

## HARMONOGRAMOWANIE ROBÓT BUDOWLANYCH W SYSTEMIE POTOKOWYM Z NIEPEWNYMI DANymi

Wojciech BOŻEJKO\*, Zdzisław HEJDUCKI\*\*  
Magdalena ROGALSKA\*\*\*, Paweł RAJBA\*\*\*\*, Mieczysław WODECKI\*\*\*\*

\* Instytut Informatyki, Automatyki i Robotyki, Politechnika Wroclawska

e-mail: wojciech.bozejko@pwr.wroc.pl

\*\* Instytut Budownictwa, Politechnika Wroclawska

e-mail: zdzislaw.hejducki@pwr.wroc.pl

\*\*\* Wydział Budownictwa i Architektury, Politechnika Lubelska

e-mail: m.rogalska@pollub.pl

\*\*\*\* Instytut Informatyki, Uniwersytet Wroclawski

e-mail: pawel@ii.uni.wroc.pl

e-mail: mwd@ii.uni.wroc.pl

Artykuł wpłynął do redakcji 30.08.2012 r. Zweryfikowaną i poprawioną wersję po recenzjach i korekcie otrzymano w październiku 2012 r.

*W pracy rozpatrujemy problem harmonogramowania przedsięwzięć budowlanych realizowanych w systemie potokowym. Są to bardzo ważne zagadnienia praktyki budowlanej, mające znaczący i bezpośredni wpływ na ostateczne koszty realizacji. Przedstawiamy założenia do harmonogramowania robót, algorytm jego rozwiązywania opartego na metodzie przeszukiwania z tabu (ang. tabu search) oraz jego modyfikację dla przypadku, gdy niepewne są czasy wykonywania prac. Porównujemy stabilność rozwiązań w przypadku, gdy niepewne dane są reprezentowane przez zmienne losowe o rozkładzie normalnym lub liczby rozmyte w trzypunktowej reprezentacji.*

**Słowa kluczowe:** niepewne dane, stabilność, harmonogramowanie, przeszukiwanie z tabu, roboty budowlane

### WSTĘP

Ważnym zagadnieniem związanym z negocjowaniem kontraktów budowlanych jest uzgodnienie z inwestorem terminu wykonania robót, ceny, szczegółowego zakresu robót oraz innych warunków realizacyjnych. Bardzo często termin jest narzucany dyrektywnie przez inwestora lub dla pojedynczych robót jest wpisywany w uzgodniony z innymi wykonawcami termin realizacji przedsięwzięcia. Dlatego podejmując decyzje o podpisaniu kontraktu, zarząd firmy budowlanej często ocenia zagrożenia wynikające z przyszłych nieznanymi czynników mogących wpłynąć na dodatkowe koszty robót i na niezbędny czas ich wykonania. W procesie negocjacji kontraktów budowlanych pojawia się konieczność jednoznacznego określenia terminów zakończenia robót. Jest to

bardzo trudne ze względu na duży stopień niepewności wynikający ze zmieniających się, w trakcie realizacji robót, wielu parametrów. Przekroczenie ustalonych terminów powoduje znaczne straty (kary umowne lub niewykorzystane zasoby). Stąd potrzeba takiego modelowania przedsięwzięć, aby możliwie dokładnie odwzorować przebieg rzeczywistych procesów budowlanych, prace: [3, 5, 6, 17, 18, 26]. Prowadzi to do złożonych dyskretno-ciągłych problemów optymalizacyjnych z niepewnymi parametrami oraz nieregularnymi funkcjami celu. Przenosząc problemy harmonogramowania przedsięwzięć budowlanych w dziedzinę klasycznej teorii szeregowania zadań, napotyka się na wiele trudności związanych z doбором właściwego modelu oraz odpowiedniego algorytmu. Są to, zazwyczaj zupełnie nowe, *silnie NP-trudne* problemy optymalizacji kombinatorycznej (zobacz [4, 7, 8]), dla których dzisiaj nie są znane algorytmy optymalne o wielomianowej złożoności obliczeniowej. W praktyce stosuje się zazwyczaj szybkie algorytmy aproksymacyjne. W ostatnich latach popularne są konstrukcje bazujące na metodach sztucznej inteligencji.

Ze względu na stosowanie nowych technik i technologii, unikalność, warunki atmosferyczne i geologiczne, często nie ma możliwości jednoznacznego określenia wartości pewnych parametrów harmonogramów robót, a w szczególności dokładnego określenia cyklu budowy. W takich przypadkach mamy do czynienia z procesem podejmowania decyzji w warunkach niepewności. Niepewność danych przekłada się bezpośrednio na wielkość ryzyka. Ponadto w trakcie realizacji procesu budowy może się okazać, że niektóre parametry różnią się od wstępnie przyjętych („typowych”), co przy braku stabilności rozwiązania prowadzi do zupełnie nieprzydatnych w praktyce rozwiązań. Niepowodzenia wynikające z bezpośredniego stosowania klasycznych algorytmów deterministycznych wskazują na konieczność uwzględnienia niepewności już na etapie budowy modelu oraz konstrukcji samego algorytmu.

Integralną częścią wielu systemów zarządzania jest planowanie przedsięwzięć budowlanych w systemach potokowych [25, 28]. Jednym z wielu występujących tam ograniczeń są sprzężenia czasowe [1, 2, 16, 11, 12, 33, 34], umożliwiające planowanie przebiegu robót w czasie i przestrzeni z zastosowaniem minimum cyklu realizacji jako funkcji celu. Dotyczą one realizacji kompleksu obiektów składających się z wielu jednakowych prac (na przykład: robót ziemnych, fundamentowych, konstrukcyjnych, dachowo-pokrywczych, montażu stolarki okiennej, elewacji, robót wykończeniowych itp.) wynikających z zakresu dokumentacji budowlanej, wykonywanych przez brygady robocze. Systemy potokowe w budownictwie są odpowiednikiem produkcji przepływowej (ang. *flow*) w przemyśle. Obiektom odpowiadają zadania, brygadam – maszyny, a pracą wykonywaną przez brygady – operacje. Kolejności wykonywania prac na obiekcie odpowiada porządek technologiczny.

Harmonogramy robót o charakterze ciągłym występujące w systemach potokowych, przedstawiane są często w formie cyklogramów [9, 10, 24, 27]. Kluczowymi wartościami w cyklogramach są czasy wykonania poszczególnych procesów na działkach roboczych oraz punkty kontrolne. Poniżej przedstawiamy podstawowe oznaczenia i definicje dotyczące cyklogramów, stosowane obecnie w literaturze np. [30, 31, 32].

- $m$  – liczba procesów,
- $n$  – liczba sektorów, działek roboczych,
- $LT$  – najkrótszy czas pomiędzy procesem w toku a kolejnym procesem, tj.

