

# HARMONOGRAMOWANIE Z OGRANICZENIAM PROJEKTÓW WSPÓLBIEŻNYCH

Bożena MARCIŃCZYK, Bożena SKOŁUD

**Streszczenie:** W artykule przedstawiono zastosowanie meta heurystycznej metody – algorytmu mrówkowego w harmonogramowaniu projektu przy nałożonych ograniczeniach czasowych i zasobowych. Zastosowanie nowoczesnych meta heurystyk pozwala na skonstruowanie efektywnych czasowo harmonogramów. Wyniki przeprowadzonego eksperymentu przedstawiono w autorskim programie.

**Słowa kluczowe:** projekt, harmonogramowanie, ograniczenia zasobowe, algorytm mrówkowy.

## 1. Wprowadzenie

Wymagania rynku i klienta są powodem dużego zróżnicowania produktów i ich indywidualizowania, co jest szczególnie zauważalne w małych i średnich przedsiębiorstwach (MŚP). Większość przedsiębiorstw tego sektora nastawiona jest na produkcję wielowersyjną lub wieloasortymentową, a w wielu przypadkach nastawiona jest na produkcję niepowtarzalną, typu projekt. Obecnie konsumenci coraz częściej zmieniają dobra konsumpcyjne (samochody, telewizory, radia, pralki itp.) nawet jeśli nie straciły one swoich użytecznych funkcji, co wpłynęło na skrócenie cyklu życia produktów. Rosnące wymagania klientów z jednej strony i coraz to większa konkurencyjność na rynku producentów z drugiej strony, wymuszają wprowadzanie zmian w zarządzaniu przedsiębiorstwem. W wielu przypadkach zlecenia produkcyjne szczególnie te dotyczące złożonych zadań traktowane są jako nowe i niepowtarzalne, a sposób zarządzania nimi jest podobny do przypadku zarządzania projektami [1]. W rzeczywistych systemach projekty zazwyczaj są realizowane współbieżnie, to znaczy, że przystępuje się do realizacji zadań nowego projektu, podczas gdy wcześniejsze projekty nie zostały jeszcze zakończone. Dodatkowo duża złożoność projektu powoduje, że szczególnie trudnym zagadnieniem staje się szeregowanie wejściowe projektów oraz szeregowanie zadań na poszczególnych zasobach przy jednoczesnym zachowaniu relacji następstwa wymaganej przy ich wykonywaniu. Harmonogramowanie w środowisku wieloprojektowym z wykorzystaniem wybranej heurystyki (algorytmu mrówkowego) jest przedmiotem niniejszego artykułu.

## 2. Zadanie typu projekt

Zadanie typu projekt zostało zdefiniowane na wiele sposobów. Przyjmujemy za Leach'em [2], że projekt to specyficzne działanie ze zdefiniowanym terminem rozpoczęcia i zakończenia. Realizacja projektu często angażuje wiele zasobów w jednym czasie i jest zadaniem długotrwałym. Projekty najczęściej przekraczają wyznaczony termin realizacji, zakres oraz budżet, dlatego też te czynniki są kluczowym punktem w planowaniu projektu [2].

Zarządzanie w tym kontekście jest zestawem działań obejmującym planowanie i podejmowanie decyzji, organizowanie, przewodzenie, tj. kierowanie ludźmi oraz kontrolowanie działań skierowanych na projekty i wykonywanych z zamiarem osiągnięcia celów organizacji w sposób sprawny i skuteczny [3].

Planowanie w organizacji zorientowanej projektowo można postrzegać jako:

- planowanie projektu – dotyczy pojedynczego projektu, obejmuje planowanie terminowe, kosztowe, jakościowe i zasobów koniecznych do realizacji projektu,
- planowanie wieloprojektowe – dotyczy zestawu projektów, obejmuje planowanie portfela projektów dla optymalnego wykorzystania zasobów organizacji oraz zrealizowania celów strategicznych i operacyjnych firmy (analogicznie do portfela zleceń produkcyjnych).

Niezależnie od wyboru podejścia projekty zajmują ograniczone zasoby firmy często na dłuższy okres, dlatego decyzja o ich realizacji jest decyzją o zaangażowaniu firmy na określonym rynku i w określone technologie.

Wśród zasobów można wyróżnić: zasoby rzeczowe, finansowe, ludzkie oraz informacyjne. Należy pamiętać, że zasoby są ograniczone. W związku z tym, w przypadku gdy zasoby wewnętrzne okazują się być niewystarczające, przedsiębiorstwa często korzystają z zasobów zewnętrznych [4], [5]. Podczas określania celów projektu organizacja powinna z jednej strony skupiać się na wykonaniu projektu w terminie, zgodnie z założonym budżetem, zakresem oraz wymaganiami jakościowymi wyznaczonymi przez klienta, z drugiej zaś powinna dążyć do efektywnego wykorzystania swoich zasobów, ograniczając marnotrawstwo i liczbę poprawek w projekcie.

