ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2008
Seria: AUTOMATYKA z. 151 Nr kol. 1797

Wojciech BOŻEJKO, Zdzisław HEJDUCKI, Michał PODOLSKI,
Mariusz UCHROŃSKI
Politechnika Wrocławska

HARMONOGRAMOWANIE PRZEDSIĘWZIĘĆ BUDOWLANYCH ZA POMOCĄ PROBLEMU PRZEPŁYWOWEGO ZE SPECYFICZNYMI CZASAMI TRANSPORTU

Streszczenie. W pracy rozpatrujemy permutacyjny problem przepływowy będący jednym z najtrudniejszych problemów kombinatorycznych. Dla zamodelowania specyficznych własności pojawiających się w praktycznych problemach harmonogramowania przedsięwzięć budowlanych zastosowaliśmy czasy transportu o ujemnej wartości. Przedstawiamy algorytm jego rozwiązywania oparty na metodzie tabu search.Wykonano wiele obliczeń dla trudnych przykładów o dużych rozmiarach zamieszczonych w pracy Taillard [14]. Otrzymane wyniki obliczeniowe porównano z powszechnie obecnie stosowanym algorytmem konstrukcyjnym NEH ([13]), wynikami dokładnego algorytmu *B&B* oraz najlepszymi znanymi w literaturze rozwiązaniami.

SCHEDULING OF CONSTRUCTION PROJECTS BY USING THE FLOW SHOP PROBLEM WITH SPECIFIC TRANSPORT TIMES

Summary. In this paper we consider a permutational flow shop problem which is one of the difficult combinatorial problems. A negative times of transports have been using to model a specific constraints of the problem. We propose a tabu search approach. Tests were done for a Taillard [14] benchmarks. Obtained results were compared to constructive algorithm NEH ([13]), exact *B&B* algorithm and the best known results from literature.

1. Wprowadzenie

Zagadnienie przedstawione w pracy dotyczy harmonogramowania przedsięwzięć budowlanych, w których - posługując się językiem automatyki - zadania powinny się rozpoczynać na następnej maszynie przed zakończeniem się na poprzedniej. W klasycznym permutacyjnym problemie przepływowym (*permutation flow shop problem*) każde z zadań należy wykonać kolejno na wszystkich maszynach, przy czym kolejność wykonywania zadań na każdej maszynie musi być taka sama, a rozpoczęcie wykonywania na maszynie następnej może nastąpić dopiero po zakończeniu jego wykonywania na poprzedniej maszynie. Optymalizacja polega na wyznaczeniu kolejności wykonywania zadań, która minimalizuje całkowity czas ich wykonywania.

Dla zamodelowania zjawiska "zakładania" się zadań zastosowano czasy transportu zadań pomiędzy maszynami, które przyjmować mogły wartości ujemne. W literaturze problem ten jest oznaczany przez $F|t_{ij}|C_{max}$ i jest on silnie NP-trudny (ponieważ dla $t_{ij} = 0$ dostajemy klasyczny problem przepływowy z kryterium C_{max}). Algorytmy optymalne rozwiązywania problemu bez czasów transportu oparte na metodzie podziału i ograniczeń pozwalają na rozwiązywanie, w rozsądnym czasie, przykładów o rozmiarze nie większym niż 20 zadań i 5 maszyn. W ostatnich latach opublikowano wiele algorytmów rozwiązywania problemu $F||C_{max}$ opartych na metodzie tabu search [6], [7] i [11]. Generalnie, jakość wyznaczanych przez te algorytmy rozwiązań zależy w dużym stopniu od czasu ich działania. Im ten czas jest większy, tym przeglądany jest (bezpośrednio) większy podzbiór zbioru rozwiązań dopuszczalnych. Zwiększa to szansę na znalezienie dobrego rozwiązania. Prace dotyczące problemu z transportami (dodatnimi) to klasyczne opracowania [8] i [12].

W dalszej części przedstawimy algorytm oparty na metodzie tabu search, którego konstrukcja bazuje na najlepszym obecnie znanym w literaturze algorytmie przybliżonym (zamieszczonym w [7]). Intensywnie badane, od ponad 30 lat, permutacyjne zagadnienie przepływowe jest bardzo trudnym, dziś już klasycznym, problemem optymalizacyjnym. Ciągle jednak brak jest zadowalających algorytmów jego rozwiązywania szczególnie, dla praktycznych przykładów o dużych rozmiarach. W kolejnych rozdziałach pracy przedstawiamy: model matematyczny rozważanego problemu szeregowania, algorytm tabu search, a na koniec wyniki obliczeniowe oraz podsumowanie.