- $LD$  – najmniejszy dystans czasowy (ang. *least time*),
- $LD$  – najmniejsza odległość miejsca pomiędzy procesem w toku a kolejnym procesem (ang. *least distance*),
- $CPH$  – droga krytyczna procesów budowlanych [dni] (ang. *controlling path*),
- $CL$  – ścieżka krytyczna – połączenie graficzne pomiędzy kolejnymi procesami (ang. *controlling link*),
- $CP$  – punkt kontrolny odpowiadający rozpoczęciu kolejnego procesu (ang. *controlling point*),
- $T$  – czas trwania procesu,
- $TT$  – całkowity czas wykonania zadania (ang. *time total*).

Poniżej przedstawiono pewne założenia dotyczące obliczeń parametrów harmonogramów [12]:

- (A) Warunek ciągłości procesów budowlanych, dla każdego rodzaju (metoda sprzężeń czasowych, TCM I (ang. *Time Coupling Methods*)).

$$LT_{1.1}, \dots, LT_{n.m} = 0.$$

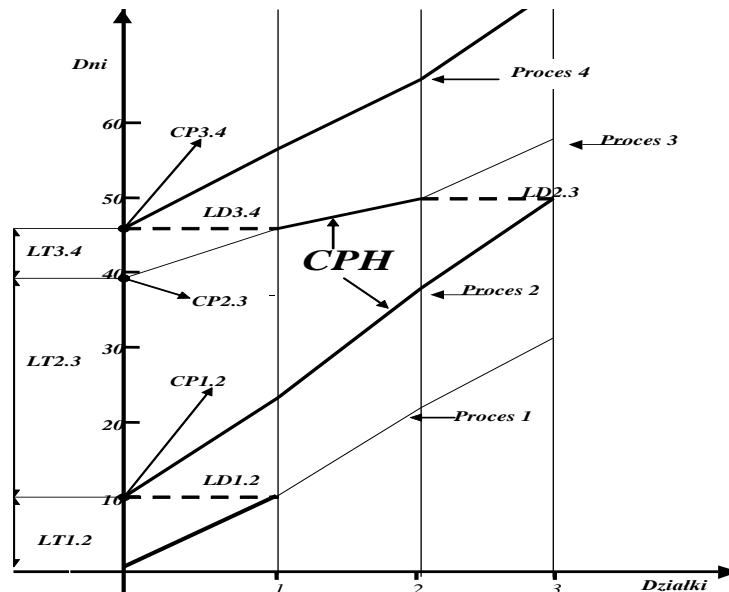
- (B) Warunek ciągłości procesów budowlanych w kolejności technologicznej, na działkach roboczych (metoda sprzężeń czasowych, TCM II)

$$LD_{1.1}, \dots, LD_{n.m} = 0.$$

- (C) Warunek utworzenia minimalnej drogi krytycznej (metoda sprzężeń czasowych TCM III, IV i V).

$$CPH = TT_{min}.$$

Na rysunku 1 zamieszczono przykładowy cyklogram (z pełnym opisem) dotyczący trzech działek roboczych oraz czterech prac.



Rys. 1. Procesy ciągłe w systemie potokowym (oznaczenia)

Źródło: Opracowanie własne

Metodę sprzężeń czasowych TCM I (warunek (A)) stosuje się w przypadku, kiedy założeniem priorytetowym jest zapewnienie ciągłości wykonywania robót budowlanych na działkach roboczych. Na rysunku 1 przedstawiono schemat cyklogramu w systemie potokowym z uwzględnieniem warunku ciągłości robót. Praktycznie wiąże się to z zapewnieniem ciągłości pracy wykonawcom lub pracy sprzętu. Ze względu na przyjętą technologię zapewnienie ciągłości procesu może stać się założeniem priorytetowym, np. w przypadku konieczności ciągłego betonowania. Ta sama sytuacja ma miejsce w odniesieniu do procesów wiodących, czyli umożliwiających rozpoczęcie wszystkich zależnych od niego prac.

W dalszej części przedstawiamy problemy optymalizacji w systemach potokowych z niepewnymi czasami wykonywania prac budowlanych. Do ich rozwiązywania stosujemy algorytmy oparte na metodzie przeszukiwania z tabu. Badamy eksperymentalnie stabilność algorytmów, tj. wpływ losowych zaburzeń parametrów procesu na zmiany wartości funkcji celu.

## 1. SYSTEM POTOKOWY W BUDOWNICTWIE

Systemowi potokowemu realizacji prac budowlanych odpowiada problem przepływowy (ang. *flow shop*) w teorii szeregowania zadań. W dalszej części pracy będziemy korzystali z definicji i oznaczeń stosowanych w rozwiązywaniu problemów szeregowania.

Rozpatrujemy przedsięwzięcie budowlane polegające na wykonaniu  $n$  obiektów ze zbioru:

$$O = \{O_1, O_2, \dots, O_n\},$$

przez  $m$  brygad ze zbioru:

$$B = \{B_1, B_2, \dots, B_m\}.$$

Każdy obiekt  $O_i \in O$  jest ciągiem  $m$  prac:

$$O_i = [P_{i,1}, P_{i,2}, \dots, P_{i,m}],$$

przy czym praca  $P_{i,j}$  ( $i=1,2,\dots,n$ ,  $j=1,2,\dots,m$ ) na  $i$ -tym obiekcie jest wykonywana przez brygadę  $B_j$  w czasie  $p_{i,j}$ . Prace na obiekcie  $O_i \in O$  należy wykonać w zadanym porządku technologicznym, tzn. dowolna praca  $P_{i,j}$  ma być wykonywana po zakończeniu  $P_{i,j-1}$ , a przed rozpoczęciem  $P_{i,j+1}$  ( $2 \leq j \leq m-1$ ). Muszą być przy tym spełnione następujące ograniczenia:

- (W1) każda praca (na obiekcie) może być wykonywana tylko przez jedną, określoną przez ciąg technologiczny, brygadę,
- (W2) żadna brygada nie może wykonywać jednocześnie więcej niż jedną pracę,
- (W3) na każdym obiekcie musi być zachowany porządek technologiczny,
- (W4) wykonywanie każdej pracy nie może być przerwane przed jej zakończeniem.

*Rozpatrywane przedsięwzięcie budowlane* (w skrócie **BP**) polega na ustaleniu kolejności wykonywania, przez każdą z brygad, prac na obiektach (tj. permutacji obiektów), dla której ustalone kryterium osiąga wartość optymalną oraz spełnione są ograniczenia **(W1)-(W4)**.

