

ALGORYTM PRIORYTETOWY ALOKACJI BUFORÓW DLA PROBLEMU HARMONOGRAMOWANIA PROJEKTU ZE ZDEFINIOWANYMI KAMIENIAMI MILOWYMI

Streszczenie

Artykuł prezentuje model harmonogramowania projektu z ograniczoną dostępnością zasobów (ang. *RCPSP – Resource-Constrained Project Scheduling Problem*) ze zdefiniowanymi terminami realizacji kamieni milowych projektu. Dla tego modelu proponowany jest priorytetowy algorytm alokacji buforów. Skuteczność opisanego algorytmu jest testowana przy użyciu zbioru instancji J30 i J90 z biblioteki *PSPLIB* (ang. *Project Scheduling Problem LIBrary*).

Słowa kluczowe: harmonogramowanie odporne, harmonogramowanie projektu z ograniczonymi zasobami, reguły priorytetowe, alokacja buforów, kamienie milowe

1. WPROWADZENIE

W artykule zaprezentowany jest model harmonogramowania projektu z ograniczoną dostępnością zasobów ze zdefiniowanymi terminami realizacji etapów przedsięwzięcia (kamiennymi milowymi projektu). Proponowany model może być wykorzystywany przy realizacji dużych projektów konstrukcyjnych czy rozwojowych. Wprowadza kontrolę przebiegu prac w trakcie całego przedsięwzięcia. Terminowa realizacja kamieni milowych, zmniejsza ryzyko niepowodzenia całego projektu.

Podczas planowania realizacji projektu stosowane jest harmonogramowanie predyktywno-reaktywne (ang. *predictive-reactive scheduling*) [1][10] – podejście wykorzystywane w dynamicznych, zmiennych systemach produkcyjnych. W fazie harmonogramowania predyktywnego generowany jest harmonogram nominalny, który następnie jest uodporniany podczas harmonogramowania proaktywnego (ang. *proactive scheduling*), zwanego też odpornym (ang. *robust schedule*). Natomiast w trakcie wykonania projektu podczas harmonogramowania reaktywnego wykonywane są niezbędne zmiany w planowanym harmonogramie, które rozwiązują konflikty czasowe czy zasobowe.

Harmonogram proaktywny ma przeciwdziałać niestabilności harmonogramów nominalnych przez uwzględnienie niepewności parametrów systemu produkcyjnego. Harmonogram odporny powinien charakteryzować się tym, że jego jakość jest akceptowalna, nawet po wystąpieniu nieprzewidywalnych zakłóceń. Jest tworzony m.in. przez wstawianie buforów czasowych w miejscach harmonogramu najbardziej narażonych na zakłócenia. Określenie odpornego przydziału buforów czasowych jest przedmiotem badań w tej pracy. Analizowana jest zdolność harmonogramu do niwelowania skutków wzrostów (lub wahań) czasów trwania czynności, które mogą być spowodowane przez niekontrolowane czynniki. Alokacja buforów czasowych ma prowadzić do minimalizacji zmian w czasach rozpoczęcia zadań w planowanym harmonogramie w trakcie jego realizacji przy równoczesnym terminowym wykonaniu wszystkich umownych etapów projektu.

* Państwowa Szkoła Wyższa w Białej Podlaskiej, Instytut Informatyki

** Akademia Górniczo-Hutnicza, Wydział Zarządzania

W niniejszym artykule jako rozwiązanie problemu odpornej alokacji zasobów proponowany jest algorytm priorytetowy, w którym buforów czasowe wstawiane są na podstawie priorytetowania zadań. Eksperymenty obliczeniowe przeprowadzone przez autorów mają na celu wyłonienie najskuteczniejszych reguł priorytetowych wykorzystywanych przy przydziale buforów. Podczas eksperymentów stosowane są zestawy *J30* i *J90* zadań testowych z biblioteki *PSPLIB* [7] z kamieniami milowymi zdefiniowanymi według procedury opracowanej przez autorów i ze zmiennością czasów trwania zadań uwzględniającą niepewność realizacji zadań projektowych.

2. SFORMUŁOWANIE PROBLEMU

Projekt to zbiór niepodzielnych zadań (czynności) wykonywany przy użyciu ograniczonej liczby zasobów odnawialnych, których ilość jest stała niezależnie od obciążeń w poprzednich okresach. W każdej chwili t wykorzystanie wszystkich typów zasobów przez czynności nie przekracza wielkości dostępnych [1][9]:

$$\sum_{i \in A(t)} r_{ik} \leq a_k \quad \forall t, \forall k \quad (1)$$

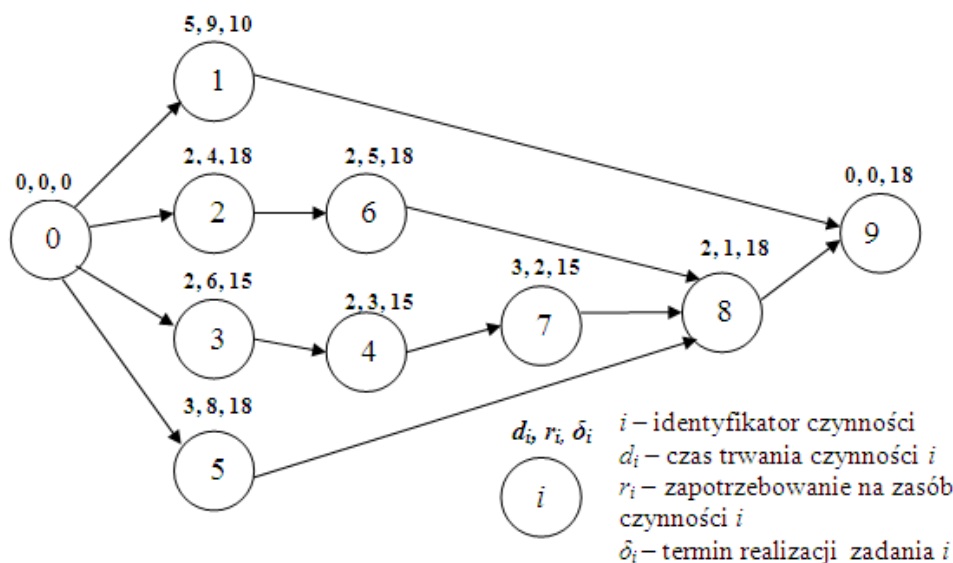
gdzie:

$A(t)$ – zbiór zadań wykonywanych w okresie $[t, t-1]$,

a_k – liczba dostępnych zasobów typu k ,

r_{ik} – zapotrzebowanie zadania i na zasób typu k .

Projekt przedstawiany jest w reprezentacji grafowej „czynność na węźle” *AON* (ang. *Activity On Node*) jako acykliczny, spójny, prosty graf skierowany $G(V, E)$, w którym V to zbiór węzłów odpowiadający czynnościom, a E to zbiór łuków opisujących relacje kolejnościowe między zadaniami. Jako wierzchołek początkowy i końcowy w grafie G dodawane są dwa pozorne zadania 0 i $n+1$ o zerowych czasach trwania i zerowym zapotrzebowaniu na zasoby (n – liczba zadań projektowych). Na rysunku 1 zaprezentowany jest przykładowy projekt w reprezentacji „czynność na węźle”.



Rys. 1. Przykładowa sieć typu *AON* dla projektu z jednym zasobem, z określonymi nieprzekraczalnymi terminami realizacji zadań

Każde z zadań ma określony jeden sposób realizacji (ang. *single-mode RCPSP*)[3][9][11]. Między zadaniami występują relacje poprzedzeń typu koniec-początek bez zwłoki, w których następnik może rozpocząć się bezzwłocznie po zakończeniu poprzednika:

$$s_i + d_i \leq s_j, \quad \forall (i, j) \in E \quad (2)$$

gdzie:

s_i – planowany czas rozpoczęcia i ,

d_i – planowany czas trwania zadania i .

Podczas harmonogramowania nominalnego szukane są czasy rozpoczęcia poszczególnych zadań s_0, s_1, \dots, s_{n+1} przy spełnieniu opisanych wzorami (1-2) ograniczeń zasobowych i kolejnościowych. Najczęstszym kryterium optymalizacyjne stosowanym w badaniach jest minimalizacja czasu wykonania całego przedsięwzięcia (ang. *makespan*).

W związku z niepewnością występującą w trakcie realizacji projektu autorzy proponują określić momenty kontroli przebiegu prac tzw. kamienie milowe (mogą to być umowne etapy projektu określone przez zleceniodawcę wykonawcy projektu), które zmniejszą ryzyko niepowodzenia lub nieterminowości realizacji przedsięwzięcia. Poszczególne zadania mają zdefiniowane nieprzekraczalne terminy ich zakończenia δ_i [4-5]:

$$z_i < \delta_i \quad (3)$$

gdzie:

z_i – planowany czas zakończenia zadania i ,

δ_i – termin realizacji najbliższego j -tego etapu projektu ($\delta_i = tm_j$), w którym zadanie i musi być już wykonane.

Dla przykładowego projektu przedstawionego na rysunku 1 określone są trzy etapy wykonania projektu:

- 1) etap 0 o terminie realizacji $tm_0=10$ (do wykonania zadanie 1),
- 2) etap 1 o terminie realizacji $tm_1=15$ (do wykonania zadania 3, 4, 7),
- 3) etap 2 o terminie realizacji $tm_2=18$ (do wykonania zadania 2, 5, 6, 8, 9 oraz etapy 0 i 1).

Przy ocenie jakości uszeregowania nominalnego istotny jest poziom zapasów czasowych występujących przy realizacji wszystkich etapów projektu. Jako kryterium oceny proponowana jest funkcja celu maksymalizująca ważoną sumę rezerw czasowych dla terminów wykonania poszczególnych kamieni milowych:

$$F_n = \sum_{i=1}^m rez_i \cdot wm_i \quad (4)$$

gdzie:

m – liczba umownych etapów projektu,

rez_i – różnica między terminem zakończenia i -tego etapu projektu tm_i a terminem realizacji wszystkich zadań związanych z tym etapem dla aktualnego harmonogramu,

wm_i – waga przypisana i -temu etapowi projektu.

