ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2000
Seria: AUTOMATYKA z. 129 Nr kol. 1474

Paweł DĄBROWSKI, Czesław SMUTNICKI
Politechnika Wrocławska

MINIMALIZACJA CZASU CYKLU W DYSKRETNYCH PROCESACH PRODUKCYJNYCH

Streszczenie. Rozważamy problem minimalizacji czasu cyklu w dyskretnych procesach produkcyjnych. Podano i przedyskutowano różne definicje czasu cyklu. Własności tej klasy problemów są omówione na podstawie przykładowych zagadnień. Podane są algorytmy rozwiązywania z uwzględnieniem szczególnych cech problemów cyklicznych. Porównano wybrane problemy cykliczne z klasycznymi w celu wykazania istotnych różnic w rozwiązywaniu omawianych problemów.

MINIMISING CYCLE TIME IN DISCRETE PRODUCTION PROCESSES

Summary. The paper deals with the problem of minimising cycle time in discrete production processes. There have been given and discussed various definitions of cycle time. Properties of this class of problems have been presented on the base of case studies. Solution algorithms have been presented taking account of special features of cyclic problems. Comparison of cyclic problems with classical ones has been carried out in order to pinpoint main differences.

1. Definicja czasu cyklu

Rozważmy dyskretny system produkcyjny, w którym przetwarzany jest zbiór zadań N typów, przy czym liczba zadań poszczególnych typów jest większa od zera. Jeśli istnieje liczba q różna od jeden, będąca wspólnym dzielnikiem liczby zadań poszczególnych typów, możemy podzielić zbiór zadań na q podzbiorów identycznych pod względem liczby elementów i ich typów. Podzbiór taki, zwany dalej MPS (ang. *Minimal Part Set*), ma takie same proporcje co cały zbiór produkcyjny, przy czym rozmiar partii jest nieistotny.

Rozważmy klasę cyklicznych uszeregowań zdefiniowaną w następujący sposób: zadania należące do pewnego MPS przechodzą przez system w ustalonym porządku, poprzedzane przez zadania poprzedniego MPS w tej samej kolejności. Pojedynczy MPS ustala cykl produkcyjny. Naszym zadaniem jest dobrać takie uszeregowanie zadań wewnątrz pojedynczego MPS, by przepustowość systemu była największa. Bardziej formalnie, jeśli zdefiniujemy S_{ij}^k jako moment rozpoczęcia wykonywania zadania j z k-tego MPS na maszynie i, to możemy powiedzieć, że stan ustalony został osiągnięty dla k-tego MPS, gdy

$$S_{ij}^{k+1} = S_{ij}^k + T \qquad i = 1,\dots,m, \; j = 1,\dots,n,$$

gdzie m - liczba maszyn w systemie, n - liczba zadań wchodzących w skład pojedynczego MPS, T - stała, którą dalej będziemy nazywać czasem cyklu.

W stanie ustalonym momenty rozpoczęcia odpowiadających sobie zadań z kolejnych MPS różnią się o stałą wartość dla wszystkich zadań i maszyn. Oznacza to, że względny czas momentów rozpoczęcia wykonania zadań wchodzących w skład pojedynczego MPS jest taki sam dla wszystkich MPS-ów. Odpowiednio, czas cyklu możemy zdefiniować następująco:

$$T = \lim_{k\to\infty} \frac{S_{11}^k - S_{11}^1}{k-1}.$$

Ponieważ stan ustalony może zostać osiągnięty w skończonej liczbie cykli, rozpatrywać będziemy system znajdujący się już w stanie ustalonym. Wtedy czas cyklu zdefiniujemy jako

$$T = S_{11}^{k+1} - S_{11}^k,$$

przy założeniu, że $S_{ij}^{k+1} - S_{ij}^k = const$, dla $i = 1,\dots,m$, $j = 1,\dots,n$. Taka definicja jest prostsza dla analizy systemów cyklicznych, gdyż ogranicza nasze zainteresowanie w zasadzie tylko do jednego cyklu, co znacznie zmniejsza rozmiar problemu szeregowania.