2. Model matematyczny

Problem przepływowy z transportem można sformułować w następujący sposób. Dany jest zbiór zadań $J = \{J_1, J_2, \ldots, J_n\}$ przeznaczonych do realizacji z wykorzystaniem zbioru maszyn $M = \{M_1, M_{2,\ldots,M_n}\}$. Każde zadanie $J_i \in J$ składa się z ciągu niepodzielnych operacji $J_i = (O_{i1}, O_{i2}, \ldots, O_{im})$. Operacja O_{ik} ma być wykonana na maszynie M_k w czasie p_{ik}. Dla każdego zadania zdefiniowany jest czas transportu zadań między maszynami. Wielkość t_{ij} oznacza czas transportu zadania i z maszyny j na maszynę $j+1$, $j = 1, \ldots, m-1$. Przypadek $t_{ij} \geqslant 0$ ma naturalne uzasadnienie praktyczne i nie wymaga komentarza. Przypadek $t_{ij} < 0$ oznacza zezwolenie na "nakładanie" kolejnych operacji zadania lub też na rozpoczęcie kolejnej operacji zadania z pewnym poślizgiem czasowym w stosunku do rozpoczęcia operacji bieżącej, a przed jej ukończeniem.

Zakładając, że kolejność wykonywania zadań określona jest przez permutację π na $\mathcal{J}$ w permutacyjnym problemie przepływowym z transportem, terminy zakończenia wykonywania zadań można wyznaczyć na podstawie warunków:

$$C_{\pi(i)j} \geqslant C_{\pi(i-1)j} + p_{\pi(i)j}, \; i = 1, \ldots, n, \; j = 1, \ldots, m, \quad (1)$$

$$C_{\pi(i)j} \geqslant C_{\pi(i)j-1} + p_{\pi(i)j} + t_{\pi(i)j-1}, \; i = 1, \ldots, n, \; j = 1, \ldots, m. \quad (2)$$

powyższe warunki prowadzą do wzoru rekurencyjnego:

$$C_{\pi(i)j} = \max\{C_{\pi(i-1)j}, C_{\pi(i)j-1} + t_{\pi(i)j-1}\} + p_{\pi(i)j}, i = 1, \ldots, n, \quad (3)$$

gdzie $\pi(0) = 0$, przy czym dla ujemnych t_{ij} należy podstawić $C_{i0} = 0$, $i = 1, \ldots, n$, $C_{01} = 0$, $C_{0j} = -\infty$ $i = 2, \ldots, m$.

3. Algorytm tabu search

Metoda ta została zaproponowana przez Glovera [3], [4] i [5], a następnie rozwijana przez innych autorów [1],[10], [11] i [15]. Tabu search (TS) jest modyfikacją metody lokalnych poszukiwań. W podstawowej wersji metoda TS rozpoczyna swe działanie od pewnego rozwiązania początkowego $x^0 \in \mathcal{X}$. W elementarnym kroku tej metody zostaje przeszukane całe sąsiedztwo $\mathcal{N}(x^i)$ rozwiązania x^i. Sąsiedztwo jest definiowane przez ruchy, które można wykonać z x^i. Celem tego poszukiwania jest znalezienie w $\mathcal{N}(x^{i+1})$ rozwiązania z najmniejszą wartością funkcji celu $K(x)$. Proces poszukiwania jest kontynuowany od najlepszego znalezionego rozwiązania.

W celu zapobieżenia cyklicznemu powtarzaniu się rozwiązań, zatrzymania w ekstremum lokalnym oraz aby skierować poszukiwania w obiecujące obszary rozwiązań, wprowadzono pamięć historii poszukiwań w postaci listy zabronień (tzw. listy tabu). Na liście tej przechowywana jest określona liczba ostatnio "odwiedzanych" rozwiązań (rozwiązania bazowe). Rozwiązania te nie są pamiętane bezpośrednio, ale w postaci pewnych ich "atrybutów". Powoduje to, że zabronienia z nich wynikające w danej iteracji dotyczą także rozwiązań, które nie były dotychczas rozwiązaniami bazowymi. W celu osłabienia tej restrykcji określana jest dodatkowo dla zabronionego rozwiązania pewna funkcja aspiracji. Jeśli wartość tej funkcji jest mniejsza od zadanego poziomu, to rozwiązania nie traktuje się jako zabronionego. W trakcie wykonywania kolejnych iteracji pamiętane jest najlepsze znalezione rozwiązanie x^* w sensie wartości funkcji celu oraz odpowiadająca mu wartość tej funkcji. Poszukiwanie zatrzymuje się w chwili zadziałania odpowiednich warunków stopu. Podsumowując, do podstawowych elementów metody tabu można zaliczyć:

ruch: funkcja, która przekształca jedno rozwiązanie w drugie,

sąsiedztwo: zbiór rozwiązań możliwych do uzyskania z ustalonego rozwiązania za pomocą klasy ruchów,

rozwiązanie początkowe: rozwiązanie, od którego algorytm rozpoczyna działanie,

lista tabu: lista, na której zapisywane są atrybuty ruchów lub rozwiązań dla ustalonej liczby ostatnio rozpatrywanych rozwiązań,

kryterium aspiracji: warunki, przy których w kolejnym kroku algorytmu można użyć rozwiązania zabronionego,

warunek zakończenia: sytuacja, w której algorytm kończy swoje działanie, np. (1) wykonana została założona z góry liczba iteracji, (2) zakończył się czas przeznaczony na działanie algorytmu, (3) w kolejnych iteracjach wartość funkcji celu nie uległa zmianie.

Poniżej zostały przedstawione zostały poszczególne kroki algorytmu tabu search.

Krok 0. Znaleźć rozwiązanie początkowe π_0;
$\pi = \pi_0$;
$\pi^* = \pi$;

Krok 1. Wyznaczyć sąsiedztwo $\mathcal{N}(\pi)$ permutacji π, usunąć z sąsiedztwa elementy zabronione przez listę T oprócz takich $\beta \in \mathcal{N}(\pi)$, dla których $C_{max}(\beta) < C_{max}(\pi^*)$;

Krok 2. Znaleźć taką permutację $\sigma \in \mathcal{N}(\pi)$,
dla której $C_{max}(\sigma) = min\{C_{max}(\beta) : \beta \in \mathcal{N}(\pi)\}$;

Tabela 1

Procentowy błąd względny algorytmu tabu search względem algorytmu NEH

$n \times m$	TS1 (1000 iter.)	TS2 (2000 iter.)	TS5 (5000 iter.)
20×5	-2.31%	-2.31%	-2.38%
20×10	-4.91%	-5.02%	-5.16%
20×20	-4.44%	-4.49%	-4.58%
50×5	-0.94%	-0.94%	-0.94%
50×10	-4.30%	-4.57%	-4.59%
50×20	-4.82%	-5.18%	-5.48%
100×5	-0.79%	-0.81%	-0.81%
średnia	-3.22%	-3.33%	-3.42%

Krok 3. Jeżeli $C_{max}(\sigma) < C_{max}(\pi^*)$, to $\pi^* = \sigma$;
Umieścić atrybuty σ na liście T;
$\pi = \sigma$;

Krok 4. Jeżeli spełniony jest warunek zakończenia, to STOP;
w przeciwnym wypadku idź do Kroku 1.

4. Eksperymenty obliczeniowe

Ze względu na brak literaturowych instancji testowych dla rozpatrywanego problemu zaprojektowane zostały odpowiednie instancje testowe. Instancje te zostały wygenerowane w oparciu o instancje autorstwa Taillarda [14], [16] dla klasycznego problemu przepływowego. Zestaw instancji Taillarda składa się ze 120 przykładów podzielonych na 12 grup o różnych rozmiarach. Dla każdego rozmiaru $n \times m$: 20×5, 20×10, 20×20, 50×5, 50×10, 50×20, 100×5, 100×10, 100×20, 200×10, 200×20, 500×20 występuje 10 instancji.

Ze względu na brak w literaturze algorytmów rozwiązania rozważanego problemu, jakość permutacji (w sensie wartości funkcji celu) generowanych przez algorytm TS została porównana z rozwiązaniami uzyskanymi za pomocą algorytmu NEH. Dodatkowo zostały stworzone instancje problemu o małych rozmiarach $n \times m$: 10×10. Dla tych instancji zostało wyznaczone rozwiązanie optymalne przez przegląd zupełny zbioru rozwiązań. Uzyskane wartości zostały porównane z wynikami, jakich dostarczyły algorytmy NEH i TS.