Niech  $\pi$  będzie pewną permutacją obiektów (elementów zbioru  $O$ ). Permutacja ta wyznacza kolejność wykonywania poszczególnych prac na obiektach, tj. brygada  $B_j \in B$  wykonuje prace  $P_{\pi(j),j}$  na obiekcie  $\pi(i) \in O$ , dopiero po wykonaniu prac  $P_{\pi(1),j}, P_{\pi(2),j}, \dots, P_{\pi(i-1),j}$  kolejno na obiektach  $\pi(1), \dots, \pi(i-1)$ , a przed wykonaniem prac  $P_{\pi(i+1),j}, P_{\pi(i+2),j}, \dots, P_{\pi(n),j}$  na obiektach  $\pi(i+1), \dots, \pi(n)$ . Oznaczmy przez  $\Phi$  zbiór wszystkich możliwych permutacji obiektów. Moc tego zbioru wynosi  $n!$ .

Jeżeli roboty na obiektach są wykonywane w kolejności  $\pi \in \Phi$  oraz  $p_{\pi(i),j}$  jest czasem wykonywania pracy  $P_{\pi(j),j}$  ( $i = 1, 2, \dots, n, j = 1, 2, \dots, m$ ) to moment zakończenia tej pracy  $C_{\pi(i),j}$  możemy wyznaczyć z następujących zależności rekurencyjnej:

$$C_{\pi(i),j} = \begin{cases} \sum_{k=1}^i p_{\pi(k),j}, & j = 1, \\ C_{\pi(i),j-1} + p_{\pi(i),j}, & i = 1, j > 1, \\ \max\{C_{\pi(i),j-1}, C_{\pi(i-1),j}\} + p_{\pi(i),j}, & i > 1, j > 1, \end{cases} \quad (1)$$

a moment rozpoczęcia jej wykonywania:

$$S_{\pi(i),j} = C_{\pi(i),j} - p_{\pi(i),j}. \quad (2)$$

Można łatwo sprawdzić, że określone przez (1) i (2) momenty rozpoczęcia i zakończenia prac na obiektach spełniają ograniczenia **(W1-W4)**, więc są rozwiązaniem dopuszczalnym problemu **BP**.

Model opisanego wyżej przedsięwzięcia budowlanego jest znany w teorii szeregowania jako problem przepływowy (ang. *flow shop problem*). Jeżeli rozpatrujemy kryterium minimalizacji czasu zakończenia wszystkich obiektów ( $C_{\max}$ ), wówczas należy on do klasy problemów *silnie NP-trudnych* [7]. Zagadnienia te, w ogólnej postaci, są rozwiązywane zazwyczaj metodami heurystycznymi. Metody i algorytmy (w tym bazujące na sztucznej inteligencji) przedstawiono w pracach [20, 22, 23, 4, 19, 8] oraz [23].

W praktyce budowlanej bardzo ważnym kryterium jest wykonanie w terminie każdej pracy lub ewentualna minimalizacja kar za przekroczenie ustalonych w kontrakcie terminów. Są to często zagadnienia z nieregularnymi funkcjami celu. W dalszej części zajmujemy się jednym z takich zagadnień.

Dla pracy  $P_{i,j}$  ( $i \in O, j \in B$ ) niech  $d_{ij}, w_{ij}$  będą odpowiednio: ustalonym w kontrakcie żądanym terminem wykonania oraz współczynnikiem kary za spóźnienie. Trójka  $\theta = \langle p, d, w \rangle$ , gdzie:

- $p = [p_{i,j}]_{n \times m}$  - macierz czasów wykonania prac,
- $d = [d_{i,j}]_{n \times m}$  - macierz żądanych terminów wykonania prac,

–  $w = [w_{i,j}]_{n \times m}$  - macierz współczynników kary,  
jest instancją danych deterministycznych dla rozpatrywanego zadania.

Jeżeli  $\pi \in \Phi$  jest kolejnością wykonywania obiektów oraz  $C_{\pi(i),j}$  jest terminem ukończenia pracy  $P_{\pi(i),j}$ , to:

$$U_{\pi(i),j} = \begin{cases} 1, & \text{gdy } C_{\pi(i),j} > d_{\pi(i),j}, \\ 0, & \text{gdy } C_{\pi(i),j} \leq d_{\pi(i),j}. \end{cases} \quad (3)$$

jest spóźnieniem, a  $w_{\pi(i),j} \cdot U_{\pi(i),j}$  karą za spóźnienie. Wówczas:

$$F(\pi) = \sum_{i=1}^n \sum_{j=1}^m w_{\pi(i),j} \cdot U_{\pi(i),j} \quad (4)$$

jest sumą kar za przekroczenie terminów wykonania prac (w skrócie, kosztem permutacji  $\pi$ ).

Rozpatrywany w pracy problem polega na wyznaczeniu optymalnej kolejności wykonywania obiektów w systemie potokowym, minimalizującej funkcję (4). Sprowadza się to do wyznaczenia permutacji  $\pi^* \in \Phi$  takiej, że:

$$F(\pi^*) = \min\{F(\pi) : \pi \in \Phi\}. \quad (5)$$

W skrócie problem ten będziemy oznaczali przez **PFS**. Dla  $m = 1$  jest on równoważny silnie *NP-trudnemu* jednomaszynowemu problemowi szeregowania zadań  $1 \parallel \sum w_i U_i$  (zobacz [8, 9]). Obecnie nie są znane algorytmy optymalne dla rozwiązywania takich problemów o wielomianowej złożoności obliczeniowej. Dlatego do rozwiązywania problemu **PFS** będziemy stosowali algorytm heurystyczny oparty na metodzie przeszukiwania z tabu.

## 2. SYSTEM POTOKOWY Z NIEPEWNYMI CZASAMI WYKONYWANIA ROBÓT

Problemy szeregowania zadań z niepewnymi parametrami rozwiązuje się, stosując metody probabilistyki lub teorii zbiorów rozmytych. W pierwszym przypadku np. [5, 29] istotna jest znajomość rozkładów zmiennych losowych. Niektóre procesy z natury mają charakter losowy. Zależą od pogody, natężenia ruchu, liczby wypadków, warunków geologicznych, awarii sprzętu itp. Jeżeli ponadto posiadają pewną „historię”, to na bazie istniejących danych statystycznych można określić ich rozkłady.

W wielu jednak zagadnieniach niepewność danych nie ma charakteru losowego, lecz wynika z powodu unikalności procesu, błędu pomiaru itp. W tym przypadku naturalnym sposobem reprezentowania niepewności są liczby rozmyte (zobacz [13, 14, 24]). Dużym problemem jest właściwy dobór funkcji przynależności oraz metoda dyfuzyfikacji. Mają one decydujący wpływ na jakość wyznaczanych rozwiązań. W tym rozdziale zastosujemy oba te podejścia do harmonogramowania prac wykonywanych w systemie potokowym.

