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

Ewa Dudek-Dyduch
Akademia Górniczo-Hutnicza

Tadeusz Dyduch
Politechnika Krakowska

WYZNACZANIE ROZWIĄZAŃ DOPUSZCZALNYCH DLA PROCESÓW MODELOWANYCH GRAFEM BERGE'A */

Streszczenie. Praca dotyczy sterowania dyskretnymi procesami produkcyjnymi. Poszukiwanie rozwiązania dopuszczalnego przedstawiono jako poszukiwanie drogi w odpowiednio zdefiniowanym grafie przejść dopuszczalnych. Przedstawiono dwa algorytmy znajdowania sterowań dopuszczalnych: pierwszy oparty na generowaniu grafu drzewowego, drugi analizujący graf Berge'a wykorzystujący procedurę ewidencjonowania wcześniej zbadanych wierzchołków. Przedstawiono organizację zbioru tych wierzchołków i procedurę szybkiego przeglądania i modyfikowania go.

1. Wstęp

Optymalizacja dyskretnych procesów produkcyjnych jest dziedziną cieszącą się w ostatnich latach wielkim zainteresowaniem. Prezentowana jest wielka ilość różnorodnych procesów. Ponieważ zadania optymalizacji większości z nich należą do klasy zadań NP - zupełnych, toteż prezentowane algorytmy są najczęściej algorytmami heurystycznymi. Równocześnie, brak matematycznych modeli zadań i klasyfikacji procesów opartych na ich matematycznych własnościach niezmiernie utrudnia analizę, porównywanie i uogólnianie idei stanowiących podstawę poszczególnych algorytmów heurystycznych. Z drugiej strony, pewna "niejasność" tych idei budzi uzasadniony sceptycyzm wobec algorytmów heurystycznych. Sytuacja byłaby korzystniejsza, gdyby algorytmy te prezentowane były na tle algorytmów dokładnych z równoczesnym zaznaczeniem, na czym polega wprowadzona przez autora heureza oraz jakie przesłanki do niej doprowadziły. Niestety, brak ogólnej metodologii uniemożliwia takie podejście dla większości skomplikowanych zadań. W szczególności dotyczy to zadań optymalizacji pośredniej posiadających następujące cechy:

- rozwiązaniem jest ciąg o nieznanej a priori długości,
- nie można a priori określić dziedziny odwzorowania A realizowanego przez algorytm zadania decyzyjnego odpowiadającego zadaniu optymalizacji.

Niniejsze opracowanie związane jest z klasą takich zadań. Celem jest przedstawienie wspólnego dla nich etapu rozwiązywania polegającego na analizie grafu przejść dopuszczalnych. Zaprezentowano dwie odmienne koncepcje oraz oparte na nich dwa podstawowe algorytmy analizy.

*/Opracowanie jest wykonane w ramach tematu "Resortowy Program Badań Podstawowych R.P.I.02" (4)

W oparciu o ściśle określony sposób rozwiązywania zadania można teraz opracowywać i definiować szereg klas algorytmów heurystycznych.

2. Graf przejść dopuszczalnych

W pracy [3] autorzy podali formalną definicję dyskretnego procesu $P = (s_0, f, S_N, S_T)$, który jest modelem szerokiej klasy dyskretnych procesów produkcyjnych. Podstawą modelu jest zdefiniowanie funkcji przejścia f oraz jej własności:

$f: U \times S \to S$, gdzie:

U - zbiór decyzji sterujących,

$S = X \times R^+$ - zbiór stanów uogólnionych,

X - zbiór stanów,

R^+ - zbiór liczb rzeczywistych nieujemnych reprezentujących czas.

$s_0 \in S$ wyróżniony uogólniony stan początkowy, S_N, S_T - wyróżnione zbiory stanów niedopuszczalnych oraz końcowych.

(W dalszym tekście pisząc o stanie uogólnionym będziemy pomijać słowo "uogólniony", jeśli notacja zapewni jednoznaczność).

Funkcję f zdefiniowano jako funkcję częściową, co pozwala uwzględnić wszystkie ograniczenia dotyczące decyzji sterujących za pomocą tzw. zbiorów sterowań możliwych w stanie $s \in S$, oznaczonych $U_p(s)$ i zdefiniowanych

$U_p(s) = \{u \in U: f(u,s) \in \mathrm{Dom} f\}$

Ograniczenia dotyczące stanów uogólnionych zostały uwzględnione za pomocą tzw. zbiorów sterowań dopuszczalnych w stanie s, oznaczonych $U_d(s)$.