Szczególnym rodzajem zasobu jest czas, którego nie można zmagazynować, ponieważ jest on zużywany bez względu na to, na co jest przeznaczony. Dlatego też większość wysiłków skierowana jest na zakończenie projektu w terminie. W tym celu tworzony jest harmonogram projektu, który określa, kiedy powinny rozpocząć się i zakończyć poszczególne etapy projektu [5,6,7].

### **3. Sformułowanie problemu**

Konkurencyjność przedsiębiorstwa zależy od jego zdolności do szybkiego podejmowania trafnych decyzji związanych z bilansowaniem potrzeb klienta i możliwości producenta.

Metody sieciowe, takie jak: CPM, PERT, MPM, GERT, nie uwzględniają ograniczeń związanych z czasowym dostępem do zasobów, czyli nie pozwalają na stwierdzenie, czy realizacja danego przedsięwzięcia nie przekroczy zadanego limitu dostępnych zasobów w zadanym horyzoncie czasu. Uzyskiwane rozwiązania odpowiadają uśrednionym, w danym horyzoncie czasu, bilansom zasobów i nie gwarantują, że w każdym momencie realizacji przedsięwzięcia, liczba zasobów niezbędnych do jego wykonania będzie wystarczająca, tym samym nie są w stanie przewidzieć przypadków związanych np. z powstaniem zakłóceń i koniecznością uwzględnienia w planie nowego zadania, rezygnacji z innego, zmian terminów rozpoczęcia zadań itp.

Przekonanie, że wszystko co jest potrzebne przy planowaniu projektu, to lista zadań do wykonania, ich sklasyfikowanie, zgromadzenie personelu do realizacji poszczególnych zadań i rozpoczęcie pracy łatwo może doprowadzić do sytuacji kryzysowej, szczególnie wtedy, gdy dotrzymanie terminów jest niemożliwe. Konieczna jest więc weryfikacja przyjętego czasu trwania każdego z zadań oraz określenie wzajemnych uzależnień i rozwiązywanie konfliktów zasobowych. Dopiero po tym można precyzyjnie prześledzić

logiczną sekwencję wykonywania zadań [10,11].

Harmonogramowanie projektu w warunkach ograniczeń zasobowych, (ang.: Resources Constrained Project Scheduling Problem - RCPSP), to klasyczny problem harmonogramowania. Na czynności nałożone są ograniczenia kolejnościowe wynikające z marszruty technologicznej oraz ograniczenia zasobowe. Główny cel w harmonogramowaniu projektu w warunkach ograniczeń zasobowych to minimalizacja długości uszeregowania (ang.: makespan) [10,11].

Podstawowym celem zarządzania projektami jest rozstrzygnięcie problemów polegających na udzieleniu odpowiedzi na pytanie, czy przy istniejących ograniczeniach zasobowych danego systemu wytwórczego i przy znanym rozkładzie mocy produkcyjnych w określonym horyzoncie czasu nowy projekt może być przyjęty do realizacji zgodnie z oczekiwaniami odbiorcy, w zadanej ilości i w określonym terminie przy jednoczesnym zachowaniu ceny produktu wynikającej z warunków rynku?

Problem rozważany w niniejszej pracy można sformułować w następujący sposób: Dany jest zbiór projektów przewidzianych do współbieżnej realizacji na zasobach o ograniczonej zdolnościach produkcyjnych i w ograniczonym czasie.

Zakłada się, że:

- Dostępność zasobów jest znana i opisana macierzą zdolności produkcyjnych dla każdego okresu planistycznego.
- Zasoby są zasobami odnawialnymi w kolejnych okresach planistycznych.
- Dany jest zbiór procesów, w których skład wchodzi określona liczba zadań.
- Kolejność wykonywanych czynności jest z góry znana i określona za pomocą relacji następstw.
- Każde zadanie może być uruchomione w chwili czasu 0 i wszystkie zadania są znane w momencie uruchomienia.
- Każde zadanie realizowane jest od momentu rozpoczęcia bez przerwy, aż do momentu zakończenia zadania.
- Czasy realizacji zadań na poszczególnych komórkach są z góry określone.
- Pomiędzy zadaniami procesów występują zależności typu zakończenie – rozpoczęcie ZR, w którym następnik może rozpocząć się bezzwłocznie po zakończeniu zadania poprzedzającego.

Czy przy takich założeniach możliwe jest utworzenie harmonogramu dopuszczalnego uwzględniającego te ograniczenia zasobowe i czasowe?