## 2.1. Probabilistyczne czasy wykonywania robót budowlanych

Niech  $\theta = \langle p, d, w \rangle$  będzie przykładem danych deterministycznych dla problemu **PFS**. Zakładamy, że czasy wykonywania prac są niezależnymi zmiennymi losowymi o rozkładzie normalnym, tj.  $\tilde{p}_{i,j} \square N(p_{i,j}, \sigma_{i,j})$ ,  $i \in O, j \in J$ . Wartość oczekiwana  $E(\tilde{p}_{i,j}) = p_{i,j}$ . Wówczas  $\tilde{\theta} = \langle \tilde{p}, d, w \rangle$ , gdzie  $\tilde{p} = [\tilde{p}_{i,j}]_{n \times m}$  jest macierzą zmiennych losowych, nazywamy danymi probabilistycznymi, a problem - probabilistycznym (w skrócie **PFSP**).

Niech  $\pi \in \Phi$  będzie pewną kolejnością wykonywania prac na obiektach. Dla uproszczenia obliczeń przyjmujemy, że momenty zakończenia poszczególnych prac mają także rozkład normalny.

$$\tilde{C}_{\pi(i),j} \square N(C_{\pi(i),j}, \alpha \sqrt{C_{\pi(i),j}^{[2]}}), \quad (6)$$

gdzie:

$$C_{\pi(i),j}^{[2]} = \begin{cases} \sum_{k=1}^i p_{\pi(i),j}^2, & j = 1, \\ C_{\pi(i),j-1}^2 + p_{\pi(i),j}^2, & i = 1, j > 1, \\ \max\{C_{\pi(i),j-1}^2, C_{\pi(i-1),j}^2\} + p_{\pi(i),j}^2, & i > 1, j > 1, \end{cases} \quad (7)$$

a parametr  $\alpha$  jest wyznaczany eksperymentalnie. Odpowiednikami spóźnień (3) są zmienne losowe:

$$\tilde{U}_{\pi(i),j} = \begin{cases} 1, & \text{gdy } \tilde{C}_{\pi(i),j} > d_{\pi(i),j}, \\ 0, & \text{gdy } \tilde{C}_{\pi(i),j} \leq d_{\pi(i),j}. \end{cases} \quad (8)$$

Przy rozwiązywaniu problemu **PFSP** (z losowymi czasami wykonywania zadań) za funkcję kosztu (4) przyjmujemy:

$$FP(\pi) = \sum_{j=1}^m \sum_{i=1}^n w_{\pi(i),j} (E(\tilde{U}_{\pi(i),j}) + D^2(\tilde{U}_{\pi(i),j})), \quad (9)$$

gdzie:

$E(\tilde{U}_{\pi(i),j})$  jest wartością oczekiwaną, a  $D^2(\tilde{U}_{\pi(i),j})$  wariancją zmiennej losowej  $\tilde{U}_{\pi(i),j}$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ . Ponieważ:

$$E(\tilde{U}_{\pi(i),j}) = P(\tilde{C}_{\pi(i),j} > d_{\pi(i),j}) = 1 - F_{\tilde{C}_{\pi(i),j}}(d_{\pi(i),j}),$$

$$D^2(\tilde{U}_{\pi(i),j}) = F_{\tilde{C}_{\pi(i),j}}(d_{\pi(i),j})(1 - F_{\tilde{C}_{\pi(i),j}}(d_{\pi(i),j})),$$

gdzie:

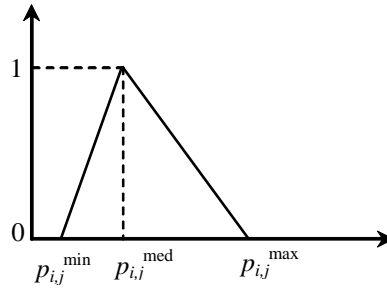
$F_X$  jest dystrybuantą zmiennej losowej  $X$  o standardowym rozkładzie normalnym ( $X \square N(0,1)$ ), więc ostatecznie:

$$FP(\pi) = \sum_{j=1}^m \sum_{i=1}^n w_{\pi(i),j} (1 - (F_{\tilde{C}_{\pi(i),j}}(d_{\pi(i),j}))^2). \quad (10)$$

Algorytm przeszukiwania z tabu rozwiązywania problemu **PFSP** (z funkcją celu (10)) będziemy nazywali probabilistycznym i oznaczali w skrócie przez **TSP**.

## 2.2. Rozmyte czasy wykonywania robót budowlanych

Niepewne czasy wykonywania robót będą w tym przypadku reprezentowane przez trzypunktowe liczby rozmyte  $p_{i,j}^{\Delta} = (p_{i,j}^{\min}, p_{i,j}^{\text{med}}, p_{i,j}^{\max})$ , gdzie  $p_{i,j}^{\min} \leq p_{i,j}^{\text{med}} \leq p_{i,j}^{\max}$  dla  $i \in J$ ,  $j \in M$ . Funkcja przynależności takiej liczby jest przedstawiona na rysunku 2.



Rys. 2. Funkcja przynależności trzypunktowej liczby rozmytej  $(p_{i,j}^{\min}, p_{i,j}^{\text{med}}, p_{i,j}^{\max})$ .

*Źródło: Opracowanie własne*

Wynikiem działań arytmetycznych na liczbach rozmytych jest liczba rozmyta [4, 15]. Jeżeli więc czas wykonywania zadania  $\pi(i)$  ( $\pi \in \Phi$ ) na maszynie  $j$  jest określony przez liczbę rozmytą:

$$p_{\pi(i),j}^{\Delta} = (p_{\pi(i),j}^{\min}, p_{\pi(i),j}^{\text{med}}, p_{\pi(i),j}^{\max}),$$

to wówczas termin jego zakończenia jest liczbą rozmytą postaci:

$$C_{\pi(i),j}^{\Delta} = (C_{\pi(i),j}^{\min}, C_{\pi(i),j}^{\text{med}}, C_{\pi(i),j}^{\max}),$$

gdzie:

$C_{\pi(i),j}^{\min}$ ,  $C_{\pi(i),j}^{\text{med}}$  oraz  $C_{\pi(i),j}^{\max}$  można wyznaczyć z następujących zależności rekurencyjnych:

$$C_{\pi(i),j}^{\min} = \max\{C_{\pi(i-1),j}^{\min}, C_{\pi(i),j-1}^{\min}\} + p_{\pi(i),j}^{\min},$$

$$C_{\pi(i),j}^{\text{med}} = \max\{C_{\pi(i-1),j}^{\text{med}}, C_{\pi(i),j-1}^{\text{med}}\} + p_{\pi(i),j}^{\text{med}},$$

$$C_{\pi(i),j}^{\max} = \max\{C_{\pi(i-1),j}^{\max}, C_{\pi(i),j-1}^{\max}\} + p_{\pi(i),j}^{\max},$$

z warunkami początkowymi:

$$C_{\pi(0),j}^{\min} = C_{\pi(0),j}^{\text{med}} = C_{\pi(0),j}^{\max} \quad j = 1, 2, \dots, m \quad \text{oraz} \quad C_{\pi(i),0}^{\min} = C_{\pi(i),0}^{\text{med}} = C_{\pi(i),0}^{\max} \quad i = 1, 2, \dots, n.$$