$U_d(s) = \{u \in U_p(s): f(u,s) \notin S_N\}$

W oparciu o model dyskretnego procesu definiuje się graf przejść dopuszczalnych $G = (S,R)$, gdzie zbiór wierzchołków S jest równy zbiorowi stanów S, zaś relacja $R \subset S \times S$ jest zdefiniowana następująco:

$(s_i, s_j) \in R \Leftrightarrow \exists u \in U_d(s_i): f(u,s_i) = s_j$

Graf przejść dopuszczalnych ma następujące własności:

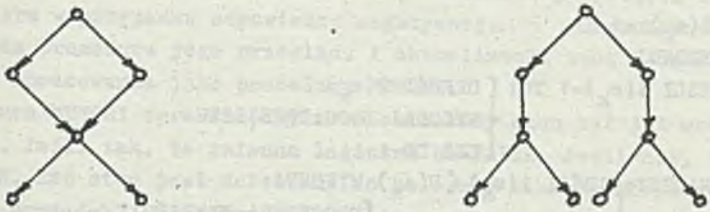
- graf nie jest podany explicite (np. zapisany za pomocą macierzy), lecz podane są zależności pozwalające generować go lokalnie,
- jest on acyklicznym digrafem, ale w ogólnym przypadku nie jest drzewem,
- graf ma wyróżniony wierzchołek początkowy oraz wierzchołek końcowy (względnie ich zbiór).

Poszukiwanie rozwiązania dopuszczalnego jest zatem równoważne poszukiwaniu drogi łączącej wierzchołek początkowy z jednym z wierzchołków końcowych, zaś optymalizacja zawiera dodatkowo wybór najlepszej z tych dróg. Ponieważ zadanie optymalizacji można przedstawić jako ciąg tzw. zadań decyzyjnych, w których poszukiwane jest rozwiązanie dopuszczalne, toteż w dalszym ciągu zajmiemy się tylko poszukiwaniem rozwiązań dopuszczalnych. Unigrafy skierowane nazywane są grafami Berge'a. Będziemy używali tej

nazwy dla zaznaczenia ogólniejszej struktury grafu przejść dopuszczalnych niż struktura drzewa.
Analogiczne zadanie poszukiwania drogi od wierzchołka początkowego do zbioru wierzchołków końcowych, gdy dane są tylko reguły generowania kolejnych wierzchołków, występuje w teorii gier ekstensywnych [4], [5], jednakże w [5] rozważane są grafy w postaci drzewa.

3. Wyznaczenie dopuszczalnego rozwiązania metodą generowania drzewa

Zadania, którym odpowiadają grafy Berge'a, analizowane są metodą przeglądu drzewowego, tzn. generowany jest graf drzewowy odpowiadający danemu grafowi Berge'a.



a) Graf Berge'a b) Odpowiadający mu graf drzewowy
a) Berge's graph b) The corresponding tree graph

Podstawowa jest metoda powrotów z uwzględnieniem eliminacji przeglądów pewnych poddrzew bądź to jako podzbiorów nieperspektywicznych (wykluczanie lub obcinanie gałęzi), bądź jako poddrzew izomorficznych (sklejanie lub łączenie gałęzi) [6]. Reguły wykluczania oraz sklejania są opracowywane dla konkretnych zadań. W przypadku grafów o znacznym rozmiarze lub zawierającym cykle przegląd ograniczony jest do pewnej kuli wokół wierzchołka początkowego. Procedura ta nosi nazwę przeszukiwania grafu do ustalonej głębokości R [7]. Pamiętana jest tylko droga łącząca badany węzeł z korzeniem. W celu przecięcia ewentualnie napotkanego cyklu analizowany węzeł porównywany jest z wszystkimi poprzednikami w generowanej drodze, i w razie tożsamości odrzucany.

Przedstawimy teraz algorytm, w którym zastosowano to podejście dla poszukiwania sterowania dopuszczalnego. Ponieważ będziemy rozważali tylko zbiory sterowań dopuszczalnych, opuścimy identyfikujący je wskaźnik dolny d. Zbiór dopuszczalnych decyzji U(s) dla stanu s wyznacza zbiór bezpośrednich następników wierzchołka s. Aby określić porządek ich przeglądania, należy wprowadzić porządek w zbiorze U(s), np. leksykograficzny.

Przyjmijmy oznaczenia:

i(s) - numer aktualnie badanej decyzji ze zbioru U(s),

U(s) - moc zbioru U(s), a więc ilość bezpośrednich następników wierzchołka s,

k - numer aktualnie badanego stanu w generowanym ciągu,

s_k - k-ty element ciągu,

count - liczba sprawdzanych węzłów drzewa,

d_k - decyzja sterująca w k-tym kroku ciągu,

R- głębokość poszukiwań.