Naszym celem będzie znalezienie takiego uszeregowania zadań z MPS, aby czas cyklu był minimalny, a co za tym idzie - przepustowość systemu była największa. Ponieważ cykl produkcyjny zawiera zwykle niewielką liczbę elementów, korzystne jest zastosowanie do rozwiązywania algorytmów dokładnych, bowiem nawet niewielka poprawa rozwiązania zwielokrotniona w kolejnych cyklach rekompensuje poniesione nakłady obliczeniowe.

2. Problem jednomaszynowy z przezbrojeniami

Jednym z prostszych problemów cyklicznych jest problem jednomaszynowy z przezbrojeniami. Oznaczmy przez N zbiór zadań, które mają zostać wykonane na tej maszynie. Zakładamy, że zadania są niepodzielne - raz rozpoczęte zadanie nie może zostać przerwane i zajmuje maszynę dopóki nie zakończy się jego wykonywanie - dokładnie przez czas p_j - czas wykonywania zadania j. W tym przypadku rozwiązanie dane jest w postaci permutacji π elementów należących do jednego MPS przy założeniu, że zadanie $\pi(j)$ rozpoczyna się bezpośrednio po zakończeniu przezbrojenia maszyny po wykonaniu zadania

$\pi(j-1)$ z permutacji π. Czasy przezbrojeń $s_{\pi(j),\pi(j+1)}$ są zależne od pary następujących po sobie w permutacji zadań $\pi(j)$, $\pi(j+1)$. Ponieważ w systemie znajduje się tylko jedna maszyna, czas cyklu jest różnicą pomiędzy momentami rozpoczęcia pierwszych zadań z następujących po sobie MPS, czyli

$$T = S^{k+1}_{\pi(1)} - S^{k}_{\pi(1)},$$

przy ograniczeniach

$$S^{k}_{\pi(j+1)} \geq S^{k}_{\pi(j)} + p_{\pi(j)} + s_{\pi(j),\pi(j+1)}, \quad j = 1,\ldots,n-1,$$

$$S^{k+1}_{\pi(1)} \geq S^{k}_{\pi(n)} + p_{\pi(1)} + s_{\pi(n),\pi(1)},$$

gdzie p_j oznacza czas wykonywania operacji j. Bez straty ogólności możemy oznaczyć $S^{k+1}_{\pi(1)}$ jako $S^{k}_{\pi(n+1)}$. Dzięki temu możemy nie uwzględniać indeksu k określającego numer MPS. Definicja problemu przyjmie następującą postać:

$$T = S_{\pi(n+1)} - S_{\pi(1)},$$

$$S_{\pi(j+1)} \geq S_{\pi(j)} + p_{\pi(j)} + s_{\pi(j)\pi(j+1)}, \quad j = 1,\ldots,n,$$

gdzie $s_{\pi(n),\pi(n+1)} = s_{\pi(n),\pi(1)}$.

Ponieważ zadania rozpoczynają się bezpośrednio po zakończeniu przezbrojenia, wartość kryterium obliczymy korzystając ze wzoru:

$$T = \sum_{j=1}^{n-1}\left(p_{\pi(j)} + s_{\pi(j),\pi(j+1)}\right) + p_{\pi(n)} + s_{\pi(n),\pi(1)} = \sum_{j=1}^{n} p_{\pi(j)} + \sum_{j=1}^{n-1} s_{\pi(j),\pi(j+1)} + s_{\pi(n),\pi(1)}.$$

Łatwo zauważyć, że człon $\sum_{j=1}^{n} p_{\pi(j)}$ jest stały i nie zależy od permutacji, zatem usuwając go otrzymamy kryterium równoważne

$$T = \sum_{j=1}^{n-1} s_{\pi(j),\pi(j+1)} + s_{\pi(n),\pi(1)}.$$

Nietrudno zauważyć, że problem ten jest znanym zadaniem komiwojażera (TSP) i jest także równoważny minimalizacji maksymalnego terminu zakończenia wykonywania zadań z MPS.

