Julia Tadej 136820 julia.tadej@student.put.poznan.pl

Jakub Raczyński 136787 jakub.raczynski@student.put.poznan.pl

Grupa: wtorki 13.30 tygodnie parzyste

Problem: Travelling salesman problem/ Problem komiwojażera

Metaheurystyka: Ant Colony Optimization

# 1. Opis algorytmu

Zgodnie z ideą algorytmu mrówkowego zaimplementowana przez nas metaheurystyka symuluje przejście kolejnych kolonii mrówek przez graf, w którym poszukujemy cyklu Hamiltona o najmniejszej długości. Mrówki każdorazowo startują z losowych wierzchołków grafu. Przed przejściem pierwszej kolonii inicjalizujemy wartości feromonu i obliczamy widoczność. Mrówki z pierwszej kolonii wybierają kolejne miasta cyklu losowo. W kolejnych koloniach z prawdopodobieństwem **q0** decydujemy o tym czy wybór następnej krawędzi ma być deterministyczny. Jeśli tak, wysyłamy mrówkę krawędzią najlepszą według zgromadzonych informacji na temat ilości feromonu i widoczności. Przy pomocy parametru **β** regulujemy ważność widoczności względem ilości feromonu. Mrówka przechodząc przez wybraną krawędź ściera część zawartego na niej feromonu, co skłania inne mrówki do zbadania dotychczas pomijanych krawędzi. Wymazywanie feromonu zależne jest od parametru **φ**. Po przejściu każdej kolonii część feromonu (odzwierciedlona wartością parametru **ρ**) paruje, a na krawędziach należących do globalnie najlepszego cyklu jego ilość jest zwiększana. Zmiana podejścia do modyfikacji feromonu względem standardowego algorytmu ACO opisywana w źródłach jako ACS (Ant Colony System) pozwoliła na znaczną poprawę jakości wyników.

**1.1.** Wzory opisujące działanie algorytmu

Oznaczenia:

f[I, J] - ilość feromonu na krawędzi (i, j)

f0 - początkowa ilość feromonu na każdej krawędzi

t+1, t - numery iteracji

LGB - długość globalnie najlepszego cyklu

LIJ - długość krawędzi (i, j), w[I, J] -widoczność krawędzi (i, j) - (1/LIJ)β

Wzory:

z prawdopodobieństwem q0 krawędź, dla której iloczyn (f[I, J])α · w[I, J]  jest największy

- wybór drogi

z prawdopodobieństwem 1- q0 wybór losowy krawędzi o przypisanym określonym  
 prawdopodobieństwie: p[I, J] = (f[I, J])α · w[I, J] / Σ p

- wymazywanie feromonu: f[I, J] = f[J, I] = (1 - φ) · f[I, J] + φ · f0

- zwiększanie feromonu: f[I, J](t + 1) = f[I, J](t) + 10/LGB

- parowanie feromonu: f[I, J] = f[J, I] = (1 - ρ) · f[I, J]

# 2. Pseudokod

**początek** ACO

ustalenie początkowej wartości feromonu na krawędziach grafu

**dopóki** (nie jest spełniony warunek zakończenia\*) **powtarzaj**:

wylosowanie miast początkowych i konstrukcja ścieżek przez mrówki na podstawie aktualnej ilości feromonu na krawędziach grafu, ścieranie części feromonu bezpośrednio po przejściu danej krawędzi

odparowanie części feromonu ze wszystkich krawędzi

naniesienie dodatkowej ilości feromonu na krawędzie należące do globalnie najlepszego cyklu, proporcjonalnie do jego długości

