

Informatyka: środa, tydzień parzysty, 9:45  
Grzegorz Szczepaniak  
numer indeksu: 121285  
mail: [grzegorz.ja.szczepaniak@student.put.poznan.pl](mailto:grzegorz.ja.szczepaniak@student.put.poznan.pl)

Wiktor Gonczaronek  
numer indeksu:  
mail:

Problem szeregowania zadań: algorytm mrówkowy (ACS)

# 1. Zadanie:

## Problem 4

Jobshop, liczba maszyn  $m=2$ , liczba zadań  $n$ ,  
operacje niewznawialne,

dla pierwszej i drugiej maszyny  $k$  okresów przestoju o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), okresy te nie mogą się na siebie nakładać (tj. czas przestoju na I maszynie nie może pokrywać się z czasem dowolnego przestoju na II maszynie),  $k \geq n/5$ ,

*minimalizacja sumy czasów zakończenia wszystkich operacji*

## 2.1 Generator instancji problemu:

Instancje generowaliśmy za pomocą utworzonej przez nas klasy Creator. Parametrami określającymi generator są:

maxExecTime – określa maksymalny czas trwania pojedynczej operacji

maxMaintenanceTime – określa maksymalny czas trwania przerw maintenance

Do generowania wartości pseudolosowych użyliśmy klasy Random.

Czas trwania operacji losujemy za pomocą wspomnianego powyżej generatora z przedziału od 1 do maxExecTime. Przydział operacji do konkretnych maszyn uzyskujemy poprzez losowanie maszyny dla pierwszego zadania, drugą operację wpisujemy automatycznie do listy operacji na przeciwnej maszynie.

Do tworzenia przerw maintenance również używamy generatora liczb pseudolosowych. W celu uniknięcia nachodzenia na siebie przerw czasy rozpoczęcia maintenance losujemy w następujący sposób:

- a) rozpoczęcie pierwszego maintenance uzależnione jest od liczby zadań oraz ich maksymalnego czasu wykonywania,
- b) kolejne maintenance rozpoczynają się w chwili : czas rozpoczęcia poprzedniego maintenance + maksymalny czas jego trwania + wartość losowa z przedziału (1,220),
- c) czas trwania losujemy z przedziału od 1 do maxMaintenanceTime.

Liczba maintenance na każdej maszynie wyznaczamy w następujący sposób:  
maksymalna liczba zadań / 5 + wartość losowa z przedziału (0,5).

Następnie utworzone obiekty są zapisywane do pliku tekstowego.

## 2.2 Generator rozwiązania losowego:

Generator tworzy rozwiązanie losowe które ma zostać ulepszone poprzez działanie algorytmu. W naszym przypadku szereguje on operacje w losowy sposób zgodnie z wymaganiami dotyczącymi zakończenia pierwszej operacji przed drugą oraz przypisaniem do konkretnych maszyn.

## 2.3 Generowanie rozwiązania za pomocą algorytmu ACS:

Algorytm ACS posiada wiele parametrów dzięki którym można zmieniać szybkość oraz dokładność jego działania:

- stopnień parowania feromonów tablicy,
- ilość dodawanych feromonów w każdej iteracji
- liczbę mrówek którą wypuszczamy w każdej iteracji algorytmu
- współczynnik alpha który jest wykorzystywany przy wyborze następnej operacji przez mrówkę, w zależności od ustawienia zwiększa bądź zmniejsza prawdopodobieństwo wybrania przez mrówkę kolejnej operacji z listy,
- paramentr gamma określający o ile rozwiązanie może być gorsze od najlepszego dotychczas, aby ulepszyć o nie tablicę feromonów
- paramentr określający minimalne polepszenie rozwiązania w określonej liczbie iteracji, jeżeli minimum nie zostanie osiągnięte resetujemy tablicę feromonów do stanu początkowego. Ma to na celu rozwiązanie problemu wpadania algorytmu w tzw minimum lokalne,

Dodatkowo dodaliśmy parametr, który określa poniżej jakiego poziomu feromonów w tablicy nie będzie on dalej dekrementowany. Miało to na celu przyspieszenie działania algorytmu, ponieważ dla 50 zadań nasza tablica feromonów ma rozmiar 100x100 i wykonywanie mnożenia na bardzo małych liczbach znacząco spowalniało nasz algorytm co prowadziło do znacznego pogorszenia jego efektywności.

Dodawanie feromonów do tablic uzależnione jest od tego, w której iteracji jest nasz algorytm. Początkowe rozwiązania mają mniejsze znaczenie, ponieważ są to w dużej mierze rozwiązania losowe. Wraz z upływem czasu działania algorytmu znaczenie kolejnych rozwiązań wzrasta i dodajemy więcej feromonu do tablicy. Dodatkowo rozwiązania które zasilą tablicę feromonów muszą spełniać następujący warunek:

$\text{gamma} * \text{wartość rozwiązania najlepszego} > \text{wartość rozwiązania aspirującego do ulepszenia tablicy}$

Warunkiem stopu algorytmu jest upłynięcie czasu określonego w parametrach naszej metaheurystyki.