Przy zastosowaniu funkcji celu F_n osiąga się równomierne rozłożenie zapasów czasowych dla poszczególnych kamieni milowych, przy czym istotne jest odpowiednie określenie wag wm_i . Wartości wag wm_i są ustalane na podstawie aktualnych rezerw czasowych dla kamieni

milowych: większa waga dla mniej zabezpieczonych (o mniejszym rez_i) etapów projektu (szczegóły ustalania wag przedstawione są w pracach [4-5]).

Maksymalne zabezpieczenie terminowej realizacji kamieni milowych jest istotne ze względu na możliwość występowania zakłóceń produkcyjnych (wydłużających czas trwania zadań) w trakcie wykonania projektu, które mogą opóźnić przebieg prac. Stworzenie uszeregowania nominalnego z rezerwami czasowymi umożliwia także przeprowadzenie alokacji buforów zabezpieczającej harmonogram w miejscach najbardziej narażonych na zakłócenia produkcyjne, o największym koszcie niestabilności.

3. ODPORNA ALOKACJA BUFORÓW

W fazie harmonogramowania nominalnego szukane jest uszeregowanie (czasy rozpoczęcia zadań) przy zastosowaniu jako kryterium jakości harmonogramu maksymalizacji zabezpieczenia terminowej realizacji kamieni milowych (4). Następnym etapem jest harmonogramowanie odporne, którego celem jest uodpornienie harmonogramu na możliwe zakłócenia produkcyjne.

Harmonogram odporny to uszeregowanie, które ze względu na swoje właściwości, jest niepodatne na zakłócenia pojawiające się w trakcie produkcji, którego jakość jest akceptowalna, nawet po wystąpieniu nieprzewidywalnych zdarzeń [2]. W fazie harmonogramowania odpornego dla problemu *RCPS*P wykonywane są alokacja zasobów (ang. *resource allocation*) i alokacja buforów [1][9]. Odporna alokacja buforów czasowych jest przedmiotem analiz w tej pracy. Odpowiednie rozmieszczenie buforów jest istotne z punktu widzenia m.in. kosztu realizacji projektu. Dla mniej zaburzanego harmonogramu występują np. niższe koszty magazynowania, czy kredytu finansującego zakup materiałów przy dostawach *just-in-time*.

Alokacja buforów odbywa się przy ustalonej alokacji zasobów do zadań i sprowadza się do wstawiania buforów czasowych przed zadaniami (lub po zadaniach) w celu uodpornienia harmonogramu m.in. przed czasową niedostępnością zasobów i przed zmiennością czasów trwania czynności. Bufory czasowe są wstawiane w newralgicznych miejscach harmonogramu nominalnego i mają zapobiegać nieznacznym wzrostom (szerzej wahaniom) czasów trwania czynności spowodowanych niekontrolowanymi czynnikami. Inne zakłócenia produkcyjne nie są rozpatrywane, gdyż zakłada się, że większość zaburzeń skutkuje wahaniami czasów trwania zadań [1][9]. W trakcie wykonywania projektu czasy trwania zadań są wydłużane z powodów niemożliwych do określenia w fazie planowania tj. błędy oszacowania czasu trwania, niekorzystne warunki atmosferyczne, awarie itp.

Dla rozpatrywanego problemu z kamieniami milowymi harmonogram powinien być buforowany tak, aby możliwie proporcjonalnie zabezpieczyć terminową realizację wszystkich etapów projektu. Przy alokacji buforów dla rozpatrywanego problemu autorzy przyjmują, że:

- mogą być buforowane jedynie czynności związane z kamieniami milowymi o $rez_i > 0$,
- wstawienie bufora nie może spowodować przekroczenia umownych terminów realizacji poszczególnych kamieni milowych.

Liczba różnych sposobów wstawienia buforów, przy przyjętych założeniach, zależy od rozmiaru rez_i dla poszczególnych kamieni milowych. Im większe rez_i , tym więcej możliwości rozłożenia zapasów czasowych dla poszczególnych zadań.

Celem alokacji buforów jest znalezienie listy buforów po czynnościach $B = (B_1, B_2, \dots, B_n)$ lub listy buforów przed czynnościami $b = (b_1, b_2, \dots, b_n)$, tak aby uzyskane uszeregowanie proaktywne było jak najbardziej odporne na zakłócenia. Kluczowe jest zatem określenie stopnia odporności harmonogramu, czyli jego zdolności do absorbowania zakłóceń

pojawiających się w trakcie działania systemu produkcyjnego. Stosowane w badaniach dotyczących RCPSP miary odporności, nie mogą być wykorzystane, gdyż nie uwzględniają umownych terminów realizacji kamieni milowych.