Zgodnie z (3), do obliczenia spóźnienia  $U_{\pi(j)}$  pracy  $\pi(i) \in J$  potrzebna jest znajomość jej terminu zakończenia. W przypadku trzypunktowej liczby rozmytej  $C_{\pi(i),j}^{\Delta} = (C_{\pi(i),j}^{\min}, C_{\pi(i),j}^{\text{med}}, C_{\pi(i),j}^{\max})$  stosujemy następującą funkcję defuzyfikacji:



$$I(C_{\pi(i),j}^{\Delta}) = \frac{1}{4} (C_{\pi(i),j}^{\min} + C_{\pi(i),j}^{\text{med}} + C_{\pi(i),j}^{\text{med}} + C_{\pi(i),j}^{\max}).$$

Wówczas spóźnienie:

$$U_{\pi(i),j}^{\Delta} = \begin{cases} 1, & \text{gdy } I(C_{\pi(i),j}^{\Delta}) > d_{\pi(i),j}, \\ 0, & \text{gdy } I(C_{\pi(i),j}^{\Delta}) \leq d_{\pi(i),j}. \end{cases}$$

Ostatecznie jako kryterium porównawcze przyjmujemy funkcję:

$$FR(\pi) = \sum_{j=1}^m \sum_{i=1}^n w_{\pi(i),j} U_{\pi(i),j}^{\Delta}. \quad (11)$$

Algorytm przeszukiwania z tabu rozwiązywania problemu **PFSR** (z funkcją celu (11)) będziemy nazywali rozmytym i oznaczali w skrócie przez **TSR**.

### 3. METODA PRZESZUKIWANIA Z TABU

Algorytmy oparte na metodzie przeszukiwania z tabu (w skrócie *TS*) są z powodzeniem stosowane do rozwiązywania *NP-trudnych* problemów optymalizacji kombinatorycznej. Metoda ta polega na iteracyjnym polepszaniu bieżącego rozwiązania poprzez lokalne przeszukiwanie. Rozpoczyna się od pewnego rozwiązania startowego  $x^0$ . W każdej iteracji, dla bieżącego rozwiązania  $x^i$ , wyznacza się jego otoczenie  $N(x^i)$  - podzbiór zbioru rozwiązań dopuszczalnych. Otoczenie jest generowane przez ruchy, tj. pewne przekształcenia rozwiązania  $x^i$ . Następnie z otoczenia wyznaczamy najlepszy element  $x^{i+1}$ , który przyjmuje się za bieżące rozwiązanie w następnej iteracji. Aby zapobiec generowaniu, w nowych iteracjach, rozwiązań niedawno rozpatrywanych (powstawaniu cykli), zapamiętuje się ruchy (ich atrybuty) na liście rozwiązań zakazanych, tzw. liście tabu. Poniżej opiszemy główne elementy tej metody.

#### Schemat algorytmu przeszukiwania z tabu (*TS*)

Niech  $\pi \in \Phi$  będzie dowolną permutacją,  $F$  – funkcją celu,  $\Delta T$  listą tabu, a  $\pi^*$  najlepszym do tej pory znalezionym rozwiązaniem (na początek przyjmujemy za  $\pi^*$  permutację  $\pi$ ).

**Krok 1:** Wyznaczyć otoczenie  $N(\pi)$  permutacji  $\pi$ , niezawierające elementów zabronionych przez listę  $\Delta T$ ;

**Krok 2:** Znaleźć permutację  $\delta \in N(\pi)$  taką, że:

$$F(\delta) = \min\{F(\beta) : \beta \in N(\pi)\};$$

**Krok 3:** if  $F(\delta) < F(\pi^*)$  then  $\pi^* \leftarrow \delta$ ;

Umieść atrybuty rozwiązania  $\delta$  na liście  $\Delta T$ ;

$\pi \leftarrow \delta$ ;

**Krok 4:** if Warunek\_Zakończenia then *STOP*

else goto Krok 1.

Sposób wyznaczania otoczenia, organizacja listy tabu i warunek zakończenia są podstawowymi elementami metody. Przy implementacji algorytmu elementy te określiliśmy następująco (dokładnie są one opisane w pracy [4]):

- *Otoczenie*  $N(\pi)$  zawiera permutacje wygenerowane z  $\pi$  przez wstawienie (ruch **insert**) pewnego zadania na inną pozycję w permutacji.
- *Lista tabu* jest cykliczna i zawiera atrybuty pewnej liczby ostatnio rozpatrywanych rozwiązań (startowych) oraz ma z góry określoną maksymalną długość. Jeżeli lista nie jest pełna, to nowy element jest do niej dopisywany. Jeśli natomiast lista osiągnęła maksymalną długość, wówczas usuwany jest najstarszy element, a nowy jest umieszczany na końcu listy.
- *Warunek zakończenia* – algorytm kończy działanie po wykonaniu z góry określonej liczby iteracji lub gdy bieżące otoczenie jest zbiorem pustym.

Złożoność obliczeniowa algorytmu opartego na metodzie przeszukiwania z tabu zależy od sposobu realizacji jego elementów, tj. metody wyznaczania otoczenia, organizacji oraz długości listy tabu, sposobu wyliczania wartości funkcji celu oraz warunku zakończenia.

#### 4. STABILNOŚĆ ALGORYTMÓW

Stabilność jest pewną miarą umożliwiającą oszacowanie wpływu zaburzeń danych na zmiany wartości funkcji celu. W pierwszej kolejności przedstawiamy metodę generowania zbioru przykładów testowych.

Niech  $\delta = \langle \mathbf{p}, \mathbf{d}, \mathbf{w} \rangle$ , gdzie:  $\mathbf{p} = [p_{i,j}]_{n \times m}$ ,  $\mathbf{d} = [d_{i,j}]_{n \times m}$  oraz  $\mathbf{w} = [w_{i,j}]_{n \times m}$  są odpowiednio macierzą: czasów wykonania prac, terminów zakończenia oraz współczynników kary, będzie pewnym przykładem danych (deterministycznych) dla problemu **PFS**. Przez  $D(\theta)$  oznaczamy zbiór danych generowanych z  $\theta$  przez zaburzenie czasów wykonywania. Zaburzenie to polega na zmianie czasów wykonywania prac  $\mathbf{p} = [p_{i,j}]_{n \times m}$ , na losowo wyznaczone wartości (tj. liczby generowane zgodnie z pewnym przyjętym rozkładem, np. jednostajnym itd.). Dowolny element zbioru  $D(\theta)$  jest postaci  $\langle \mathbf{p}', \mathbf{d}, \mathbf{w} \rangle$  gdzie zaburzone elementy macierzy  $\mathbf{p}' = [p'_{i,j}]_{n \times m}$ , są wyznaczone losowo. Wobec tego, zbiór  $D(\theta)$  zawiera przykłady danych deterministycznych dla problemu **PFS**, różniące się pomiędzy sobą jedynie wartościami czasów wykonywania prac.