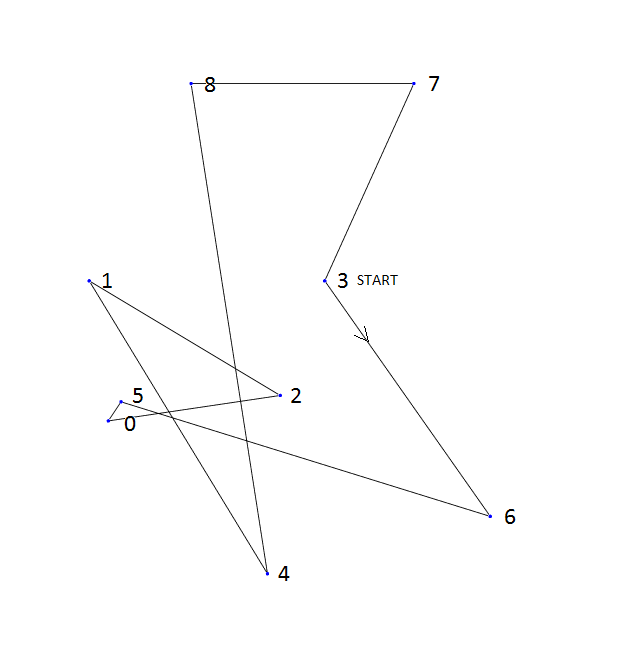
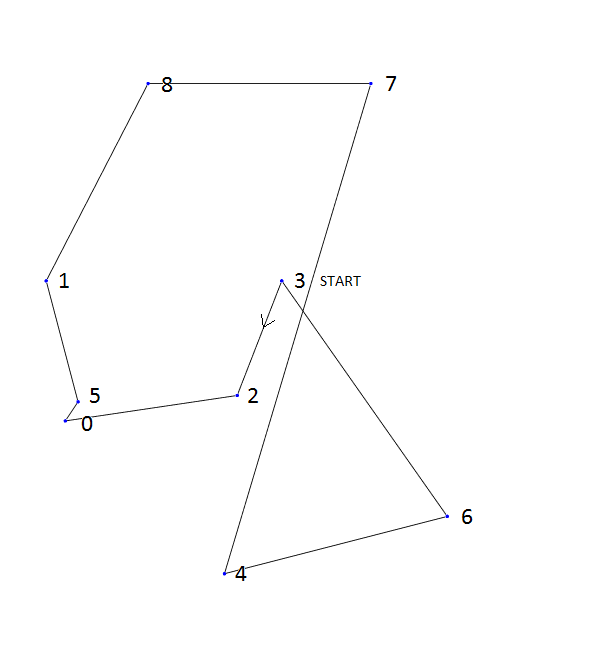
**koniec** ACO

\*W przypadku naszego kodu warunkiem zakończenia jest liczba iteracji mieszcząca się w zadanym czasie

# 3. Przykłady działania

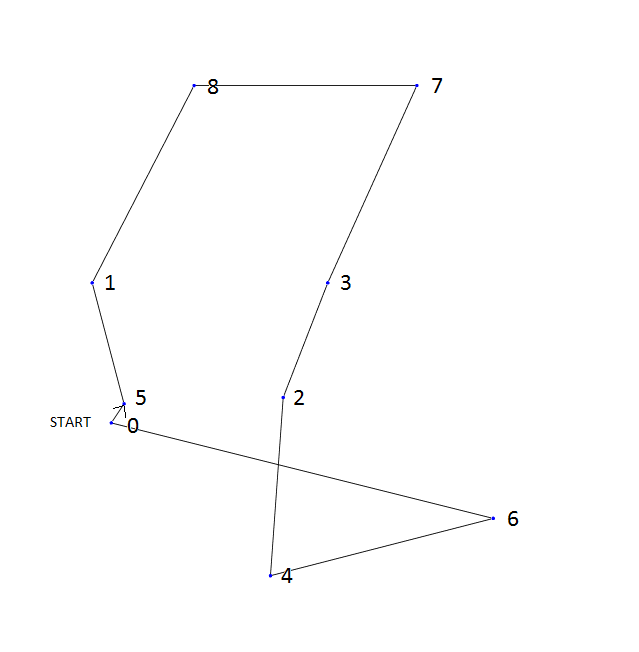
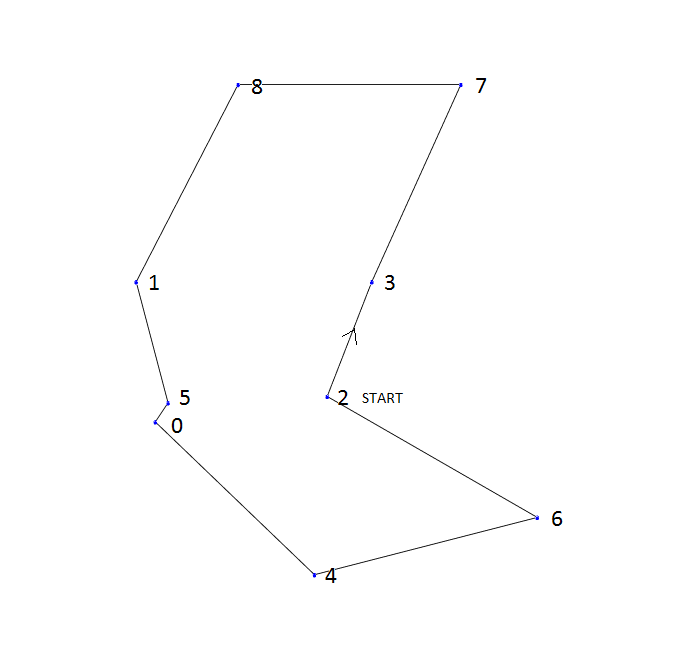
Zobrazujemy działanie naszego algorytmu poprzez pokazanie globalnie najlepszego cyklu po przejściu kolejnych kolonii. Przedstawia on krawędzie na których zwiększamy liczbę feromonu w czasie działania algorytmu. Przy każdym kroku zamieszczamy także długość przedstawianego cyklu. Działanie algorytmu prezentujemy dla niewielkiej losowej instancji złożonej z 9 wierzchołków oraz jednej instancji wybranej z biblioteki - berlin52. W pierwszym przypadku możemy zaprezentować wszystkie kroki, w drugim - kilka wybranych.