Do wyznaczenia permutacji początkowej dla algorytmu TS został użyty odpowiednio zaadoptowany algorytm NEH [13]. Adaptacja tego algorytmu do potrzeb rozważanego problemu polegała na uwzględnieniu czasów transportu zadań między maszynami podczas wyznaczania wartości funkcji celu dla każdej permutacji częściowej przetwarzanej w algorytmie.

Dla każdej instancji problemu zostały obliczone następujące wielkości:

- C^A - wartość funkcji celu otrzymana algorytmem TS,
- $PRD(A) - 100\%(C(A) - C^{ref})/C^{ref}$ procentowa różnica wartości funkcji celu uzyskanej algorytmem A względem wartości C^{ref} uzyskanej algorytmem NEH.

Tabela 2

Wartości funkcji celu dla rozwiązania optymalnego oraz rozwiązań uzyskanych za pomocą algorytmu NEH i tabu search

instancja		C_{max}			PRD[%]	
nazwa	$n \times m$	OPT	NEH	TS	NEH	TS
TM21	10×10	908	950	908	4.63	0.00
TM22	10×10	784	803	784	2.42	0.00
TM23	10×10	883	912	883	3.28	0.00
TM24	10×10	922	971	922	5.31	0.00
TM25	10×10	884	884	884	0.00	0.00
TM26	10×10	888	919	888	3.49	0.00
TM27	10×10	911	915	911	0.44	0.00
TM28	10×10	835	861	835	3.11	0.00
TM29	10×10	881	929	881	5.45	0.00
TM30	10×10	825	867	825	5.09	0.00
średnia					3.32	0.00

Tabela 3

Procentowy błąd względny algorytmu tabu search i NEH względem rozwiązań Taillarda

$n \times m$	NEH	TS1 (1000 iter.)	TS5 (5000 iter.)
20×5	3.36%	0.08%	0.02%
20×10	4.60%	0.27%	0.20%
20×20	3.73%	0.30%	0.22%
50×5	0.71%	0.16%	0.16%
50×10	4.14%	-0.02%	-0.12%
50×20	5.70%	0.52%	0.01%
100×5	0.50%	-0.01%	-0.01%
średnia	3.25%	0.19%	0.07%

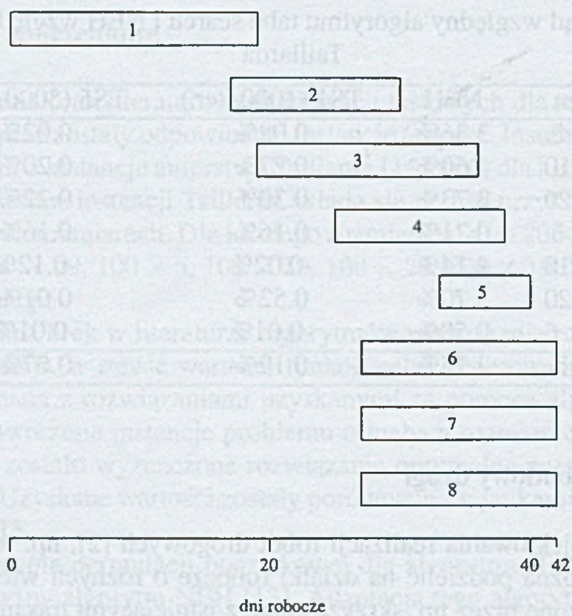
5. Harmonogram budowy drogi

Podczas projektowania realizacji robót drogowych [2], np. remontu drogi, całe przedsięwzięcie można podzielić na działki robocze o różnych wielkościach, których granice są wyznaczone przez np. skrzyżowania z istniejącymi trasami drogowymi. Kolejność zajmowania działek przez brygady wykonujące procesy wpływać będzie na czas całego przedsięwzięcia, czas przestoju maszyn lub brygad roboczych. Problem ustalenia optymalnej kolejności prowadzenia robót na poszczególnych działkach ze względu na założone kryterium, np. minimalny czas realizacji przedsięwzięcia, minimalny czas przestoju brygad roboczych czy koszt realizacji, jest zagadnieniem szeregowania zadań.

