem jest znalezienie harmonogramu produkcji, który minimalizuje całkowite koszty związane z produkcją i przezbrajaniem.

Rozwiązanie zagadnienia jest reprezentowane przez zbiór S planów elementarnych oznaczanych przez S_β ($\beta \in \mathcal{B}$). Dla każdego planu elementarnego S_β określony jest przydział zadań do maszyn $[v_{ij}^\beta]$, gdzie

$$v_{ij}^\beta = \begin{cases} 1, & \text{jeżeli zadanie } j \text{ jest wykonywane na maszynie } i \text{ w planie } S_\beta, \\ 0 & \text{w pp,} \end{cases}$$

i czas trwania planu Δ_β.

Każdy plan elementarny S_β musi spełniać następujące ograniczenia:

$$\sum_{j \in N_i} v_{ij}^\beta \leq 1, \qquad i = 1, \ldots, m \tag{1}$$

$$\sum_{j \in J_r} \sum_{i \in M_j} \alpha_{ijr} v_{ij}^\beta \leq W_r, \qquad r = 1, \ldots, p \tag{2}$$

$$v_{ij}^\beta \in \{0, 1\}, \qquad j = 1, \ldots, n, \quad i \in M_j \tag{3}$$

Ograniczenia (1) zapewniają, że każda maszyna wykonuje co najwyżej jedno zadanie w danej chwili. Warunki (2) ograniczają szybkość zużywania zasobów.

3. OPIS STOSOWANYCH ALGORYTMÓW

W prezentowanej pracy stosujemy algorytm generacji kolumn w celu otrzymania harmonogramu bez przezbrojeń oraz algorytm genetyczny w celu znalezienia takiej kolejności planów elementarnych, która minimalizuje koszty przezbrojeń.

3.1. Algorytm generacji kolumn

Aby skonstruować harmonogram minimalizujący koszt produkcji, tzn. koszt czasu produkcji i koszty bezpośrednie, należy rozwiązać następujący problem:

$$\min \sum_{\beta \in \mathcal{B}} \Delta_\beta (c_0 + \sum_{j=1}^{n} \sum_{i \in M_j} c_{ij} r_{ij} v_{ij}^\beta) \tag{4}$$

przy ograniczeniach

$$\sum_{\beta \in \mathcal{B}} \Delta_\beta \sum_{i \in M_j} r_{ij} v_{ij}^\beta = d_j, \qquad j = 1, \ldots, n \tag{5}$$

oraz ograniczeniach (1)-(3), które musi spełniać każdy plan elementarny S_β ($\beta \in \mathcal{B}$). Warunki (5) zapewniają, że wszystkie zadania zostaną wyprodukowane w wymaganych ilościach.

Stosując technikę generacji kolumn pomijamy trudności związane z generowaniem wszystkich kolumn danego problemu. Pracujemy tylko z podzbiorem kolumn i dodajemy nowe kolumny tylko wtedy, gdy poprawiają wyniki.

Niech B oznacza zbiór kolumn (opisujących plany elementarne), tworzących aktualne dopuszczalne rozwiązanie, π_j ($j = 1, \ldots, n$) oznacza zmienne dualne odpowiadające ograniczeniom (5). Zastosowany w tej pracy algorytm generacji kolumn na przemian rozwiązuje problem *programowania liniowego* (4)-(5) z nowo wprowadzoną kolumną i poszukuje następnej najlepszej kolumny. W celu znalezienia najlepszej kolumny należy zmaksymalizować $z_{0\beta} = \sum_{j=1}^{n} \sum_{i \in M_j} (\pi_j - c_{ij}) r_{ij} v_{ij}^\beta - c_0$ przy ograniczeniach (1)-(3). Jeżeli $z_{0\beta} > 0$, to nowa kolumna jest dołączana do zbioru B, $z_{0\beta} \leq 0$ oznacza, że optymalne rozwiązanie zostało znalezione.

3.2. Algorytm genetyczny

Wynik działania algorytmu generacji kolumn nie zależy od kolejności wykonywania planów elementarnych. Naszym celem jest teraz znalezienie takiej kolejności planów elementarnych, otrzymanych przy pomocy algorytmu generacji kolumn, która zminimalizuje całkowity koszt przezbrojeń.