3. Czas cyklu w komórce produkcyjnej

Innym "względnie prostym" problemem należącym do omawianej klasy jest zagadnienie minimalizacji czasu cyklu pojedynczego stanowiska (komórki) złożonego z m maszyn równoległych, identycznych funkcjonalnie; zbiór maszyn oznaczmy przez M. Każde zadanie jest niepodzielne i przechodzi przez dokładnie jedną, dowolną maszynę z komórki. Rozwiązanie problemu ma postać pary wektorów (P,S), $P=[P_1,P_2,\dots P_n]$, $S=[S_1,S_2,\dots S_n]$, gdzie P_j oznacza numer maszyny, na której ma zostać wykonane zadanie j, a S_j oznacza moment rozpoczęcia wykonywanie zadania j. W celu określenia obciążenia maszyn dokonujemy podziału zbioru N na m podzbiorów $N_i \subset N$; każdy z nich jest skojarzony z odpowiednią maszyną $i \in M$. Uporządkowanie zadań na maszynie będzie zatem dane przez permutację $\pi_i = (\pi_i(1), \pi_i(2), \dots, \pi_i(n_i))$, gdzie n_i oznacza liczebność podzbioru N_i - liczbę zadań przyporządkowanych do maszyny i.

W sytuacji występowania więcej niż jednej maszyny pojawia się problem określania wpływu obciążenia poszczególnych maszyn na wartość kryterium. Zgodnie z przyjętą przez nas definicją czasu cyklu interesuje nas okres czasu, jaki upływa między rozpoczęciem wykonywania pierwszego zadania jednego MPS a momentem rozpoczęcia wykonywania pierwszego zadania z następnego MPS. Ponieważ rozpatrujemy system znajdujący się w stanie ustalonym, kryterium może być wyznaczone na podstawie dowolnych dwóch odpowiadających sobie zadań z kolejnych cykli produkcyjnych. W komórce produkcyjnej zależności kolejnościowe występują jedynie pomiędzy zadaniami z jednej maszyny. Aby wyznaczyć wartość kryterium, skupimy się jedynie na pierwszych zadaniach $\pi_i(1)$ ze wszystkich maszyn. Cykle produkcyjne następują bezpośrednio po sobie, bez przestoju. Istnieje zatem maszyna krytyczna i, na której bezpośrednio po zakończeniu wykonywania ostatniego zadania z pierwszego cyklu rozpoczyna się pierwsze zadanie w cyklu następnym.

$$S^{k+1}_{\pi_i(1)} = C^{k}_{\pi_i(n_i)},$$

gdzie C^k_j - moment zakończenia wykonywania zadania j z cyklu k. Ponieważ czas cyklu możemy określić

$$T = S^{k+1}_{\pi_i(1)} - S^{k}_{\pi_i(1)},$$

po przekształceniu otrzymujemy:

$$T = C^{k}_{\pi_i(n_i)} - S^{k}_{\pi_i(1)}$$

Czas cyklu jest zatem równy okresowi, jaki maszyna potrzebuje na wykonanie wszystkich przypisanych jej zadań z pojedynczego MPS i możemy go wyliczyć jako:

$$T = \sum_{j=1}^{n_i} p_{\pi_i(j)}.$$

Ponieważ suma czasów wykonywania zadań na pozostałych maszynach jest mniejsza lub równa czasowi cyklu, kryterium możemy obliczyć w następujący sposób:

$$T = \max_{i \in M} T_i, \text{ gdzie } T_i = \sum_{j=1}^{n_i} p_{\pi_i(j)}.$$

Nietrudno zauważyć, że wartość kryterium zależy jedynie od przydziału zadań do maszyn, a nie zależy od kolejności wykonywania zadań na maszynach. Problem ten jest równoważny problemowi minimalizacji terminu momentu zakończenia wykonywania zadań.

Problem minimalizacji czasu cyklu w komórce produkcyjnej komplikuje się, jeśli uwzględnimy występowanie czasów przezbrojeń. Rozważania na temat metody wyznaczania kryterium wymagają wprowadzenia niewielkich zmian. Pomiędzy momentem zakończenia wykonywania ostatniego zadania jednego cyklu a momentem rozpoczęcia wykonywania pierwszego zadania cyklu następnego na tej samej maszynie musi upłynąć dodatkowo czas potrzebny na przezbrojenie maszyny $s_{\pi_i(n_i)\pi_i(1)}$. W skład czasu potrzebnego maszynie na wykonanie wszystkich przypisanych jej zadań wchodzą nie tylko czasy wykonywania tych zadań, lecz także czasy przezbrojeń między nimi.

$$T_i = \sum_{j=1}^{n_i - 1} \left(p_{\pi_i(j)} + s_{\pi_i(j)\pi_i(j+1)} \right) + p_{\pi_i(n_i)} + s_{\pi_i(n_i)\pi_i(1)}, \text{ oraz } T = \max_{i \in M} T_i.$$

