

HARMONOGRAMOWANIE ROBÓT BUDOWLANYCH Z MINIMALIZACJĄ ŚREDNIEGO POZIOMU ZATRUDNIENIA

Wojciech BOŻEJKO, Zdzisław HEJDUCKI,
Michał PODOLSKI, Mariusz UCHROŃSKI

Streszczenie: w pracy proponujemy zastosowanie funkcji kryterialnej mierzącej zróżnicowanie poziomu zatrudnienia podczas wykonywania prac budowlanych. Dla zaproponowanego podejścia przedstawiamy algorytm oparty na metodzie poszukiwania z zabronieniami (*tabu search*) oraz przeprowadzamy eksperymenty obliczeniowe oraz analizę przypadku.*

Słowa kluczowe: algorytm poszukiwania z zabronieniami, szeregowanie zadań.

1. Wstęp

W pracy rozważany jest problem zaczerpnięty z praktyki budowlanej, a mianowicie harmonogramowania robót budowlanych na frontach roboczych. w literaturze najczęściej rozpatruje się to zagadnienie w kontekście *problemu przepływowego* (ang. *flow shop*), dla którego najpopularniejszą stosowaną funkcją kryterialną jest maksymalny czas zakończenia zadań C_{max} . Takie podejście prezentowane jest w pracach Bożejko i in. [2,3], Rogalska i oraz. [5,6]. w praktyce występuje szereg dodatkowych ograniczeń (zobacz [1,4]) – np. na placu budowy trzeba ustawić pewną liczbę baraków dla określonej (maksymalnej) liczby pracowników. Choć z harmonogramu wynika, że np. na budowie w pewnym momencie będzie aż 50 robotników, to w praktyce średnio jest ich np. 5. Ale ilość baraków i innych zasobów musi być przygotowana dla 50, choć ten moment będzie sporadyczny. w niniejszej pracy skupiliśmy się więc na badaniu średniego poziomu zatrudnienia i jego minimalizacji, jako na praktycznym aspekcie minimalizacji kosztów wykonania robót budowlanych.

2. Opis problemu

Zadanie optymalizacyjne dotyczy zagadnienia, gdy zadany jest czas trwania przedsięwzięcia budowlanego, dla którego należy znaleźć najlepszy, w sensie przyjętego kryterium optymalizacji, poziom zatrudnienia pracowników. Jednym z kryterium optymalizacji może być uwzględnienie średniej wartości zapotrzebowania na zasoby w konstruowanej funkcji celu. Jako miarę nierównomierności zapotrzebowania na pracowników można przyjąć średnie odchylenie od średniego dziennego zapotrzebowania, w celu poszukiwania minimalnej wartości. Funkcja celu ma postać:

* Praca finansowana z projektów badawczych MNiSW nr N N514 232237 oraz N N514 470439.

$$f(x) = \frac{1}{T} \sum_{j=1}^T \left| \frac{q_j(x) - r_{avg}}{r_{avg}} \right| \quad (1)$$

gdzie:

$$r_{avg} = \frac{1}{T} \sum_{i=1}^n d_i r_i \quad (2)$$

jest średnią ilością pracowników zatrudnionych w każdym z T dni.

$x \in R^n$, $x = (x_1, x_2, \dots, x_n)$ – wektor momentów rozpoczęcia wykonywania zadań;