Złożoność problemów harmonogramowania sprawia, że przy wykorzystaniu algorytmów dokładnych liczba iteracji rośnie wykładniczo wraz ze wzrostem liczby zadań. Dlatego też proponuje się metody, które pozwolą na znalezienie zadowalającego rozwiązania w żądanym przedziale czasu i przy danych ograniczeniach zasobowych. Grupą metod, które dają możliwość sukcesywnego poprawiania istniejącego rozwiązania są metody heurystyczne. Metody te umożliwiają rozwiązywanie trudnych kombinatorycznych zagadnień m.in. takich jak harmonogramowanie i alokacja zasobów. Zastosowanie tych metod skraca czas rozwiązywania problemu wraz ze wzrostem liczby ograniczeń.

Do metod heurystycznych należą algorytm rojowe. Inspiracją do ich powstania były obserwacje zachowań kluczy ptaków, ławic ryb i rojów owadów. Ptaki lecące w stadzie potrafią całym stadem zmienić kierunek ruchu, przegrupowując się, skupiając lub rozpraszając. Algorytm mrówkowy (należący do grupy algorytmów rojowych), wzoruje się na zachowaniu kolonii żywych mrówek, które znajdują najkrótszą drogę między źródłem pożywienia a mrowiskiem [12,13].

Z tego też powodu do harmonogramowania projektów wybrano algorytm optymalizacji

mrowiskowej, który dostarczy rozwiązania przybliżonego, umożliwi rozdział czynności na ograniczonych zasobach w ograniczonym przedziale czasu.

#### 4. Algorytm optymalizacji mrowiskowej

Inteligentne zachowanie osobników roju wynika ze współdziałania wielu prostych osobników. Istnieje wiele przykładów organizmów żywych, w których pojedyncze osobniki są nieświadome całości zadania, w jakim uczestniczą, natomiast ich liczność oraz charakterystyczne zachowanie prowadzi do wykonania tego zadania [12].

Algorytm mrówkowy inspirowany jest naturalnym zachowaniem mrówek. Sztuczne mrówki, tak samo jak prawdziwe w kolonii, współpracują ze sobą w trakcie poszukiwania optymalnych rozwiązań. Elementy łączące sztuczne i prawdziwe mrówki to [14,15]:

- ślad feromonowy i siła jego oddziaływania,
- wyszukiwanie najkrótszej ścieżki do pożywienia,
- przypadkowe przemieszczenia indywidualnych mrówek w stanie początkowym wyszukiwania ścieżki.

W zaproponowanym algorytmie mrówkowym zastosowano globalną regułę wzmocnienia śladu feromonowego – po każdej zakończonej iteracji algorytmu poziom feromonu jest wzmocniony, zgodnie z zasadą [16]:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t, t+n) \quad (1)$$

gdzie:  $\tau_{ij}(t)$  – wielkość śladu feromonowego,

$\rho$  – współczynnik parowania feromonu,  $(1-\rho)$  to poświata feromonu; przyjmuje wartość z przedziału  $<0,1>$ ,

$(i,j)$  – krawędź najlepszej z tras znalezionej w jednej iteracji,

$m$  – liczba mrówek, które przeszły przez  $(i,j)$  pomiędzy chwilami  $(t)$  i  $(t+n)$ ,

gdzie  $n$  – liczba cykli, składających się na jedną iterację algorytmu,

$$\Delta\tau_{ij}(t, t+n) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+n),$$

$$\Delta\tau_{ij}^k(t, t+n) = \begin{cases} Q/L_k & \text{jeżeli mrówka } k \text{ wybierze krawędź } (i, j) \\ 0 & \text{w przeciwnym wypadku} \end{cases}$$

$Q$  – stała wartość feromonu,

$L_k$  – długość trasy  $k$ -tej mrówki.

Za [16] przyjmuje się, że algorytm osiąga najlepsze efekty gdy jest liczba mrówek wprowadzana do systemu ( $m$ ) na podobną wartość do liczby węzłów ( $w$ ).