W celu dokonania właściwego podziału robót niezbędne jest określenie rodzaju robót według klasyfikacji ogólnej. Ogólna klasyfikacja robót w budownictwie drogowym i mostowym jest następująca:

1. roboty przygotowawcze,
2. roboty ziemne,
3. zagęszczenie gruntów oraz nawierzchni,
4. budowa profilowanych nawierzchni gruntowych i żwirowych,
5. wzmacnianie nawierzchni dróg gruntowych oraz podłoża,
6. budowa nawierzchni tłuczniowych,
7. budowa nawierzchni z betonu asfaltowego,
8. budowa nawierzchni z betonu cementowego,
9. naprawa, konserwacja, utrzymanie i rekonstrukcja nawierzchni,
10. wydobywanie, przerabianie i uszlachetnianie kruszywa łącznie z zagospodarowaniem kruszyw pozyskanych w procesie recyklingu,
11. roboty załadunkowe, wyładunkowe i transportowe,
12. budowa mostów i przepustów,
13. wytwarzanie i przesył energii,
14. rozbiórka starych obiektów.

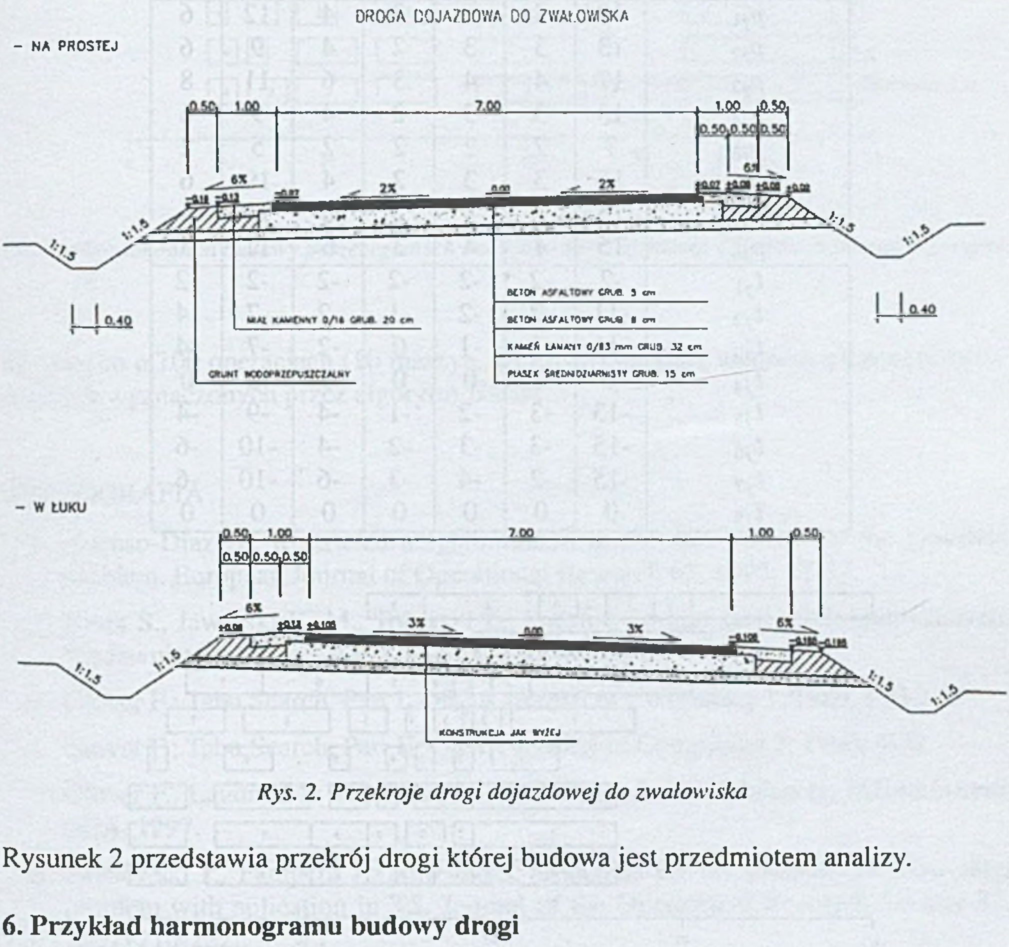
Na rysunku 1 został przedstawiony harmonogram robót dla pojedynczego odcinka drogi o długości 766 m. Czas wykonania wszystkich robót na tym odcinku wynosi 42 dni robocze. Kolejność technologiczna robót dla tego odcinka jest następująca:



Rys. 1. Przykład harmonogramu dla odcinka drogi o długości 766 m

1. roboty ziemne - 19 dni roboczych,
2. wykonanie warstwy odsączającej z piasku - 13 dni roboczych,
3. wykonanie podbudowy z tłucznia lub kamienia łamanego - 17 dni roboczych,
4. wykonanie warstwy wiążącej z betonu asfaltowego średnioziarnistego - 13 dni roboczych,

5. wykonanie warstwy ścieralnej z betonu asfaltowego drobnioziarnistego - 7 dni roboczych,
6. wykonanie poboczy z miału kamiennego - 15 dni roboczych,
7. wykonanie odwodnienia - 15 dni roboczych,
8. plantowanie i ewentualne humusowanie terenu - 15 dni roboczych.



Rys. 2. Przekroje drogi dojazdowej do zwałowiska

Rysunek 2 przedstawia przekrój drogi której budowa jest przedmiotem analizy.

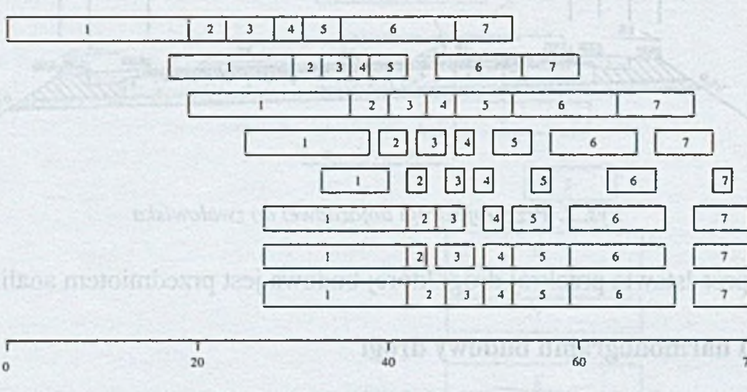
6. Przykład harmonogramu budowy drogi

Na podstawie danych praktycznych zamieszczonych powyżej zostały wygenerowane dane dotyczące budowy odcinków drogi o różnych długościach. W tabeli 4 zostały umieszczone dane w postaci czasów trwania poszczególnych operacji na odcinkach roboczych wyrażone w dniach roboczych. Dla danych z tabeli 4 został uruchomiony algorytm TS. Wynikiem działania algorytmu jest permutacja $\pi = (4, 2, 5, 7, 6, 1, 3)$, dla której wartość funkcji celu wynosi $C_{max} = 75$ (dni roboczych). Na rysunku 4 został przedstawiony harmonogram budowy poszczególnych odcinków drogi dla uzyskanej permutacji π. Wartość funkcji celu w przypadku harmonogramu dla permutacji naturalnej (rysunek 3, $C_{max} = 78$) jest większa niż dla harmonogramu uzyskanego na podstawie wyników działania algorytmu TS (rysunek 4, $C_{max} = 75$).

Tabela 4

Dane do przykładu

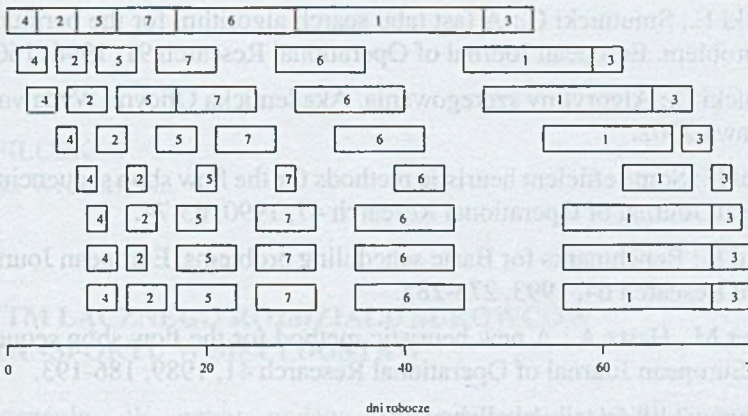
	zadania						
	1	2	3	4	5	6	7
dł. odcinka [m]	750	150	175	100	200	500	300
p_{j1}	19	4	5	3	4	12	6
p_{j2}	13	3	3	2	4	9	6
p_{j3}	17	4	4	3	6	11	8
p_{j4}	13	3	3	2	4	9	6
p_{j5}	7	2	2	2	2	5	2
p_{j6}	15	3	3	2	4	19	6
p_{j7}	15	2	4	3	6	10	6
p_{j8}	15	4	4	3	6	11	10
t_{j1}	-2	-2	-2	-2	-2	-2	-2
t_{j2}	-11	-2	-2	-1	-2	-7	-4
t_{j3}	-11	-1	-1	0	-2	-7	-4
t_{j4}	-5	0	0	0	0	-3	0
t_{j5}	-13	-3	-2	-1	-4	-9	-4
t_{j6}	-15	-3	-3	-2	-4	-10	-6
t_{j7}	-15	-2	-4	-3	-6	-10	-6
t_{j8}	0	0	0	0	0	0	0