$x_i \in [a_i, b_i]$

a_i – najwcześniejszy moment rozpoczęcia wykonywania zadania i

b_i – najpóźniejszy moment rozpoczęcia wykonywania zadania i

T – horyzont czasowy

$q_j(x)$ – ilość osób zatrudnionych w dniu j , $j = 1, 2, \dots, T$,

d_i – czas trwania procesu i

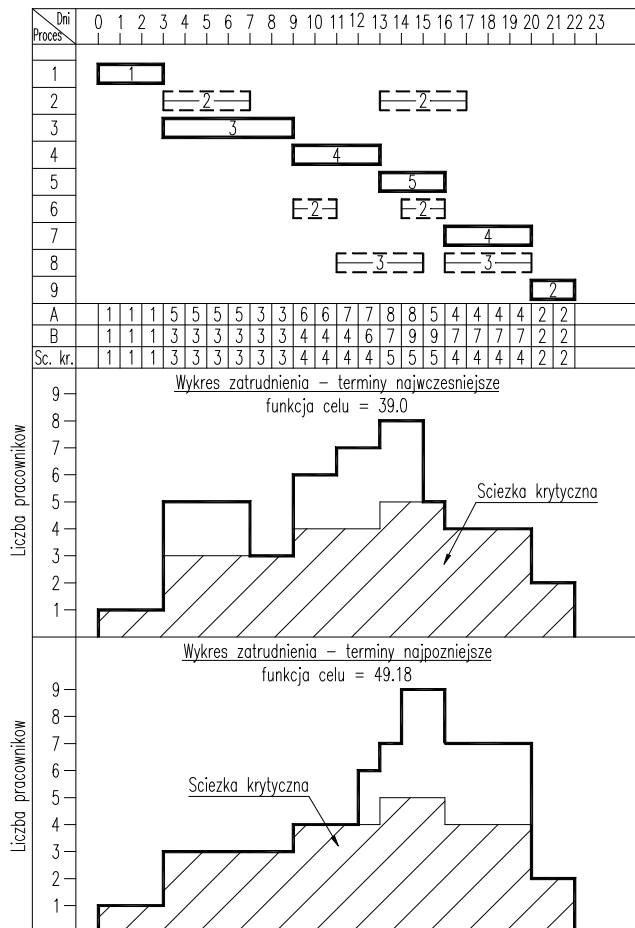
r_i – ilość zatrudnionych pracowników do wykonania procesu i

Przeprowadzenie obliczeń optymalizacyjnych wymaga wyznaczenia dla każdej iteracji wartości funkcji celu, która może być również obliczona jako średnie procentowe odchylenie dziennej ilości zatrudnionych pracowników od wartości średniego zatrudnienia w ciągu wszystkich T dni, zatem: $f(x) \in [0, 1]$, dla standaryzowanej funkcji celu.

Tab. 1. Dane dotyczące procesów 1-9

Numer procesu	Czas trwania procesu [dni]	Najwcześniejszy termin rozpoczęcia	Najwcześniejszy termin zakończenia	Najpóźniejszy termin rozpoczęcia	Najpóźniejszy termin zakończenia	Ilość pracowników niezbędnych do wykonania zadania
1	3	0	3	0	3	1
2	4	3	7	12	16	2
3	6	3	9	3	9	3
4	4	9	13	9	13	4
5	3	13	16	13	16	5
6	2	9	11	14	16	2
7	4	16	20	16	20	4
8	4	11	15	16	20	3
9	2	20	22	20	22	2

Przykładowe dane dla rozważanego zagadnienia wraz z pokazaniem poziomu zatrudnienia (przy najwcześniejszych i najpóźniejszych terminach rozpoczęcia wykonywania zadań) pokazano w Tabeli 1. oraz na Rysunku 1.



Rys. 1. Wykres Gantta dla najwcześniejszych i najpóźniejszych terminów z wartościami funkcji celu, odpowiednio: $f=39,0$ i $f=49,18$; dane pochodzą z Tabeli 1

3. Hybrydowy algorytm poszukiwania z zabronieniami

Zaproponowany w pracy algorytm składa się z dwóch modułów:

- moduł odpowiedzialny za szeregowanie zadań,
- moduł odpowiedzialny za optymalizację poziomu zatrudnienia.