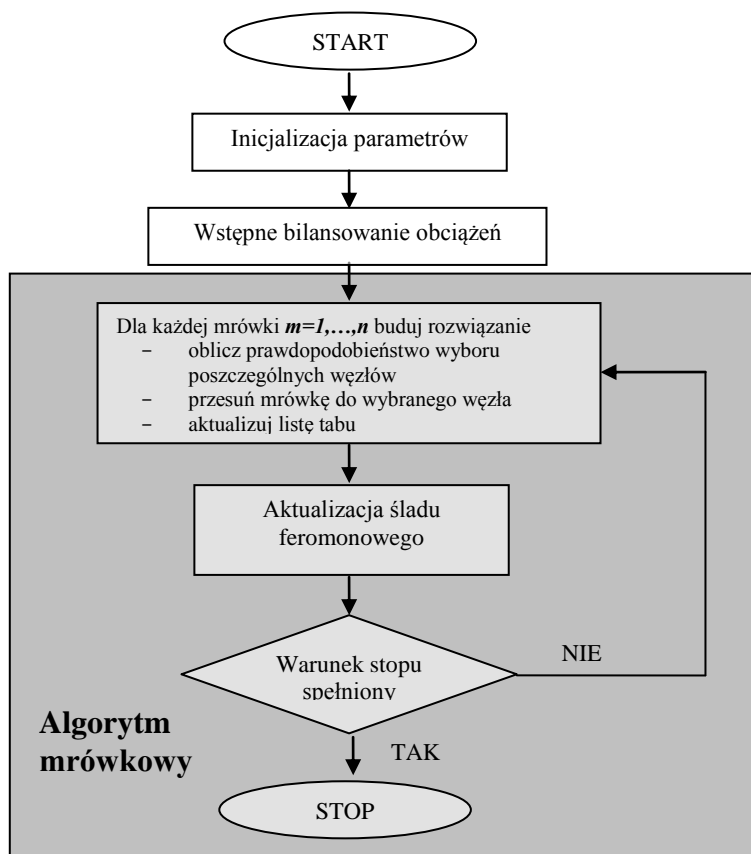
Przyjęto też, że stała wartość feromonu  $Q=1$ , przeprowadzone badania wykazały niewielki wpływ tego parametru na globalne wyniki algorytmu [16]. Parametry pozwalające użytkownikowi sterować względną ważnością intensywności śladu i jego widoczności to odpowiednio parametry  $\alpha$  i  $\beta$ . Jeżeli wpływ wartości feromonowej ( $\alpha$ ) równy jest 0, wówczas brana jest pod uwagę tylko wartość heurystyczna ( $\beta$ ); dla  $\beta = 0$  wpływ ma tylko wartość feromonu na trasie [17].

Tab.1. Różnice pomiędzy zachowaniem prawdziwych i sztucznych mrówek [16]

PRAWDZIWE MRÓWKI	SZTUCZNE MRÓWKI
Poruszają się w sposób asynchroniczny	Poruszają się w sposób synchroniczny
Zostawiają feromon na gruncie gdziekolwiek się poruszają	Osadzają sztuczny feromon na ich drodze powrotnej do mrowiska

Zachowanie plądrowania oparte na domniemanej ocenie rozwiązania; krótsze ścieżki będą ukończone wcześniej niż dłuższe	Oceniając rozwiązania mają wzgląd na jakościową miarę, używaną do rozstrzygnięcia siły feromonu
---	---

Algorytm mrówkowy zostaje uruchomiony dopiero po wstępnej ocenie dostępności zasobów w czasie, w którym zadania powinny być wykonane (rys.1). W algorytmie mrówkowym pojedyncza mrówka generuje swoje rozwiązanie niezależnie od pozostałych osobników. Sztuczna mrówka wyrusza z losowo wybranego miejsca, które umieszczane jest jako pierwsze na liście odwiedzonych miejsc. Następnie w sposób losowy, zgodnie z prawdopodobieństwem wyboru, wybiera kolejną pozycję, umieszczając każde odwiedzone stanowisko na liście tabu – lista zakazanych miejsc (zabezpieczenie przed wyborem tego samego miejsca). Po każdej iteracji algorytmu następuje uaktualnienie śladu feromonowego zgodnie z (1). Warunek stopu kończy działanie algorytmu, jeśli warunek nie zostanie spełniony rozpoczyna się kolejna iteracja algorytmu (rys. 1).



Rys. 1. Schemat działania algorytmu

## 5. Algorytm mrówkowy w harmonogramowaniu zadań typu projekt

W niniejszym rozdziale przedstawiono implementację bilansowania zadań i zdolności produkcyjnych i algorytmu mrówkowego w celu odpowiedzi na pytanie dotyczące możliwości przyjęcia projektu do realizacji oraz odpowiedzi na pytanie dotyczące końcowego terminu realizacji projektu. Podejście to zostało zaimplementowane w autorskim programie.

Przeprowadzenie procedury harmonogramowania projektów wymaga specyfikacji danych wejściowych, dotyczących zarówno parametrów środowiska wieloprojektowego, jak i parametrów wynikających z wybranej metody meta heurystycznej.

Specyfikacja parametrów dotycząca środowiska wieloprojektowego określa wymagania w zakresie ograniczeń czasowych, zasobowych oraz kolejnościowych, wynikających z zadanej marszruty technologicznej dla implementowanych do systemu projektów.

Specyfikacja parametrów dotycząca wybranej metody meta heurystycznej określa dobór parametrów początkowych w zaproponowanej metodyce opartej na zastosowaniu algorytmu mrówkowego do harmonogramowania zadań w środowisku wieloprojektowym. Opracowany program pozwala Użytkownikowi systemu na wprowadzenie danych w oknie dialogowym definiowania kodu implementowanych do systemu projektów. Parametry wejściowe do systemu zostały wyznaczone doświadczalnie, w sposób pozwalający na uzyskanie wyników zbieżnych dla jak największej liczby algorytmów.

