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 (gdzie Σ p to suma prawdopodobieństw dla wszystkich krawędzi)

- 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

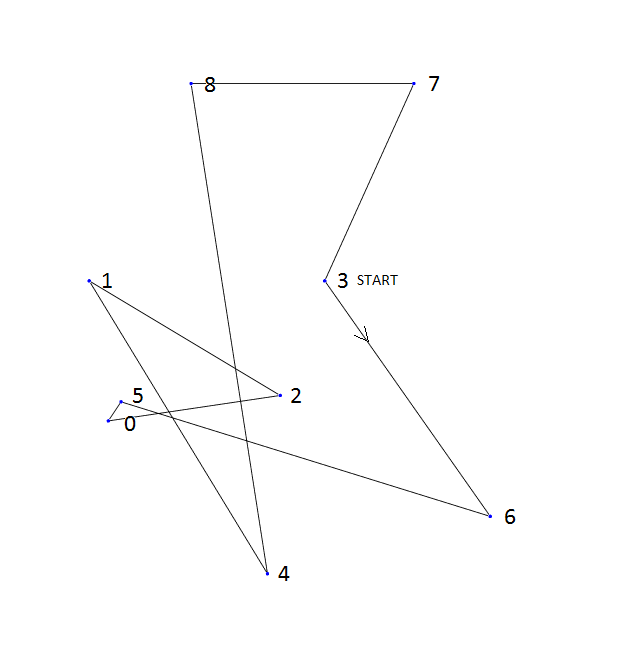
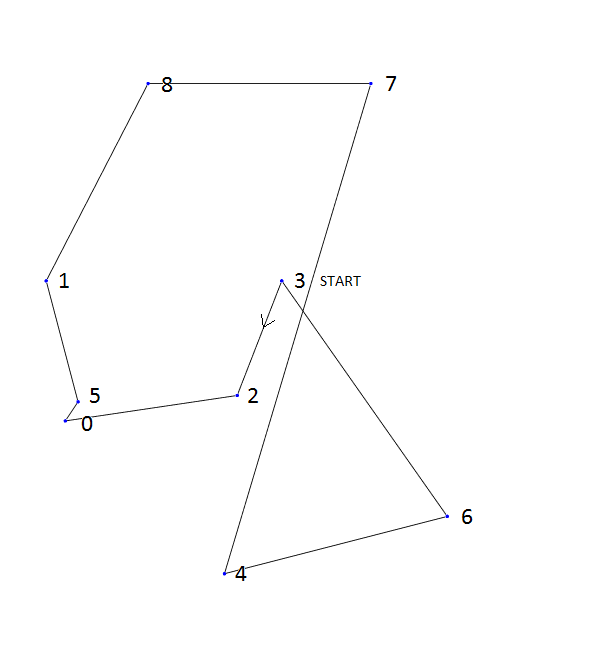
**koniec** ACO