Niech  $A = \{TS, TSP, TSR\}$ . Algorytmy  $TS$ ,  $TSP$  i  $TSR$  są odpowiednio: deterministycznym, probabilistycznym i rozmytym. Przez  $\pi_{\delta}^A$  oznaczamy rozwiązanie (permutację) wyznaczoną przez algorytm  $A$  dla danych  $\delta$ . Wartość wyrażenia  $W(\pi_{\delta}^A, \varphi)$  jest wartością funkcji celu dla przykładu danych deterministycznych  $\varphi$ , gdy obiekty są wykonywane w kolejności (permutacji)  $\pi_{\delta}^A$  (tj. w kolejności wyznaczonej przez algorytm  $A$  dla danych  $\delta$ ). Wówczas:

$$\Delta(A, \delta, D(\delta)) = \frac{1}{|D(\delta)|} \sum_{\varphi \in D(\delta)} \frac{W(\pi_{\delta}^A, \varphi) - W(\pi_{\varphi}^{TS}, \varphi)}{W(\pi_{\varphi}^{TS}, \varphi)} \cdot 100\%, \quad (12)$$

nazywamy *stabilnością rozwiązania*  $\pi_\delta^A$  wyznaczonego przez algorytm  $A$  na zbiorze danych zaburzonych  $D(\delta)$ . Wyznaczając  $\pi_\varphi^{TS}$ , za rozwiązanie startowe algorytmu  $TS$  przyjęto  $\pi_\varphi^A$  i wówczas  $W(\pi_\delta^A, \varphi) - W(\pi_\varphi^{TS}, \varphi) \geq 0$ . Stąd  $\Delta(A, \delta, D(\delta)) \geq 0$ . Wartość wyrażenia  $\Delta(A, \delta, D(\delta))$  jest średnim błędem względnym najlepszego rozwiązania  $\pi_\delta^A$  do najlepszych rozwiązań wyznaczonych, dla każdego przykładu danych zaburzonych  $\varphi \in D(\delta)$ .

Niech  $\Omega$  będzie pewnym zbiorem przykładów deterministycznych dla problemu **PFS**. *Współczynnik stabilności algorytmu  $A$*  na zbiorze  $\Omega$  definiujemy następująco:

$$S(A, \Omega) = \frac{1}{|\Omega|} \sum_{\delta \in \Omega} \Delta(A, \delta, D(\delta)). \quad (13)$$

Im ten współczynnik jest mniejszy, tym wyznaczone przez algorytm  $A$  rozwiązania są stabilniejszej, tj. niewielkie zmiany wartości danych (zaburzenia przyjętych czasów wykonywania prac) powodują niewielkie zmiany wartości funkcji celu.

## 5. EKSPERYMENTY OBLICZENIOWE

Przedstawione w pracy algorytmy zostały zaprogramowane w języku C++. Eksperymenty obliczeniowe wykonano na komputerze osobistym z procesorem 2.2GHz. Dane deterministyczne wygenerowano na bazie 31 przykładów o rozmiarach  $(n \times m)$  od  $11 \times 5$  do  $75 \times 20$  zamieszczonych na stronie ORLiB [21]. Do każdego przykładu (macierzy czasów wykonywania prac  $p = [p_{i,j}]_{n \times m}$ ) dodano macierz żądanych terminów zakończenia prac  $d = [d_{i,j}]_{n \times m}$  oraz macierz współczynników kary za spóźnienie  $w = [w_{i,j}]_{n \times m}$ . Elementy obu macierzy generowano losowo zgodnie z rozkładem jednostajnym odpowiednio ze zbioru  $\{1, 2, \dots, \sum_{i=1}^n \sum_{j=1}^m p_{ij}\}$  oraz  $\{1, 2, \dots, 10\}$ . Niech  $\Omega$  będzie zbiorem tak wygenerowanych przykładów. Na bazie przykładów z  $\Omega$  wygenerowano dane:

- rozmyte, dla algorytmu **TSR** (dokładnie jest to opisane w 3.2),
- probabilistyczne, dla algorytmu **TFP** (dokładnie jest to opisane w 3.1).

Następnie dla każdego przykładu danych deterministycznych  $\delta \in \Omega$  wygenerowano 100 przykładów danych zaburzonych (elementy zbioru  $D(\delta)$ ). Zaburzenie czasu  $p_{ij}$  ( $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ ) polega na zastąpieniu go nową wartością wylosowaną zgodnie z rozkładem normalnym  $N(p_{ij}, p_{ij}/10)$ . W sumie wyznaczono 3100 przykładów danych zaburzonych. Przy uruchomieniu każdego algorytmu za permutację startową przyjęto  $\pi = (1, 2, \dots, n)$ , a ponadto:

- długość listy ruchów zakazanych:  $n$ ,
- liczba iteracji algorytmu:  $n/2$  lub  $n$ .

Dla każdego rozwiązania wyznaczonego przez algorytm  $A = \{TS, TSP, TSR\}$  wyznaczono procentowy błąd względny (ang. *percentage relative deviation*):

$$\varepsilon(A) = \frac{W_A - W_{NEH}}{W_{NEH}} \cdot 100\% ,$$

gdzie:

$W_A$  jest wartością rozwiązania wyznaczonego przez algorytm  $A$ , a  $W_{NEH}$  wartością wyznaczoną przez najlepszy algorytm konstrukcyjny  $NEH$  (Navaz i in. [9]) dla problemu przepływowego z kryterium  $C_{max}$ . Średnie wartości tych błędów przedstawiono w tabeli 1. Wartości rozwiązań wyznaczone przez każdy z trzech przedstawionych algorytmów są lepsze od rozwiązań wyznaczonych przez  $NEH$ . Najlepszym okazał się algorytm deterministyczny (średnia poprawa 6,1%), a najgorszym – algorytm probabilistyczny (średnia poprawa 1,7%). Następnie, wyznaczono współczynniki stabilności (11) algorytmu deterministycznego  $TS$ , rozmytego  $TSR$  oraz probabilistycznego  $TSP$ . Obliczenia wykonano dla liczby iteracji algorytmu  $n/2$  oraz  $n$ . Dla największych przykładów jest to nie więcej niż 70 iteracji. Dzięki temu całkowity czas obliczeń trzech badanych algorytmów nie przekracza 5 minut. Przy większej (np.  $n^2$ ) liczbie iteracji, stabilność poszczególnych algorytmów ulega jedynie niewielkiemu zwiększeniu. Wzrasta za to znacznie czas obliczeń. Dokładne wyniki zamieszczono w tabeli 1.