Stąd wyraźnie widać, że na wartość kryterium istotny wpływ ma także kolejność, w jakiej będą wykonywane zadania - decyduje to o sumarycznym czasie przezbrojeń. Z tego powodu problem ten jest trudny i złożony. Oprócz podziału zbioru zadań na powiązane z maszynami podzbiory musimy także ustalić optymalną kolejność wykonywania zadań. Zakładając, że nie występują żadne zależności kolejnościowe pomiędzy zadaniami, każda z maszyn może być traktowana oddzielnie i problem ustalenia kolejności zadań na maszynie może być traktowany jak omawiany problem minimalizacji czasu cyklu na jednej maszynie z przezbrojeniami.

4. Klasyczny problem przepływowy z czasem cyklu

Aby zrozumieć główne różnice między problemami klasycznymi, w których rozpatrywany jest cały zbiór przetwarzanych zadań, a problemami cyklicznymi, w których skupiamy się jedynie na zadaniach wchodzących w skład jednego MPS, przyjrzyjmy się problemowi przepływowemu. W celu porównania rozpatrywać będziemy NP-trudny problem przepływowy z kryterium $C_{\max}$ - maksymalny czas zakończenia wykonywania zadań.

W problemie przepływowym zadania ze zbioru N przechodzą kolejno przez wszystkie maszyny ze zbioru M, przy czym kolejność maszyn jest ustalona i jednakowa dla wszystkich zadań. Zadania są niepodzielne. Zadanie j nie może zostać rozpoczęte na maszynie i, dopóki nie zostanie ono zakończone na maszynie $i-1$. Kolejność wykonywania zadań na poszczególnych maszynach może być jednakowa - problem permutacyjny lub dowolna - problem niepermutacyjny. Dalsze rozważania ograniczymy do problemu permutacyjnego, w którym rozwiązanie jest reprezentowane permutacją π elementów ze zbioru N, oznaczającą kolejność wykonywania zadań na maszynach. Matematycznie problem definiujemy następująco

$$S_{i,\pi(j+1)} \geq S_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\ldots,m, \ j = 1,\ldots n-1,$$

$$S_{i+1,\pi(j)} \geq S_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\ldots,m-1, \ j = 1,\ldots,n,$$

$$C_{\max}(\pi) = \max_{j \in N, i \in M} C_{i,j}.$$

Należy wyznaczyć permutację, która zminimalizuje wartość $C_{\max}(\pi)$.

W problemie cyklicznym pojawiają się dodatkowe ograniczenia wynikające z zależności kolejnościowych wykonywania zadań należących do następujących po sobie cykli produkcyjnych. Zmienia się także postać kryterium, mianowicie:

$$S^k_{i,\pi(j+1)} \geq S^k_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\ldots,m, \ j = 1,\ldots n-1,$$

$$S^k_{i+1,\pi(j)} \geq S^k_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\ldots,m-1, \ j = 1,\ldots,n,$$

$$S^{k+1}_{i,\pi(1)} \geq S^k_{i,\pi(n)}, \quad i = 1,\ldots,m,$$

$$T = S^{k+1}_{1,\pi(1)} - S^k_{1,\pi(1)}.$$

Podobnie jak postępowaliśmy rozwiązując problem z poprzedniego rozdziału, tutaj także skupimy naszą uwagę na pierwszym zadaniu z permutacji na każdej maszynie. Różnica między czasami rozpoczęcia wykonywania pierwszego zadania na maszynach w kolejnych MPS jest stała - system znajduje się w stanie ustalonym. Ponieważ kolejne cykle następują bezpośrednio po sobie, istnieje maszyna krytyczna i, na której pierwsze zadanie rozpoczyna się bezpośrednio po zakończeniu ostatniego cyklu.

