Kierunek Informatyka, Wydział Informatyki	**Dane osobowe** Katarzyna Jóźwiak Nr indeksu: 127237 Piotr Pawlaczyk Nr indeksu: 127245 Środa, tydzień nieparzysty 9:45-11:15 Email kontaktowy : katarzyna.jozwiak@student.put.poznan.pl	**Data realizacji** Styczeń 2017

Problem 6 :
„Flowshop, liczba maszyn m=2, liczba zadań n,
operacje niewznawialne,
dla pierwszej maszyny k okresów przestoju o losowym czasie rozpoczęcia i trwania (określonym przez generator instancji problemu), k >= n/4,
czas gotowości dla każdej operacji nr 1 każdego zadania, nieprzekraczający połowy sumy czasów wszystkich operacji dla maszyny I ,
minimalizacja sumy czasów zakończenia wszystkich operacji"

Sprawozdanie

1) Metoda rozwiązania problemu

Problem zaimplementowaliśmy przy użyciu wektora wektorów uszeregowań, z których każdy odpowiada pojedynczej maszynie (wektory uszeregowań maszyn pierwszych mają indeksy parzystych, wektory uszeregowań maszyn drugich to wektory o indeksach nieparzystych). Każdy indeks wektora odpowiada pojedynczej jednostce czasu. Aby reprezentować uszeregowanie użyliśmy wartości liczbowych takich, że liczby dodatnie reprezentują odpowiedni numer zadania, liczby ujemne odpowiednie maintenance , a zera to okresy bezczynności maszyny.

Metodą, którą zdecydowaliśmy się rozwiązać zadany problem jest algorytm mrówkowy (ACO). Na początku wygenerowaliśmy dane wejściowe, a następnie na ich podstawie utworzyliśmy określoną ilość losowych uszeregowań. Do tego celu użyliśmy funkcji generatora losowego.

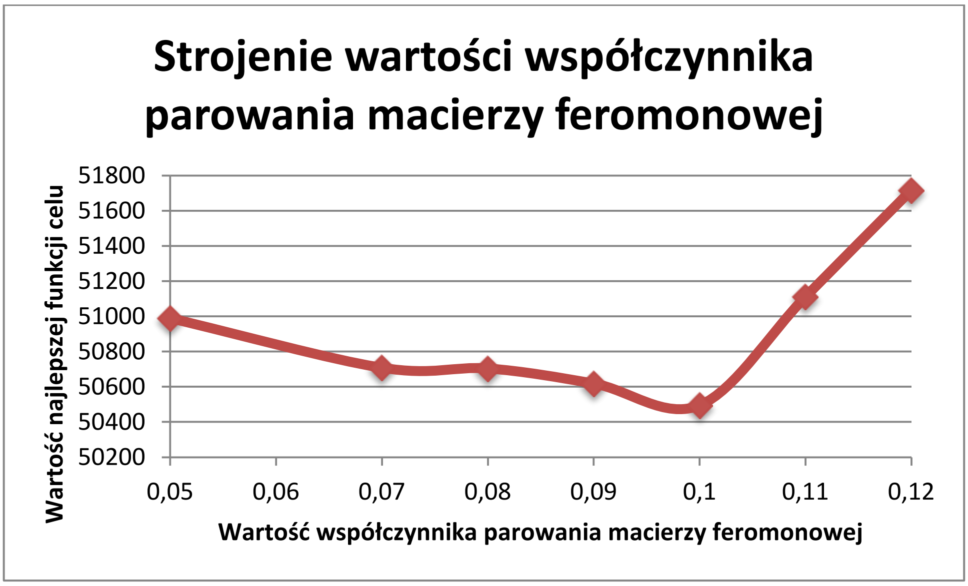
Po wykonaniu określonej ilości uszeregowań wykonanych przez generator, odrzucamy metodą turnieju połowę z nich. Losujemy dwa losowe uszeregowania i zostawiamy w wektorze rozwiązań te z lepszą funkcją celu. Następnie na ich podstawie wypełniamy macierz feromonową i zapisujemy wartość funkcji celu najlepszego z uszeregowań generatora losowego, która później jest zapisywana do pliku wynikowego.

