ZESZYTY NAUKOWE POLITECHNIKI ŚLASKIEJ 2006
Seria: AUTOMATYKA z. 144 Nr kol. 1727

Paweł ANTCZAK, Arkadiusz ANTCZAK, Tadeusz WITKOWSKI
Politechnika Warszawska

ZASTOSOWANIE ALGORYTMU SYMULOWANEGO WYŻARZANIA DO OPTYMALIZACJI HARMONOGRAMOWANIA PRODUKCJI MAŁOSERYJNEJ

Streszczenie. W pracy przedstawiono jedną z klas gniazdowego problemu szeregowania zadań. Do rozwiązania tego problemu zaimplementowano algorytm symulowanego wyżarzania. Eksperymenty były porównane z zadaniami testowymi i otrzymanymi przy użyciu procedury GRASP.

USED SIMULATED ANNEALING ALGORITHM FOR OPTIMIZATION JOB SHOP SCHEDULING PROBLEM

Summary. One class of problem, that presents general case of scheduling problem. The algorithm is presented for solving the flexible job shop problem. The algorithm is based on the simulated annealing metaheuristic modified appropriatelly. Experiments were compared with test problem and results obtained with GRASP procedure.

1. Wprowadzenie

Zbadano jedną z klas zadania harmonogramowania, przedstawiająca ogólny przypadek zadania harmonogramowania z określonym porządkiem wykonania operacji na grupach technologicznie zamiennych maszyn. Dla rozwiązania tej klasy zadań opracowano algorytm na podstawie algorytmu symulowanego wyżarzania. Przedstawiono sformułowanie rozpatrywanego zadania i określono etapy opracowanego algorytmu. Przeprowadzono eksperymenty i opisano otrzymane wyniki, porównując je z wynikami otrzymanymi za pomocą innych algorytmów przy rozwiązywaniu testowego zadania.

2. Sformułowanie zadania

Przedstawmy sformułowanie rozwiązywanego zadania dla problemu gniazdowego. Matematycznie może być on przedstawiony w następujący sposób.

Określono zbiór maszyn M (m – moc zbioru M), zbiór operacji O, elementami którego są poszczególne operacje technologiczne σ^i, $i = 1..n$, gdzie n – moc zbioru O. Każdej operacji $\sigma^i \in O$ przypisano podzbiór maszyn $M^i \in M$, które

mogą je wykonywać. Zbiór operacji O jest częściowo uszeregowany. Oznaczmy to częściowe uszeregowanie symbolem $\prec$. Z punktu widzenia ograniczeń technologicznych dla operacji σ^i, $\sigma^i \in O$ relacja («$\sigma^i \prec \sigma^j$» oznacza, że operacja σ^i powinna być wykonana przed rozpoczęciem operacji σ^j. Oznaczmy przez $K(\sigma^i)$ klasę elementów ze zbioru O operacji σ^i. Niech dla operacji σ^i: $p(\sigma^i)$ – czas, niezbędny do wykonania operacji (jednakowy dla wszystkich maszyn), $t(\sigma^i)$ – czas, niezbędny do przezbrojenia maszyny przed wykonaniem danej operacji.

Zadanie opracowania harmonogramu polega na tym, aby wybrać dla każdej operacji $\sigma^i \in O$ maszynę ze zbioru $M^i (i = 1..n)$ i następnie określić sekwencję wykonania operacji na maszynach ze zbioru M w taki sposób, aby określony harmonogram minimalizował sumaryczny czas wykonania wszystkich prac (kryterium Johnsona).

Oznaczmy: $S(\sigma^i)$ – czas rozpoczęcia wykonania operacji technologicznej σ^i, $F(\sigma^i) = S(\sigma^i) + p(\sigma^i)$ – czas zakończenia jej wykonania, m^i – maszyna wybrana ze zbioru M^i dla wykonania operacji σ^i. Wtedy matematyczne sformułowanie zadania opracowania harmonogramu dla wykonania operacji technologicznych będzie mieć następującą postać:

$$\min F \tag{1}$$

przy ograniczeniach:

$$F \geq F(\sigma^i), \forall \sigma^i \in O \tag{2}$$

$$F(\sigma^i) \leq S(\sigma^i), \forall \sigma^i \prec \sigma^j \tag{3}$$

$$S(\sigma^i) \geq t(\sigma^i), \forall \sigma^i \in O \tag{4}$$

$$F(\sigma^i) = S(\sigma^i) + p(\sigma^i), \forall \sigma^i \in O \tag{5}$$

$$F(\sigma^i) \leq S(\sigma^j) - t(\sigma^j) \vee$$
$$\vee F(\sigma^j) \leq S(\sigma^i) - p(\sigma^i),$$
$$\forall \sigma^i, \sigma^j \in O, \text{ takich że}$$
$$m^i = m^j, (\sigma^i \cup \sigma^j) \notin k^l, l = 1...K \tag{6}$$

$$F(\sigma^i) \leq S(\sigma^j) \vee F(\sigma^j) \leq S(\sigma^i),$$
$$\forall \sigma^i, \sigma^j \in O, \text{ takich, że}$$
$$m^i = m^j, (\sigma^i \cup \sigma^j) \in k^l, l \in \{1...K\} \tag{7}$$