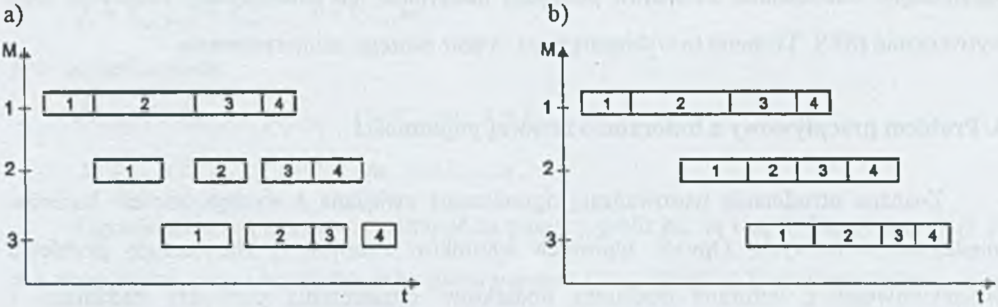
$$S^{k+1}_{i,\pi(1)} = C^k_{i,\pi(n)}.$$

Możemy zatem zapisać, że na maszynie krytycznej zachodzi zależność:

$$T = C_{i,\pi(n)} - S_{i,\pi(1)}.$$

Ponieważ różnica ta na pozostałych maszynach będzie nie większa niż dla maszyny krytycznej, możemy obliczyć czas cyklu z następujących zależności:

$$T = \max_{i \in M} T_i, \quad \text{gdzie} \quad T_i = C_{i,\pi(n)} - S_{i,\pi(1)}.$$



Rys.1. Wykres Gantta dla problemu przepływowego a) C_{max}, b) czas cyklu
Fig 1. Gantt chart for flow shop problem a) C_{max}, b) cycle time

Widzimy, że analogicznie przy optymalizacji pracy komórki produkcyjnej tutaj także czas cyklu zależy od obciążenia maszyny w ramach jednego MPS. Dla ilustracji posłużymy się przykładem z rys.1, gdzie zostały pokazane rozwiązania problemów C_{max} oraz problemu cyklicznego dla permutacji naturalnej. W komórce produkcyjnej minimalizowaliśmy obciążenie maszyn decydując o przydzieleniu zadania do konkretnej maszyny. W przypadku problemu przepływowego przez każdą maszynę przechodzą wszystkie zadania, a czas ich wykonywania jest ustalony. Czas zajętości maszyny możemy rozłożyć na następujące składniki:

$$T_i = \sum_{j=1}^{n} p_{i,\pi(j)} + \sum_{j=1}^{n-1} e_{i,\pi(j),\pi(j+1)},$$

gdzie $e_{i,\pi(j),\pi(j+1)} \geq 0$ - czas przestoju maszyny i po zakończeniu zadania j z permutacji π, a przed rozpoczęciem zadania $j+1$. Ponieważ suma czasów wykonywania zadań na maszynie jest stała i niezależna na permutacji, zmniejszenie obciążenia maszyny możemy uzyskać jedynie przez zmniejszenie bezużytecznego czasu maszyny. W problemie C_{max} wartości te są zależne od permutacji, ponieważ rozwiązanie tego problemu należy do grupy rozwiązań dosuniętych w lewo, co wynika z charakteru zastosowanego kryterium. Rozwiązania w problemie minimalizacji czasu cyklu nie mają tej cechy. Zatem, jak widać na rys.1b, możemy zmniejszyć (praktycznie do zera) bezużyteczny czas maszyny przez odpowiednie opóźnienie rozpoczęcia wykonywania zadań, niezależnie od uszeregowania. Stąd jest w problemie minimalizacji czasu cyklu każda permutacja optymalna, zaś potrzebne jest tylko dobranie odpowiednich terminów rozpoczęcia wykonywania zadań. Terminów takich jest nieskończenie wiele i możemy je generować niezależnie dla każdej maszyny, spełniając jedynie ograniczenia problemu. Wyboru odpowiedniego rozwiązania dokonujemy, kierując się dodatkowym kryterium, np. ograniczając składowanie elementów pomiędzy maszynami lub minimalizując całkowity czas wytwarzania MPS. Techniki te wybiegają poza obszar naszego zainteresowania.