Moduł szeregowania zadań jest oparty na metodzie poszukiwań z zabronieniami, a moduł optymalizujący poziom zatrudnienia oparty jest na metodzie poszukiwań losowych. w każdym kroku algorytmu generowane jest otoczenie dla bazowego uszeregowania zadań, następnie wybierane jest uszeregowanie dla którego wartość C_{max} jest najmniejsza. Dla wybranego rozwiązania uruchamiany jest algorytm odpowiedzialny za optymalizację poziomu zatrudnienia. Jako rozwiązanie bazowe w następnej iteracji wybierane jest rozwiązanie z otoczenia o najmniejszej wartości C_{max} . Poniżej zostały przedstawione kolejne kroki algorytmu:

Krok 0. Wyznacz rozwiązanie startowe π^0 ;

$$\pi = \pi^0;$$

$$\pi^* = \pi;$$

Krok 1. Wyznacz sąsiedztwo $N(\pi)$ permutacji π , nie zawierające elementów zabronionych przez listę T;

Krok 2. Znaleźć permutację $\sigma \in N(\pi)$, dla której

$$C_{max} = \min \{C_{max}(\beta) : \beta \in N(\pi)\};$$

Krok 3. Dla permutacji σ uruchomić procedurę optymalizacji poziomu zatrudnienia;

Krok 4. Jeśli $C_{max}(\sigma) < C_{max}(\pi^*)$, to $\pi^* = \sigma$;

Umieść atrybuty σ na liście T;

$$\pi = \sigma;$$

Krok 5. Jeśli spełniony jest warunek zakończenia to STOP.

W przeciwnym przypadku idź do kroku 1.

4. Eksperymenty obliczeniowe

Algorytm został zaimplementowany w języku C++ i uruchomiony na serwerze Hewlett-Packard z procesorem Intel Core 2 Duo 3,16GHz. Eksperymenty obliczeniowe zostały przeprowadzone dla instancji testowych o różnych rozmiarach: 10×7 , 10×11 , 10×16 , 20×7 , 20×11 , 20×16 . Jako rozwiązania referencyjne zostały przyjęte rozwiązania dla permutacji naturalnej. w tabeli 1 i 2 poszczególne kolumny oznaczają:

- NEH – procentowy błąd względny algorytmu NEH względem rozwiązań referencyjnych,
- TS (C_{max}) - procentowy błąd względny algorytmu NEH względem rozwiązań referencyjnych dla najlepszej wartości C_{max} ,
- TS (Koszt) - procentowy błąd względny algorytmu NEH względem rozwiązań referencyjnych dla najlepszej wartości kosztu.

Tab. 2. Procentowy błąd względny zaproponowanego algorytmu dla C_{max}

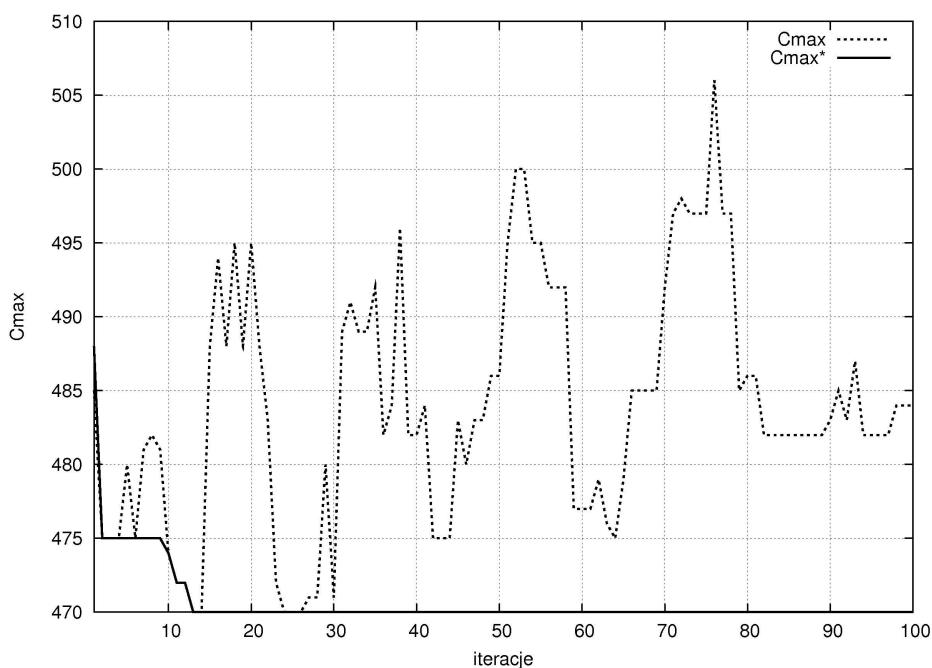
nxm	NEH	TS (C_{max})	TS (Koszt)
10x7	-16,0194	-21,1068	-19,4709
10x11	-8,19827	-13,4704	-11,5646
10x16	-12,2765	-16,331	-15,1352
20x7	-16,1756	-20,4049	-19,9597
20x11	-15,3085	-19,917	-19,5984
20x16	-14,7666	-20,0193	-20,0193
Średnia	-13,7908	-18,5416	-17,6247