Tabela 1. Średni błąd względny  $\varepsilon(A)$  oraz stabilność  $S(A, \Omega)$  algorytmów: deterministycznego  $TS$ , rozmytego  $TSR$  oraz probabilistycznego  $TSP$

Liczba iteracji	Średni błąd względny			Stabilność		
	Algorytm					
	$TS$	$TSR$	$TSP$	$TS$	$TSR$	$TSP$
$n/2$	-4,8	-1,5	-0,6	6,1	2,8	6,0
$n$	-7,4	-4,8	-2,7	7,8	4,0	7,4
Średnia	<b>-6,1</b>	<b>-3,2</b>	<b>-1,7</b>	<b>6,9</b>	<b>3,49</b>	<b>6,4</b>

Źródło: Opracowanie własne

Najbardziej stabilny jest algorytm rozmyty  $TSR$ , dla którego współczynnik stabilności  $S(TSR, \Omega) = 3,4\%$ . Po zaburzeniu danych średnio wartość funkcji celu wzrasta o 3,4% w stosunku do wartości najlepszej wyznaczonej przez algorytm  $TS$ . Algorytmy  $TS$  i  $TSP$  mają podobne wartości współczynników stabilności  $S(TS, \Omega) = 6,9\%$  oraz  $S(TSP, \Omega) = 6,4\%$ . Są jednak znacznie mniej stabilne niż algorytm  $TSR$ .

Przy porównaniu wyników, pewnym zaskoczeniem może być zwiększanie się współczynnika stabilności (tj. pogorszenie stabilności) wraz ze wzrostem liczby iteracji. Dotyczy to zarówno algorytmu deterministycznego, jak i probabilistycznego. Wynika to z faktu, że lepsze rozwiązania (a takie otrzymano po dwukrotnym zwiększeniu liczby iteracji) są bardziej wrażliwe na wszelkie zaburzenia danych.

Na podstawie zamieszczonych wyników możemy stwierdzić, że rozwiązania (tj. permutacje, czyli kolejności wykonywania obiektów) wyznaczone przez algorytm

rozmyty *TSR* są znacznie bardziej stabilne. Niepewność szacunków czasów wykonywania obiektów, dla rozwiązania uzyskanego przez algorytm rozmyty, ma znacznie mniejszy wpływ na ostateczne terminy ich zakończenia (a więc i ewentualne spóźnienia) niż w przypadku pozostałych algorytmów. Zatem można wnioskować, że gdy kolejność zadań jest wyznaczona przez algorytm z rozmytymi parametrami, znacznie mniej zmienia się wartość funkcji celu (suma kar za przekroczenie ustalonych terminów zakończenia poszczególnych prac) przy zaburzeniu czasów wykonywania prac.

## 6. STUDIUM PRZYPADKU

Zadanie inwestycyjne dotyczy realizacji kompleksu dwunastu budynków mieszkalnych ( $n=12$ ). Projekty budynków charakteryzują się podobnym zestawem robót budowlanych tworzących uporządkowany dziewięcioelementowy ( $m=9$ ) ciąg prac rozpoczynających się od robót ziemnych, a kończący na robotach wykończeniowych. Na podstawie Katalogu Nakładów Rzeczowych wyznaczono czasy (macierz  $p$ ) wykonywania robót (w dniach roboczych):

7	8	7	7	7	8	7	7	6	7	5	4
8	11	8	9	9	11	8	9	8	9	8	8
8	11	10	9	9	11	10	9	11	9	9	9
7	8	7	7	8	8	7	7	8	8	8	7
6	7	7	7	7	7	7	7	7	7	8	15
11	14	11	13	13	14	11	13	14	13	14	8
9	14	9	11	10	13	9	11	8	10	11	9
4	8	6	7	5	7	7	8	9	9	9	5
6	9	5	9	7	5	8	9	8	7	7	7

W trakcie realizacji okazało się, że rzeczywiste czasy (macierz  $p'$  – dane zaburzone) różnią się od wcześniej wyznaczonych. Zgodnie z opisem zamieszczonym w rozdziale 4 wyznaczono terminy zakończenia prac (macierz  $d$ ) oraz wagi funkcji kary (macierz  $w$ ). Następnie, na bazie macierzy  $p$ , wygenerowano przykład danych rozmytych oraz probabilistycznych. Korzystając z algorytmu deterministycznego (*TS*), rozmytego (*TSR*) i probabilistycznego (*TSP*), wyznaczono trzy permutacje (kolejności wykonywania budynków). Dla danych zaburzonych (macierz  $p'$ ) i każdej permutacji obliczono wartość funkcji kary. Wzrost wartości funkcji kary, dla algorytmu rozmytego, probabilistycznego i deterministycznego wynosił odpowiednio: 3,6%, 9,1% i 17,4%. Najbardziej stabilny okazał się algorytm rozmyty *TSR*. Zmiana czasów wykonania prac (w stosunku do wyznaczonych) spowodowała zwiększenie się kary za przekroczenie ustalonych terminów zakończenia prac o 3,6%.

## PODSUMOWANIE

W pracy przedstawiono metodykę harmonogramowania prac budowlanych z uwzględnieniem niepewnych danych dotyczących czasów wykonania robót. W związku z tym, że bardzo trudno określić z góry jednoznacznie czasy wykonywania poszczególnych robót, zaproponowano wykorzystanie do ich oszacowania liczb rozmytych oraz zmiennych losowych.

Do rozwiązywania zadania optymalizacyjnego zastosowano algorytm oparty na metodzie przeszukiwania z tabu. Na podstawie przeprowadzonej analizy wyników stwierdzono, że zaburzenie czasów wykonywania zadań dla rozwiązania wyznaczonego

przez algorytm rozmyty, ma znacznie mniejszy wpływ na termin oraz koszt wykonania prac niż w przypadku rozwiązań wyznaczonych przez pozostałe algorytmy.

Niniejsza praca stanowi kontynuację studiów autorów nad zastosowaniem algorytmów metaheurystycznych w harmonogramowaniu robót budowlanych z niepewnymi parametrami.

