ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1990
Seria: AUTOMATYKA z.100 Nr kol.1082

Adam Janiak
Politechnika Wrocławska

PROBLEM MINIMALIZACJI PRZESTOJÓW MASZYNY PRZY ZALEŻNYCH OD ZASOBÓW TERMINACH GOTOWOŚCI ZADAŃ NA PRZYKŁADZIE PROCESU PRODUKCYJNEGO POMIĘDZY STALOWNIĄ KONWERTOROWĄ A ZGNIATACZEM [1]

Streszczenie. W pracy, wychodząc z problemu minimalizacji przestojów walcarki zgniatacza w procesie produkcyjnym w hucie na odcinku: stalownia konwertorowa - zgniatacz, sformułowano problem szeregowania niepodzielnych zadań na jednej maszynie krytycznej z terminami gotowości zadań liniowo zależnymi od zasobów podzielnych w sposób ciągły (gaz). Wykazano, że problem ten jest NP-trudny nawet dla identycznych parametrów kierunkowych modeli gotowości zadań. Znaleziono szereg wielomianowo-rozwiązywalnych przypadków tego problemu. Do rozwiązania ogólnego problemu zaproponowano pięć algorytmów aproksymacyjnych, dla których przeprowadzono analizę najgorszego przypadku i analizę eksperymentalną.

1. Wstęp. Przykład zastosowań

W wielu dyskretnych i dyskretno-ciągłych procesach produkcyjnych występuje często tylko jedna maszyna krytyczna (np. z dużym kosztem zakupu lub eksploatacji), na której problem szeregowania zadań (i ewentualnego rozdziału zasobów) jest rzeczywiście istotnym problemem. Natomiast z punktu widzenia rozpatrywanych kryteriów optymalności problemy szeregowania zadań na pozostałych maszynach są praktycznie nieistotne i można w przybliżeniu rozpatrywać te maszyny jako maszyny o nieograniczonej przepustowości. Zatem w wielu praktycznych przypadkach można ograniczyć się do problemu szeregowania zadań i rozdziału zasobów tylko na jednej krytycznej maszynie. Czasy obróbki rozpatrywanych elementów na maszynach poprzedzających (w sensie technologicznego procesu) maszynę krytyczną mogą być uwzględnione w tak zwanych terminach (czasach) dostępności (gotowości) danych elementów do obróbki na maszynie krytycznej. Natomiast czasy obróbki rozpatrywanych elementów na maszynach, występujących w ciągu technologicznym po maszynie krytycznej, mogą być uwzględnione w tak zwanych czasach dostarczenia zadań, które muszą upłynąć po zejściu zadań z maszyny krytycznej, aby były one całkowicie zakończone. Czasy dostarczenia zadań - w porównaniu z terminami ich dostępności - są znacznie mniej istotne z punktu widzenia problematyki szeregowania zadań (istotna jest krytyczna maszyna)

[1] Praca częściowo finansowana w ramach problemu R.P.I.02.

i dlatego są one rzadko rozpatrywane.

Terminy dostępności zadań do obróbki na maszynie krytycznej - traktowane jako czasy obróbki rozpatrywanych elementów na maszynach poprzedzających w ciągu technologicznym maszyną krytyczną - mogą zmieniać się w sposób ciągły w określonych granicach wynikających z technologicznych uwarunkowań, zależąc od ilości zasobów podzielnych w sposób ciągły, przydzielonych tym operacjom lub maszynom, na których są one wykonywane. Przykładowo, proces produkcyjny występujący w hutach stali na odcinku produkcyjnym: stalownia konwertorowa - walcarka zgniatacz, polegający na podgrzewaniu wlewków za pomocą gazu w piecach wgłębnych, a następnie walcowaniu tych wlewków na walcarce zgniataczu, można sprowadzić do jednomaszynowego problemu szeregowania zadań na walcarce zgniataczu, z terminami dostępności zadań zależnymi od ilości przydzielonego do pieców wgłębnych podzielnego w sposób ciągły zasobu (gazu).

Proces ten krótko można scharakteryzować w następujący sposób [11,18].

Po zakończeniu cyklicznego procesu wytapiania stali w jednym z konwertorów (zwykle jest kilka konwertorów) płynny metal wlewany jest do kadzi, a następnie do wlewnic (specjalnych form), gdzie krzepnie i później zastyga. Wlewnice znajdują się na tzw. zestawach kołowych, na torach krystalizacji. Stygnięcie i starzenie się wlewków trwa od kilku do kilkudziesięciu godzin. Następnie po procesie stryperowania wlewki stygną na wolnym powietrzu, a później są odpowiednio segregowane na zestawy i transportowane do właściwej baterii pieców wgłębnych w celu ich podgrzania do wymaganej temperatury. Każdy zestaw wlewków składa się z kilku wlewków zazwyczaj o podobnej temperaturze, zbliżonym formacie i rodzaju materiału, z którego są zrobione. Każdy zestaw wlewków jest podgrzewany za pomocą gazu w oddzielnym piecu wgłębnym. Wygrzewanie wsadu wlewków w piecu wgłębnym trwa od kilku do kilkunastu godzin. Poszczególne piece wgłębne mogą być dostępne do podgrzewania w różnych momentach czasu (cykliczny proces). Następnie, poszczególne zestawy wlewków - po osiągnięciu pożądanej temperatury - są wyjmowane z pieców wgłębnych i wlewki po kolei za pomocą specjalnej suwnicy są dostarczane do walcarki zgniatacza, gdzie są walcowane na gorąco. Jeden wlewek jest walcowany około kilku minut i następnie w postaci płyty jest przekazywany do dalszej obróbki na wydział walcowni. Zwykle jest tylko jeden zgniatacz w całej linii produkcyjnej, ponieważ jest on urządzeniem bardzo drogim (znacznie droższym niż wszystkie piece wgłębne). Z powyższego powodu może on być traktowany jako maszyna krytyczna w tej linii produkcyjnej. Piece wgłębne można tutaj praktycznie potraktować jako maszyny o nieograniczonej przepustowości. Czas podgrzewania każdego zestawu wlewków w każdym z pieców wgłębnych jest odwrotnie proporcjonalny do ilości spalonego w nim gazu. Czas ten także zależy od liczby, formatu i typu surówki wlewków w rozpatrywanym zestawie, zależy on również od stanu technicznego rozpatrywanego pieca

wgłębnego. Te ostatnie zależności są zwykle uwzględniane w stałych parametrach modelu czasu podgrzewania wlewków. Każdy zestaw wlewków, podgrzewanych w oddzielnym piecu wgłębnym, tworzy oddzielne zadanie produkcyjne. Czas walcowania (na gorąco) każdego zestawu wlewków może być różny dla różnych zestawów wlewków, ponieważ zależy on od liczby, formatu, temperatury i gatunku stali wlewków w zestawie. Jednakże można go z góry łatwo ustalić i dla każdego zestawu wlewków przyjmuje się, że jest on zadany. Czas podgrzewania każdego zestawu wlewków (z uwzględnieniem czasu dostępności rozpatrywanego pieca wgłębnego oraz czasu zakrzepnięcia wlewków z rozpatrywanego zestawu, liczonych od chwili rozpoczęcia optymalizacji badanego procesu produkcyjnego) można potraktować jako termin (czas) dostępności rozpatrywanego zestawu wlewków (zadania) na walcarkę zgniatacz (w celu walcowania jej na gorąco). Maksymalne natężenie przepływu gazu, dostarczanego zarówno do każdego pieca wgłębnego, jak i do całej piecowni, jest ograniczone.