Wprowadźmy następujące oznaczenia. Niech $\mathcal{K}$ będzie zbiorem wszystkich możliwych kolejności planów elementarnych, $k \in \mathcal{K}$, $k(l)$ jest równe indeksowi β planu elementarnego, który znajduje się na l-tej pozycji w ciągu k ($l = 1, \ldots, L$, gdzie L oznacza liczbę planów elementarnych). Dla kolejności $k \in \mathcal{K}$ całkowity koszt przezbrojeń dany jest wzorem

$$h_k = \sum_{l=1}^{L} \sum_{(i,j):\, x_{ij}^{k(l)}=1} (x_{ij}^{k(l)} - x_{ij}^{k(l-1)})\sigma_{ij},$$

gdzie

$$x_{ij}^{k(l)} = \begin{cases} v_{ij}^{k(l)}, & \text{jeżeli } \exists\, s \in N_i \;\; v_{is}^{k(l)} = 1, \\ x_{ij}^{k(l-1)}, & \text{jeżeli } \forall\, s \in N_i \;\; v_{is}^{k(l)} = 0, \end{cases}$$

$$x_{ij}^{0} = 0,$$

dla $j = 1, \ldots, n$, $i \in M_j$.

Minimalizację całkowitego kosztu przezbrojeń, polegającą na znalezieniu $h_{min} = \min_{k \in \mathcal{K}} h_k$, przeprowadzamy przy pomocy algorytmu genetycznego (AG), którego charakterystyka jest, jak następuje:

1. Potencjalne rozwiązanie problemu (nazywane *chromosomem*) jest reprezentowane jako ciąg liczb całkowitych (*genów*), oznaczających indeksy planów elementarnych S_β. Pozycje poszczególnych genów określają kolejność wykonywania planów.
2. Populacja, stanowiąca podzbiór rozwiązań problemu, ma stały rozmiar, tj. liczba chromosomów g w populacji nie zmienia się podczas procesu optymalizacji.
3. Populacja początkowa jest generowana w sposób losowy.
4. Ponieważ celem jest znalezienie kolejności planów elementarnych o minimalnej wartości h_k, w celu maksymalizacji przystosowania stosowana jest następująca transformacja: $f_k = 1.05 h_{max} - h_k$, gdzie $h_{max} = \max_{1 \le k \le g} h_k$. Dla otrzymanej wartości przystosowania stosuje się skalowanie liniowe (współczynnik skalowania oznaczamy przez λ).
5. Reprodukcja jest przeprowadzana według reguły ruletki.
6. Operacja krzyżowania jest wykonywana z prawdopodobieństwem P_{crs} dla dwóch losowo wybranych chromosomów przy użyciu dwupunktowego operatora *PMX* [4].
7. Operacja mutacji, przeprowadzana z prawdopodobieństwem P_{mut}, polega na zamianie miejscami wartości dwóch losowo wybranych genów w danym chromosomie.
8. Jeśli dla zadanej z góry liczby kolejnych pokoleń nie uzyskuje się poprawy najlepszego znalezionego do tej pory rozwiązania, to natępuje zakończenie poszukiwań.

Działanie AG zależy od doboru wartości parametrów takich jak wielkość populacji, prawdopodobieństwo krzyżowania (P_{crs}), prawdopodobieństwo mutacji (P_{mut}), współczynnik skalowania (λ) i liczba pokoleń. Dlatego też, przeprowadzone zostały testy obliczeniowe, na podstawie których określono wartości powyższych parametrów pozwalające na uzyskanie jak najlepszej kolejności planów elementarnych.

4. EKSPERYMENTY OBLICZENIOWE

Rysunek 1 przedstawia przykład harmonogramów produkcji otrzymanych w kolejnych etapach dwu-etapowej metody. W harmonogramie (b) widoczna jest znaczna redukcja

(1)

Koszty przezbrojeń σ_{ij}.

	Zadanie									
maszyna	1	2	3	4	5	6	7	8	9	10
1	6.30	3.99	4.37	5,1	4.75	1.94	7.39	1.93	2.61	8.69
2	8.99	5.2	7.44	3.93	7.69	1.88	3.88	5.51	3.46	7.39
3	1.06	5.62	8.16	5.44	1.68	3.8	3.99	3.76	5.69	5.07

Szybkości produkcji r_{ij}.