W tabeli 2 zawarte zostały wartości błędu względnego względem rozwiązań referencyjnych dla kryterium C_{max} . Różnica między wartością tego błędu dla najlepszej wartości kryterium C_{max} , a wartościami dla najlepszego kosztu jest niewielka. W tabeli 3 zostały przedstawione wartości błędu względnego dla kryterium kosztu. Podobnie jak w tabeli 2 występuje różnica między wartościami dla najlepszego kosztu i najlepszych wartości C_{max} . Dla większych rozmiarów problemu (20×16) wartości błędów względnych są sobie równe.

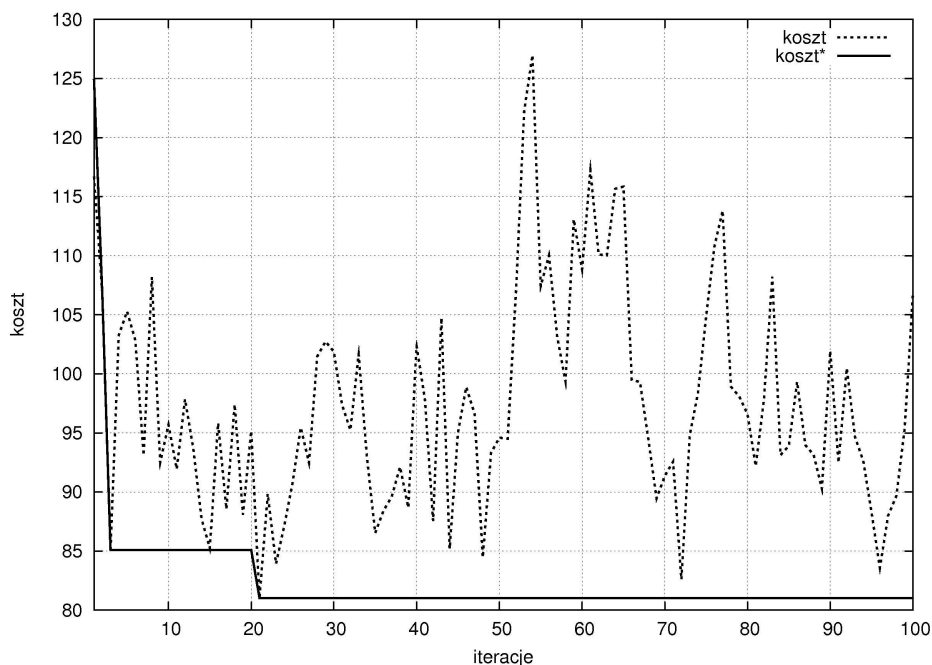
Tab. 3. Procentowy błąd względny zaproponowanego algorytmu dla kosztu