W artykule przedstawiono autorski program komputerowy, który wykorzystuje mechanizm działania algorytmu mrówkowego. Program komputerowy utworzony na potrzeby realizacji niniejszych badań obejmuje swoją funkcjonalnością: bilansowanie obciążeń z dysponowanymi zdolnościami produkcyjnymi oraz harmonogramowanie w środowisku wieloprojektowym przy nałożonych ograniczeniach czasowych, zasobowych i kolejnościowych z wykorzystaniem algorytmu mrówkowego. Aplikacja została zaprojektowana i napisana w środowisku w Borland C++ Builder 6/2007. Aplikacja komputerowa składa się z kilku modułów ułatwiających jego obsługę:

- moduł bilansowania,
- moduł wizualnej edycji procesów,
- moduł wizualnej reprezentacji harmonogramu.

Główne cechy programu to:

- Możliwość równoczesnej pracy nad dowolną liczbą projektów.
- Możliwość definiowania nowego projektu dzięki wbudowanemu kreatorowi.
- Każdemu projektowi można przydzielić dowolną liczbę komórek produkcyjnych, procesów i rozpatrywanych okresów planistycznych.
- Możliwość wstępnego bilansowania odpowiadającego na pytanie czy zakończenie zbioru zadań w danym okresie planistycznym jest w ogóle możliwe.
- Edycja parametrów algorytmu mrówkowego użytego do harmonogramowania
- Wizualna reprezentacja macierzy bilansu.
- Wizualna reprezentacja i edycja przebiegu procesu, a także możliwość ich zapisu do plików \*.hrc.
- Wizualna reprezentacja rozwiązania harmonogramowania.
- Możliwość eksportu wynikowego raportu rozwiązania do pliku (\*.txt, \*.html).
- Możliwość zapisu i odczytu projektu do formatu plików stworzonych na potrzeby aplikacji (\*.aco).

Za pomocą wbudowanego kreatora definiuje się dowolne środowisko wieloprojektowe. Jak już wspomniano pierwszym etapem tworzenia harmonogramu jest przeprowadzenie

bilansu, który pozwala na wstępne oszacowanie zdolności produkcyjnych.

Wykonanie projektów w środowisku wieloprojektowym wymaga jednoczesnego wykorzystania różnych zasobów, np. rzeczowych, finansowych oraz ludzkich i jest uwarunkowane dostępem do tych ograniczonych zasobów w czasie, gdyż w danym horyzoncie planistycznym część zasobów jest zaangażowana w realizację różnych przedsięwzięć. [8].

Przed przystąpieniem do sporządzenia harmonogramu, należałoby ocenić, czy dysponowane zasoby w ogóle dają szansę na wykonanie zadań w oczekiwanym czasie.

Można tego dokonać przez bilansowanie obciążeń z dysponowanymi zdolnościami produkcyjnymi (Z) i porównać na przykład tworząc macierze reprezentujące zdolności produkcyjne oraz planowane zadania dla każdego okresu planistycznego [9].

Jeśli dla każdej komórki produkcyjnej spełniony jest warunek, że jej dysponowany fundusz czasu jest większy od sumy wszystkich zadań planowanych do wykonania w niej (wliczając zadania już rozpoczęte) w danym okresie planistycznym to bilans jest spełniony.

Należy pamiętać o tym, że takie bilansowanie jest jedynie wstępnym szacowaniem i nie gwarantuje wykonania zadań w terminie. Jednym z powodów jest to, że nie są uwzględniane relacje następstwa.

W module wizualnej edycji procesu użytkownik definiuje ograniczenia zasobowe, kolejnościowe (wynikające z marszruty technologicznej) oraz czasowe. Edytor pozwala na utworzenie połączeń między zadaniami w trybie wzajemnego wykluczania i w trybie spotkaniowym, pozwalając na stworzenie dowolnej konfiguracji ograniczeń kolejnościowych.

Po zdefiniowaniu wszystkich zasobów, ograniczeń i czynności oraz okresów planistycznych, w tym procesów branych pod uwagę w danym okresie planistycznym, można przejść do modułu wizualnej reprezentacji harmonogramu. Moduł ten nie tylko pozwala na wizualizację harmonogramu, ale również na edycję ustawień algorytmu mrówkowego i wyświetlenie raportu dla najlepszego rozwiązania.

Liczba iteracji algorytmu wprowadzona do systemu jest równa liczba iteracji po jakiej algorytm musi zostać zakończony. Jeśli wartość parametru „liczba iteracji algorytmu” ustawiona jest na wartość „0” algorytm będzie działał bez końca, a zakończyć go może tylko użytkownik naciskając przycisk „Zatrzymaj”. Jeśli jednak wartość tego parametru będzie większa „0” algorytm sam zakończy pracę po osiągnięciu warunku stop.