## LITERATURA

1. Afanasjev V.A, Afanasjev A.V., *Potocnaja organizacija rabot v stroitelstwie*, Sankt-Petersburg 2000.
2. Arditi D., Tokdemir O.B., Suh K., *Scheduling system for repetitive unit construction using line-of-balance technology*, Department of Civil and Architectural Engineering, Illinois Institute of Technology, Department of Civil Engineering, Honan University, Honam, South Korea 2002.
3. Bożejko W., Hejducki Z., Rogalska M., Wodecki M., *Scheduling of construction projects with a hybrid evolutionary algorithm's application*, [in:] *Evolutionary Algorithms*, edited Eisuke Kita, INTECH, Rijeka 2011.
4. Bożejko W., Wodecki M., *Solving Flow Shop Problem by Parallel Simulated Annealing*, LNCS, Springer-Verlag, 2328, 2002, pp. 236-244.
5. Dean B.C., *Approximation algorithms for stochastic scheduling problems*, PhD thesis, MIT, 2005.
6. Dubois D., Prade H., *Theorie des Possibilites. Applications a la representation des connaissances en informatique*, MASSON, Paris 1988.
7. Garey M.R., Johnson D.S., Seti R., *The complexity of flowshop and jobshop scheduling*, [in:] "Mathematics of Operations Research", no. 1/1976, pp. 117-129.
8. Grabowski J., Wodecki M., *A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion*, [in:] "Computers & Operations Research", no. 31/2004, pp. 1891-1909.
9. Hamerlink D.J., Rowings J.E., *Linear scheduling model: Development of controlling activity path*, [in:] "Journal of Construction Engineering and Management", no. 124(4)/1998, pp. 266-268.
10. Harris R.B., Ionnou P.G., *Scheduling projects with repeating activities*, [in:] "Journal of Construction Engineering and Management", no. 124(4)/1998, pp. 269-278.
11. Hejducki Z., *Zarządzanie czasem w procesach budowlanych z zastosowaniem modeli macierzowych*, Oficyna Wydawnicza PWR, Wrocław 2004.
12. Hejducki Z., Rogalska M., *Time coupling methods*, Oficyna Wydawnicza PWR, Wrocław 2011.
13. Ishibuschi H., Murata T., *Scheduling with Fuzzy Duedate and Fuzzy Processing Time*, [in:] *Scheduling Under Fuzziness*, edited Słowiński R., Hapke M., Springer-Verlag, 2000, pp. 113-143.
14. Ishii H., *Fuzzy combinatorial optimization*, [in:] "Japanese Journal of Fuzzy Theory and Systems", vol. 4, no. 1/1992.

15. Karp R.M., *Reducibility among combinatorial problems*, [in:] *Complexity of Computer Computation*, edited Miller R.E., Thatcher J.W., Plenum Press, New York 1972, pp. 85-104.
16. Mattila K.G., Acse A.M., Park A., *Comparison of Linear Scheduling Model and Repetitive Scheduling Method*, [in:] "J.Constr. Eng. Manage.", no. 129(1)/2003, pp. 56-64.
17. Mattila K.G., Abraham D.M., *Resource leveling of linear schedules using integer linear programming*, [in:] "Journal of Construction Engineering and Management", ASCE, no. 120(4)/1998, pp. 232-244.
18. Navaz M., Enscore E.E., Ham I., *A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem*, [in:] "OMEGA", no. 11/1/1983, pp. 91-95.
19. Nowicki E., Smutnicki C., *A fast tabu search algorithm for the permutation flow-shop problem*, [in:] "European Journal of Operational Research", no. 91/1996, pp. 160-175.
20. Ogbu F., Smith D., *The Application of the Simulated Annealing Algorithm to the Solution of the n/m/Cmax Flowshop Problem*, [in:] "Computers & Operations Research", no. 17/3/1990, pp. 243-253.
21. OR Library, [online]. [dostęp: 2012]. Dostępny w Internecie: <http://people.brunel.ac.uk/mastjbjeb/orlib/flowshopinfo.html>.
22. Osman I., Potts C., *Simulated Annealing for Permutation Flow-Shop Scheduling*, [in:] "OMEGA", no. 17/6/1989, pp. 551-557.
23. Reeves C., *A Genetic Algorithm for Flowshop Sequencing*, [in:] "Computers & Operations Research", no. 22/1/1995, pp. 5-13.
24. Rogalska M., Bożejko W., Hejducki Z., Wodecki M., *Harmonogramowanie robót budowlanych z zastosowaniem algorytmu tabu search i rozmytymi czasami wykonania robót*, [w:] „Przegląd Budowlany”, 7/8/2007, Warszawa 2007, s. 76-80.
25. Senouci, A.B., Eldin, N.N., *Dynamic programming approach to scheduling of nonserial linear project*, [in:] "Journal of Computing in Civil Engineering", no. 10/1996, pp. 106-144.
26. Skorupka D., *Risk Management in Building Projects*, [in:] AACE International Transaction, (CSC.1.91– CSC.1.96), The Association for the Advancement of Cost Engineering, USA, Orlando 2003.
27. Thabet W.Y., Beliveau Y.J., *HVLS: horizontal and vertical logic scheduling for multistory projects*, [in:] "Journal of Construction Engineering and Management", no. 120/1994, ASCE, pp. 875-892.
28. Ustinovicus L., *Decision-support System for Determining the Efficiency of Investments in Construction: Summary of the research report presented for habilitation*, Technika, Vilnius 2003.
29. Vondrák J., *Probabilistic methods in combinatorial and stochastic optimization*, PhD, MIT, 2005.
30. Vorster M. C., Parvin C. M., *Linear scheduling for highway contractors and state DOT's*, Videotapes, P&W Publications, Richmond, Va, 1990.

31. Vorster M.C., Beliveau Y.J., Bafna, T., *Linear scheduling and visualization*, [in:] "Transp. Res. Rec.", no. 1351/1992, pp. 32-39.
32. Wang C., Huang Y., *Controlling activity interval times in LOB scheduling*. [in:] "Construction Management and Economics", no. 16/1998, pp. 5-16.
33. Zavadskas E.K., *Book review. Methods and models of research in construction projects engineering*, [in:] "Journal of Business Economics and Management", no. 9(3)/2008, pp. 240-243.
34. Zavadskas E.K., *History and evolving trends of construction colloquia on sustainability and operational research*, [in:] "Technological and Economic Development of Economy", no. 14(4)/2008, pp. 578-592.

## CONSTRUCTION SCHEDULING WITH TABU SEARCH ALGORITHM AND UNCERTAIN DATA

### Summary

*This paper deals with some problems of synchronizing construction activities differing in their execution times. The methodology of calculating the times of execution of the activities, ensuring that there will be no collisions between them, is presented. The methodology is illustrated with numerical examples showing the successive steps of the algorithm and it is applied to complex works modelled as the flow shop problem. The tabu search algorithm is adapted to solve the problem.*

**Keywords:** *uncertain data, stability, scheduling, tabu search, construction works*