Następnie przechodzimy do głównej pętli algorytmu. Tworzymy kolejne uszeregowania , zmniejszając co określoną liczbę iteracji prawdopodobieństwo korzystania z generatora losowego , a zwiększając prawdopodobieństwo korzystanie z macierzy feromonowej . W późniejszym czasie dokonujemy mutacji na wektorze rozwiązań , którym dysponujemy, tworząc nowe uszeregowania.
Po mutacji dokonujemy kolejnego turnieju, zestawiając losowo w pary uszeregowania i zostawiając lepsze. W dalszej kolejności ponownie uzupełniamy macierz feromonową, wygładzamy jej wartości oraz wykonujemy w jej zabieg parowania i przechodzimy do

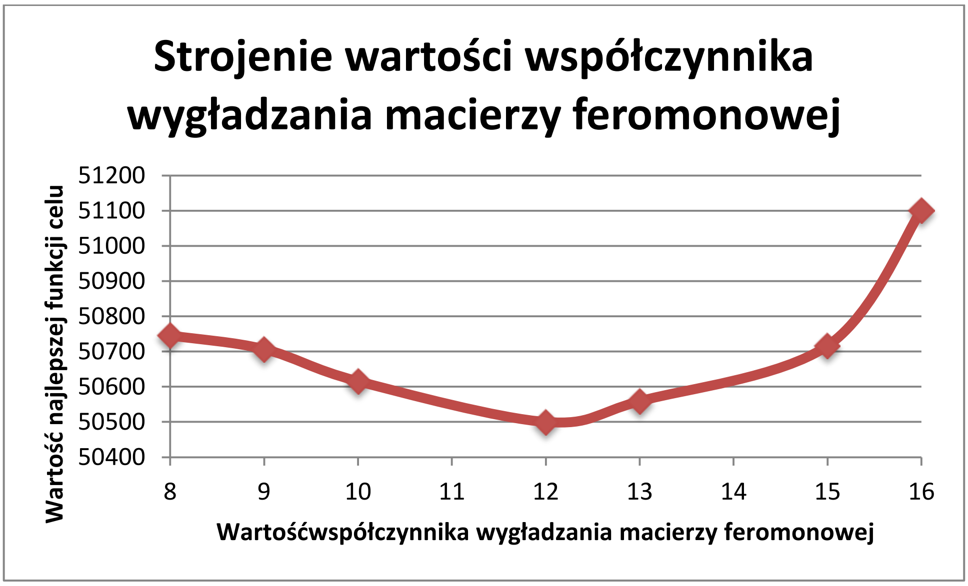
kolejnej iteracji pętli.
Na sam koniec zapisujemy najlepsze uszeregowanie, wraz z jego parametrami do pliku wynikowego.

2) Strojenie

Aby metaheurystyka działała w prawidłowy i jak najbardziej optymalny sposób należy zacząć od przeprowadzenia operacji strojenia występujących w niej parametrów. Aby to zrobić wygenerowaliśmy jedną instancję testową danych (50 zadań i 15 mainenance). W strojeniu danej wartości tylko ta wartość się zmienia, reszta danych pozostaje niezmienna.