W zaproponowanym programie możliwy jest wybór harmonogramowania:

- w przód – rozwiązaniem jest najwcześniejszy termin zakończenia
- w tył – rozwiązaniem jest najpóźniejszy termin rozpoczęcia

Kiedy algorytm zakończy swoje działanie poprzez osiągnięcie zdefiniowanej liczby iteracji, lub na życzenie użytkownika pojawią się okna dialogowe, które informują w przypadku harmonogramowania w przód o najlepszym terminie zakończenia. Dla harmonogramowania w tył w wyświetlonym komunikacie zawarta zostanie informacja o najpóźniejszym terminie rozpoczęcia robót, aby nie został przekroczony termin dopuszczalny.

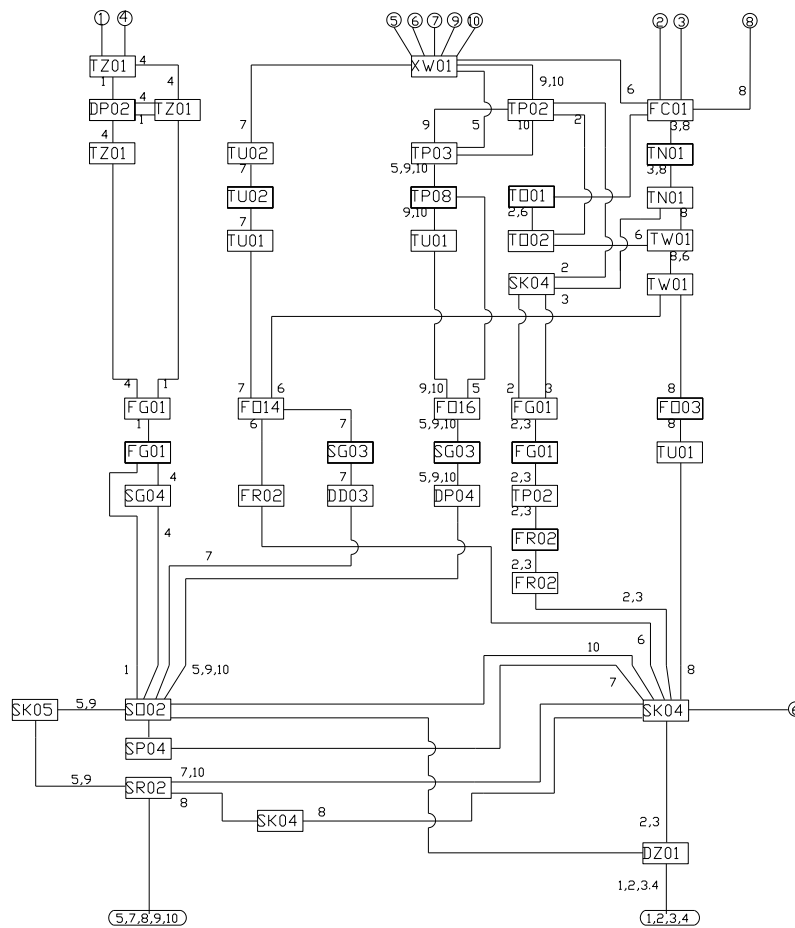
Tab. 2 Domyślne parametry algorytmu w programie (por. rozdz.5.)

Liczba mrówek	10
Alfa - współczynnik wyparowania feromonu	0,01
Beta - parametr względnej ważności pomiędzy śladem feromonowym, a odwrotnością odległości	1,0

Współczynnik eksploracja-eksploatacja – parametr $q_0$	0,5
Maksymalna wartość feromonu	10
Rodzaj rozwiązania	Global Best
Liczba iteracji algorytmu	0
Typ harmonogramowania	W przód

## 6. Eksperyment

Dany jest system pracujący w środowisku wieloprojektowym, w którym rozpatrzono 10 przedsięwzięć, operujących na 64 zasobach. Zakłada się, że na zasobach nie planowano innych zadań. Czasy jednostkowe –  $t_j$  zaczerpnięto z [18]. Schemat przepływu ilustruje rys. 2.



Rys. 2. Schemat przepływu



Na podstawie przeprowadzonych testów algorytmu warunek stop ustalono na postawie powtarzalności wyników dla rozpatrywanego problemu na poziomie:

Termin preferowany – 330

dopuszczalny – 340

maksymalny – 350

zakłada się ponadto: 500 iteracji algorytmu.

Oczekuje się znalezienia jak najlepszego harmonogramu, ze względu na przyjęte kryterium optymalizacji, którym jest minimalna długość uszeregowania wszystkich przedsięwzięć (Cmax).