5. Problem przepływowy z buforami o zerowej pojemności

Znaczne utrudnienie wprowadzają ograniczenia związane z występowaniem buforów międzystanowiskowych. Oprócz typowych warunków znanych z klasycznego problemu przepływowego z buforami dochodzą dodatkowe ograniczenia pomiędzy zadaniami z kolejnych cykli produkcyjnych, mające istotny wpływ na harmonogram wewnątrz pojedynczego MPS. Zajmiemy się tutaj najprostszym przypadkiem ze wspomnianej grupy, a mianowicie problemem przepływowym z buforami o zerowej pojemności. Matematycznie ograniczenia problemu zapisujemy następująco:

$$S_{i,\pi(j+1)} \geq S_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\dots,m, \ j = 1,\dots,n-1,$$

$$S_{i+1,\pi(j)} \geq S_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\dots,m-1, \ j = 1,\dots,n,$$

$$S_{i,\pi(j+1)} \geq S_{i+1,\pi(j)}, \quad i = 1,\dots,m-1, \ j = 1,\dots,n-1,$$

$$S^{k+1}_{i,\pi(1)} \geq S^{k}_{i,\pi(n)} + p_{i,\pi(n)}, \quad i = 1,\dots,m-1,$$

$$S^{k+1}_{i,\pi(1)} \geq S^{k}_{i+1,\pi(n)}, \quad i = 1,\dots,m-1.$$

Dla wygody, w warunkach odnoszących się do zadań z jednego MPS pominęliśmy indeks cyklu. Z drugiego i trzeciego warunku otrzymujemy pierwszy

$$S_{i,\pi(j+1)} \geq S_{i+1,\pi(j)} \geq S_{i,\pi(j)} + p_{i,\pi(j)},$$

możemy zatem usunąć ograniczenie pierwsze zastępując je dwoma pozostałymi,

$$S_{i+1,\pi(j)} \geq S_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\dots,m, \ j = 1,\dots,n,$$

$$S_{i,\pi(j+1)} \geq S_{i+1,\pi(j)}, \quad i = 1,\dots,m, \ j = 1,\dots,n-1.$$

$$S^{k+1}_{i,\pi(1)} \geq S^{k}_{i+1,\pi(n)}, \quad i = 1,\dots,m.$$

Aby po przekształceniu warunki nadal odpowiadały pierwotnemu problemowi, musimy rozpatrywać dodatkowo maszynę $m+1$, dzięki której będziemy mogli zinterpretować czas zakończenia wykonywania zadań na maszynie m. Dla ułatwienia oznaczmy pierwsze zadanie z $k+1$-go cyklu jako zadanie z cyklu k-tego o numerze $n+1$. Zwiększamy wprawdzie w ten sposób rozmiar problemu, natomiast ograniczamy nasze rozważania tylko do jednego MPS. Formalnie nasz problem zapiszemy w następującej postaci:

$$T = \max_{i \in M} T_i,$$

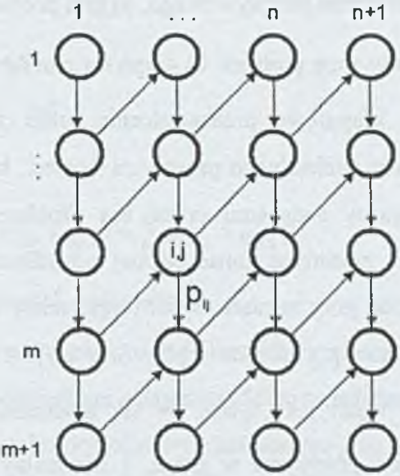
$$T_i = S_{i,\pi(n+1)} - S_{i,\pi(1)}, \quad i = 1,\dots,m,$$

przy ograniczeniach

$$S_{i+1,\pi(j)} \geq S_{i,\pi(j)} + p_{i,\pi(j)}, \quad i = 1,\dots,m, \ j = 1,\dots,n,$$