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

# 3. Przykłady działania

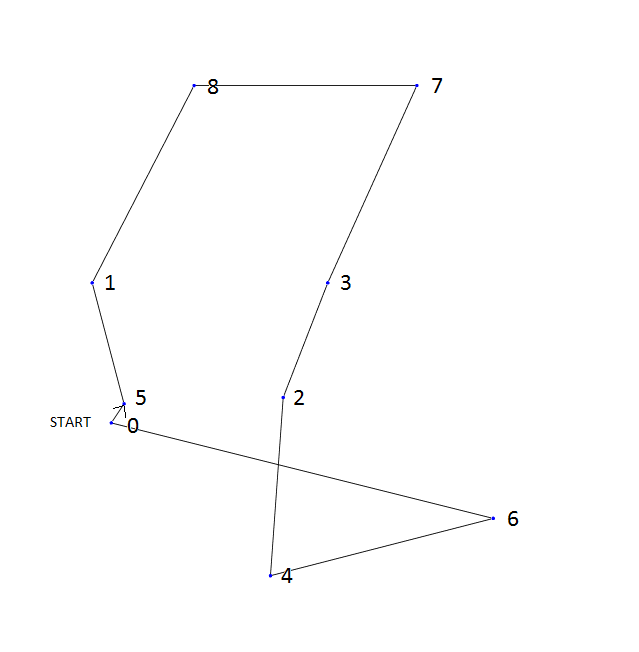
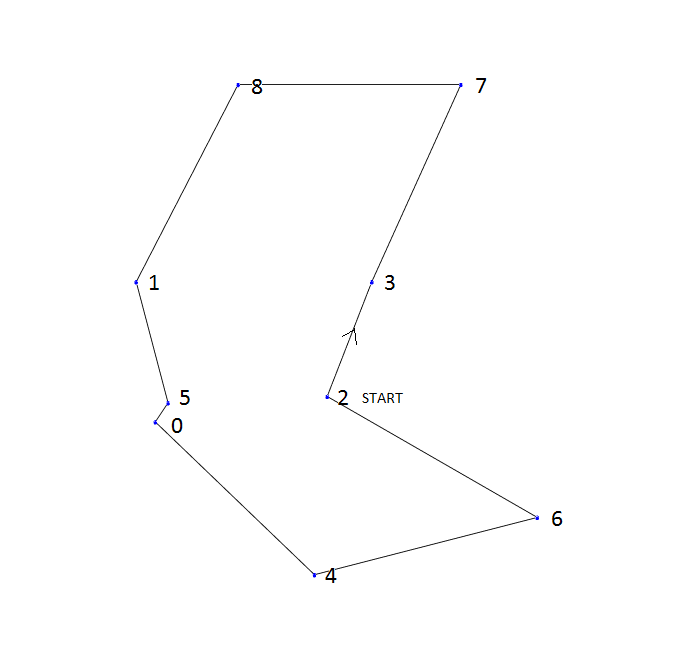
Zobrazujemy działanie naszego algorytmu poprzez pokazanie globalnie najlepszego cyklu po przejściu kolejnych kolonii. Przedstawia on krawędzie na których zwiększamy liczbę feromonu w czasie działania algorytmu. Przy każdym kroku zamieszczamy także długość przedstawianego cyklu. Działanie algorytmu prezentujemy 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

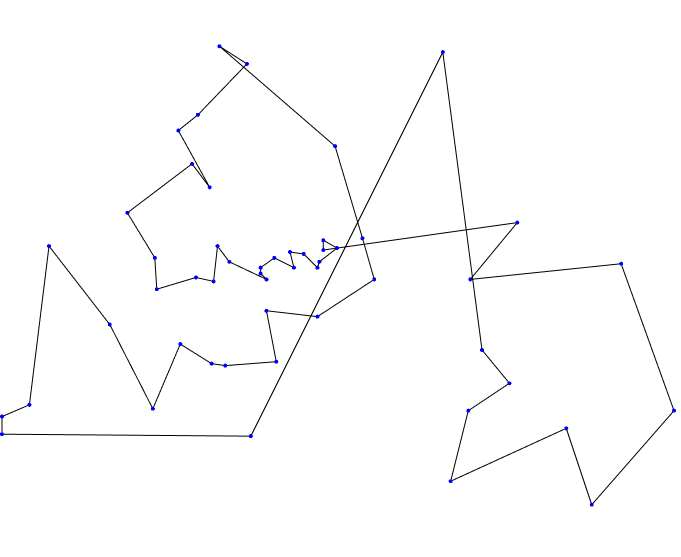
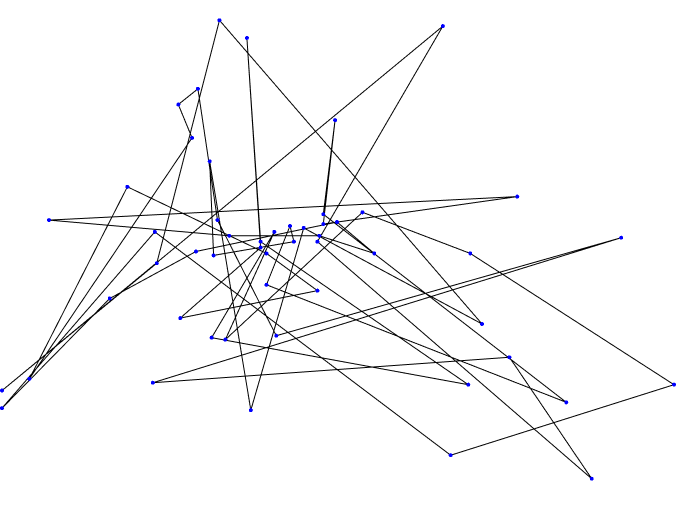
1. kolonia  
372,6