	Zadanie									
maszyna	1	2	3	4	5	6	7	8	9	10
1	0.35	0.43	0.39	0.37	0.35	0.35	0.38	0.45	0.39	0.43
2	0.47	0.29	0.13	0.39	0.17	0.19	0.29	0.17	0.17	0.47
3	0.35	0.13	0.43	0.40	0.10	0.21	0.43	0.34	0.27	0.28

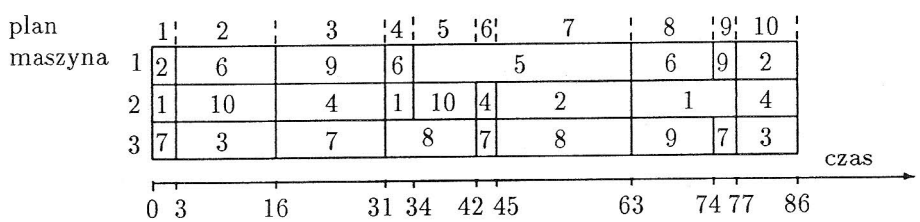
Przydział typów zasobów do zadań.

Zadanie	1	2	3	4	5	6	7	8	9	10
Typ zasobu	2	3	2	1	2	4	4	1	3	3

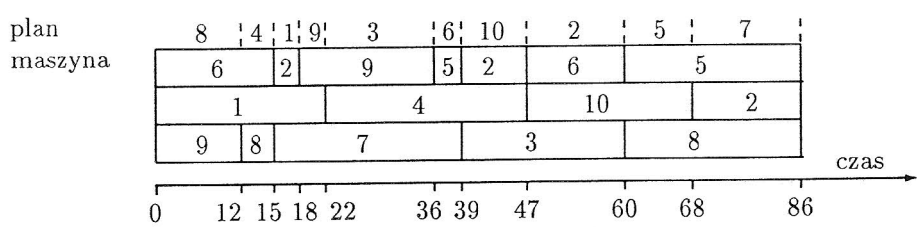
Pozostałe parametry: $c_0 = 1.5$, $\alpha_{ijr} \in \{0,1\}$, $W_1 = W_2 = W_3 = W_4 = 1$.

(2)

(a)



(b)



Rys. 1. Przykładowe dane (1) i harmonogramy (2) ilustrujące rozwiązania otrzymane w kolejnych etapach dwu-etapowej metody. (a) Harmonogram (o przypadkowej kolejności planów elementarnych) o łącznym koszcie (produkcji i przezbrojeń) równym 256.86 otrzymany za pomocą algorytmu generacji kolumn. (b) Harmonogram o łącznym koszcie 203.70 otrzymany w wyniku zmiany kolejności ustawienia planów elementarnych po zastosowaniu AG. Redukcja kosztów przezbrojeń przez AG wynosi w tym przypadku 41.53%.

Tablica 1. Wartości redukcji kosztów przezbrojeń, $\delta = \frac{K_c - K_g}{K_c} \times 100\%$, w zależności od parametrów AG

		P_{crs}		
P_{mut}	λ	0.4	0.6	0.8
0.001	1.6	25.26 (2.93)	25.78 (2.86)	27.82 (4.38)
	2.2	23.57 (4.15)	25.62 (4.06)	28.54 (3.42)
	2.8	26.33 (4.77)	26.25 (3.77)	27.14 (3.96)
0.005	1.6	27.46 (3.93)	27.89 (2.94)	27.98 (3.79)
	2.2	27.71 (3.89)	28.89 (3.01)	30.06 (2.87)
	2.8	28.11 (3.21)	28.97 (3.98)	29.23 (4.21)
0.01	1.6	29.87 (3.96)	27.61 (5.70)	23.70 (5.58)
	2.2	28.92 (3.73)	29.04 (3.86)	28.20 (3.63)
	2.8	29.06 (3.53)	30.03 (3.26)	30.08 (4.26)

Tablica przedstawia wartości średnie i odchylenia standardowe dla 10 losowo wygenerowanych problemów o następujących parametrach: 20 zadań, 3 maszyny, 4 typy zasobów, $d_j = 10$, $W_r = 2$ oraz $c_0 = 3$. Wartości parametrów r_{ij} i σ_{ij} wygenerowano losowo zgodnie z rozkładem równomiernym opowiednio z przedziałów [0.1,0.5] i [1,9]. Każdemu zadaniu przydzielono losowo (z przedziału [1,4]) indeks typu zasobu.

liczby przezbrojeń o wysokich kosztach w porównaniu z harmonogramem (a) .