Schemat poszukiwania przedstawia ALGORYTM 1.

ALGORYTM 1

```
k:=1
s_k:=s_o
DLA k=1,2..R  i(s_k):=1
1 JEŚLI k>O WYKONUJ:
          ⎧JEŚLI i(s_k)=1 TO: ⎧OKREŚL U(s_k)
          ⎪                   ⎨WYKONAJ PROC.PRZEJŚCIE
          ⎪                   ⎩IDŹ DO 1
          ⎩INACZEJ:⎧JEŚLI i(s_k) < U(s_k) WYKONUJ:
                   ⎪                      ⎧PROCEDURA PRZEJŚCIE
                   ⎪                      ⎩IDŹ DO 1
                   ⎩INACZEJ:⎧k:=k-1
                            ⎩IDŹ DO 1
INACZEJ: STOP (nie znaleziono rozwiązania)
```

Procedura PRZEJŚCIE

```
x:=f(s_k, u_i(s_k))
d_k:=u_i(s_k)
i(s_k):=i(s_k)+1
count:=count+1
JEŚLI s ∈ S_F TO: ⎧DRUKUJ (s_1,s_2...s_k,s), (d_1,d_2,..d_k)
                  ⎩STOP
INACZEJ: ⎧k:=k+1
         ⎨JEŚLI k ≤ R TO: s_k:=s
         ⎩INACZEJ: k:=k-1
```

Efektywność procedury silnie zależy od stosunku głębokości poszukiwań R do długości L poszukiwanego ciągu. Musi zachodzić $R \geq L$, lecz równocześnie ze wzrostem R liczba badanych węzłów wzrasta wykładniczo. Niestety na ogół nie istnieją sposoby dokładnego szacowania L i wybór R jest dokonywany zupełnie arbitralnie, raczej w zależności od będącej w dyspozycji mocy obliczeniowej.

4. Przegląd grafu Berge'a

Łatwo pokazać, że takie podejście dopuszcza możliwość wielokrotnego przeglądania (i generowania!) końcowych odcinków pewnych dróg. Dokładnie, odcinek rozpoczynający się od wierzchołka s występuje w analizie tyle razy, ile istnieje różnych dróg bez cykli o długości $1<R$ prowadzących do wierzchołka s od wierzchołka początkowego. W celu uniknięcia wielokrotnego analizowania tych samych odcinków dróg zaproponujemy metodę generowania i przeglądania nie drzewa, lecz zdefiniowanego powyżej grafu przejść procesu. Istota jej polega na sprawdzaniu, czy nowo generowany stan kiedykolwiek (a nie tylko na aktualnej drodze) wystąpił. Jeśli tak, to następuje powrót do stanu poprzedniego i sprawdzenie następnego sterowania. Podejście takie wymaga organizacji zbioru zbadanych wierzchołków w sposób umożliwiający szybkie ustalenie, czy nowy wierzchołek już wcześniej wystąpił i dołączenie go do zbioru w przypadku odpowiedzi negatywnej. Organizacja zbioru stanów oraz procedura jego przeglądu i aktualizacji będą opisane w dalszej części opracowania jako procedura MEMORY.
Procedura MEMORY sprawdza, czy nowo obliczony stan był już wcześniej wygenerowany. Jeśli tak, to zmienna logiczna bl=FALSE. Jeśli nie, to zmienna bl=TRUE, zaś stan jest dołączany do zbioru w taki sposób, aby zachowany pozostał porządek leksykograficzny zbioru.

Przedstawimy teraz ALGORYTM 2 wyznaczający rozwiązanie dopuszczalne wykorzystujący porównywanie stanów (procedurę MEMORY).
Przyjęte oznaczenia:
bl - zmienna logiczna obliczana w procedurze MEMORY o znaczeniu określonym powyżej,
pozostałe nazwy są takie same jak w ALGORYTMIE 1.

ALGORYTM 2
$k:=1$
$s_k:=s_0$
DLA $k=1,2,..R$ $i(s_k):=1$
JEŚLI $k>0$ TO:
1. JEŚLI $i(s_k)=1$ TO: OKREŚL $U(s_k)$
   WYKONAJ PROC. TRIAL
   IDŹ DO 1
   INACZEJ: JEŚLI $i(s_k)\le \bar{U}(s_k)$ TO:
      WYKONAJ PROC.TRIAL
      IDŹ DO 1
   INACZEJ: $k:=k-1$
      IDŹ DO 1
INACZEJ: STOP (nie znaleziono rozwiązania)