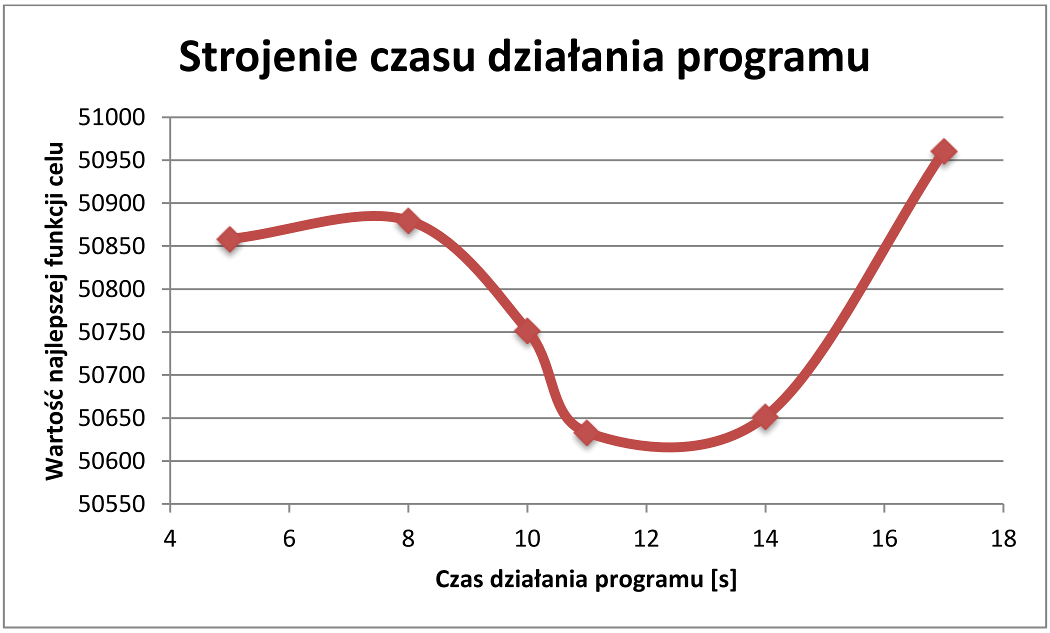


Pierwszą wartością, od której zaczęliśmy strojenie jest współczynnik parowania macierzy feromonowej wskazujący na to jaka część macierzy zanika po każdej iteracji głównej pętli algorytmy. Według naszych testów wartość optymalna wynosi on 0,1, ponieważ dla tej wartości funkcja celu przyjmuje najmniejszą wartość. Dla większych wartości parametru funkcja celu gwałtownie rośnie, ponieważ zbyt dużo feromonu zanika i trudniej za jego pomocą generować kolejne rozwiązania.

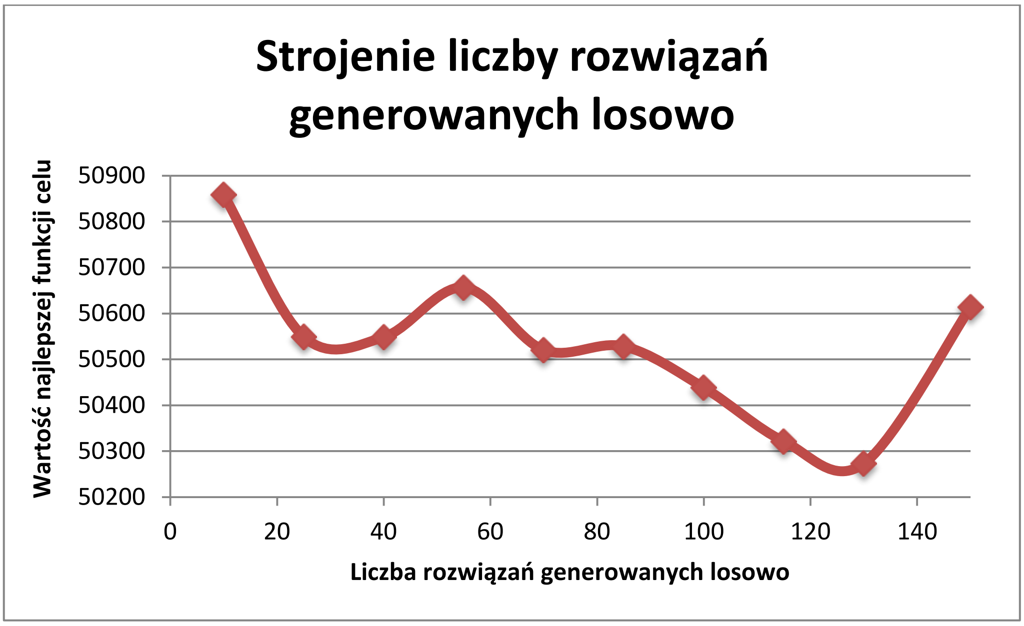


Kolejną strojoną wartością jest współczynnik wygładzenia macierzy feromonowej . Współczynnik wygładzenia macierzy jest to podstawa logarytmu wykorzystywana we wzorze na wygładzanie macierzy . Im większy jest to współczynnik, tym większy efekt wygładzania . W metaheurystyce nie może jednak wystąpić zbyt duży efekt wygładzania, ponieważ nie byłby wtedy widoczny efekt przechodzenia poszczególnych uszeregowań przez macierz. Według przeprowadzonych przez nas testów najbardziej optymalną wartością tego współczynnika jest 12 .

Wykorzystaliśmy wzór: $F_{ij} = F_{min}(1 + log_p \frac{F_{ij}}{F_{min}})$,gdzie F_{ij} to dana komórka macierzy, F_{min} to minimalna ilość feromonu w danym wierszu, a p jest to wartość współczynnika wygładzania macierzy.



Inną zmienną, której wartość staraliśmy się zoptymalizować jest czas głównej pętli algorytmu metaheurystycznego. Początkowo wartość funkcji celu wraz ze wzrostem argumentu ulega stopniowej poprawie. Optymalną wartość osiąga po 11 sekundach działania , ponieważ dłuższe jej działanie nie poprawiało w efektywny sposób wartości otrzymywanych przez nas funkcji celu.