## Losowa instancja złożona z 9 wierzchołków

2. kolonia  
301,5

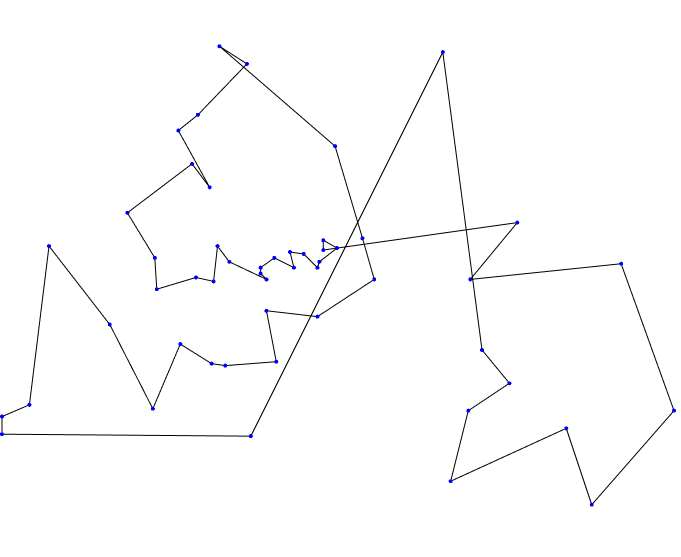
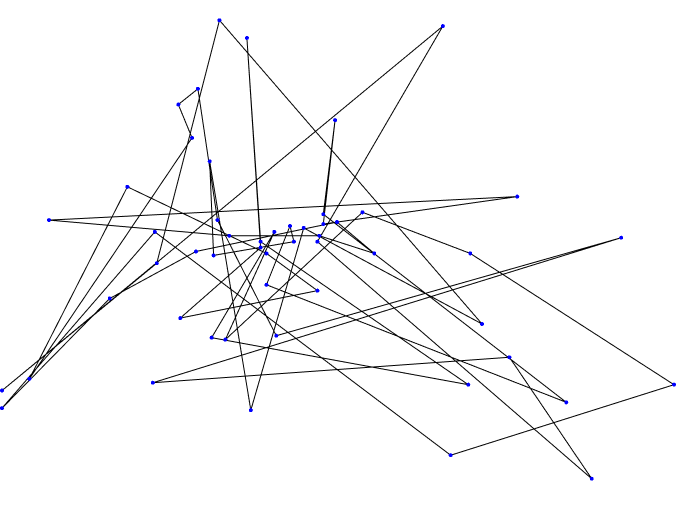
1. kolonia  
372,6