Rys. 3. Harmonogram budowy poszczególnych odcinków drogi dla permutacji naturalnej

7. Podsumowanie

W pracy przedstawiliśmy konstrukcję algorytmu rozwiązywania zagadnienia harmonogramowania przedsięwzięć budowlanych modelowanego za pomocą problemu przepływowego z transportami $F|t_{ij}|C_{max}$ opartą na metodzie przeszukiwania z zabronieniami (*tabu search*). Otrzymane wyniki (po niewielkiej liczbie iteracji) dla dużych przykładów z literatury tylko nieznacznie różnią się od najlepszych obecnie znanych, a



Rys. 4. Harmonogram budowy poszczególnych odcinków drogi uzyskany w wyniku działania algorytmu TS

dla danych o 100 operacjach (10 maszyn, 10 zadań) osiągają wartości rozwiązań optymalnych wyznaczonych przez algorytm $B\&B$.

BIBLIOGRAFIA

1. Adenso-Diaz B.: Restricted neightborthood in the tabu search for the flowshop problem. European Journal of Operational Research 62, 1992, 27-37.
2. Biruk S., Jaworski K. M., Tokarski Z.: Podstawy organizacji robót budowlanych. Wydawnictwo Naukowe PWN, Warszawa 2007.
3. Glover F.: Tabu Search. Part I, ORSA Journal of Computing 1, 1989, 190-206.
4. Glover F.: Tabu Search. Part II, ORSA Journal of Computing 2, 1990, 4-32.
5. Glover F., Laguna M.: Tabu Search. Kluwer Academic Publishers, Massachusetts USA, 1997.
6. Grabowski J., Pempera J.: New block properties for the permutation flow-shop problem with aplication in TS. Journal of the Operational Research Society 52, 2001, 210-222.
7. Grabowski J., Wodecki M.: A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. Computers & Operations Research 31, 2004, 1891-1909.
8. Gupta J. N. D., Stafford Jr E. F.: Flowshop scheduling research after five decades. European Journal of Operational Research 169, 2006, 699-711.
9. Hurnik J.: Makespan minimization for flow-shop problem with transportation times and a single robot, Discrete Applied Mathematics 112, 2001, 199-216.
10. Nowicki E.: Metoda tabu w problemach szeregowania zadań produkcyjnych. Prace Naukowe ICT PWr, Seria Monografie, 1999.

11. Nowicki E., Smutnicki C.: A fast tabu search algorithm for the permutation flow-shop problem. European Journal of Operational Research 91, 1996, 160-175.
12. Smutnicki C.: Algorytmy szeregowania. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002.
13. Taillard E.: Some efficient heuristic methods for the flow shop sequencing problem. European Journal of Operational Research 47, 1990, 65-74.
14. Taillard E.: Benchmarks for Basic scheduling problems. European Journal of Operational Research 64, 1993, 278-285.
15. Widmer M., Hertz A.: A new heuristic method for the flow shop sequencing problem. European Journal of Operational Research 41, 1989, 186-193.
16. http://www2.lifl.fr/ talbi/challenge2007

Recenzent: Dr hab. inż. Mirosław Zaborowski

Abstract

In this paper we consider a permutational flow shop problem which is one of the difficult combinatorial problems. A negative times of transports have been using to model a specific constraints of the problem. We propose a tabu search approach. Tests were done for a Taillard [14] benchmarks. Obtained results were compared to constructive algorithm NEH ([13]), exact *B&B* algorithm and the best known results from literature.