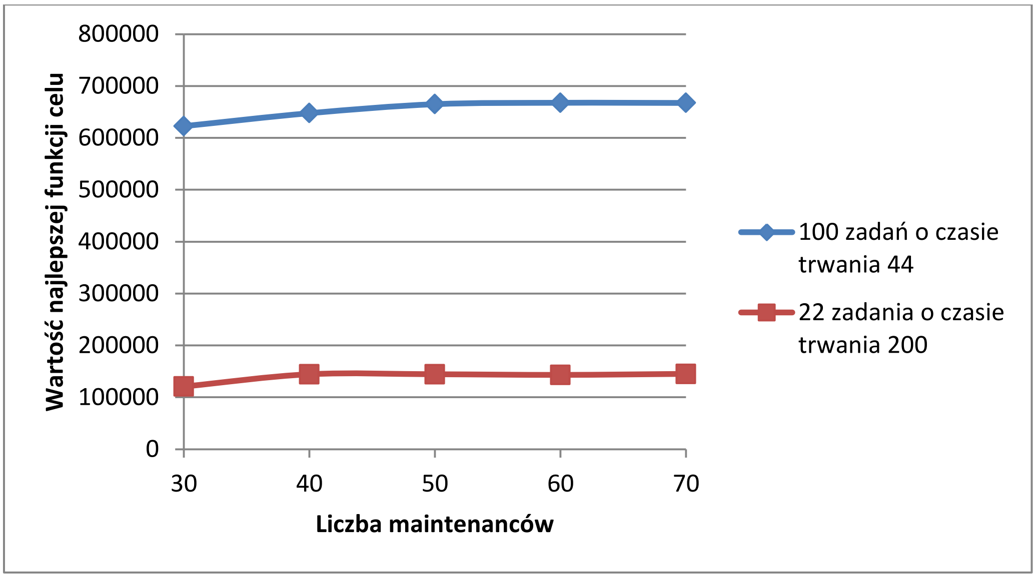


Ostatnim testowanym przez nas parametrem była ilość rozwiązań generowanych losowo, która wypełnia macierz feromonową przed rozpoczęciem głównej pętli algorytmu metaheurystcznego.
Po wykonanych przez nas testach okazało się, że stopniowe zwiększanie tego parametru powoduje polepszenie wartości funkcji celu, jednak jak widać na wykresie przy wartości 120 osiągamy optimum i dalsze zwiększanie liczby tych rozwiązań, nie przynosi efektu, wręcz przeciwnie, funkcja celu rośnie.