Autorzy proponują jako funkcję celu harmonogramowania odpornego minimalizację funkcji F , określonej wzorem (5), uwzględniającej zarówno terminy realizacji kamieni milowych projektu jak i stabilność uszeregowania (zmiany w planie dla poszczególnych czynności). Funkcja celu F wyznaczana jest w drodze symulacji dla różnych scenariuszy przebiegu produkcji według następującego wzoru:

$$F = \alpha \cdot \sum_{i=1}^n |s_i^R - s_i^0| + (1 - \alpha) \cdot \sum_{i=1}^m (f_i \cdot \max(0, tm_i^R - tm_i)) \quad (5)$$

gdzie:

s_i^0 – planowany moment rozpoczęcia zadania i (na podstawie harmonogramu odpornego),

s_i^R – rzeczywisty moment rozpoczęcia zadania i ,

tm_i^R – rzeczywisty termin wykonania i -tego etapu projektu,

f_i – funkcja kary za przekroczenie terminu zakończenia i -tego etapu projektu,

α – parametr, określający znaczenie stabilności harmonogramu w stosunku do jego jakości.

W funkcji F zakłada się, że zakłócenie każdej z czynności ma taki sam wpływ na stabilność produkcji. Możliwe jest przypisanie wag dla poszczególnych zadań określających znaczenie terminowości ich wykonania [9], nie jest to jednak przedmiotem rozważań w tej pracy.

Współczynnik α należy do przedziału $\langle 0, 1 \rangle$ i określa jakie znaczenie w danym przedsięwzięciu ma stabilność realizacji planu produkcji (uwzględniając np. specyfikę tego projektu, charakterystykę danej branży itp.):

- dla $\alpha = 0$ – problem sprowadza się do terminowej realizacji wszystkich etapów projektu (kryterium wydajnościowe),
- dla $\alpha = 1$ – problem sprowadza się do uodpornienia harmonogramu na możliwe zakłócenia w celu minimalizacji całkowitego odchylenia czasów rozpoczęcia zadań pomiędzy planowanym a zrealizowanym harmonogramem (kryterium stabilności).

Dla poszczególnych etapów projektu określona jest funkcja kary f_i , która może być zależna od wpływu opóźnień na realizację projektu (np. zależna od wysokości kar umownych za przekroczenie terminu wykonania danego etapu projektu). W tej pracy przyjmowane są równe wartości $f_i = n$ dla każdego etapu projektu: przekroczenie o 1 jednostkę czasową umownego terminu wykonania etapu projektu ma taki sam wpływ na wartość funkcji celu jak opóźnienie o 1 rozpoczęcia wszystkich czynności w projekcie (przy współczynniku $\alpha=0.5$). Przy takich wartościach kar f_i terminowość wykonania każdego z etapów projektu jest równie istotna. Możliwe jest przyjęcie innych wartości f_i , ale nie jest to przedmiotem analiz w tej pracy.

4. ALGORYTM PRIORYTETOWY ODPORNEJ ALOKACJI BUFORÓW

Dla problemu harmonogramowania projektu stosowane są różne metody alokacji buforów [1] np. metoda łańcucha krytycznego oraz zarządzania buforami (ang. *CC/BM – Critical Chain/Buffer Management*), metody szacowania rozmiarów buforów na podstawie wiedzy statystycznej dla problemu minimalizacji ważonego kosztu niestabilności harmonogramu (algorytmy alokacji buforów czasowych przy zmiennych czasach trwania

zadań, algorytmy alokacji buforów czasowych lub/i zasobowych przy zmiennej dostępności zasobów)

Stosowane w badaniach procedury alokacji buforów czasowych rozkładają bufory równomiernie [6] lub proporcjonalnie do kosztu niestabilności związanego z poszczególnymi zadaniami [1][9]. Dla rozpatrywanego nowego modelu *RCPS*P z kamieniami milowymi wskazane jest opracowanie procedury uwzględniającej także zabezpieczenie terminowej realizacji wszystkich etapów projektu.

Proponowany przez autorów jest algorytm *BufP*, w którym bufory czasowe wstawiane są na podstawie prostych reguł priorytetowych uwzględniających np. podatność na zakłócenie czy koszt niestabilności poszczególnych zadań. Stosowanie reguł priorytetowych to nowe podejście do problemu alokacji buforów.

Schemat działania algorytmu *BufP* przedstawiony jest na rysunku 2.

```
TL := ∅
//Krok 1: Inicjalizacja danych

//Kroki iteracyjne alokacji kolejnych buforów jednostkowych
repeat
  Wybierz zadanie job do buforowania //Krok 2
  if (Wstawienie bufora dla job jest możliwe) then
    b[job] := b[job] + 1 //Krok 3a
    Modyfikuj czasy rozpoczęcia czynności
  else
    Dodaj job do TL //Krok 3b
until (liczba elementów w TL jest mniejsza od n)
```

Rys. 2. Schemat działania algorytmu alokacji buforów *BufP*

W pojedynczym kroku algorytmu testowane jest wstawienie bufora jednostkowego dla wybranej czynności *job*. Jeśli dodanie bufora dla czynności *job* jest możliwe (sprawdzana terminowość realizacji kamieni milowych), bufor jest wstawiany i modyfikowany jest harmonogram. W przeciwnym wypadku zadanie *job* jest umieszczane na liście czynności *TL*, która zawiera zadania, które nie mogą już być buforowane. Algorytm kończy działanie, gdy wszystkie czynności nie mogą być już buforowane (liczba elementów w zbiorze *TL* wynosi *n*). Kluczowy dla działania algorytmu jest Krok 2, w którym wybierana jest czynność do wstawienia bufora. Bufory są wstawiane przed lub po analizowanej w danej iteracji czynności (w kroku 3a) w zależności od stosowanej reguły priorytetowania.