Ograniczenia (1),(2) określają kryterium optymalizacji (kryterium minimalizacji wykonania wszystkich zadań – kryterium Johnsona). Ograniczenia (3) określają ograniczenia kolejności zgodnie z technologicznym porządkiem wykonania. Ograniczenia (4) wymagają wykonania przezbrojenia maszyny przed rozpoczęciem wykonania operacji. Ograniczenia (5), (6) przedstawiają ograniczenia na zasoby (maszyna może wykonywać jednocześnie tylko jedna operację). Te ograniczenia także uwzględniają czas niezbędny na przezbrajanie maszyn przy wykonaniu operacji. Ograniczenia (7) wymagają, aby operacje były wykonywane na odpowiednich, wstępnie określonych maszynach technologicznie zamiennych z danej grupy.

Dalej przedstawiono zastosowanie algorytmu symulowanego wyżarzania (SA) do harmonogramowania w przypadku istnienia grup technologicznie zamiennych maszyn.

Rozwiązania zadania (1)-(7) można poszukiwać w postaci uporządkowanego zbioru F_{queue} par w postaci (σ^i, m), gdzie $\sigma^i \in O$, $m \in M^i$.

3. Przeszukiwanie lokalne

Istotę algorytmu SA, podobnie jak wielu innych algorytmów wykorzystujących przeszukiwanie lokalne, stanowi operacja zamiany danego rozwiązania na rozwiązanie z tego sąsiedztwa. Przez sąsiedztwo zwykle rozumiemy zbiór rozwiązań otrzymanych za pomocą określonych przekształceń nad rozwiązaniem wyjściowym. Większość sąsiedztw dla zadań harmonogramowania, w których jako kryterium optymalności występuje kryterium Johnsona, wykorzystuje określenie drogi krytycznej.

Różne rodzaje sąsiedztw opisano w [3,4]. Sąsiedztwo zaproponowane w [3] składa się z rozwiązań otrzymanych z rozwiązania wyjściowego poprzez zmianę porządku uszeregowania dwóch pierwszych lub dwóch ostatnich operacji w blokach krytycznych. Oznaczmy $N^{NS}(x)$ zbiór rozwiązań z tego sąsiedztwa dla rozwiązania x. Zastosowanie tego sąsiedztwa do zadania (1)-(7) jest utrudnione, gdyż nie uwzględnia ono możliwości wykonania operacji na jednej z technologicznie zamiennych maszyn danej grupy. Dlatego też, stosując lokalne przeszukiwanie, wykorzystujące to sąsiedztwo, dla zadania (1)-(7) wybór maszyn do wykonania operacji pozostaje niezmienny. W celu zaadaptowania tego sąsiedztwa rozszerzamy je dodając do niego rozwiązania otrzymane w wyniku przeniesienia operacji krytycznej na inną maszynę. W przypadku przeniesienia operacji σ^i na inną maszynę, którą oznaczymy przez m', powinniśmy także określić porządek, w jakim będą wykonywane operacje na nowej maszynie m' po takiej zamianie. Ponieważ liczba rozwiązań w tym sąsiedztwie będzie bardzo duża, to należy ograniczyć liczbę wariantów przeszukiwania poprzez dokładny i efektywny wybór miejsca wstawienia σ^i do uszeregowania wykonania operacji na nowej maszynie.

Dla każdej operacji $\sigma^i \in O$ określimy zbiór $JP(\sigma^i)$, zawierający operacje $\sigma^j \in O$ takie, że $\sigma^j \prec \sigma^i$ i nie istnieje operacja $\sigma^l \in O$ taka, że $\sigma^j \prec \sigma^l \prec \sigma^i$. Oznaczymy przez $MP(\sigma^i)$ operację, dla której $(\sigma^i, m^i), (MP(\sigma^i), m^i) \in F_{queue}$, $(MP(\sigma^i), m^i) = (\sigma^i, m^i)$ oraz nie istnieje operacja $\sigma^l \in O$ taka, że $(\sigma^l, m^i) \in F_{queue}$ i $(MP(\sigma^i), m^i) = (\sigma^l, m^i) = (\sigma^i, m^i)$. Zbiór JP dla każdej operacji pozostaje niezmienionym, a MP zależy od konkretnego rozwiązania określonego przez zbiór F_{queue}. Dalej oznaczmy przez $JP(\sigma^i)$ zbiór operacji $\sigma^j \in O$, dla których $\sigma^j \in JP(\sigma^i)$ oraz $MS(\sigma^i)$ – operację, dla której $MP(MS(\sigma^i)) = \sigma^i$.