5. kolonia  
255,3

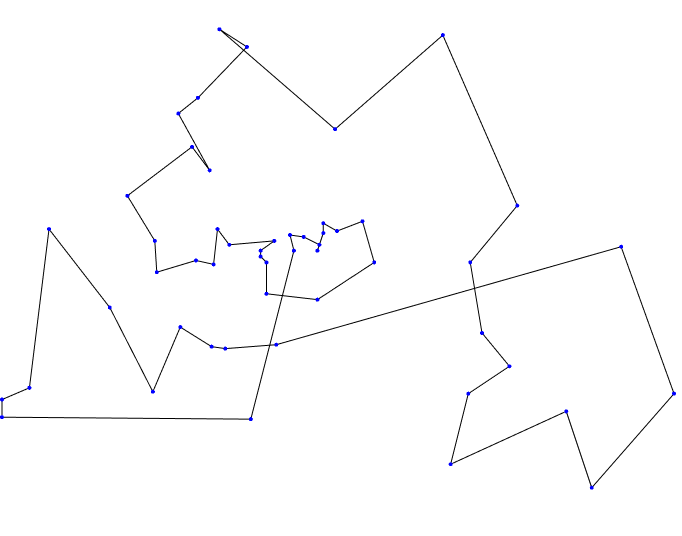
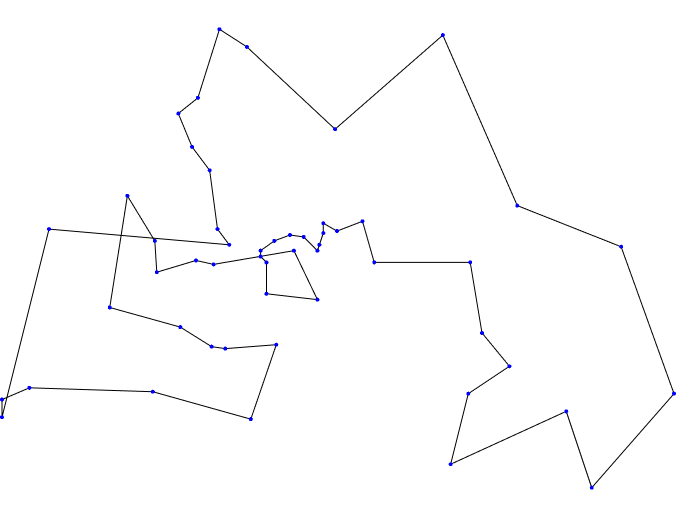
3. i 4. kolonia  
272,5

## Berlin52



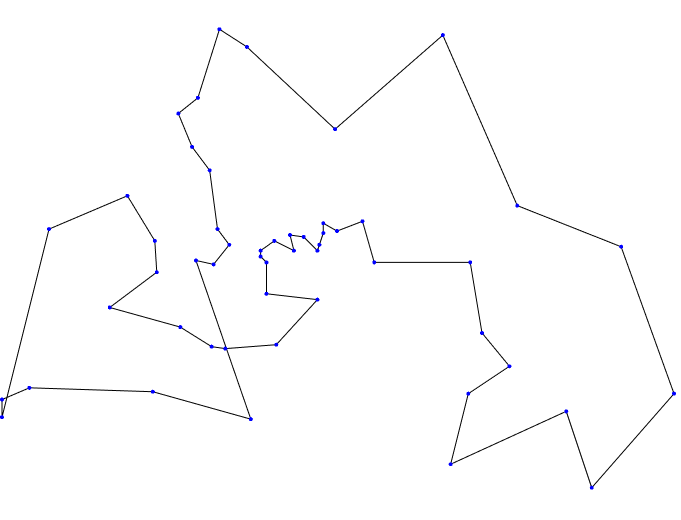
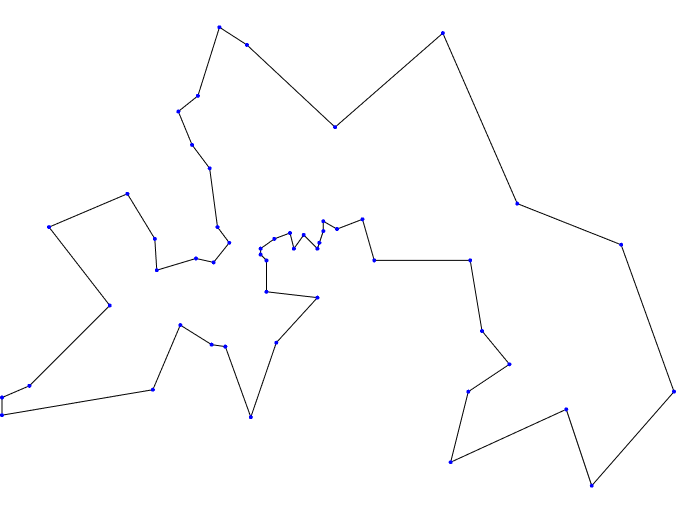
1. kolonia  
26545

