AUTOMATYKA • 2011 • Tom 15 • Zeszyt 2

Roger Książek*

Przedstawienie wyników działania algorytmu genetycznego dla zadania CLSP planowania wielkości i szeregowania partii produkcyjnej

1. Wprowadzenie

Podstawowym celem grupy zadań planujących wielkości i szeregujących partie produkcyjne jest wyznaczenie minimalnych kosztów. Koszty te związane są z kosztami utrzymywania zapasów i kosztami tworzenia partii produkcyjnych. Sformułowane modele dla tych zadań poszukują rozwiązania pomiędzy tymi dwoma przeciwstawnymi celami. Tworzenie dużych partii produkcyjnych pozwala na rzadkie ponoszenie kosztów ich uruchamiania, jednak prowadzi do wysokich zapasów. Wysokie zapasy generują większe koszty swojego utrzymania. Jeżeli zapasy są niskie, i niskie koszty ich utrzymania, to partie produkcyjne są małe ale jest ich dużo, powstają częste koszty związane z uruchomieniem partii. Przegląd prac na temat zadań planowania wielkości i szeregowania partii produkcyjnej można znaleźć między innymi w pracy Karimi *et al.*, 2003) [1] oraz Quadt i Kuhn, 2008 [2].

Zaprezentowany algorytm genetyczny dostosowano do rozwiązania najstarszego a zarazem podstawowego modelu planowania wielkości i szeregowania partii produkcyjnej CLSP (*Capacitated Lot Sizing Problem*). Model ten jest najczęściej stosowanym modelem z długimi okresami. W tym modelu koszty i czas przezbrojenia są naliczane za każdym razem, kiedy maszyna jest w stanie gotowość do wykonania wyrobu. W tabelach 1 i 2. przedstawione zostało zestawienie parametrów i zmiennych zadania CLSP.

* AGH Akademia Górniczo-Hutnicza, Wydział Zarządzania, Katedra Badań Operacyjnych i Technologii Informacyjnych

Tabela 1
Parametry zadania CLSP

$\mathcal{T}$	=	$\{1,...,T\}$ – zbiór okresów, gdzie T to liczba okresów,
$\mathcal{N}$	=	$\{1,...,n\}$ – zbiór produktów, gdzie n to liczba produktów,
C_t	–	zdolność produkcyjna, długość okresu t,
ST_j	$\leq$	C_t, czas przezbrojenia na wyrób j,
SC_j	–	koszt przezbrojenia na wyrób j,
p_j	–	jednostkowy czas wykonania wyrobu j,
h_j	–	jednostkowy koszt magazynowania wyrobu j,
d_{jt}	–	popyt na wyrób j w okresie t

Tabela 2
Zmienne zadania CLSP

x_{jt}	–	wielkość produkcji wyrobu j w okresie t,
y_{jt}	=	1, jeżeli w okresie t maszyna jest w gotowości do wykonania wyrobu j 0, w przeciwnym przypadku
I_{jt}	–	wielkość zapasu wyrobu j na koniec okresu t

Model PLCM zadania CLSP możemy sformułować w następujący sposób:

$$\min \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{T}} \left(SC_j y_{jt} + h_{jt} I_{jt} \right) \tag{1}$$

p.o.

$$I_{j,t-1} + x_{jt} - d_{jt} = I_{jt} \qquad t \in \mathcal{T},\ j \in \mathcal{N} \tag{2}$$

$$p_j x_{jt} + ST_j y_{jt} \leq C_t y_{jt} \qquad t \in \mathcal{T},\ j \in \mathcal{N} \tag{3}$$

$$\sum_{j\in\mathcal{N}} \left(p_j x_{jt} + ST_j y_{jt} \right) \leq C_t \qquad t \in \mathcal{T},\ i \in \mathcal{M} \tag{4}$$

$$I_{jt}, x_{jt} \geq 0, \qquad t \in \mathcal{T},\ j \in \mathcal{N} \tag{5}$$

$$y_{ijt} \in \{0, 1\} \qquad t \in \mathcal{T},\ j \in \mathcal{N} \tag{6}$$

Funkcja celu zadania (1) reprezentuje całkowite koszty uruchomienia partii i utrzymywania zapasów. Kolejne ograniczenie (2) opisuje bilans zapasów produkcji i popytu. Ograniczenie (3) pozwala na produkcję wtedy, gdy maszyna jest w gotowości do produkcji. Ograniczenie (4) zapewnia, że łączne obciążenie wynikające z produkcji w danym okresie nie przekroczy limitu dla tego okresu.