Dane wejściowe do algorytmu przedstawia tab. 3

Tab. 3. Dane wejściowe do algorytmu – eksperyment

Liczba mrówek	64(por.tab.2)
Alfa	0,01
Beta	0,5
Współczynnik eksploracja-eksploatacja	0,5
Maksymalna wartość feromonu	10
Rodzaj rozwiązania	Global Best
Liczba iteracji algorytmu	500
Typ harmonogramowania	W przód

Rozwiązanie:

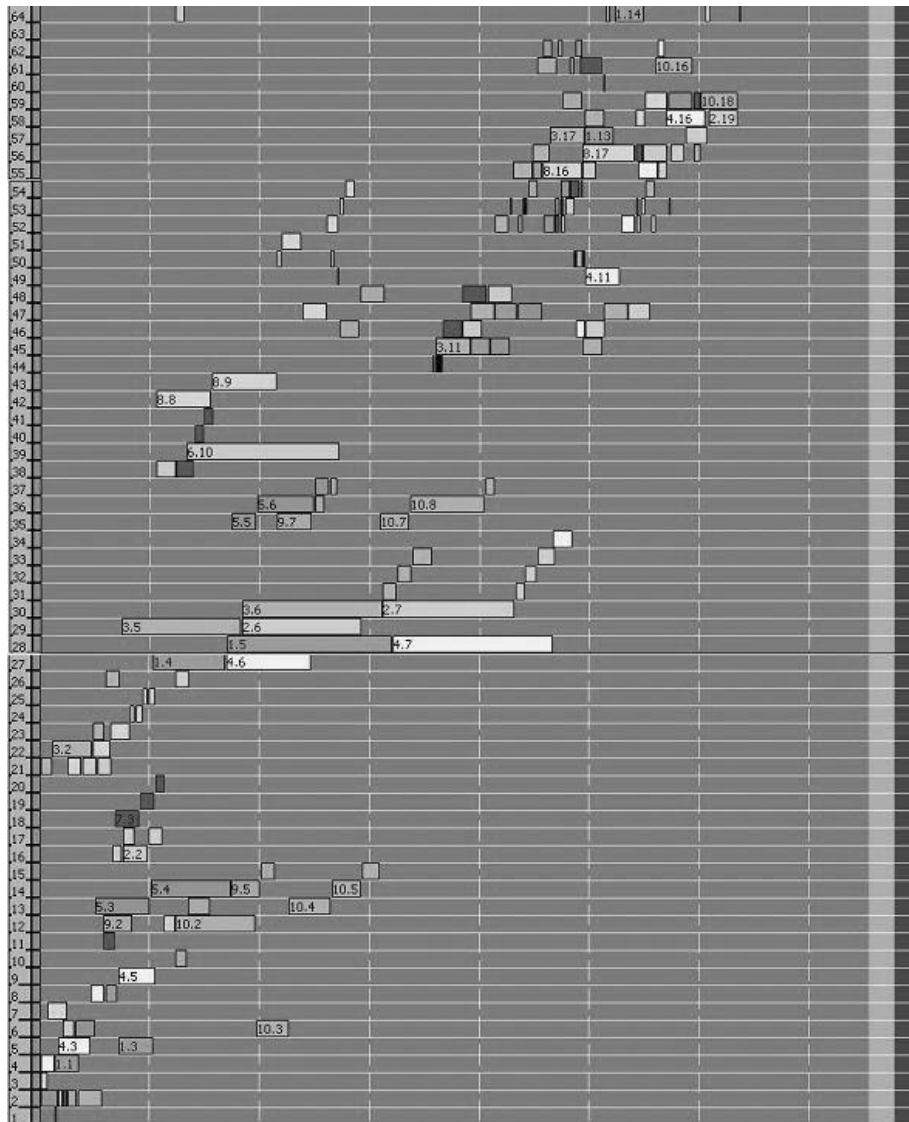
Przeprowadzono 10 eksperymentów. Po przeprowadzonym wstępnym bilansowaniu obciążeń z dysponowanymi zdolnościami produkcyjnymi porównano zdolności produkcyjne oraz planowane zadania dla każdego okresu planistycznego - dla każdej komórki produkcyjnej spełniony jest warunek, że jej dysponowany fundusz czasu jest większy od sumy wszystkich zadań planowanych do wykonania w niej. w danym okresie planistycznym. Dla każdego zasobu okres zajętości przez planowane zadania jest mniejszy niż termin preferowany.

W drugiej części przeprowadzone zostały testy algorytmu mrówkowego. Dla zdefiniowanych parametrów początkowych (por. tab.3) oraz przy nałożonych ograniczeniach zasobowych, czasowych i kolejnościowych algorytm znalazł rozwiązanie w czasie 32,2 sekundy. Najlepszy wynik uzyskano w pierwszym eksperymencie symulacyjnym - termin zakończenia w pierwszym okresie planistycznym wynosi 279 jednostek czasu dla mrówki 56. Tabela 4 przedstawia wyniki symulacji 10 eksperymentów:

Tab. 4. Testy algorytmu mrówkowego

Nr testu	Wynik	Mrówka
<b>1</b>	<b>279</b>	<b>56</b>
2	287	13
3	285	60
4	302	2
5	294	54
6	285	39
7	284	2
8	292	28
9	293	7
10	299	20

Po zakończonej pracy algorytmu wygenerowany zostaje wykres Gantta (rys.3).