Algorytm *BufP* to uniwersalny, elastyczny sposób alokacji na podstawie listy priorytetowej zadań. Jeśli znana jest specyfika optymalizowanego systemu produkcyjnego może ona być uwzględniona w regułach priorytetowania, co może przyczynić się do zwiększenia odporności harmonogramu. Na przykład można w ten sposób skutecznie rozwiązać problem ze stochastycznymi czasami trwania czynności, czy ze stochastyczną dostępnością zasobów.

W algorytmie *BufP* stosowane są dwa rodzaje reguł priorytetowych:

- statyczne – priorytety obliczane przed alokacją buforów dla harmonogramu nominalnego, zadanie zbuforowane w danej iteracji umieszczane jest na koniec listy priorytetowej,
- dynamiczne – priorytety modyfikowane po każdej modyfikacji harmonogramu, gdyż uwzględniają aktualną alokację buforów (większy koszt obliczeniowy niż priorytetowania statycznego), w każdym kroku modyfikacja priorytetu zadania zbuforowanego w tym kroku.

W tabelicy 1 zaprezentowane są reguły priorytetowania statyczne i dynamiczne wstawiania buforów przed czynnościami lub po czynnościach, wykorzystywane w algorytmie *BufP*.

Tabela 1. Reguły priorytetowe dla problemu odpornej alokacji buforów

Reguła	Wzór na priorytet zadania i	Opis reguły
RB ₀	$rb_0(i) = random$	Losowe priorytety zadań
RB ₁	$rb_1(i) = \sum_{k=1}^K r_{ik}$	Maksymalna zasobochłonność czynności
RB ₂	$rb_2(i) = \sum_{k=1}^K r_{ik} + \sum_{j \in ZNA_i} \left(\sum_{k=1}^K r_{jk} \right)$	Maksymalna suma zapotrzebowania na zasoby czynności i jej następników w sieci przepływu zasobów
RB ₃	$rb_3(i) = -(LS_i - s_i)$	Minimalna różnica między najpóźniejszym a planowanym czasem rozpoczęcia czynności
RB ₄	$rb_4(i) = \sum_{k \in K} r_{ik} \cdot \sum_{j \in P(i)} \max(s_j + d_j + \Delta d_j - s_i, 0)$	Maksymalny szacowany koszt niestabilności
RB ₅	$rb_5(i) = \sum_{j \in P(i)} \max(s_j + d_j + \Delta d_j - s_i, 0)$	Maksymalny szacowany koszt niestabilności
RB ₆	$rb_6(i) = \frac{\sum_{k \in K} r_{ik} \cdot \sum_{j \in P(i)} \max(s_j + d_j + \Delta d_j - s_i, 0)}{b_i + 1}$	Maksymalny szacowany koszt niestabilności z uwzględnieniem aktualnego bufora przed czynnością
RB ₇	$rb_7(i) = \frac{\sum_{j \in P(i)} \max(s_j + d_j + \Delta d_j - s_i, 0)}{b_i + 1}$	Maksymalny szacowany koszt niestabilności z uwzględnieniem aktualnego bufora przed czynnością
RB ₈	$rb_8(i) = d_i \cdot \sum_{k=1}^K r_{ik}$	Maksymalny iloczyn czasu trwania i zasobochłonności danej czynności
RB ₉	$rb_9(i) = d_i$	Maksymalny czas trwania czynności
RB ₁₀	$rb_{10}(i) = \frac{d_i \cdot \sum_{k=1}^K r_{ik}}{B_i + 1}$	Maksymalny iloczyn czasu trwania i zasobochłonności danej czynności z uwzględnieniem aktualnego bufora po czynności.
RB ₁₁	$rb_{11}(i) = \frac{d_i}{B_i + 1}$	Maksymalna suma iloczynu czasu trwania i zasobochłonności danej czynności i jej wszystkich następników
RB ₁₂	$rb_{12}(i) = \lambda \cdot d_i - B_i$	Maksymalny czas trwania czynności z uwzględnieniem aktualnego bufora po czynności
RB ₁₃	$rb_{13}(i) = \frac{\sum_{j \in N(i)} \max(s_i + d_i + \Delta d_i - s_j, 0)}{B_i + 1}$	Maksymalny wskaźnik narażenia na opóźnienie ze względu na wydłużenie poprzedników

RB ₁₄	$rb_{14}(i) = \frac{\sum_{j \in N(i)} \left(\sum_{k=1}^K r_{jk} \cdot \max(s_i + d_i + \Delta d_i - s_j, 0) \right)}{B_i + 1}$	Maksymalny wskaźnik narażenia na opóźnienie ze względu na wydłużenie poprzedników
------------------	---	---

gdzie:

$P(i)$ - zbiór poprzedników zadania i ,

Δd_j - korekta czasu trwania zadania j wyliczana następująco: $\Delta d_j = \lambda \cdot d_j$,

λ - szacowany wskaźnik wzrostu rzeczywistego czasu trwania zadania w stosunku do planu.