Problem tutaj polega na znalezieniu takiej kolejności walcowania wlewków z rozpatrywanych zestawów wlewków na walcarce zgniataczu i takiego rozdziału natężenia przepływu gazu (lokalnie i globalnie ograniczonego) pomiędzy rozpatrywane piece wgłębne w piecowni, aby przestoje walcarki zgniatacza były minimalne.

Można pokazać [16], że minimalizacja przestojów dowolnej maszyny jest równoważna minimalizacji czasu zakończenia wykonywania zadań na tej maszynie.

Z punktu widzenia procesów zachodzących w hutach jest to bardzo istotny problem, bowiem - jak wcześniej już wspomniano - walcarka zgniatacz jest bardzo drogim urządzeniem i przestoje w jego pracy są niesłychanie kosztowne. Godzina jego postoju kosztuje wiele milionów złotych, bowiem przestoje zgniatacza są tutaj równoznaczne z bezczynnością całej walcowni z wyjątkiem wykańczalni i pieców wgłębnych. Według zbiorczych statystyk [12] prowadzonych przez huty ponad 50% postojów zgniatacza jest spowodowane okresowym brakiem odpowiednio podgrzanych wlewków. Przestoje z winy pieców wgłębnych wynoszą od 20 do 50 godzin miesięcznie. Niewielkie nawet usprawnienia w sterowaniu tym procesem mogą przynieść znaczne efekty ekonomiczne, bowiem wielkość strumienia przepływającego w tym procesie materiału wynosi w ciągu roku kilka milionów ton stali . Podejmowano już wiele różnych prób rozwiązania tego problemu, i tak np. pewne metody symulacyjne przedstawiono w [2], natomiast algorytmy suboptymalne proponowano w [3]. Niestety żadna z tych prób nie dała w pełni zadowalających rezultatów.

Podobne problemy pojawiają się także na innych wydziałach hut, np. na wydziale kuźniczym.

Niniejsza praca poświęcona jest właśnie problemowi szeregowania zadań na jednej maszynie, przy założeniu, że terminy dostępności zadań na tę ma-

szynę liniowo zależą od podzielnych w sposób ciągły ograniczonych zasobów. Głównym zagadnieniem będzie minimalizacja czasu zakończenia wszystkich zadań na tej maszynie (równoważna minimalizacji przestojów tej maszyny).

Badanie tego problemu istotne jest nie tylko z praktycznego punktu widzenia, ale również i z teoretycznego punktu widzenia, bowiem problemy takie pojawiają się w kontekście wyznaczenia dolnych oszacowań przy rozwiązywaniu wielomaszynowych problemów szeregowania zadań i rozdziału zasobów [7].

W następnym rozdziale badany problem zostanie precyzyjnie sformułowany. W rozdziale trzecim zostanie pokazane, że problem ten jest NP-trudny (decyzyjna wersja tego problemu należy do klasy problemów NP-zupełnych). W rozdziale czwartym zostaną podane wielomianowo rozwiązywalne szczególne przypadki tego problemu oraz algorytmy optymalnego rozdziału zasobów przy ustalonej kolejności wykonywania zadań. Rozdział szósty poświęcony jest aproksymacyjnym algorytmom rozwiązania tego problemu. W rozdziale tym krótko została również przedstawiona analiza najgorszego przypadku proponowanych algorytmów oraz ich analiza eksperymentalna. W ostatnim (siódmym) rozdziale niniejszej pracy naszkicowano dalsze kierunki badań rozpatrywanej jednomaszynowej problematyki, z uwzględnieniem - w szczególności - minimalizacji zużycia całkowitej ilości zasobów przy zadanym ograniczeniu na maksymalną wartość przestojów maszyny oraz dwukryterialnego podejścia, w którym jednocześnie minimalizuje się całkowitą ilość zużytego zasobu i przestoje maszyny.

2. Sformułowanie problemu

W niniejszym rozdziale zostanie precyzyjnie sformułowany jednomaszynowy problem szeregowania z terminami dostępności zadań zależnymi od zasobów, który został opisowo przedstawiony w poprzednim rozdziale.

Dany jest zbiór n niepodzielnych zadań $J_1,\ldots,J_j,\ldots,J_n$, które mają być wykonane na maszynie o przepustowości jednostkowej (tzn. w każdej chwili czasu maszyna ta może wykonywać co najwyżej jedno zadanie). Zadanie J_j jest wykonywane na rozpatrywanej maszynie w czasie p_j, $j = 1,2,\ldots,n$. Każde zadanie J_j jest dostępne (gotowe) do wykonywania na rozpatrywanej maszynie w terminie r_j, zakłada się, że jest to czas wykonywania tego zadania na maszynie poprzedzającej maszynę rozpatrywaną. Przy czym czas ten może zmieniać się w sposób ciągły w określonych granicach wynikających z technologicznych uwarunkowań, zależąc od ilości zasobów podzielnych w sposób ciągły, przydzielonych tym zadaniom lub maszynom, na których są one wykonywane. Dalej będziemy zakładali, że zależność ta jest liniowa, tzn.:

$$r_j \triangleq r_j(u_j) \triangleq b_j - a_j u_j, \quad j = 1,2,\ldots,n \tag{1}$$

gdzie u_j jest ilością zasobu przydzieloną do zadania J_j, a_j, b_j ($a_j > 0$, $b_j > 0$) są znanymi parametrami. (Tego typu modele terminów gotowości zadań po raz pierwszy zostały wprowadzone do problemów szeregowania zadań (nawet w ogólniejszej postaci) w pracy [6]. Później w pewnym sensie podobne modele (typu koszt/czas) zostały wprowadzone w pracach innych autorów [14,15]). Ponadto będziemy zakładali, że ilość zasobów przydzielona zadaniom na maszynach poprzedzających maszynę krytyczną, jest lokalnie:

$$\alpha_j \leq u_j \leq \beta_j, \quad j = 1,2,\ldots,n \tag{2}$$

oraz globalnie:

$$\sum_{j=1}^{n} u_j \leq R \tag{3}$$

ograniczona, gdzie α_j, β_j ($0 \leq \alpha_j \leq \beta_j \leq b_j/a_j$) są zadanymi technologicznymi (lub ekonomicznymi -nasycenie) ograniczeniami na zakres ilości zasobu przydzielanego do zadania J_j, celem jego realizacji na maszynie poprzedzającej maszynę krytyczną; R jest ograniczeniem na maksymalną ilość zasobów, które można wykorzystać do realizacji wszystkich zadań na maszynach poprzedzających maszynę krytyczną. Ogólnie mogą to być zasoby takie, jak: gaz, paliwo, energia, tlen, katalizator lub ich natężenia przepływu, a także surowce i środki finansowe.