Procedura TRIAL
$s:=f(s_k,u_{i(s_k)})$

$i(s_k):=i(s_k)+1$

JEŚLI $s \in S_P$ TO: { DRUKUJ $(s_1,s_2,\ldots s_k,s)$, $(d_1,d_2,\ldots d_k)$
{ STOP

INACZEJ: WYKONAJ PROC.MEMORY

{ JEŚLI bl=FALSE TO:

{ JEŚLI $i(s_k) \leq U(s_k)$ TO WYKONAJ PROC.TRIAL

{ INACZEJ $k:=k-1$

INACZEJ: { $d_k:=u_{i(s_k)}$

$k:=k+1$

JEŚLI $k \leq R$ TO $s_k:=s$

INACZEJ: $k:=k-1$

Stosowanie takiej metody jest celowe, gdy:

a) Średnia ilość bezpośrednich poprzedników wierzchołka jest odpowiednio duża lub

b) ilość poprzedników jest znaczna dla wierzchołków występujących w początkowej części ciągu.

Własności te powodują:

w przypadku a - zwiększenie wielokrotności generowania tych samych odcinków dróg,

w przypadku b - zwiększenie długości powtarzanych odcinków dróg.

Ponadto, argumentem przemawiającym za stosowaniem takiej metody jest fakt, że dla pewnych procesów generowanie stanu wymaga pewnej ilości obliczeń; między innymi ma to miejsce w przypadku symulacji asynchronicznej.

5. Numeryczny zapis grafu Berge'a

Metody przeszukiwania grafu Berge'a wymagają m.in. rozwiązania problemu ewidencjonowania przebytych węzłów grafu w taki sposób, aby z jednej strony opisać połączenia węzła z innymi węzłami grafu, a z drugiej umożliwić generowanie fizycznych opisów następników węzła i efektywnie porównywać je z takimi opisami wszystkich już przebytych węzłów grafu. Rozwinięta w ramach niniejszego opracowania metoda zapisu grafu Berge'a spełnia powyższe postulaty.

Metoda polega na umieszczeniu informacji o każdym napotkanym węźle badanego grafu Berge'a w dwóch różnych zbiorach. W zwykłym zbiorze sekwencyjnym umieszczone są informacje opisujące strukturę połączeń grafu, sterowania dopuszczalne $U(s)$ oraz informacje związane ze stanem przeszukiwania grafu. Natomiast fizyczny opis stanu wraz z numerem (nazwą) węzła umieszczony jest w zbiorze uporządkowanym leksykograficznie i pogrupowanym w rekordy (podzbiory) o stałej długości z adresacją względną. Procedura obsługująca ten zbiór realizuje zadanie ustalenia lokalizacji dla zadanego opisu stanu i umieszczenia tam opisu wraz z odpowiednią aktualizacją zbioru. lub podania informacji zwrotnej o istnieniu opisu identycznego z zadanym z podaniem numeru istniejącego węzła. Lokalizacja odbywa się dwuetapowo, metodą połowienia przedziałów. W pierwszym etapie korzystając z informacji podanej w ogranicznikach rekordów (wskaźników) ustalany jest rekord właściwy, a następnie wyznaczone jest miejsce wewnątrz rekordu. Podział zbioru na rekordy radykalnie zmniejsza liczbę przemieszczeń (relokacji) koniecznych dla aktualizacji zbioru, czyli przemieszczeń elementów występujących za nowo wprowadzonym elementem (w uporządkowaniu leksykograficznym) niezbędnych dla zrobienia mu miejsca. Rekordy przeciętnie wypełnione są w 75%, toteż wystarcza przemieszczanie części zbioru w obrębie jednego rekordu. Jeśli dany rekord został całkowicie wypełniony, połowa jego zawartosci zostaje przeniesiona tworząc nowy rekord.

Zbiór wraz z obsługującym go algorytmem charakteryzują następujące parametry:

- przeciętna liczba porównań przy lokalizacji, bliska absolutnemu optimum $l = \log_2 n$ - gdzie n - liczba elementów,
- przeciętna liczba przemieszczeń przy długości rekordu m, $n \gg m$

$$p = \frac{1}{2} \cdot \frac{3}{4}\, m + \frac{\frac{m}{2}}{m} = \frac{3}{8}\, m + \frac{1}{2}$$

Opis procedury MEMORY

Zakłada się, że w zbiorze stanów S określony jest porządek leksykograficzny.