Z podanych reguł priorytetowych można wyróżnić następujące grupy reguł:

- reguły statyczne przy alokacji buforów przed czynnościami (RB₁-RB₃),
- reguły dynamiczne przy alokacji buforów przed czynnościami (RB₄-RB₆),
- reguły statyczne przy alokacji buforów po czynnościach (RB₇-RB₉),
- reguły dynamiczne przy alokacji buforów przed czynnościami (RB₁₀ - RB₁₄).

Zaprezentowane reguły priorytetowe RB₁-RB₁₄ uwzględniają następujące założenia.:

- zakłócenie rozpoczęcia czynności bardziej zasobochłonnej powoduje większy koszt niestabilności (w związku z dezorganizacją pracy większej liczby zasobów),
- im więcej zasobów realizuje czynność, tym jest ona bardziej podatna na zakłócenia, istnieje większe prawdopodobieństwo wystąpienia czasowej niedostępności zasobów (awarii maszyny, choroby pracownika itp.),
- im czynność jest dłuższa, tym bardziej zwiększa się możliwość jej wydłużenia, dłuższe zadania wykazują się większą zmiennością bezwzględna.

Opracowane reguły RB₀-RB₁₄ stosowane są w algorytmie *BufP* w celu znalezienia najlepszych strategii buforowania.

5. WYNIKI EKSPERYMENTÓW

Celem eksperymentów jest znalezienie najlepszych zasad priorytetowania zadań wykorzystywanych przez algorytm *BufP* do alokacji buforów.

Ekspertymenty prowadzone są przy użyciu 960 instancji testowych z zestawów *J30* i *J90* (po 480 dla problemów złożonych z 30 i 90 zadań) biblioteki *PSPLIB*, o jednym sposobie wykonania zadań z zasobami odnawialnymi z kryterium minimalizacji czasu trwania projektu. Do każdego z projektów z biblioteki *PSPLIB* dodawane są losowo cztery kamienie milowe (i związane z nimi zadania) z określonymi terminami realizacji tm_i , rozłożonymi równomiernie w trakcie realizacji całego projektu. Poza definicją umownych etapów projektu w symulacjach istotne jest określenie zmienności czasów trwania zadań. Analiza czasów trwania zadań w rzeczywistych projektach (zwłaszcza realizowanych przez zasoby ludzkie) wskazuje, że krzywa prawdopodobieństwa zakończenia zadania ma kształt krzywej β . Skumulowany rozkład β jest używany w planowaniu projektów do modelowania możliwych czasów ukończenia przy danym oczekiwanym czasie ukończenia i jego zmienności. W niniejszej pracy przyjęte są takie same parametry opisujące zmienność czasów trwania zadań jak w innych badaniach [1][9]. Dla każdego zadania rzeczywisty czas wykonania zadania jest losowany ze zdyskretyzowanego, prawoskośnego rozkładu β o parametrach 2 i 5, ze średnią równą planowanemu czasowi trwania zadania. Do jednoznacznego zdefiniowania parametrów należy zdefiniować także wartość minimalną i maksymalną. Rozważane są: duża i mała zmienność czasów trwania zadań. Dla każdego zadania i ($i = 1, 2, \dots, n$):

- przy dużej zmienności wartość minimalna wynosi $0,25 \cdot d_i$, zaś wartość maksymalna wynosi $2,85 \cdot d_i$,

- przy małej zmienności wartość minimalna wynosi $0,75 \cdot d_i$, zaś wartość maksymalna wynosi $1,625 \cdot d_i$.

Ekspertyzy prowadzone są na komputerze klasy Pentium z procesorem 1,7 GHz przy użyciu programu zaimplementowanego w języku C# w środowisku Visual Studio.NET. Harmonogram nominalny generowany jest przy użyciu metaheurystyki symulowanego wyżarzania SA (ang. *Simulated Annealing*) z funkcją celu harmonogramowania nominalnego określoną wzorem (4). Eksperymentalnie ustalone są najlepsze parametry algorytmu i najlepsze techniki przeszukiwania rozwiązań [4]. Podczas harmonogramowania nominalnego uzasadnione jest wykorzystanie algorytmów przybliżonych, takich jak algorytm SA, ponieważ *RCPSP* jest problemem silnie *NP*-trudnym [11].