Oznaczymy przez $N^{NEW}(x)$ zbiór rozwiązań, które odpowiadają zbiorowi uporządkowanych zbiorów, otrzymanych z F_{queue} poprzez zamianę pary (σ^j, m^j) na parę (σ^j, l^j), gdzie σ^j – operacja krytyczna, $l^j \in M^j$ oraz $l^j \neq m^j$, a uszeregowanie pozostaje poprzednie. W takim przypadku wszystkie rozwiązania przynależące do $N^{NEW}(x)$ będą dopuszczalne, z powodu wstępnie przeprowadzonego przeszeregowania F_{queue}. W ten sposób określamy sąsiedztwo dla rozwiązania x, które jest bardziej uniwersalne.

Najbardziej pracochłonnym etapem lokalnego przeszukiwania jest ocena rozwiązań z danego sąsiedztwa. W pracy do realizacji lokalnego przeszukiwania w sąsiedztwie $N^{NEW}()$ wykorzystano schemat szybkiej oceny rozwiązań przedstawiony w [7].

4. Algorytm symulowanego wyżarzania

W pracy dla rozwiązania zadania harmonogramowania przedstawiono algorytm wykorzystujący metodę symulowanego wyżarzania [4,8]. Niżej przedstawiono ogólny schemat tego algorytmu do opracowania harmonogramu.

```
Określić wejściowe parametry algorytmu
k = 1
while k <=K do
 for iter = 0 do L do
  losowo wybrać rozwiązanie y ∈N^NS (x)
    if f (y) < f (x) then
     przejście do rozwiązania y, tj. x = y
     else
     przejście do rozwiązania y z prawdopodobieństwem
     exp [ - {f (y) – f (x)}/ T]
     end if
  if iter % freq = 0 then
        { % - reszta od dzielenia}
     improvement = true
     while improvement do
       if istnieje rozwiązanie y ∈ N^NEW (x) takie, że
 f (y) < f (x) then
       x = y
       improvement = true
      else
       improvement = false
      end if
     end while
  end if
 end for
 k = k + 1
 T = T * r
end while
```

5. Eksperyment komputerowy

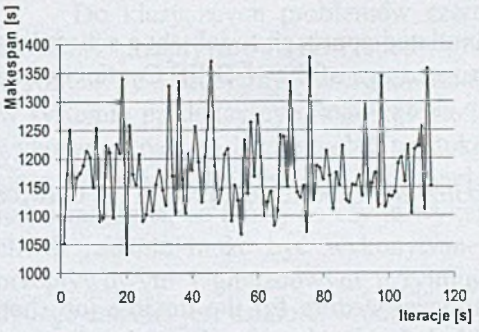
W algorytmie SA występuje wiele parametrów, których wartości muszą być ustalone. Brak jest jednak ogólnych metod ich doboru. Przyjmuje się, że początkowa temperatura T_o powinna być wystarczająco wysoka, tak aby zagwarantować odpowiednio dużą wartość prawdopodobieństwa p_o zaakceptowania ruchów, które nie

prowadzą do poprawy rozwiązania. Można przyjąć wartość $p_o = 0,9$, chociaż niektórzy autorzy proponują tylko wartość 0,4. Duży wpływ na czas przetwarzania mają współczynnik schładzania r oraz liczba iteracji ze stałą temperaturą.

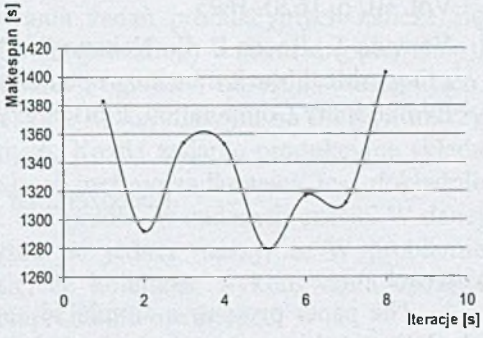
W przedstawionym algorytmie sąsiedztwo $N^{NS}(g)$ wykorzystuje się dla realizacji cyklu temperatury algorytmu symulowanego wyżarzania. Korzystne przejścia z tego sąsiedztwa są od razu stosowane do rozwiązania, a niekorzystne stosowane są z prawdopodobieństwem zależnym od bieżącej temperatury i stopnia pogorszenia rozwiązania. Każda liczba freq iteracji przeprowadza lokalne przeszukiwanie w sąsiedztwie $N^{NEW}(g)$, co pozwala istotnie polepszyć rozwiązania.