W dalszej części pracy - w celu uproszczenia zapisu - będziemy zakładali, że dolne ograniczenia na przydzielane ilości zasobu do poszczególnych zadań są równe zeru, tzn. $\alpha_j = 0$, $j = 1,2,\ldots,n$, bowiem przypadek z $\alpha_j > 0$ można łatwo sprowadzić do przypadku z $\alpha_j = 0$ poprzez odpowiednią modyfikację parametrów b_j oraz β_j, a mianowicie poprzez następujące podstawienie: $b_j := b_j - a_j\beta_j$, $\beta_j := \beta_j - \alpha_j$. Wszystkie własności i algorytmy prezentowane w niniejszym rozdziale przy założeniu, że $\alpha_j = 0$, $j = 1,2,\ldots,n$, można trywialnie uogólnić na przypadek, gdy $\alpha_j \geqslant 0$.

Zbiór wszystkich dopuszczalnych rozdziałów zasobów, tzn. rozdziałów spełniających ograniczenia (2) oraz (3), będziemy oznaczali przez U.

Niech π oznacza pewną kolejność wykonywania (uporządkowanie, permutację) zadań $J_1, J_2, \ldots, J_n$, a $J_{\pi(j)}$ zadanie, które jest na pozycji j-tej w uporządkowaniu π. Zbiór wszystkich (dopuszczalnych) kolejności wykonywania zadań π będzie oznaczony Π.

Dla zadanego sterowania dopuszczalnego, tzn. zadanej dopuszczalnej kolejności wykonywania zadań $\pi \in \Pi$ oraz zadanego dopuszczalnego rozdziału zasobu $u \in U$, czas zakończenia $C_{\pi(j)}(\pi, u)$ zadania $J_{\pi(j)}$, $j = 1,2,\ldots,n$, w π można zdefiniować w następujący rekurencyjny sposób:

$$C_{\pi(1)}(\pi,u) \triangleq r_{\pi(1)}(u_{\pi(1)}) + p_{\pi(1)}, \tag{4}$$

$$C_{\pi(j)}(\pi,u) \triangleq \max\{r_{\pi(j)}(u_{\pi(j)}),\ C_{\pi(j-1)}(\pi,u)\} + p_{\pi(j)}, \tag{5}$$

$j = 2,3,\ldots,n$. Zatem łatwo zauważyć, że czas zakończenia wykonania zadani $J_{\pi(j)}$ można wyznaczyć w następujący sposób:

$$C_{\pi(j)}(\pi,u) = \max_{1\le i\le j}\{r_{\pi(j)}(u_{\pi(j)}) + \sum_{l=i}^{j} p_{\pi(l)}\}, \tag{6}$$

a czas zakończenia wykonania wszystkich zadań, oznaczany przez $C_{max}(\pi,u)$ dla $\pi\in\Pi$ oraz $u\in U$, można wyznaczyć z następującego wzoru:

$$C_{max}(\pi,u) = \max_{1\le j\le n}\{b_{\pi(j)} - a_{\pi(j)}u_{\pi(j)} + \sum_{i=j}^{n} p_{\pi(i)}\}. \tag{7}$$

Rozpatrywany jednomaszynowy problem szeregowania zadań i rozdziału zasobów polega na znalezieniu takiego dopuszczalnego sterowania, tzn. takiej dopuszczalnej kolejności wykonywania zadań $\pi^*\in\Pi$ oraz takiego dopuszczalnego rozdziału zasobów $u^*\in U$, aby czas zakończenia wykonania wszystkich zadań $C_{max}(\pi,u)$ był minimalny, tzn. aby zachodziła następująca zależność:

$$\min_{\pi\in\Pi}\ \min_{u\in U} C_{max}(\pi,u) = C_{max}(\pi^*,u^*).$$

Oczywiste jest, że badany w niniejszej pracy problem jest uogólnieniem klasycznego jednomaszynowego problemu z ustalonymi terminami gotowości zadań, rozpatrywanego w [16], jako "dualna" wersja problemu rozpatrywanego przez Jacksona w [5].

Jak łatwo zauważyć, rozpatrywany problem dla ustalonej kolejności wykonywania zadań π redukuje się do pewnego problemu, który można sformułować jako problem programowania liniowego (w rozdziale czwartym do jego rozwiązania zostanie zaproponowany bardzo prosty algorytm o złożoności obliczeniowej $O(n^2)$). Z kolei, dla ustalonego dopuszczalnego rozdziału zasobów $u\in U$, problem ten można trywialnie rozwiązać w $O(n \log n)$ krokach za pomocą algorytmu Jacksona [5].

Jednakże w ogólnym przypadku problem ten, nawet dla identycznych parametrów kierunkowych modeli, tzn. dla $a_j = a$, $j = 1,2,\ldots,n$, staje się NP-trudny, co zostanie wykazane w następnym rozdziale.

3. Złożoność obliczeniowa

Obecnie wykażemy, że decyzyjna wersja (oznaczana dalej przez DWP) badanego (optymalizacyjnego) problemu jest NP-zupełna, gdzie DWP jest zdefiniowane w sposób następujący:

DWP: dany jest jednomaszynowy problem szeregowania niepodzielnych zadań o czasach wykonywania równych p_j, $j = 1,2,\ldots,n$, dane są modele (1) terminów dostępności zadań do wykonania z parametrami a_j, b_j, β_j, $j = 1,2,\ldots,n$ oraz liczbami R i y, gdzie n, b_j, β_j, p_j, dla $j = 1,2,\ldots,n$, są nieujemnymi liczbami całkowitymi, natomiast R, y, a_j $(j = 1,2,\ldots,n)$ są nieujemnymi liczbami wymiernymi.

Pytanie: Czy istnieje dopuszczalna kolejność wykonywania zadań $\pi \in \Pi$ oraz dopuszczalny rozdział zasobów $u \in U$, takie, że $C_{max}(\pi,u) \leq y$?

Twierdzenie 1

Decyzyjna wersja (DWP) badanego problemu jest NP-zupełna dla $a_j = a$, $j = 1,2,\ldots,n$.

Dowód: Oznaczmy przez $DW\,1||\sum T_j$ decyzyjną wersję klasycznego jednomaszynowego problemu szeregowania zadań z kryterium suma opóźnień zadań.