Parametry: s - badany stan (wierzchołek),
nr - numer węzła,
bl - wskaźnik odnalezienia stanu;

zmienne: Szafa (NxMx2), wykaz (Nx3), Ip - liczba rekordu
Szafa (•,•,1) - opisy wierzchołków (stanów),
Szafa (•,•,2) - numery węzłów (nazwy),
Wykaz (•,1) - opisy ostatnich węzłów rekordów (ograniczników),
Wykaz (•,2) - liczba pozycji w rekordzie M,
Wykaz (•,3) - pozycja rekordów w Szafie (adres względny).

```
PROCEDURA MEMORY
bl:=FAŁSZ
"wyznaczenie numeru rekordu n"
a:=0, b:=Ip+1
PÓKI b-a > 1 ORAZ NIE bl WYKONAJ:
   { n:=ENTIER (a+b)/2
   { JEŚLI s=Wykaz (n,1) TO m:=Wykaz (n,3); bl:=PRAWDA;
   {                                        POWRÓT
   { INACZEJ { JEŚLI s > Wykaz (n,1) TO a:=n
             { INACZEJ b:=n;
n:=b;
```

```
"wyznaczenie pozycji w rekordzie m"
a:=0, b:=Wykaz (n,2), n1:=Wykaz (n,3)
PÓKI b-a >1 ORAZ NIE bl WYKONAJ:
    {m:=ENTIER (a+b)/2
    {JEŚLI s=Szafa(n1,m,1) TO bl:=PRAWDA; POWRÓT
    {INACZEJ {JEŚLI s > Szafa (n1,m,1) TO a:=m
             {INACZEJ b:=m
m:=b;
"przesuniecie elementów rekordu i wpisanie elementu s"
DLA i=Wykaz (n,2) COFAJ DO m
DLA j=1 DO 3
Szafa (n1,i+1,j):=Szafa(n1,i,j);
Szafa(N1,m,1):=s
Wykaz (n,2):=Wykaz (n,2)+1;
nr:=nr+1
Szafa (n1,m,2):=nr;
"ewentualne rozdzielenie rekordu"
JEŚLI Wykaz (n,2)=M  TO
DLA i=1 DO M/2
DLA j=1 DO 3
Szafa (Ip,i,j):=Szafa (n1,i+M/2,j);
DLA i=Ip COFAJ DO n+1
DLA j=1  DO  3
Wykaz (i,j):=Wykaz (i-1,j);
Wykaz (n,1):=Szafa (n1,M/2,1);
Wykaz (n,2):=Wykaz (n+1,2):=M/2;
Wykaz (n,3):=Ip;
KONIEC PROCEDURY
```

LITERATURA

[1] Deo N.: Teoria grafów i jej zastosowania w technice i informatyce, Prentice - Hall 1974, Wyd.pol. PWN, Warszawa 1980.

[2] Dudek-Dyduch E.: Symulation of some class of discrete manufacturing processes, Proc.of European Congress on Simulation, Praga 1987.

[3] Dudek-Dyduch E., Dyduch T.: Scheduling some class of discrete processes Proc.of 12th I MACS World Congress on Scientific Computation, Paris 1988.

[4] Kazimierczak J.: Teoria gier w cybernetyce, WP, Warszawa 1973.

[5] Owen G.: Teoria gier. Londyn 1968. Wyd.pol. PWN, Warszawa 1975.

[6] Reingold E., Nievergelt J., Deo N.: Algorytmy kombinatoryczne, Prentice - Hall 1977, Wyd.pol. Warszawa 1985.

[7] Komputer no 5 1986: Programowanie gier logicznych, metody heurystyczne, Kraszek J.

Recenzent: Doc.dr h.inż.M.Zaborowski

Wpłynęło do Redakcji do 1988-04-30.

ОПРЕДЕЛЕНИЕ ДОПУСКАЕМЫХ РЕШЕНИЙ ДЛЯ ПРОЦЕССОВ МОДЕЛИРУЕМЫХ ГРАФОМ БЕРЖЕ

Р е з ю м е

В работе обсуждается вопрос управления дискретными производственными процессами. Поиски допускаемого управления представлены как поиски в соответствующе сформулированном графе допускаемых переходов. Даются два алгоритма нахождения допускаемых управлений. Первый - основанный на генерировании дерева и второй - анализирующий граф Берже, но использующий процедуру ведения учёта исследованных раньше вершин. Представлена организация множества этих вершин и процедура быстрого просмотра и его моделирования.

SEARCHING FEASIBLE SOLUTION OF PROCESSES MODELLED BY MEANS OF BERGE'S GRAPH

S u m m a r y

The paper deals with the control of discrete manufacturing processes, especially with searching feasible control decisions. Two algorithms for finding the feasible solutions are presented. The first consists in generating a tree graph, the second analises a Berge's graph and utilizes the special procedure that keeps a record of the tested vertixes. The procedure is presented too.