Wykonane badania testowe [5, 8] dla różnych wariantów algorytmu symulowanego wyżarzania z różnymi sąsiedztwami i z kryterium minimalnego czasu przetwarzania, przy różnych metodach strojenia algorytmu, wykazały umiarkowany optymizm w ocenach postaci: jakość rozwiązania / czas obliczeń. Z tego też względu kolejne prace badawcze zmierzały w kierunku algorytmów hybrydowych, otrzymanych głównie poprzez wbudowanie dodatkowej procedury przeszukiwania lokalnego (pełny przegląd subotoczenia) poprzedzającej wybór końcowego rozwiązania. Podobne podejście zastosowano w pracy.

Dla rozwiązania zadania (1)-(7) opracowano oprogramowanie realizujące algorytm wykorzystujący metodę SA, którego szczegóły przedstawiono wyżej. Eksperymenty obliczeniowe przeprowadzone dla danych przedstawionych w [5] pokazały dużą efektywność tego algorytmu. Liczba operacji dla tych danych wynosi 160, a liczba maszyn 26. Eksperymenty komputerowe były przeprowadzone z wykorzystaniem komputera z procesorem Pentium 733 MHz z 256 MB pamięci operacyjnej z następującymi parametrami: K – liczba zmian temperatury, L – liczba iteracji ze stałą temperaturą, T – temperatura początkowa, r – parametr zmniejszenia temperatury, *freq* – częstość lokalnego przeszukiwania w sąsiedztwie N^{NEW} (K=100, L=100, T=100, r=0.95). Jako początkowe rozwiązanie przyjmowano najlepsze rozwiązania znalezione na wszystkich iteracjach.



Rys. 1. Wyniki obliczeń dla FT10



Rys. 2. Wyniki obliczeń dla FT20

Wstępne eksperymenty przeprowadzone na testowych zadaniach typu FT10 i FT20 (Fischer i Thompson) dają wyniki zbliżone do wyników otrzymanych za pomocą innych efektywnych algorytmów, np. [1], chociaż gorsze niż otrzymane np. w [3]. Dokładniejsza analiza parametrów algorytmu zapewne przyczyni się do

poprawy otrzymanych wyników. Na rysunkach 1 i 2 przedstawiono wartości funkcji kryterialnej (makespan) dla rozwiązania zadań typu FT10 i FT20. Przy czym liczba iteracji dla tego samego czasu obliczeń (ok. 7000 s) problemu FT10 i FT20 wzrosła kilkanaście razy (odpowiednio 9 i 113).

Celem naszej pracy było głównie zastosowanie efektywnego algorytmu do rozwiązania problemu gniazdowego z grupami technologicznie zamiennych maszyn. Dokładne obliczenia praktycznego zadania (10 zadań, 16 operacji i 27 maszyn) dla innych algorytmów, m.in. algorytmów genetycznych, GRASP zaprezentowano w [6,7]. Rozwiązania otrzymane za pomocą algorytmu SA należą do najlepszych.

LITERATURA

1. Domdorf U., Pesch E.: Evolution Based learning in a Job Shop Environment. Computers and Operations Research, Vol. 22, 1995, p. 25–40.
2. Janiak A.: Wybrane problemy i algorytmy szeregowania zadań i rozdziału zasobów. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1999.
3. Nowicki E., Smutnicki Cz.: A Fast Taboo Search Algorithm for the Job Shop Problem. Management Science, Vol. 42, No 6, 1996, p. 797–813.
4. Van Laarhoven P., Aarts E, and Lenstra J.: Job shop scheduling by simulated annealing. Operations Research, 40, 1992, p. 113–125.
5. Witkowski T.: Decyzje w zarządzaniu przedsiębiorstwem. WNT, Warszawa 2000.
6. Witkowski T., Antczak P., Antczak A.: Random and Evolution Algorithms of Tasks Scheduling and the Production Scheduling. Proceedings of International Joint Conference on Fuzzy Systems – Fuzz – IEEE 2004, Budapest 2004, Vol.2, p. 727–732.
7. Witkowski T., Antczak A., Antczak P.: Taboo search and GRASP used in hybrid procedure for optimize the flexible job shop problem. In Proceedings of the Eleventh International Fuzzy Systems Association World Congress, Beijing 2005 Vol. III, p. 1620–1625.
8. Yamada T., Rosen B. E., Nakano R.: A simulated approach to job-shop scheduling using critical block transition operators. Proceedings of the 1st IEEE Conference on Evolutionary Computation, Florida 1994, p. 4687–4692.

Recenzent: Dr hab.inż: M. Zaborowski, prof. IITiS Gliwice

Abstract

The paper presents a simulated annealing procedure for the flexible job shop scheduling problem. One class of problem with defined order of realization operations on groups technologically exchangeable machines, was searched. To solve this class of problem an algorithm was created based on simulated annealing procedure. Simulations studies have been performed to evaluate the performance of the algorithm and to compare the solution results with those obtained by using GRASP procedure.