NP-zupełność DWP będzie dowodzona poprzez pokazanie, że $DW1||\sum T_j$ (będąca NP-zupełną [1]) jest wielomianowo transformowalna do DWP. Rozważmy $DW1||\sum T_j$: dany jest jednomaszynowy problem szeregowania zadań z m niepodzielnymi zadaniami o czasach wykonywania zadań równych q_i oraz pożądanymi terminami ich zakończenia d_i, $i = 1,2,\ldots,m$, spełniającymi następujące zależności $d_i \leq \sum_{i=1}^{m} q_i$, dana jest także liczba z (gdzie m,z,q_i,d_i, $i = 1,2,\ldots,m$, są nieujemnymi liczbami całkowitymi).

Pytanie: Czy istnieje kolejność wykonywania zadań σ, taka, że

$$\sum_{i=1}^{m} T_i \triangleq \sum_{i=1}^{m} \max\left(0,\sum_{i=1}^{1} q_{\sigma(i)} - d_{\sigma(1)}\right) \leq z \ ?$$

Odpowiednia instancja DWP jest zdefiniowana w sposób następujący: $n = m$, $p_j = q_j$, $b_j = \sum_{j=1}^{m} q_j - d_j$, $\beta_j = b_j$, $a_j = a = 1$, $j = 1,2,\ldots,n$, $R = z$, $y = \sum_{j=1}^{m} q_j$. (Transformacja ta jest w pewnym sensie podobna do pewnej transformacji rozpatrywanej w [14,15]). Obecnie pokażemy, że dla DWP zachodzi $C_{max}(\pi,u) \leq y$ (pod warunkiem, że $u \in U$), wtedy i tylko wtedy, gdy dla $DW||\sum T_i$ można znaleźć kolejność wykonywania zadań σ taką, że zachodzi: $\sum_{i=1}^{m} T_i \leq z$.

(i) Jeżeli $\sum_{i=1}^{m} T_i \leq z$ dla pewnego σ, tzn. jeśli $DW1||\sum T_i$ ma rozwiązanie, wówczas kolejność wykonywania zadań $\hat{\pi}$ oraz rozdział zasobów $\hat{u} \in U$, przy których dla DWP zachodzi:

$C_{max}(\hat{\pi},\hat{u}) \leq y$, są następujące:

$$\hat{\pi}(i) = \sigma(n-(i-1)),\ i = 1,2,\ldots,n, \tag{8}$$

$$\hat{u}_{\hat{\pi}(i)} = \max\left(0,\sum_{j=1}^{n-(i-1)} q_{\sigma(j)} - d_{\sigma(n-(i-1))}\right),\ i = 1,2,\ldots,n. \tag{9}$$

Warto zauważyć, że rozdział $\hat{u}$ jest optymalny dla $\hat{\pi}$, tzn. $\hat{u} = u^*_{\hat{\pi}}$, a dla sterowania $(\hat{\pi},\hat{u})$ zachodzi:

$$\sum_{j=1}^{n} \hat{u}_{\hat{\pi}(j)} = R = \sum_{j=1}^{n} T_{\sigma(n-(j-1))} = z \quad \text{oraz}$$

$$C_{max}(\pi,u) = \sum_{j=1}^{n} p_j = \sum_{j=1}^{m} n_j = y.$$

(ii) Na odwrót, jeśli OW $1||\sum T_i$ nie ma rozwiązania, tzn. jeśli $\sum_{i=1}^{m} T_i - z = \varepsilon > 0$ zachodzi dla wszystkich dopuszczalnych kolejności wykonywania zadań σ, wówczas łatwo zauważyć, że dla każdego uporządkowania π postaci (8)((8) wyczerpuje wszystkie możliwe dopuszczalne uporządkowania zadań) i dla każdego dopuszczalnego rozdziału zasobów $u \in U$ zachodzi: $C_{max}(\pi,u) > y$ (wskazówka: $\sum_{i=1}^{n} u_i = R < z + \varepsilon$).

Wykazanie, że OWP $\in$ NP jest trywialne.

Uwaga 1. W podobny sposób [8] można pokazać, że jeśli parametry kierunkowe modeli są dowolne, to wówczas decyzyjna wersja badanego problemu jest silnie NP-trudna. W tym przypadku pseudowielomianowa transformacja przeprowadzana jest z decyzyjnej wersji problemu $1||\sum w_j T_j$, która jest silnie NP-trudną [13].

4. Przypadki wielomianowo rozwiązywalne

Przez $u^*_\pi \in U$ będziemy dalej oznaczali optymalny rozdział zasobów dla ustalonej kolejności wykonywania zadań π, tzn. rozdział zasobów, dla którego zachodzi: $C_{max}(\pi,u^*_\pi) = \min_{u \in U} C_{max}(\pi,u)$.

Łatwo zweryfikować [10], że rozdział ten, tzn. $u^*_\pi \in U$, można uzyskać w $O(n^2)$ krokach za pomocą następującego algorytmu:

Algorytm 1

Krok 1. Podstaw $u^*_{\pi(i)} := 0$, $C_{\pi(i)} := b_{\pi(i)} + \sum_{j=i}^{n} p_{\pi(j)}$, dla $i = 1,2,\ldots,n$, oraz $I := \{\pi(i): i = 1,2,\ldots,n\}$, $l := 0$, $C_0 := 0$, $P_0 := 0$ i przejdź do kroku 2.

Krok 2. Podstaw $l := l+1$ i znajdź zbiór $P_l := \{\pi(i) \in I: C_{\pi(i)} = \min_{\pi(j) \in I} C_{\pi(j)}\}$, następnie podstaw $I := I - P_l$. Jeśli zbiór I jest pusty, wówczas podstaw $Q := P_l$ i przejdź do kroku 3; w przeciwnym wypadku wróć do kroku 2.

Krok 3. Jeśli $R = 0$ lub $l = 0$ lub $\min_{j \in Q} \{\beta_j - u^*_j\} = 0$, wówczas Stop - u^*_π jest poszukiwanym optymalnym rozdziałem zasobów dla π; w przeciwnym wypadku przejdź do Kroku 4.

Krok 4. Podstaw $x := \min\{C_q - C_p, R/\sum_{j \in Q}(1/a_j), \min_{j \in Q}\{a_j(\beta_j - u^*_j)\}\}$, gdzie q oraz p są indeksami zadań odpowiednio ze zbiorów Q oraz P_{l-1}. Następnie podstaw $u^*_j := u^*_j + x/a_j$, dla $j \in Q$, $R := R - \sum_{j \in Q} x/a_j$, $l := l-1$, $Q := Q \cup P_l$ i wróć do kroku 3.