2. Algorytm genetyczny

2.1. Przegląd literatury

Algorytm genetyczny stanowi jeden z typów grupy algorytmów ewolucyjnych (*Evolutionary Algorithms*). Przeszukiwanie i optymalizacja opiera się tu na pewnych zasadach zaczerpniętych z teorii ewolucji. Sam algorytm jest prosty w działaniu i dlatego chętnie wykorzystuje się go do rozwiązywania problemów optymalizacji.

Rozwiązywaniem zadań wyznaczania wielkości i szeregowania partii produkcyjnej za pomocą heurystyk opartych na algorytmie genetycznym zajęli się A. Kimms, 1998) [3] oraz N. Dellaert *et al*, 2000 [4]. Wykorzystali oni heurystyki zbudowane w oparciu o zasady działania algorytmów genetycznych do rozwiązania zadań wielopoziomowych oraz zadań z maszynami równoległymi w planowaniu wielkości i szeregowaniu partii produkcyjnych. W pracy (Jinxing Xie i Jiefang Dong, 2001) [5] zaproponowano sposób rozwiązania zadania GCLDP (*General Capacitated Lot Sizing Problem*) za pomocą zbudowanego algorytmu genetycznego. Zastosowane w tych pracach algorytmy genetyczne uważano za efektywne i przydatne w ich stosowaniu, zwłaszcza do rozwiązywania zadań o większych rozmiarach.

Każda z prac przedstawia pewien indywidualny sposób kodowania chromosomów niosących informacje o rozwiązaniu, oraz własny sposób zdefiniowania operatorów krzyżowania i mutacji.

2.2. Zastosowany w pracy algorytm genetyczny

W rozwiązaniu zadania CLSP i zestawieniu wyników wykorzystano algorytm genetyczny zaproponowany przez autora.

Dla rozwiązywanego problemu CLSP, na początku wyznacza się rozwiązanie dopuszczalne za pomocą prostego algorytmu uzupełniania produkcji x_{jt} względem zapotrzebowania na produkt d_{jt}, poruszając się od ostatniego okresu planowania do okresu początkowego. Uwzględnia się przy tym ograniczenie (4) zapewniające, że łączne obciążenie wynikające z produkcji w danym okresie nie przekroczy limitu dla tego okresu.

Następnie w oparciu o wyznaczone rozwiązanie początkowe, buduje się populacje chromosomów. Każdy chromosom niesie ze sobą informacje o wielkości produkcji x_{jt}, czyli pewne dopuszczalne rozwiązanie zadania CLSP. Następnie populacja jest poddawana selekcji, krzyżowaniu i mutacji. Tylko podczas krzyżowania i mutacji w chromosomach dopuszcza się przechowywanie rozwiązania mogącego naruszać ograniczenia zadania. W ostatnim etapie jednego pełnego cyklu życia populacji chromosomy są naprawiane w celu wyeliminowania rozwiązań niespełniających ograniczeń zadania.

Celem tej pracy jest przedstawienie wyników obliczeń uzyskanych przy zastosowaniu opracowanego algorytmu. Dlatego dokładny sposób działa algorytmu, wykorzystanie użytych operatorów i funkcji oraz ich wpływ na otrzymywane rozwiązania zostanie opisany w następnej publikacji.

3. Eksperymenty obliczeniowe

3.1. Wykorzystane dane

Do obliczeń wykorzystano 20 zestawów danych dla których wartość parametrów zostały wygenerowane w sposób losowy, zgodnie z rozkładem równomiernym i zadanymi przedziałami dopuszczalnych wartości. Zestawy danych zostały zróżnicowane pod względem horyzontu planowania oraz liczby wyrobów (tab. 3). Po wygenerowaniu każdy zestaw został sprawdzony pod względem możliwości wyznaczenia rozwiązania dopuszczalnego.

Tabela 3

Parametry generowania zestawów z danymi

Zestaw	N	T	d_{jt}	C_t	p_j	ST_j	h_{jt}	SC_j
1 - 4	12	3	(2 .. 7)	50	(1..2)	(2..10)	(1..2)	(50..300)
6 - 10	16	5	(2 .. 7)	50	(1..2)	(2..10)	(1..2)	(50..300)
11 - 15	24	7	(2 .. 7)	50	(1..2)	(2..10)	(1..2)	(50..300)
16 - 20	31	12	(2 .. 7)	100	(1..2)	(2..10)	(1..2)	(50..300)