Nxm	NEH	TS (C_{max})	TS (Koszt)
10x7	-18,9503	-30,264368	-42,4768
10x11	-14,6481	-22,189902	-35,1655
10x16	-15,4126	-22,426648	-27,237
20x7	-24,6287	-34,136267	-42,3067
20x11	-18,0113	-28,832973	-33,4277
20x16	-14,7666	-20,019334	-20,0193
Średnia	-17,7363	-26,311582	-33,4388

Na rysunku 2 i 3 zostały przedstawiony przykładowy przebieg zaproponowanego w pracy algorytmu. Przedstawione wykresy pozwalają stwierdzić, że kryterium C_{max} i kryterium kosztu dla rozpatrywanego problemu nie są przeciwstawne. z przeprowadzonych eksperymentów również wynika, że najlepszej wartości kryterium C_{max} nie odpowiada najlepsza wartość kosztu. Przeprowadzenie badań numerycznych dla kryterium kosztu pozwoliłoby uzyskać pełniejszy obraz zależności kryterium C_{max} i kosztu.



Rys. 2. Bieżąca i najlepsza wartość C_{max}



Rys. 3. Bieżąca i najlepsza wartość kosztu

Przeprowadzone eksperymenty obliczeniowe pokazały, że czas działania algorytmu jest duży, ze względu na metodę użytą do optymalizacji poziomu zatrudnienia. Zastosowanie szybkiego algorytmu konstrukcyjnego lub wykorzystanie specyficznych własności problemu (własności blokowe) pozwoliłoby na znaczne skrócenie czasu działania algorytmu. Innym podejściem pozwalającym na skrócenie czasu obliczeń jest implementacja zaproponowanego w pracy algorytmu w środowisku obliczeń równoległych (klaster, GPU).

5. Przykład obliczeniowy

Wykonawca na zlecenie inwestora ma zrealizować przedsięwzięcie budowlane składające się $n = 12$ budynków mieszkalnych znajdujących się w obrębie jednego osiedla. Każdy z budynków wymaga wykonania $m = 11$ robót w ustalonej poniżej sekwencji, która wynika z przyjętej technologii:

1. roboty przygotowawcze, usunięcie humusu, roboty ziemne,
2. wykonanie ław i ścian fundamentowych wraz z izolacją,
3. wykonanie ścian, kominów, stropów budynku,
4. wykonanie więźby dachowej z pokryciem i odwodnieniem,
5. wykonanie stolarki okiennej i drzwiowej,
6. wykonanie instalacji wewnętrznych w budynku,
7. wykonanie elewacji, tynków wewnętrznych, posadzek oraz zabudowy poddasza,
8. wykonanie ogrodzenia, podjazdu i chodnika,
9. ułożenie glazury, malowanie,

10. wykonanie podłóg,
11. biały montaż.

Wykonawca dysponuje grupami roboczymi (brygadami) wyspecjalizowanymi do wykonywania jednego rodzaju robót i prowadzącymi roboty w każdym z budynków ze stałą wydajnością. Aktualnie dysponuje on po jednej grupie (brygadzie) dla danego rodzaju robót. Na podstawie pracochłonności robót w poszczególnych budynkach oraz liczby pracowników i wydajności grup roboczych wykonawcy ustalono czasy trwania realizacji robót w obiektach, które podano w tabeli 4. Celem wykonawcy tego przedsięwzięcia będzie jak najszybsze jego zrealizowanie zakładając, że ma on możliwość przekazywania w dowolnej kolejności budynków inwestorowi. Dodatkowym celem jest prowadzenie robót przedsięwzięcia tak, aby w największym stopniu wyeliminować skoki liczby pracowników na placu budowy, które powodują wzrost ogólnych kosztów realizacji przedsięwzięcia. Tak więc zadaniem planisty wykonawcy będzie ustalenie harmonogramu przebiegu robót o najkrótszym czasie trwania oraz o możliwie najmniejszych wahanach liczby pracowników na placu budowy.