Można łatwo pokazać, że w przypadku, gdy $b_j = b$, dla $j = 1,2,\ldots,n$, powyższy algorytm redukuje się do algorytmu (oznaczanego dalej jako Algorytm 1') o złożoności obliczeniowej $O(n)$.

Zachodzi następująca własność:

Własność 1

1° Przypadki z:

(i) $b_i = b$, $a_i = a$, $\beta_i = \beta$, $i = 1,2,\ldots,n$,
(ii) $b_i = b$, $\beta_i = \beta$, $p_i = p$, $i = 1,2,\ldots,n$,
(iii) $b_i = b$, $a_i = a$, $p_i = p$, $i = 1,2,\ldots,n$,
można rozwiązać w $O(n \log n)$ krokach szeregując zadania zgodnie, odpowiednio, z (i) nierosnącymi wartościami p_i, (ii) nierosnącymi wartościami a_i, (iii) nierosnącymi wartościami β_i i rozdzielając zasoby zgodnie z Algorytmem 1'.

2° Przypadek z $a_i = a$, $\beta_i = \beta$, $p_i = p$, $i = 1,2,\ldots,n$, można rozwiązać w $O(n^2)$ krokach szeregując zadania zgodnie z niemalejącymi wartościami b_i i rozdzielając zasoby zgodnie z Algorytmem 1.

5. Algorytmy aproksymacyjne. Analiza eksperymentalna i najgorszego przypadku

W rozdziale tym, dla badanego w niniejszej pracy problemu, przedstawimy pięć algorytmów przybliżonych (w wersji prostej i zmodyfikowanej) wraz z analizą eksperymentalną i analizą najgorszego przypadku.

Będzie stosowany następujący ogólny schemat algorytmu aproksymacyjnego:

Ogólny schemat algorytmu aproksymacyjnego A.

Krok 1. Wybierz kolejność wykonywania zadań $\pi_o^A \in \Pi$.

Krok 2. Za pomocą Algorytmu 1 wyznacz optymalny rozdział zasobu $u^*_{\pi_o^A} \in U$ dla π_o^A ($u_o^A = u^*_{\pi_o^A}$).

Dokładność powyższego podejścia aproksymacyjnego można zwiększyć poprzez dodanie następującego kroku (otrzymując w efekcie zmodyfikowane aproksymacyjne podejście M):

Krok 3. Dla terminów dostępności zadań wyznaczonych przy rozdziale zasobu u_o^A, otrzymanym w kroku 2, wyznacz za pomocą algorytmu Jacksona [5] kolejność wykonywania zadań $\pi_1^M \in \Pi$, a następnie dla π_1^M za pomocą Algorytmu 1 wyznacz optymalny rozdział zasobu $u_1^M \in U$. Wykonaj co najwyżej zadaną liczbę L kroku 3. Jeśli $C_{max}(\pi_{l+1}^M, u_{l+1}^M) = C_{max}(\pi_l^M, u_l^M)$, wówczas Stop - (π_l^M, u_l^M) oraz $C_{max}(\pi_l^M, u_l^M)$ są, odpowiednio, przybliżonym sterowaniem oraz przybliżoną wartością rozpatrywanego kryterium.

Łatwo zauważyć, że kolejne kroki zmodyfikowanego podejścia przybliżonego M dają następujące wartości rozpatrywanego kryterium:
$C_{max}(\pi_o^A, u_o^A) \geqslant C_{max}(\pi_1^M, u_o^A) \geqslant C_{max}(\pi_1^M, u_1^M) \geqslant \ldots$.

Mając na uwadze prezentowane w poprzednim rozdziale wielomianowo rozwiązywalne przypadki rozpatrywanego problemu, następujące aproksymacyjne algorytmy wyznaczania kolejności wykonywania zadań π_o^A są intuicyjnie oczywiste:

Wybierz π_o^A zgodnie z jedną z następujących reguł listowego porządkowania zadań:

π^{A1} - według nierosnących wartości p_j,
π^{A2} - według nierosnących wartości a_j,
π^{A3} - według nierosnących wartości β_j,
π^{A4} - według niemalejących wartości b_j,
π^{A5} - według niemalejących wartości $b_j - a_j\beta_j$.

Łatwo zauważyć, że złożoność obliczeniowa każdego prostego algorytmu aproksymacyjnego A_k (A_k otrzymane jest dla początkowego uporządkowania zadań π^{Ak}), k = 1,2,3,4,5, wynosi $O(n^2)$. A złożoność obliczeniowa każdego zmodyfikowanego algorytmu aproksymacyjnego M_k (M_k jest otrzymywane dla π_o^{Ak}), k = 1,2,3,4,5, wynosi $O(L\cdot n^2)$.

Warto także zauważyć, że dla każdego zmodyfikowanego algorytmu aproksymacyjnego M_k, jeśli $C_{max}(\pi_{l+1}^{Mk}, u_{l+1}^{Mk}) = C_{max}(\pi_l^{Mk}, u_l^{Mk})$, wówczas sterowanie (π_l^{Mk}, u_l^{Mk}) spełnia warunki konieczne optymalności dla rozpatrywanego problemu, tzn.

zachodzi $\min_{u\in U} C_{max}(\pi_l^{Mk}, u) = C_{max}(\pi_l^{Mk}, u_l^{Mk})$

oraz $\min_{\pi\in\Pi} C_{max}(\pi, u_l^{Mk}) = C_{max}(\pi_l^{Mk}, u_l^{Mk})$.

Analiza najgorszego przypadku

Można pokazać [8], że zachodzi następująca własność (C_{max}^{Ak} oznacza wartość $C_{max}(\pi^{Ak}, u_{\pi^{Ak}}^*)$, otrzymaną przez zastosowanie algorytmu A_k, a $C_{max}^* = C_{max}(\pi^*, u^*)$):

Własność 2

Dla każdego k = 1,2,3,4,5 zachodzi:

$$C_{max}^{Ak}/C_{max}^* \le 2 \tag{10}$$

i ograniczenie to jest najlepsze z możliwych.

Analiza eksperymentalna

Mimo iż osiągalny współczynnik najgorszego przypadku dla wszystkich pięciu algorytmów jest taki sam i wynosi 2, analiza eksperymentalna wykazała, że z praktycznego punktu widzenia, algorytmy te różnią się w sposób istotny. A mianowicie, zarówno każdy z prostych, jak i ze zmodyfikowanych algorytmów aproksymacyjnych był testowany przy użyciu tych samych 400 losowo wygenerowanych przykładów. Po sto przykładów było generowanych dla każdej liczby zadań n = 25, 50, 75 i 100. Każdy wygenerowany przykład był badany przy trzech (skrajnie) różnych wartościach globalnej ilości zasobu R, zatem - tak naprawdę - każdy z algorytmów aproksymacyjnych (prostych i zmodyfikowanych) był testowany przy użyciu tych samych 1200 przykładów. Dla każdego zmodyfikowanego algorytmu M_k przyjęto, że L = 20.