W celu oceny jakości algorytmu genetycznego wprowadziliśmy wielkość

$$\delta = \frac{K_c - K_g}{K_c} \times 100\%, \tag{6}$$

gdzie K_c oznacza koszt przezbrojeń dla harmonogramu o przypadkowej kolejności planów elementarnych utworzonego przez algorytm generacji kolumn, K_g — koszt przezbrojeń po zastosowaniu AG.

Wartość δ wskazuje o ile procent został zredukowany całkowity koszt przezbrojeń po zastosowaniu AG w stosunku do kosztu przezbrojeń dla harmonogramu o przypadkowej kolejności planów elementarnych otrzymanego przez algorytm generacji kolumn.

Analiza kilkunastu wstępnie wykonanych testów obliczeniowych pozwoliła przyjąć rozmiar populacji równy 80 oraz kryterium stopu w ten sposób, że poszukiwanie lepszego rozwiązania zostaje zakończone z chwilą, gdy w kolejnych 50 pokoleniach nie następuje poprawa najlepszego dotychczas znalezionego rozwiązania. Dla tak przyjętych wartości przeprowadzono eksperyment obliczeniowy, którego wyniki przedstawione są w tablicy 1. Eksperyment ten polegał na rozwiązaniu za pomocą AG z różnymi wartościami operatorów P_{crs}, P_{mut}, λ dziesięciu losowo wygenerowanych problemów. Wyniki przedstawione w tablicy 1 pokazują, że najlepsze rezultaty (największe δ przy niewielkiej wartości odchylenia standardowego) zostały uzyskane dla algorytmu o wartościach (P_{crs}, P_{mut}, λ)=(0.8, 0.005, 2.2). Powyższe wartości zostały przyjęte dla AG w dalszych obliczeniach.

W tablicy 2 przedstawione zostały wyniki eksperymentów obliczeniowych polegających na rozwiązaniu 100 losowo wygenerowanych problemów o różnych parametrach. Wartości w trzech ostatnich kolumnach tablicy 2 stanowią średnie z 10 problemów (w nawiasach podane są odchylenia standardowe) i oznaczają odpowiednio: redukcję kosztów przezbrojeń przez AG (w procentach), czas (w sekundach) generowania planów elementarnych przez

Tablica 2. Redukcja kosztów przezbrojeń przez AG i czasy obliczeń dla algorytmu generacji kolumn i AG

liczba zadań	liczba maszyn	liczba zasobów	koszt jednego przezbrojenia	redukcja kosztów przezbrojeń $\delta = \frac{K_c - K_g}{K_c} \times 100\%$	czas obliczeń [s] generacja planów	czas obliczeń [s] szeregowanie planów przez AG
10	3	4	[1,9]	31.49 (9.06)	0.35 (0.15)	2.44 (0.41)
			const	28.50 (8.80)	0.29 (0.13)	2.22 (0.37)
20	3	4	[1,9]	31.85 (4.70)	0.87 (0.16)	6.74 (2.26)
			onst	30.27 (5.60)	0.84 (0.15)	5.04 (1.57)
20	6	6	[1,9]	34.50 (7.56)	1.22 (0.46)	6.70 (1.97)
			const	37.80 (3.70)	1.55 (0.43)	5.55 (1.31)
30	3	6	[1,9]	31.71 (3.01)	1.74 (0.31)	12.11 (4.44)
			const	26.69 (4.020	1.92 (0.41)	8.00 (1.50)
30	6	8	[1,9]	36.38 (6.79)	4.83 (2.04)	11.36 (2.76)
			const	32.20 (4.60)	5.87 (0.97)	10.67 (1.96)

Parametry σ_{ij} były generowane losowo zgodnie z rozkładem równomiernym z przedziału [1,9] lub równe 5, r_{ij} — generowane losowo z przedziału [0.1, 0.5], W_r i α_{ijr} (całkowitoliczbowe) — generowane losowo odpowiednio z przedziałów [10, 15] i $[0.4W_r, 0.6W_r]$, $d_j = 10$. Każde zadanie używa średnio dwóch losowo wybranych typów zasobów oraz może być wykonywane średnio na dwu losowo wybranych maszynach.

algorytm generacji kolumn oraz czas szeregowania planów elementarnych przez AG.