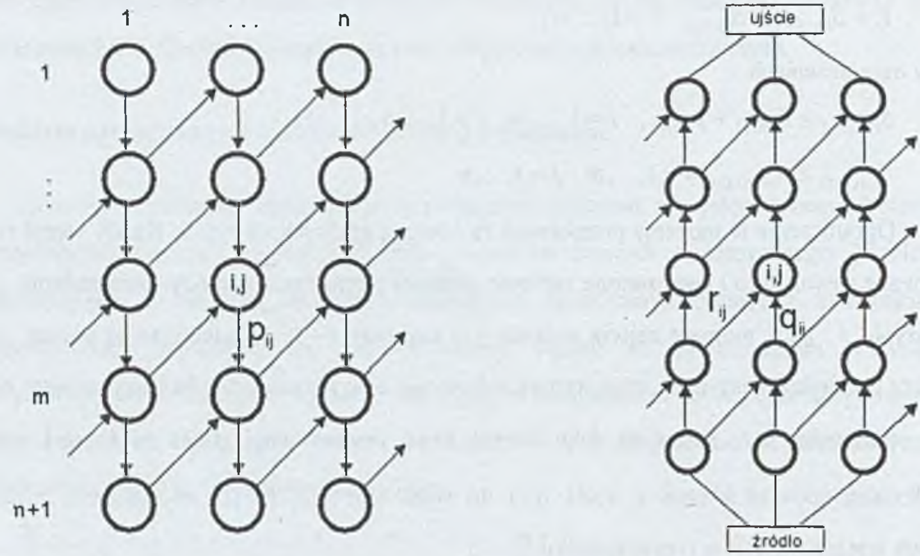
$$S_{i,\pi(j+1)} \geq S_{i+1,\pi(j)}, \quad i = 1,\dots,m, \ j = 1,\dots,n.$$

Ograniczenia te możemy przedstawić za pomocą grafu jak na rys. 2. Każdy węzeł (i, j) ma wagę równą zero i reprezentuje zarówno moment rozpoczęcia wykonywania zadania j na maszynie i, jak i moment zejścia zadania j z maszyny $i-1$. Momenty te są równe, gdyż bufory pomiędzy maszynami mają zerową pojemność. Łuk pionowy wychodzący z węzła (i, j) ma obciążenie $p_{i,j}$, natomiast łuki ukośne mają zerowe wagi. Czas cyklu jest równy najdłuższej spośród ścieżek z węzła $(i,1)$ do węza $(i, n+1)$, $i = 1,\dots,m$. Długość każdej z takich ścieżek wynosi w tym przypadku T_i.



Rys.2. Graf ograniczeń kolejnościowych dla problemu przepływowego z zerowymi buforami
Fig.2. Precedence constraints graph for flow shop with zero capacity buffers

Zauważmy, że $n+1$ kolumna w grafie jest identyczna z pierwszą pod względem obciążeń łuków, reprezentują one bowiem czasy wykonywania pierwszego zadania na maszynach. Możemy zatem dokonać "zawinięcia" grafu w ten sposób, że usuniemy węzły i łuki z ostatniej kolumny, a dochodzące do nich łuki ukośne dochodzić będą teraz do odpowiednich węzłów w kolumnie pierwszej, rys. 3a. Wartość T_i reprezentowana będzie teraz jako najdłuższa ścieżka wokół cylindra taka, że wychodzi z i kończy się w węźle $(i,1)$ oraz okrąża cylinder dokładnie jeden raz.



Rys.3. a) Przekształcony graf problemu przepływowego, b) graf problemu minimalnego przepływu
Fig.3. a) Converted graph for flow-shop problem, b) graph for min flow problem

To z pozoru zbędne i kłopotliwe przekształcenie grafu jest konieczne w celu transformacji danych do problemu minimalnego przepływu w sieci, który, jak to zostanie za chwilę pokazane, jest równoważny badanemu przez nas problemowi. Aby rozpatrywać problem minimalnego przepływu, musimy dokonać pewnej modyfikacji grafu cylindrycznego. Łuki pionowe będą wskazywać do góry zamiast na dół i będą miały wagę zerową, natomiast łuki ukośne będą miały obciążenie $p_{i,j}$. Dolne węzły $(m+1,j)$ traktować będziemy jako źródło, natomiast górne $(1,j)$ węzły jako ujście. W tak zdefiniowanym grafie obciążenie łuków jest dolnym ograniczeniem przepływu w gałęzi, a minimalny przepływ w grafie jest równoważny czasowi cyklu.