Dla każdego generowanego przykładu (j), j = 1,2,...,400 parametry $p_i^{(j)}$, $\alpha_i^{(j)}$, $\beta_i^{(j)}$, $a_i^{(j)}$, i = 1,2,...,n były generowane przez generator o rozkładzie równomiernym z przedziałów: $p_i^{(j)} \in [10,30]$, $\alpha_i^{(j)} \in [8,12]$, $\beta_i^{(j)} \in [60,90]$

$a_i^{(j)} \in [1,8]$, $b_i^{(j)} \in [180 + a_i^{(j)} \cdot \beta_i^{(j)}, 900 + a_i^{(j)} \cdot \alpha_i^{(j)}]$, charakterystycznych dla danych dotyczących wspomnianego we wstępie procesu technologicznego na odcinku produkcyjnym stalownia konwertorowa - walcarka zgniatacz w Hucie "Katowice". W rzeczywistym procesie jednorazowo rozpatruje się partię wlewków składającą się z 20 do 40 zestawów wlewków (jeden zestaw tworzy zadanie). Natomiast tutaj w celu dokładniejszego przebadania rozpatrywanych algorytmów aproksymacyjnych oprócz n = 25 i 50, zbliżonych do rzeczywistej sytuacji, będziemy badali - jak to zostało już wcześniej zaznaczone - także większą liczbą zadań, tzn. n = 75 i 100 (odpowiednio zwiększając dysponowaną do rozdziału globalną ilość zasobu). A mianowicie, każdy generowany przykład (j) był testowany dla trzech wartości R, tzn. dla $R := z \cdot R^*$, gdzie $z := 0.2, 0.5, 0.9$, a $R^* \triangleq \sum_{i=1}^{n} \beta_i^{(j)}$. Dla każdego testowanego przykładu (j) (j = 1,2,...,1200) oraz każdego algorytmu aproksymacyjnego $A := A_k, M_k$, k = 1,2,3,4,5, był wyznaczany następujący współczynnik (będący górnym oszacowaniem błędu względnego):

$$\varrho_A^{(j)} = [(C_A^{(j)} - LB^{(j)})/LB^{(j)}] \cdot 100\ \%, \tag{11}$$

gdzie $C_A^{(j)}$ oznacza wartość rozpatrywanego kryterium C_{max}, znalezioną za pomocą algorytmu A dla j-tego przykładu, a $LB^{(j)}$ - dolne oszacowanie na minimalną wartość kryterium, tzn. na wartość $C_{max}^{(j)}(\pi^*, u^*)$, dla tegoż przykładu.

Przyjęto następujące dolne oszacowanie:

$$LB \triangleq \min_{\pi \in \Pi} C_{max}(\pi, \beta), \tag{12}$$

które wyznacza się w O(n log n) krokach poprzez uporządkowanie zadań według niemalejących wartości $b_i - a_i \cdot \beta_i$. (Oczywiste jest, że dla każdego (j) $LB^{(j)} \leq C_{max}^{(j)}(\pi^*, u^*)$).

Badane tutaj algorytmy aproksymacyjne były zaimplementowane w języku FORTRAN 77 i uruchomione na komputerze IBM PC/AT. Dla każdego n:=25, 50, 75, 100, każdego z:= 0.2, 0.5, 0.9 oraz każdego algorytmu A ($A := A_k, M_k$, k = 1,2,3,4,5) były wyznaczane: średnia arytmetyczna stu wartości ϱ_A^a, ich maksymalna wartość ϱ_A^{max}, jak również średnia wartość (stu) czasów obliczeń T_A^o oraz ich maksymalna wartość T_A^{max}. Z kolei dla każdego zmodyfikowanego algorytmu przybliżonego M_k, k = 1,2,3,4,5 średnia arytmetyczna wykonanej liczby powtórzeń kroku 3 l^a i ich maksymalna wartość l_{max} były także wyznaczane dla każdego n:= 25, 50, 75, 100 oraz każdego z:= 0.2, 0.5, 0.9. Ze względu na ograniczoną ilość miejsca przedstawione zostaną rezultaty tylko dla n = 50 i 100 (Tabela 1, 2). Rezultaty uzyskane dla n = 25 i 75 są podobne do prezentowanych.

Analizując przedstawione w Tabelach 1 i 2 wyniki przeprowadzonego eksperymentu, łatwo zauważyć, że (z wyjątkiem algorytmów A_4 i M_4) zmodyfikowane

Tabela 1

Rezultaty testowania pięciu prostych (A_k) oraz pięciu zmodyfikowanych (M_k) algorytmów aproksymacyjnych dla liczby zadań n = 50 (każdy wynik reprezentuje sto losowo wygenerowanych przykładów)

Ilość zasobów	R:= 0.2 $\sum \beta_i$					R:=0.5 $\sum \beta_i$					R:= 0.9 $\sum \beta_i$				
k-indeks prostego algor. przybl. A_k	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
$\rho^a_{A_k}$ (%)	47,9	54,2	50,8	22,0	44,6	36,4	34,0	36,2	13,8	30,0	33,9	17,4	32,5	11,6	16.1
$\rho^{max}_{A_k}$ (%)	59,3	61,2	61,5	65,1	53,9	55,8	58,6	56,3	77,8	65,8	52,3	52,4	51,2	32,9	46,8
$T^a_{A_k}$ (sek)	0.05	0.10	0.07	0.11	0.07	0.08	0.08	0.08	0.05	0.04	0.07	0.06	0.02	0.06	0.05
$T^{max}_{A_k}$ (sek)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
k-indeks zmodyf. algor. przybl. M_k	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
$\rho^a_{M_k}$ (%)	22.0	22.0	22.0	22.0	22.0	13.7	12.6	13.6	13.8	13.1	12.2	9.4	12.2	11.6	5.2
$\rho^{max}_{M_k}$ (%)	65.1	65.1	65.1	65.1	65.1	77.8	77.8	77.8	77.8	77.8	22.1	19.4	32.9	32.9	32.9
T_k (sek)	0.17	0.16	0.17	0.04	0.16	0.20	0.24	0.17	0.13	0.21	0.13	0.16	0.30	0.21	0.08
$T^{max}_{M_k}$ (sek)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
l^a	2	2	2	1	2	2	2	2	1	2	2	2	2	1	1
l^{max}	4	3	4	1	4	5	5	5	1	7	4	5	6	1	1

algorytmy przybliżone M_k wykazują znacznie wyższą dokładność obliczeń niż ich proste odpowiedniki A_k, k = 1,2,3,4,5. Przypadek algorytmów A_4 i M_4 nietrudno zinterpretować, ponieważ - jak łatwo zauważyć - kolejność wykonywania zadań $\pi_0^{A_4}$ uzyskana w pierwszym kroku algorytmu A_4 nie ulega zmianie w kroku 3 zmodyfikowanego algorytmu przybliżonego M_4 (tzn. $\pi_0^{A_4} = \pi_1^{M_4} = \pi_2^{M_4} = \ldots$). Wśród prostych algorytmów aproksymacyjnych zdecydowanie najlepszym - w sensie średniej wartości górnego oszacowania na błąd względny ρ^a_A - jest algorytm A_4. Widoczne jest to dla każdego n, jak i dla każdej wartości maksymalnego poziomu zużycia zasobu R. Trudno natomiast znaleźć metodę najlepszą lub najgorszą, jeśli chodzi o maksymalną wartość górnego