Następnym etapem jest odporna alokacja zasobów. Przydział zasobów wykonywany jest przy użyciu algorytmu *ISH*² [8]. Algorytm *ISH*² przydziela aktualnie analizowaną czynność *j* do zasobów (łańcuchów), w których ostatnia czynność jest bezpośrednim poprzednikiem czynności *j*. Gdy zapotrzebowanie na zasoby czynności *j* jest większe niż poprzedników tej czynności, pozostałe łańcuchy wybierane są zgodnie z algorytmem *ISH* [8], który sprowadza się do przydzielania czynności o zapotrzebowaniu na dany zasób ($r_{jk} > 1$), w taki sposób, aby zmaksymalizować liczbę wspólnych łańcuchów z ostatnimi czynnościami w dostępnych łańcuchach.

Harmonogram nominalny znaleziony algorytmem symulowanego wyżarzania z alokacją zasobów wygenerowaną procedurą *ISH*² jest uodporniany przy użyciu analizowanego algorytmu *BufP*. Kryterium oceny harmonogramów z alokacją buforów jest funkcja celu *F* określona wzorem (5). Funkcja *F* wyznaczana jest symulacyjnie dla czasów trwania wygenerowanych z rozkładu β o dużej lub małej zmienności. W fazie realizacji projektu stosowana jest najprostsza metoda reharmonogramowania z przesunięciem prawostronnym (ang. *right-shift rescheduling*) [10], która sprowadza się do przesuwania w prawo zakłóconych czynności bez zmiany alokacji zasobów.

W tabelicy 2 przedstawione są wyniki działania algorytmu *BufP* dla różnych reguł priorytetowych. Zaprezentowane są średnie wartości funkcji celu *F* wyliczanej przy założonym mniejszym ($\alpha = 0.25$) lub większym ($\alpha = 0.75$) znaczeniu stabilności realizacji poszczególnych zadań zgodnie z planem w stosunku do znaczenia terminowości wykonania kamieni milowych. Do celów porównawczych w tabelicy umieszczone są także średnie wartości *F* uzyskane dla harmonogramów nominalnych bez wstawionych buforów czasowych.

Tabela 2. Średnie wartości funkcji celu *F* dla różnych reguł priorytetowych dla zestawów J30 i J90

Reguła	mała zmienność				duża zmienność			
	J30		J90		J30		J90	
	$\alpha = 0.25$	$\alpha = 0.75$	$\alpha = 0.25$	$\alpha = 0.75$	$\alpha = 0.25$	$\alpha = 0.75$	$\alpha = 0.25$	$\alpha = 0.75$
RB ₀	15.8	12.5	585.5	302.5	82.5	61.5	4084.5	2060.9
RB ₁	15.9	12.7	587.2	302.8	79.9	60.8	4068.1	2051.7
RB ₂	21.7	14.5	624.9	310.8	135.4	78.0	4377.3	2111.1
RB ₃	16.3	12.9	593.7	305.4	87.6	65.9	4161.0	2089.2
RB ₄ , $\lambda=0.1$	13.4	11.5	563.6	294.1	59.5	53.3	3857.1	1971.7
RB ₄ , $\lambda=0.4$	13.7	10.9	573.0	296.1	65.8	51.3	3955.1	2003.5
RB ₅ , $\lambda=0.1$	12.9	11.3	558.8	293.2	58.6	53.6	3838.5	1967.5
RB ₅ , $\lambda=0.4$	13.0	10.6	568.2	294.3	64.3	49.9	3934.8	1994.6
RB ₆ , $\lambda=0.1$	17.7	22.0	546.5	304.2	84.1	118.8	3764.2	2033.5
RB ₆ , $\lambda=0.4$	17.0	18.8	567.5	303.9	81.4	103.5	3839.4	2036.9
RB ₇ , $\lambda=0.1$	17.1	23.4	538.6	304.7	80.8	126.3	3753.0	2044.5
RB ₇ , $\lambda=0.4$	18.8	19.1	578.5	303.7	84.0	100.2	3808.9	1989.2
RB ₈	12.9	12.6	554.3	305.0	61.4	54.2	3916.0	2061.8
RB ₉	12.0	12.2	551.9	304.2	61.7	53.6	3921.1	2062.7

RB ₁₀	15.8	22.1	534.0	311.6	75.9	117.4	3760.5	2119.8
RB ₁₁	16.3	24.0	538.9	319.3	81.9	132.9	3788.3	2171.7
RB ₁₂ , $\lambda=0.1$	15.2	21.7	530.8	310.3	76.6	124.7	3752.9	2132.5
RB ₁₂ , $\lambda=0.4$	14.3	20.7	532.1	309.8	72.9	113.3	3775.7	2104.8
RB ₁₃ , $\lambda=0.4$	17.7	19.7	561.4	311.8	84.5	103.6	3925.0	2105.5
RB ₁₃ , $\lambda=0.1$	15.4	19.6	542.0	306.8	77.1	103.2	3817.4	2062.4
RB ₁₄ , $\lambda=0.4$	17.0	18.9	562.7	308.8	81.7	102.6	3880.2	2097.1
RB ₁₄ , $\lambda=0.1$	15.9	18.3	551.4	304.3	77.5	103.4	3824.5	2052.2
nominalny	30.3	79.9	498.1	416.7	194.0	513.7	3810.4	2932.1

Stosowanie alokacji buforów w większości konfiguracji parametrów zwiększa stabilność uszeregowania w porównaniu z realizacją harmonogramu nominalnego bez buforów czasowych. Przy większym znaczeniu stabilności realizacji poszczególnych zadań ($\alpha = 0.75$), etap alokacji buforów czasowych przynosi istotniejszą poprawę wartości funkcji celu F . Im lepsze przeciętne rezultaty (mniejsze średnie wartości funkcji F), tym mniejszy koszt niestabilności np. niższe koszty dezorganizacji produkcji i niższe kary umowne za nieterminową realizację kamieni milowych.