Na podstawie przeprowadzonych eksperymentów obliczeniowych można stwierdzić, że czasy wykonywania obliczeń przez AG oraz algorytm generacji kolumn są w przybliżeniu proporcjonalne do liczby zadań. Czas wykonywania obliczeń przez AG nie zależy od liczby maszyn, natomiast czas obliczeń dla algorytmu generacji kolumn wzrasta wraz ze wzrostem liczby maszyn. Dla problemów, w których liczba zadań jest równa 20, dwukrotny wzrost liczby maszyn (z 3 do 6) spowodował zwiększenie czasu obliczeń dla algorytmu generacji kolumn o mniej niż 100%, natomiast dla problemów, w których liczba zadań równa była 30, zwiększenie liczby maszyn z 3 do 6 spowodowało w przybliżeniu trzykrotne zwiększenie czasu obliczeń. Obserwowana jest niewielka zależność czasu obliczeń dla AG od rodzaju kosztów przezbrojeń. Przy stałych kosztach przezbrojeń czas obliczeń dla AG jest nieco krótszy niż w przypadku kosztów przezbrojeń zależnych od zadania i maszyny.

Zastosowanie AG do wyznaczania kolejności planów elementarnych przyczyniło się do zredukowania całkowitego kosztu przezbrojeń średnio o 32.14% w stosunku do kosztu przezbrojeń dla przypadkowej kolejności planów elementarnych otrzymanych przez algorytm generacji kolumn.

Eksperymenty obliczeniowe zostały przeprowadzone na komputerze PC z procesorem Pentium.

5. WNIOSKI

Eksperyment obliczeniowy pokazał przydatność proponowanej dwu-etapowej metody łączącej dwie techniki optymalizacji, generację kolumn i algorytm genetyczny, do

rozwiązywania problemów szeregowania zadań podzielnych na maszynach równoległych przy niezerowych kosztach przezbrojeń i ograniczeniach zasobowych. Stosując dwuetapową metodę można otrzymać rozwiązania przy rozsądnym nakładzie obliczeń dla zagadnień o rzeczywistych rozmiarach (liczba zadań w rzeczywistych systemach rzadko przekracza trzydzieści). Zastosowanie AG umożliwia efektywne rozwiązywanie problemów o dowolnych wartościach cen przezbrojeń oraz zapewnia ocenę jakości znalezionych harmonogramów.

Podziękowania. Pragnę złożyć podziękowania prof. E. Toczyłowskiemu za wprowadzenie w tematykę i dyskusje.

LITERATURA

1. Dantzig, G. B., Wolfe, P.: Decomposition principle for linear programs, *Operations Research* **8**, 101-111 (1960)
2. Figielska E., Toczyłowski E.: Algorytm szeregowania operacji podzielnych o zmiennych intensywnościach wykonywania przy ograniczeniach zasobów zużywalnych i odnawialnych, *Materiały Czwartej Konferencji Badań Operacyjnych i Systemowych*, Gdynia (1995).
3. Gilmore, P. C., Gomory, R. E.: A linear programming approach to the cutting-stock problem, *Operations Research* **9**, 849-859 (1961).
4. Goldberg, D. E.: *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Pub. Company, Inc.,N.Y. (1989).
5. Hindi, K. S., Toczyłowski, E.: Detailed scheduling of batch production in a cell with parallel facilities and common renewable resources, *Computers ind. Engng* **28**, 839-850 (1995).
6. Michalewicz Z.: *Genetic algorithms + data structures = evolution programs*, Springer-Verlag, Berlin (1992).
7. Słowinski R.: Two approaches to problems of resource allocation among project activities — a comparative study, *J. Opl Res. Soc.* Vol. 31, 711-723 (1980).
8. Toczylowski, E.: *Niektóre metody strukturalne optymalizacji do sterowania w dyskretnych systemach wytwarzania*, WNT, Warszawa (1989).
9. Toczylowski, E.: Algorithms for preemptive scheduling of independent tasks in the presence of general renewable and consumable resources, *Zesz. Nauk. AGH, ser. Automatyka* **59**, 263-272 (1991).