Tabela 2

Rezultaty testowania pięciu prostych (A_k) oraz pięciu zmodyfikowanych (M_k) algorytmów aproksymacyjnych dla liczby zadań n = 100
(każdy wynik reprezentuje sto losowo wygenerowanych przykładów)

Ilość zasobów	$R := 0.2 \cdot \sum \beta_i$					$R := 0.5 \sum \beta_i$					$R := 0.9 \sum \beta_i$				
k-indeks prostego algor. przybl. A_k	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
$\rho^a_{A_k}$ (%)	26.6	29.6	28.4	9.7	24.1	21.1	19.7	20.2	4.3	16.7	19.3	11.1	17.8	2.8	8.5
$\rho^{max}_{A_k}$ (%)	32.9	33.7	32.5	55.9	29.8	31.1	32.3	30.3	64.9	29.3	29.3	35.4	28.1	6.1	26.1
$T^a_{A_k}$ (sek)	0.10	0.12	0.18	0.21	0.11	0.16	0.12	0.11	0.19	0.18	0.12	0.20	0.10	0.22	0.09
$T^{max}_{A_k}$ (sek)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
k-indeks zmodyf. algor. przybl. M_k	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
$\rho^a_{M_k}$ (%)	9.7	9.7	9.7	9.7	9.7	4.3	4.2	4.2	4.3	4.1	2.8	2.9	2.9	2.8	1.3
$\rho^{max}_{M_k}$ (%)	55.9	55.9	55.9	55.9	55.9	64.9	64.9	64.9	64.9	64.9	5.7	6.3	5.7	6.1	5.2
$T^a_{M_k}$ (sek)	0.35	0.32	0.36	0.17	0.33	0.30	0.45	0.44	0.18	0.38	0.21	0.31	0.61	0.25	0.19
$T^{max}_{M_k}$ (sek)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
l^a	2	2	2	1	2	2	2	2	1	2	2	2	2	1	1
l^{max}	2	2	2	1	3	3	5	5	1	6	4	5	5	1	4

oszacowania ρ^{max}_A na błąd względny.

Natomiast jeśli chodzi o zmodyfikowane algorytmy przybliżone M_i zarówno wartości średnie, jak i maksymalne górnego oszacowania na błąd względny są zbliżone dla wszystkich metod, chociaż w wielu przypadkach, szczególnie przy dużych wartościach maksymalnego poziomu zużycia zasobu, algorytm M_5 wykazuje pewną przewagę nad pozostałymi algorytmami. To ostatnie spostrzeżenie jest łatwe do zinterpretowania, bowiem w metodzie M_5 zadania szeregowane są według niemalejących wartości $b_i - a_i \beta_i$.

Jak łatwo zauważyć, dokładność algorytmów przybliżonych zarówno prostych, jak i zmodyfikowanych, wzrasta wraz ze wzrostem liczby zadań, co jest bardzo korzystną własnością, mając na względzie możliwości praktycznych zastosowań tych algorytmów. Ponadto, dla tej samej ilości zadań dokładność wszystkich algorytmów wzrasta wraz ze wzrostem zadanego maksymalnego poziomu zużycia zasobu R. To nie jest cechą charakteryzującą prezentowane algorytmy, ale jest skutkiem przyjętego dolnego oszacowania na wartość optymalną, ponieważ analizując sposób wyznaczenia dolnego oszacowania, łatwo zauważyć, że wartość optymalna kryterium zbliża się do jego wartości wraz ze wzrostem globalnej ilości zasobu R.

Jeżeli chodzi o czas obliczeń, to maksymalna jego wartość nie przekraczała 1 sekundy dla wszystkich algorytmów przybliżonych (zarówno prostych, jak i zmodyfikowanych).

Jeśli z kolei chodzi o liczby wykonywanych iteracji kroku 3 (w zmodyfikowanym podejściu przybliżonym M), to są one podobne dla wszystkich algorytmów (z wyjątkiem wspomnianego już powyżej algorytmu M_4, dla którego zawsze $l^a = l^{max} = 1$). Maksymalna liczba wykonywanych iteracji kroku 3 nie przekracza 7.

Reasumując, chociaż wszystkie badane tutaj proste algorytmy przybliżone są sobie równoważne w sensie analizy najgorszego przypadku (dla wszystkich ograniczenie wynosi 2, i jest ono osiągalne), to jednak analiza eksperymentalna wykazuje znaczną przewagę algorytmu A_4 nad pozostałymi. Z kolei, zmodyfikowane algorytmy niewiele różnią się w sensie dokładności obliczeń (tzn współczynnika ρ), tylko algorytm M_5 wykazuje, przy dużych wartościach R, niewielką przewagę nad pozostałymi algorytmami.

Biorąc pod uwagę przeprowadzoną powyżej analizę eksperymentalną uzyskanych wyników, wszystkie badane tutaj algorytmy zmodyfikowane oraz prosty algorytm A_4 można rekomendować do zastosowań praktycznych.

6. Uwagi końcowe

W poprzednich rozdziałach badaliśmy problem minimalizacji czasu zakończenia wykonania wszystkich zadań na jednej maszynie przy założeniu, że globalna ilość zasobów, dysponowana do realizacji tychże zadań na maszynach poprzedzających maszynę rozpatrywaną, jest ograniczona. Wracając do naszego przykładu zastosowań z pierwszego rozdziału, innymi słowy, minimalizowaliśmy przestoje walcarki zgniatacza(co jest równoważne minimalizacji czasu zakończenia walcowania wszystkich rozpatrywanych wlewków na zgniataczu), przy zadanym poziomie zużycia gazu. Z punktu widzenia rozpatrywanego procesu technologicznego na odcinku produkcyjnym piecownia - zgniatacz, sensowne jest postawienie także zadania dualnego (odwrotnego), a mianowicie przy z

góry ustalonym górnym ograniczeniu na dopuszczalną wielkość przestojów walcarki zgniatacza (proces cykliczny) można minimalizować zużycie gazu. Problem taki był badany w [9], gdzie osiągnięto w pewnym sensie podobne wyniki do prezentowanych w niniejszej pracy.