Tab. 4. Dane do przykładu obliczeniowego

Numer roboty $k =$	Liczba pracowników w grupie roboczej	Obiekty $j =$											
		1	2	3	4	5	6	7	8	9	10	11	12
1	4	17	24	12	16	27	21	18	12	12	16	8	8
2	4	28	45	37	21	23	45	21	16	33	41	41	40
3	6	55	35	55	52	29	40	47	75	84	38	79	66
4	4	24	18	35	12	12	33	25	34	15	24	33	25
5	3	20	32	27	24	24	22	26	19	21	30	20	9
6	4	10	8	4	6	3	5	5	18	11	5	5	8
7	9	53	75	77	90	89	66	35	58	42	76	60	61
8	5	25	34	29	27	11	15	37	27	17	39	13	39
9	5	20	23	31	9	26	31	20	27	30	23	13	23
10	6	11	14	5	5	14	11	7	7	9	14	16	14
11	3	9	3	6	8	11	13	6	9	11	3	15	4

Dla powyższego przykładu zaproponowany algorytm pozwala na uzyskanie następujących wyników: C_{\max} - 949, koszt - 239,058. Zaproponowany algorytm pozwala na poprawę wartości C_{\max} o 6,59 % oraz wartości kosztu o 13,52% w stosunku do rozwiązania referencyjnego stanowiącego permutację naturalną.

6. Wnioski

Kierunkiem dalszych badań może być przeprowadzenie eksperymentów dla całkowitego kosztu projektu budowlanego (tj. energii wbudowanej) jako kryterium optymalizacyjnego. Innym kierunkiem badań może być implementacja algorytmu równoległego oraz algorytmu wykorzystującego specyficzne własności problemu (własności blokowe) co pozwoliłoby na znaczne skrócenie czasu obliczeń.

Literatura

1. Al-Tabtabai H., Alex. A.: Using genetic algorithms to solve optimization problems in construction, *Engineering, Construction and Architectural Management* 1999/2, 121-132.
2. Bożejko W., Hejducki Z., Wodecki M.: Harmonogramowanie kompleksu robót drogowych w warunkach niepewności, *Komputerowo Zintegrowane Zarządzanie* (red. R. Knosala), Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Tom I, Opole 2008, ISBN 978-83-923797-3-7, 109-118.
3. Bożejko W., Hejducki Z., Podolski M., Uchroński M.: Harmonogramowanie przedsięwzięć budowlanych za pomocą problemu przepływowego ze specyficznymi czasami transportu, *Automatyka, Zeszyty Naukowe Politechniki Śląskiej, Gliwice* 2008, z. 151, 93-102.
4. Chan W.T., Chua D.K.H., Kannan G.: Construction Resource Scheduling with Genetic Algorithms, *Journal of Construction Engineering and Management*, czerwiec 1996, 125-132.
5. Rogalska M., Bożejko W., Hejducki Z.: Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling, *Automation in Construction* 18, Elsevier, 2008, 24-31.
6. Rogalska M., Bożejko W., Hejducki Z., Wodecki M.: Harmonogramowanie robót budowlanych z zastosowaniem algorytmu Tabu Search z rozmytymi czasami wykonania zadań, *Przegląd Budowlany* Nr 7-8, 76-80.

Dr Wojciech BOŻEJKO
Mgr inż. Mariusz UCHROŃSKI
Instytut Informatyki, Automatyki i Robotyki
Politechnika Wrocławska
e-mail: wojciech.bozejko@pwr.wroc.pl
mariusz.uchronski@pwr.wroc.pl

Dr hab. inż. Zdzisław HEJDUCKI
Dr inż. Michał PODOLSKI
Instytut Budownictwa
Politechnika Wrocławska
e-mail: zdzislaw.hejducki@pwr.wroc.pl
michal.podolski@pwr.wroc.pl