Rys. 3. Wykres Gantta dla otrzymanego rozwiązania

## 7. Podsumowanie

Algorytm wzorujący się na zachowaniu kolonii mrówek – algorytm mrówkowy – jest alternatywną meta heurystyczną metodą poszukiwania optymalnych rozwiązań. Jedyną analogią pomiędzy sztucznymi a żywymi mrówkami to sposób komunikacji za pomocą śladu feromonowego – substancji zapachowej pozostawianej na trasie, która uatrakcyjnia

ścieżkę pomiędzy źródłem pożywienia a gniazdem. Im więcej feromonu znajduje się na trasie, tym częściej dana ścieżka jest wybierana przez kolejnych wędrujących agentów.

Heurystyka dana w algorytmie mrówkowym jest to minimalna wiedza potrzebna do rozwiązywania konkretnego problemu, która traci ważność w miarę wzrostu doświadczenia osiąganego przez mrówki podczas generowania kolejnych rozwiązań.

Efektywność w poszukiwaniu rozwiązań problemów należących do klasy NP zawdzięczają naśladowaniu rzeczywistych zachowań mrówek. Systemy mrowiskowe cechują się dużą dokładnością i szybkością w zawężaniu przestrzeni rozwiązań oraz wyznaczaniu rozwiązań możliwie dobrych. Przy wykorzystaniu feromonu jako środka komunikacji, system mrówkowy buduje dobry mechanizm kooperacji wewnątrz mrowiska. Na obecny etapie pracy rozwiązania generowane przez zaproponowany algorytm nie są weryfikowane z innymi metodami optymalizacji harmonogramowania projektów w środowisku wieloprojektowym, przy nałożonych ograniczeniach czasowych, zasobowych i kolejnościowych.

## Literatura

1. Skołod B.: Zarządzanie operacyjne. Produkcja w małych i średnich przedsiębiorstwach. Wydawnictwo Politechniki Śląskiej, 2007, ss. 238.
2. Leach L.P.J.: Critical Chain Project Management. Artech Hous, Norwood, 2000.
3. Griffin R.W.: Podstawy zarządzania organizacjami. Wydawnictwo Naukowe PWN, Warszawa 1996.
4. Pawlak M.: Zarządzanie projektami. Ekonomika i Organizacja Przedsiębiorstwa, Warszawa 2001.
5. [www.strategie.info.pl](http://www.strategie.info.pl).
6. Goldratt E. M.: Łańcuch Krytyczny. Werbel, Warszawa 2000.
7. Milian Z.: Łańcuch krytyczny w budownictwie. Czasopismo Techniczne Budownictwo, Wydawnictwo Politechniki Krakowskiej, Kraków 2004.
8. Rand G.K., Critical Chain: the theory of constraints applied to project management. International Journal of Project Management 18, 2000.
9. Wróblewski K. J.: Podstawy sterowania przepływem produkcji. Warszawa: WNT, 1993.
10. Kosturbić A.: Harmonogramowanie realizacji projektów – przegląd modeli, w [www.zie.pg.gda.pl/koipsp/4adamkosturbić.pdf](http://www.zie.pg.gda.pl/koipsp/4adamkosturbić.pdf).
11. Wall M.B.: A Genetic Algorithm for Resource – Constrained Scheduling. Doctoral Dissertation for Mechanical Engineering at the Massachusetts Institute of Technology.
12. Dorigo M., Maniezzo V., Colnani A.: The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Systems, Man and Cybernetics, Italy, 1996.
13. Merkle D., Middendorf M., Schmeck H.: Ant colony optimization for resource-constrained project scheduling. IEEE Transactions on Evolutionary Computation, Vol. 6, pp. 333–346, 2002.
14. Maniezzo V.: Gambardella L.M., F. de Luigi, Ant Colony Optimization. New Optimization Techniques in Engineering, Springer-Verlag Berlin Heidelberg, 101-117, 2004.
15. Dorigo M. Do Caro G. Gambardella L.M.: Ant Algorithms for Discret optimization. Artificial Life 5: 137-172, 1999.
16. Boryczka U.: Algorytmu optymalizacji mrowiskowej. Wydawnictwo Uniwersytetu Śląskiego, Katowice, 2006.

17. Dorigo M., Stutzle T.: Ant Colony Optimization. Massachusetts Institute of Technology, London, 2004.
18. Skołud B.: Planowanie wielosortymentowej produkcji rytmicznej. Zeszyty naukowe Politechniki Śląskiej. Mechanika z.136, Gliwice, 2000.

Mgr inż. Bożena MARCIŃCZYK  
Prof. dr hab. inż. Bożena SKOŁUD  
Instytut Automatykacji Procesów Technologicznych  
i Zintegrowanych Systemów Wytwarzania  
Politechnika Śląska  
44-100 Gliwice, ul. Konarskiego 18A  
tel./fax: (0-32) 237 16 24  
e-mail: bozena.skolud@polsl.pl  
bozena.marcinczyk@polsl.pl