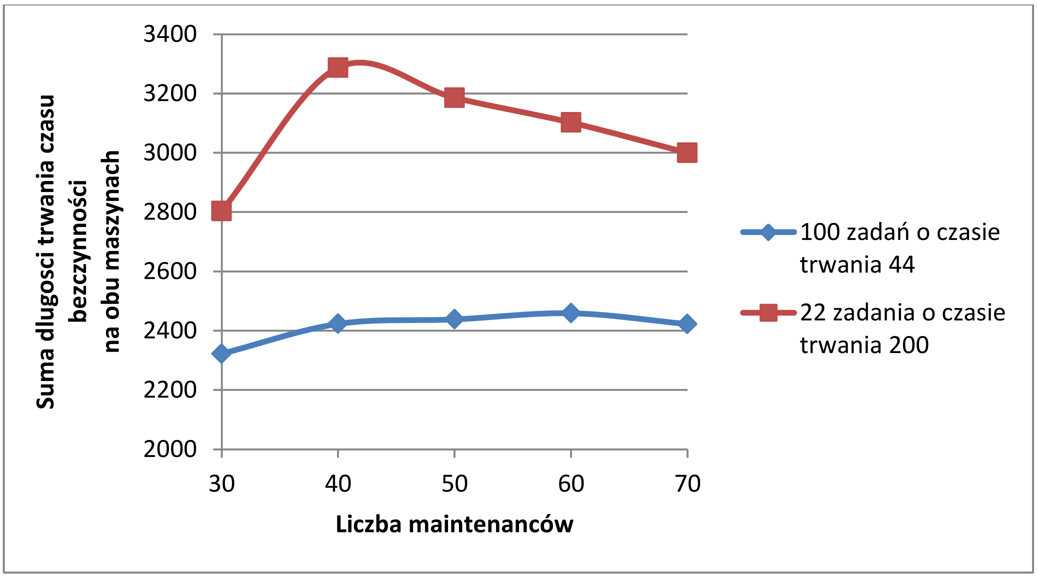
3) Inne testy

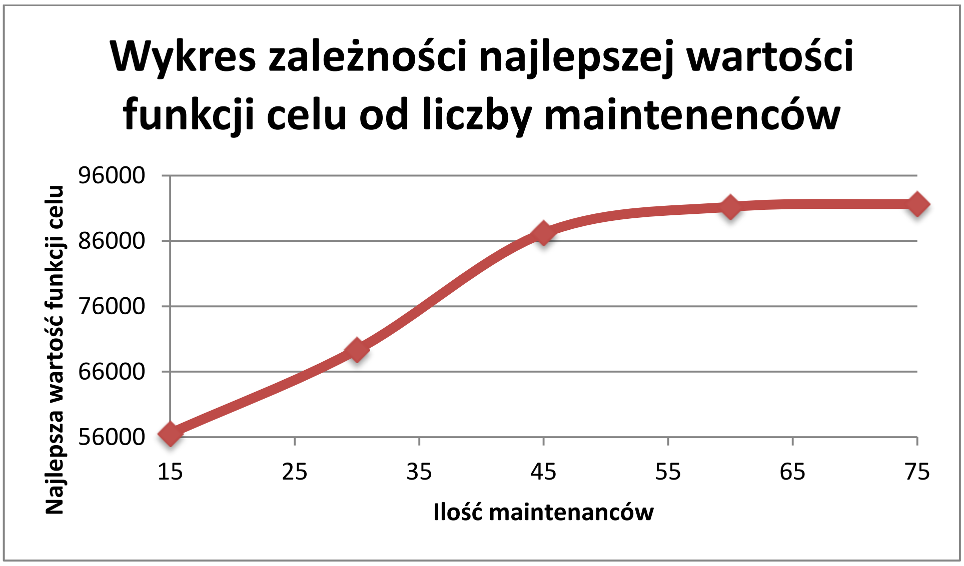
Pierwszym analizowanym przez nas problem było to czy łatwiej uszeregować więcej zadań o krótszym czasie trwania czy mniej zadań o dłuższym czasie trwanie.
Aby odpowiedzieć na to pytanie porównaliśmy najlepsze funkcje celu instancji o identycznym sumarycznym czasie trwania zadań równym 4400 jednostek czasu i po 2200 jednostek czasu dla każdej sumy trwania operacji danej maszyny.
Jak widać na poniższym wykresie dużo łatwiej uszeregować mniejsza liczbę zadań o dłuższym czasie trwania, niż więcej zadań o krótszym czasie trwania. Wbrew oczekiwaniom wartość funkcji celu dla obu wykresów nie rośnie znacząco wraz ze wzrostem liczby maintenance. Jest to spowodowane tym, że wartości początkowe przerw technicznych na maszynie 1 są

losowe.



Odwrotnie do funkcji celu prezentują się wyniki dla sumy czasów przestoju maszyny 1 jak i maszyny 2. Dzieje się tak, ponieważ zadaniami o krótszym czasie trwania łatwiej wypełnić jest wolne przestrzenie w uszeregowaniu pomiędzy maintenancami.





Kolejnym przeprowadzonym przez nas testem jest zależność najlepszej wartości funkcji celu od liczby maintenance. Na poniższym wykresie możemy zauważyć pewną zależność. Im więcej maintenance tym gorsza wartość funkcji celu. Jednakże pogorszenie się tej wartości nie jest stałe. Obserwujemy, że wraz z dodawaniem tej samej liczby maintenance wartość funkcji celu ulega pogorszeniu o mniejsze wartości (wykres nie rośnie liniowo) .

4) Podsumowanie

Każdy punkt na dowolnym wykresie w naszym sprawozdaniu jest to średnia z dziesięciu instancji.
Dobieranie odpowiedniej wartości współczynnika wygładzania macierzy feromonowej jak i jej parowania pozwala generowanym przez nas rozwiązaniom na szerszą eksploatacje macierzy feromonowej i zapobiega tworzeniu tych samych rozwiązań wraz z upływem czasu.
Podczas przeprowadzanych przez nas testów zauważyliśmy, że długość trwanie programy nie zawsze wpływa znacząco na polepszenie otrzymywanych rozwiązań.
Ważnym czynnikiem była decyzja o tym by wygrany z losowej pary uszeregowań w turnieju mógł ponownie brać udział w losowaniu. Pozwoliło nam to na zapisywanie lepszych uszeregowań do macierzy feromonowej .