3.2. Wyniki obliczeń

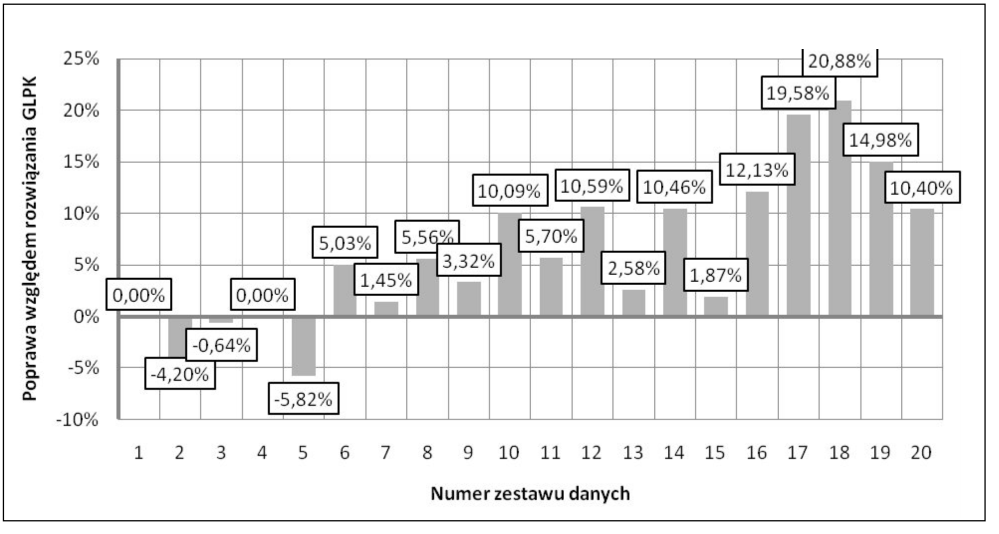
Eksperyment obliczeniowy polegał na policzeniu tego samego zastawu danych za pomocą zaimplementowanego algorytmu genetycznego oraz na rozwiązaniu zadania PLCM za pomocą solvera GLPK (*GNU Linear Programming Kit*). Algorytm genetyczny został zaimplementowany z wykorzystaniem języka JAVA SE 6. Natomiast GLPK jest to darmowy program do rozwiązywania zadań programowania całkowitoliczbowego. Obliczenia wykonano na komputerze z procesorem Intel® Core™2 Dou 2.00 GHz, z pamięciom RAM 4 GB.

Warunkiem granicznym szukania rozwiązania w obu przypadkach był z góry narzucony limit czasowy. Dla pierwszych 4 zestawów zadań, solver GLPK wyznaczył rozwiązania optymalne. W pozostałych przypadkach narzucony limit czasowy przerywał działanie solvera. Algorytm genetyczny szukał rozwiązań przez cały wyznaczony limit czasu. Tabela 4 przedstawia limity czasowe dla poszczególnych zestawów z danymi. Dla obu algorytmów w każdym przypadku były one takie same. Dla zadań trudniejszych limity te były stopniowo zwiększane.

Wyniki obliczeń dla 20 zestawów danych z zachowanymi limitami czasowymi przedstawiono na rysunku 1. Opisują one względną poprawę wartości funkcji celu wyznaczoną przez algorytm genetyczny, w odniesieniu do wartości obliczonej przez solver GLPK. Dla pierwszych 4 zestawów danych, gdzie liczba wyrobów wynosiła 4 oraz horyzont planowania 12 okresów, solver wyznaczył rozwiązania optymalne. Algorytm genetyczny w przypadku zestawu 1 i 4 również wyznaczył takie rozwiązania, jednak już dla zestawu 2 i 3 znalazł rozwiązania gorsze. Dlatego na wykresie pojawiły się wartości ujemne.

Tabela 4
Limity czasu poszukiwania rozwiązania

Zestaw	Limit czasu działania [s]
1 - 4	100
6 - 10	150
11 - 15	200
16 - 20	300



Rys. 1. Względna poprawa rozwiązania znalezionego przez algorytm genetyczny

Wraz ze wzrostem rozmiarów zadania algorytm genetyczny znajdował rozwiązania względnie coraz lepsze od rozwiązań uzyskanych za pomocą solvera GLPK. Dla małych zadań, których rozwiązanie nie stanowi problemu dla metod programowania całkowitoliczbowego, użycie tej heurystyki jest nie uzasadnione. Jednak wraz ze wzrostem złożoności zadania widać, że zaimplementowany algorytm genetyczny pozwalał na wyznaczanie coraz lepszych rozwiązań względem uzyskanych za pomocą solvera GLPK. Dodatkową zaletą stosowania algorytmu genetycznego jest to, że zwiększanie rozmiaru zadania nie wpływa gwałtownie na zwiększanie skomplikowania całości algorytmu. W przeciwieństwie do zadania programowania liniowego. Rozmiar zadania w pierwszym przypadku wpływa głównie na długość chromosomu, co ma małe znaczenie przy wyszukiwaniu lepszych rozwiązań. Natomiast w programowaniu całkowitoliczbowym wraz ze wzrostem zmiennych, zwiększa się rozmiar drzewa rozwiązań dopuszczalnych. Ma to duży wpływ na to, w jakim czasie zostanie dokonany przegląd rozwiązań i nastąpi wyznaczenie kolejnego lepszego rozwiązania.