Niech $r_{i,j}$ oznacza przepływ w łukach ukośnych od węza $(i+1,j-1)$ do (i,j), a $q_{i,j}$ oznacza przepływ w łukach pionowych od $(i+1,j)$ do (i,j). Problem minimalnego przepływu definiujemy następująco:

$$\min\sum_{j=1}^{n}(q_{m,j}+r_{m,j}),$$

przy ograniczeniach dla $i = 1,\dots,m$, $j = 1,\dots,n$.

$$r_{i,j} + q_{i,j} = r_{i-1,j+1} + q_{i-1,j},$$

$$r_{i,j} \geq p_{i,j},$$

$$q_{i,j} \geq 0,$$

Problemy minimalnego przepływu i minimalizacji czasu cyklu są tak podobne, że można dokonać konwersji jednego do drugiego za pomocą liniowej transformacji. W tym celu zdefiniujmy: $r_{i,j} = S_{i+1,j} - S_{i,j}$ - czas, jaki zadanie j spędza na maszynie i oraz $q_{i,j} = S_{i,j+1} - S_{i+1,j}$ - pusty czas maszyny po zakończeniu zadania j na maszynie i.

Jeśli S są dopuszczalne, to $r_{i,j} + q_{i,j} = S_{i,j+1} + S_{i,j}$ określa czas między rozpoczęciem zadania j a rozpoczęciem zadania $j+1$ na maszynie i. Zatem $\sum_{j=1}^{n}(r_{m,j} + q_{m,j})$ jest równe $S_{m,n+1} - S_{m,1} = T$, czyli czasowi cyklu. Odwrotnie, jeśli $r_{i,j}$ i $q_{i,j}$ są dopuszczalne, definiujemy:

$$S_{i,j} = \sum_{k=1}^{j-1}(r_{1,k} + q_{1,k}) + \sum_{l=1}^{i-1} r_{l,j}.$$

Wtedy

$$S_{i+1,j} - S_{i,j} = r_{i,j} \geq p_{i,j},$$

$$S_{i,j+1} - S_{i+1,j} = q_{i,j} \geq 0$$

oraz

$$S_{i,n+1} - S_{i,1} = \sum_{k=1}^{n}(r_{1,k} + q_{1,k}) = \sum_{k=1}^{n}(r_{m,k} + q_{m,k}),$$

co dowodzi równoważności problemów.

Chociaż nasze rozważania prowadziliśmy dla systemu z buforami o zerowej pojemności, możemy zastosować omówione własności do przypadków niezerowych buforów z regułą obsługi FIFO modelując bufory jako sztuczne maszyny, na których czasy wykonywania zadań są równe zeru.

LITERATURA

1. Karmakar U.S.: Lot sizes, lead times and in-process inventories, Management Science, 1987, 33, 409-418.
2. Kubiak W.: Minimizing variation of production rates in just-in-time systems: A survery, Europen Journal of Operational Research, 1993, 259-271.

3. McCormick S.T., Pinedo M.L., Shenker S., Wolf B.: Sequencing in an Assembly Line with Blocking to Minimize Cycle Time, Operations Research 1989, 37, 925-935.
4. Porteus E.L.: Optimal lotsizing, process quality improvement and set-up cost reduction, Operations Resaerch, 1986, 34, 137-144.
5. Sawik T.: A scheduling algorithm for flexible flow lines with limited intermediated buffers. Applied Stochastic Models and Data Anlysis, 1993, 9, 127-138.
6. Sawik T.: Scheduling flexible flow lines with no in-process buffers. International Journal of Production Research, 1995, 33, 1359-1370.
7. Smutnicki C.: Optimization and control in Just-In-Time manufacturing systems, Oficyna Wydawnicza Politechniki Wrocławskiej, 1997, 37-41, 131-143.
8. Toczyłowski E.: Modelling FMS operational scheduling problems by m.-processors, Elektrotechnika, 1995, 14, 429-436.

Recenzent: Prof.dr hab.inż. M.Zaborowski

Abstract

The paper deals with the problem of minimising cycle time in discrete production processes. There have been given and discussed various definitions of cycle time. Properties of this class of problems have been presented on the base of case studies, namely, there have been discussed single-machine problem with sequence-dependent set-up times, parallel-machine problem with set-up times, flow-shop problem with infinite and finite capacity buffer between stages. Solution algorithms have been presented taking account of special features of cyclic problems. The first considered problem is equivalent to TSP, whereas the second – is equivalent to minimising the makespan while scheduling independent jobs on parallel machines. The third has trivial solution, whereas the fourth – is hard and cannot be converted to any known scheduling problem. For this problem we seek the optimal permutation of jobs in the cycle, whereas for the fixed permutation the minimum cycle time can be found by solving special max flow problem. Comparison of cyclic problems with classical ones has been carried out in order to pinpoint main differences.