Do rozpatrywanego hutniczego procesu technologicznego można zastosować także dwukryterialne podejście, w którym równocześnie minimalizowano by oba przeciwstawne kryteria: przestoje walcarki zgniatacza (reprezentujące jakość sterowania) i ilość (poziom) zużytego gazu (reprezentującą w pewnym sensie koszt sterowania). Można tutaj poszukiwać całego zbioru sterowań Pareto optymalnych, każde składające się z uporządkowanych zadań i rozdziału zasobów. Problem ten był badany w [9]. W ogólnym przypadku jest on bardzo trudnym problemem, co najmniej tak trudnym jak silnie NP-trudny problem dualny do badanego w niniejszej pracy.

Problem ten dla czterech przypadków rozpatrywanych we własności 1 (rozdział 4) upraszcza się znacznie i redukuje się do zagadnienia znalezienia tzw. krzywej punktów kompromisowych, która jest wypukłą, malejącą i odcinkami liniową krzywą. Dla każdego ze wspomnianych czterech przypadków Pareto optymalne uporządkowanie zadań jest identyczne dla wszystkich punktów wspomnianej krzywej punktów kompromisowych. W każdym z tych czterech przypadków punkty załamania krzywej punktów kompromisowych (wraz z odpowiadającymi tym punktom Pareto optymalnymi rozdziałami zasobów) można wyznaczyć w wielomianowym czasie. A użytkownik ze zbioru znalezionych wszystkich sterowań Pareto optymalnych może wybrać dla siebie rozwiązanie najbardziej go satysfakcjonujące.

Do wspomnianego dwukryterialnego problemu, można również zastosować inne podejście, w którym nie wyznacza się całego zbioru sterowań Pareto optymalnych, ale poszukuje się tylko jednego rozwiązania Pareto optymalnego najbliższego (np. w sensie ważonej normy Czebyszewa) pewnym zadanym przez użytkownika celom sterowania. Ciekawe podejście w tym względzie prezentowane jest w [17]. Do rozwiązania rozpatrywanego dwukryterialnego problemu można także zastosować,powszechnie znane w literaturze, inne podejścia [4].

LITERATURA

[1] Du J., Leung J.Y.-T.: Minimizing total tardiness on one machine is NP-hard. Math. Oper. Res. (to appear) 1989.

[2] Duda J.: Dobór parametrów dyskretnego procesu przepływu materiałów na przykładzie procesu w piecowni zgniatacza Inst. Aut. Polit. Śl., Gliwice 1978 (rozprawa doktorska).

[3] Filipowicz B.: Metody i algorytmy syntezy optymalnego sterowania dyskretnych procesów przemysłowych. Z.N.AGH, s. Automatyka, Nr 19, 1979.

[4] Galas Z., Nykowski I., Żółkiewski Z.: Programowanie wielokryterialne, PWN, 1987.

[5] Jackson J.R.: Scheduling a production line to minimize maximum tardiness, Research Report, The University of California at Los Angeles, 1955.

[6] Janiak A.: Time-optimal control in a single machine problem with resource constraints, Automatica 22, 745-747, 1986.

[7] Janiak A.: General flow-shop scheduling with resource constraints. International Journal on Production Research , vol. 26, No 6, 1089-1103, 1988.

[8] Janiak A.: On single machine sequencing with linear models of release dates, Raport ICT Politechniki Wrocławskiej Nr 93/88, 1988.

[9] Janiak A.: Minimization of resource consumption under a given deadline in a single machine scheduling problem with linear models of release dates, E.J.O.R. 1990, (przyjęto do druku).

[10] Janiak A.: Single machine sequencing with linear models of jobs subject to precedence constraints. Archiwum Aut. i Telem. t.XXXIII, 203-210, 1988.

[11] KowalowskiH.: Automatyzacja dyskretnych procesów przemysłowych. WNT, Warszawa 1984.

[12] Korfel M.: Sterowanie odcinkiem stalownia martenowska - zgniatacz minimalizujące postoje walcarki na dogrzewanie wsadu, w: Mat. Konf. Współczesne Problemy Automatyki i informatyki, Sekcja B, Gliwice, 35-51,1973.

[13] Lestra J.K., Rinnooy Kan, Brucker P.: Complexity of machine scheduling problems, Ann. Discr. Math. 1, 343-362, 1977.

[14] Nowicki E.: Minimalizacja kosztu w jednomaszynowym problemie szeregowania ze zmiennymi czasami gotowości zadań, w: Mat. Konf. pt. "Problematyka budowy i eksploatacji maszyn w ujęciu systemowym", Kraków, 70-75, 1986.

[15] Nowicki E.; Zdrzałka S.: A survey of results for sequencing problems with controllable processing times Discr. Appl. Math. (przyjęto do druku), 1990.

[16] Rinnooy Kan A.H.G.: Machine scheduling problems: Clasyfication, Complexity and Computations. Nijhoff, Hague, 1976.

[17] Wierzbicki A.P.: On the completeness and Constructiveness of Parametric Characterizations to Vector Optimization Problems. OR Spectrum, 8, 73-87, 1986.

[18] Williams T.J.: Analysis and design of hierarchical control systems: with special reference to steel plant operations, North-Holland, 1986.

Recenzent: Doc.dr h.inż.A.Swierniak

Wpłynęło do Redakcji do 1990-04-30.

THE MINIMIZATION PROBLEM OF MACHINE STANDANDSTILLS WITH RELEASE DATES DEPENDEND ON RESOURCE - ON THE EXAMPLE OF THE PRODUCTION PROCESS BETWEEN CONVERTER PLANTS AND A BLOOMING MILL

S u m m a r y

On the basis of the minimization problem of the blooming mill standstills in the ingot preheating and hot rolling process in steel mill, the single machine sequencing problem with linear models of release dates was formulated. It is shown that the problem is NP-hard. Some efficiently solvable cases of the problem are found. The worst case and experimental analyses of five approximation algorithms proposed are given.

ПРОБЛЕМА МИНИМАЛИЗАЦИИ ПРОСТОЕВ МАШИНЫ ПРИ ЗАВИСИМЫХ ОТ РЕСУРСОВ МОМЕНТАХ ГОТОВНОСТИ РАБОТ, НА ПРИМЕРЕ ПРОИЗВОДСТВЕННОГО ПРОЦЕССА: СТАЛЕЛИТЕЙНЫЙ МАРТЕНОВСКИЙ ЦЕХ - СЛЯБИНГ

Р е з ю м е

На базе проблемы минимализации простоев прокатного стана слябинга в процессе подогрева и горячего проката слитков на металлургическом заводе сформулировано для одной машины проблему расписания работ с линейными моделями моментов готовности задач. Доказана НП - сложность этой проблемы. Найдены решения отдельных случаев этой проблемы с использованием многочленов. Проведен анализ худшего случая и экспериментальный анализ пяти предложенных алгоритмов аппроксимации.