Wstawienie buforów przed czynnościami jest skuteczniejsze od alokacji buforów po czynnościach przy opracowanych regułach priorytetowych. Najskuteczniejszy jest algorytm wstawiający bufor przed czynnościami wykorzystujący regułę dynamiczną RB₅. Przy wstawianiu buforów po czynnościach najskuteczniejsze są reguły RB₉ i RB₁₂. Zatem efektywne jest wstawianie buforów w pierwszej kolejności zabezpieczając harmonogram przed wydłużeniem zadań najdłuższych.

Wybór odpowiedniej reguły priorytetowej ma wpływ na wartości funkcji celu F . Zastosowanie nieefektywnych zasad priorytetowania prowadzi do alokacji buforów gorszych niż przy zastosowaniu losowej reguły RB₀.

6. ZAKOŃCZENIE

W artykule zaprezentowany jest model harmonogramowania projektu z ograniczoną dostępnością zasobów ze zdefiniowanymi terminami realizacji zadań związanych z kamieniami milowymi projektu. Dla opracowanego modelu proponowany jest priorytetowy algorytm odpornej alokacji buforów czasowych. Wyniki eksperymentów obliczeniowych potwierdzają skuteczność zaproponowanego algorytmu. Harmonogramy z buforami czasowymi wyznaczonymi przez priorytetowy algorytm alokacji ze skutecznymi regułami priorytetowania są mniej zakłócane podczas wykonania projektu niż harmonogram nominalny.

Odporna alokacja buforów jest istotna z punktu widzenia m.in. minimalizacji kosztu realizacji projektu. Odpowiednie rozmieszczenie buforów czasowych prowadzi do minimalizacji wpływu nieprzewidywalnych wahań czasów trwania czynności na realizację poszczególnych zadań i wszystkich etapów projektu.

LITERATURA

- [1] Herroelen W., Leus R.: *Robust and reactive project scheduling: a review and classification of procedures*. International Journal of Production Research 42(8), s. 1599-1620, 2004.
- [2] Jensen M.T.: *Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures*. Applied Soft Computing 1, s.35-52, 2001.

- [3] Józefowska J., Węglarz J. (Red.): *Perspectives in Modern Project Scheduling*, Springer, 2006.
- [4] Klimek M., Łebkowski P.: *Algorytmy metaheurystyczne dla problemu harmonogramowania projektu z kamieniami milowymi*. Zeszyty Naukowe Politechniki Śląskiej, Seria: Automatyka, z. 150, s. 63–72, 2008.
- [5] Klimek M., Łebkowski P.: *Predictive scheduling, with determined terms of milestone achievement*. Total Logistic Management, AGH University of Science and Technology Press, Annual 1, Kraków, s. 69-80, 2008.
- [6] Kobylański P., Kuchta D.: *A note on the paper by M. A. Al-Fawzan and M. Haouari about a bi-objective problem for robust resource-constrained project scheduling*. International Journal of Production Economics, 107, s.496-501, 2007.
- [7] Kolisch R., Sprecher A.: *PSPLIB – a project scheduling library*. European Journal of Operational Research 96, s. 205-216, 1997.
- [8] Policella N., Oddi A., Smith S., Cesta A.: *Generating robust partial order schedules*, In *Proceedings of CP2004*, Toronto, Kanada, 2004.
- [9] Van de Vonder S., Demeulemeester E., Herroelen W., Leus R.: *The use of buffers in project management: The trade-off between stability and makespan*. International Journal of Production Economics 97 s. 227-240, 2005.
- [10] Vieira G.E., Herrmann J.W., Lin E.: *Rescheduling manufacturing systems: a framework of strategies, policies and methods*. Journal of scheduling, 6(1), s. 35-58, 2003.
- [11] Węglarz J. (Red.): *Project Scheduling: Recent Models, Algorithms and Applications*, Kluwer Academic Publishers, 1998.

PRIORITY ALGORITHM OF BUFFER ALLOCATION FOR PROJECT SCHEDULING PROBLEM WITH DEFINED MILESTONES

Abstract

This paper presents a model of the *Resource-Constrained Project Scheduling Problem (RCPSP)* with defined terms of milestones achievement. For this model is proposed priority algorithm of buffer allocation. Effectiveness of described algorithm is tested using a set of *J30* and *J90* problem instances from the *Project Scheduling Problem LIBrary (PSPLIB)*.

Keywords: robust scheduling, resource-constrained project scheduling, priority rules, buffer allocation, milestones