Dla analizowanych danych możemy zauważyć znaczącą korzyść ze stosowania algorytmu genetycznego zwłaszcza dla zestawów od 16 do 20, czyli zadań z długim horyzontem planowania i z większą liczbą wyrobów. W tym samym czasie działania, algorytm genetyczny znalazł w każdym przypadku rozwiązania lepsze od rozwiązań uzyskanych za pomocą GLPK. Dla zestawu 18 różnica ta wynosiła aż 20%.

3.3. Parametry algorytmu

Tabela 5
Parametry algorytmu genetycznego

Zestaw	**Prawdopodobieństwo**	
	Krzyżowania	**Mutacji**
1 - 4	0,7	0,01
6 - 10	0,7	0,01
11 - 15	0,7	0,005
16 - 20	0,7	0,001

Wadą algorytmu genetycznego jest na pewno czas, jaki należy poświęcić na wyznaczenie odpowiednich wartość jego parametrów. W obliczeniach dla wszystkich zestawów danych algorytm działał na 10 chromosomach stanowiących populacje. Dodatkowo w każdym przypadku wykorzystywano selekcję proporcjonalną (*roulette wheel selection*).

Pewnym problemem było ustawienie prawidłowych wartość parametrów operatorów mutacji i krzyżowania. Dla policzenia analizowanych zestawów danych użyto ustawień przedstawionych w tabeli 5. Dobranie tych parametrów miało duże znaczenie w uzyskaniu dobrych wyników działania algorytmu genetycznego. Wraz ze wzrostem rozmiarów zadania należało obniżać prawdopodobieństwo mutacji. Zbyt duża mutacja „*psuła*” całą populację, rozrzucając rozwiązania daleko od siebie i cały algorytm zaczynał zachowywać się w sposób losowy. Mutacja zbyt mała pozwalała algorytmowi na utknięciu w minimum lokalnym i zatrzymywała poprawę rozwiązania. Poprawnie dobrany współczynnik mutacji sprawiał, że cała populacja zachowywała się prawidłowo, dając stopniowo ulepszone rozwiązania. Operator krzyżowania miał mniejszy wpływ na zachowanie się populacji. Jednak całkowity jego brak hamował i spowalniał ulepszanie rozwiązań, z drugiej strony pełne krzyżowanie osobników wprowadzało podobnie jak zbyt duża mutacja, losowość rozwiązań.

4. Podsumowanie

Wykorzystanie algorytmu genetycznego w rozwiązaniu zadań wyznaczania wielkości i szeregowania partii produkcyjnej jest uzasadnione dla zadań dużych, których rozwiązanie za pomocą metod programowania całkowitoliczbowego jest czasochłonne. Jak pokazały eksperymenty obliczeniowe, odpowiednio dopasowany algorytm genetyczny może znajdować efektywnie rozwiązania lepsze od metody programowania całkowitoliczbowego w tym samym czasie. Oczywiście uzyskanie dobrych wyników jest możliwe przy prawidłowym doborze parametrów algorytmu genetycznego. Przedstawione w pracy obliczenia uzyskane algorytmem zaproponowany przez autora pokazały, że wykorzystanie tej metody może być przydatne dla zadań CLSP wyznaczania wielkości i szeregowania partii produkcyjnej o większych rozmiarach.

Literatura

[1] Karim B., Fatemi Ghomi S., Wilson J., *The capacitated lot sizing problem: a review of models and algorithms.* Omega, 31, 2003, 365–378.
[2] Quadt D., Kuhn H., C*apacitated lot-sizing with extentions: a review.* OR, 6, 2008, 61–83.
[3] Kimss A., *A genetic algorithm for multi-level, multi-machine lot sizing and scheduling.* Computers & Operations Research, 26, 1999, 829–848.
[4] Dellaert N., Jeunet J., Jonard N., *A genetic algorithm to solve the general multi-level lot-sizing problem with time-varying costs*. Int. J. Production Economics, 68, 2000, 241–257.
[5] Jinxing Xie, Jiefang Dong, *Heuristic genetic algorithm for general* c*apacitated lot-sizing problems.* Computer & Mathematics with Applications, 44, 2002, 263–276.