2. kolonia  
9671

3. kolonia  
9015

6. kolonia  
8238

8. kolonia  
7906

476. kolonia  
7544

# 4. Wykresy i testy

**4.1.** Aby jeszcze lepiej zobrazować działanie zaimplementowanego przez nas algorytmu przedstawimy na początek wpływ parametrów metaheurystyki na jakość wyniku dla wybranej przez nas instancji. Dla parametrów α i β kluczowy jest stosunek ich wartości - zatem wartość α będzie stała i równa 1, a wartość parametru β będziemy zmieniać. Kluczowe dla jakości rozwiązania są także parametry φ i q0.

W wykonanych przez nas dodatkowych testach zauważyliśmy brak znaczącego wpływu parametru ρ na jakość wyniku. W testach będzie miał więc stałą wartość ρ =

Postanowiliśmy przetestować zaimplementowany przez nas algorytm na kilku płaszczyznach.

* Porównanie najlepszej uzyskanej wartości z alg. zachłannym (w ograniczonym czasie? w ograniczonej liczbie prób?)
* Porównanie najlepszej uzyskanej wartości (w ograniczonym czasie? w ograniczonej liczbie prób?) ze znanym optimum
* dla wybranej instancji - pokazanie wpływu parametrów na wynik (max 3 parametry po 5 wartości?)
* Zależność jakości wyniku od wielkości instancji w dwóch wariantach: 1. wszystkim instancjom zadajemy tę samą liczbę pokoleń, 2. wszystkim instancjom dajemy ten sam czas

Lista instancji EUC\_2D

- a280

- eil51

- eil76

- eil101

- kroa200

- krob150

- kroc100

- lin318

- pcb442/pr439

# 5. Źródła:

- "ACO Algorithms for the Traveling Salesman Problem" - T. Stäutzle, M. Dorigo

- "Bio-inspired Algorithms for TSP and Generalized TSP" - Zhifeng Hao, Han Huang, Ruichu Cai

- "Improved ACO Algorithm with Pheromone Correction Strategy for the Travelling Salesman Problem" - M. Tuba, R. Jovanovic

- "On Optimal Parameters for Ant Colony Optimization algorithms" - D. Gaertner, K. Clark

1,2. Porównanie najlepszego/średniego wyniku z 10 prób (każda 3 minuty) z wynikiem alg. zachłannego dla X losowych instancji

3,4. Porównanie najlepszego/średniego wyniku z 10 prób (każda 3 minuty) ze znanym optimum (dla wszystkich instancji o znanym optimum)

5. Dla jednej wybranej instancji - pokazanie wpływu parametrów na wynik (nwm ile parametrów zmieniamy chyba 2 albo 3 i myślałam o 5 różnych wartościach dla każdego ale to zaraz uzgodnimy)

6. Zależność jakości wyniku od wielkości instancji - proponuję żeby przedstawić to jako jakiś błąd względny dzięki temu wykorzystamy ponownie dane z wykresu 3. tj różnica między najlepszym wynikiem a optimum / wielkość optimum czy coś takiego

7. Chciałam zrobić to samo co w 6stce ale olać na chwilę ograniczenie 3 minut i dać dla wszystkich instancji tą samą liczbę pokoleń żeby pokazać że później wynik jest spoko xd ale w sumie szkoda czasu na takie testy raczej