5. kolonia  
255,3

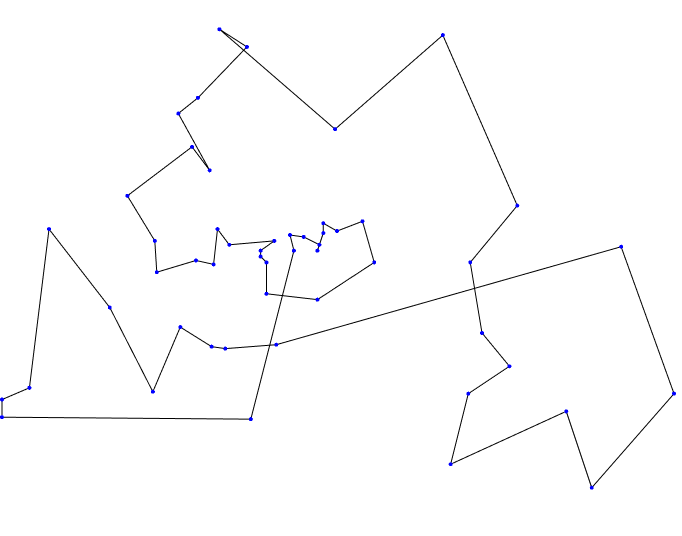
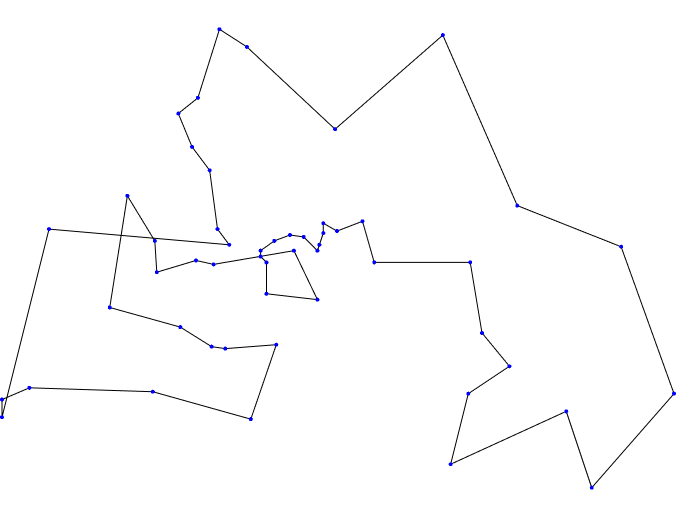
3. i 4. kolonia  
272,5

## Berlin52



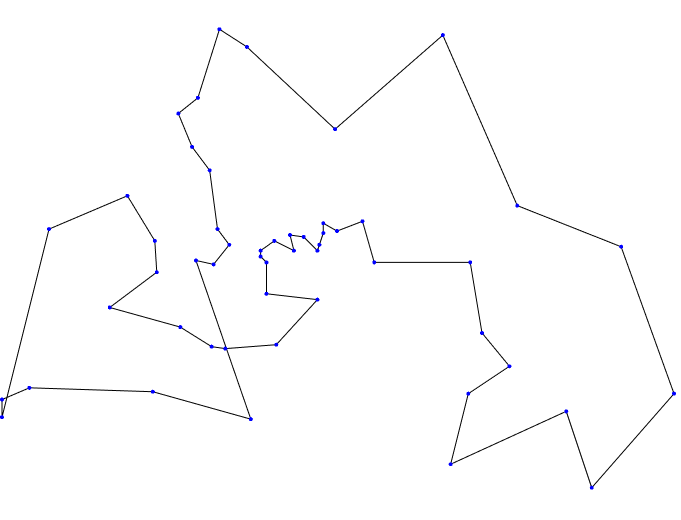
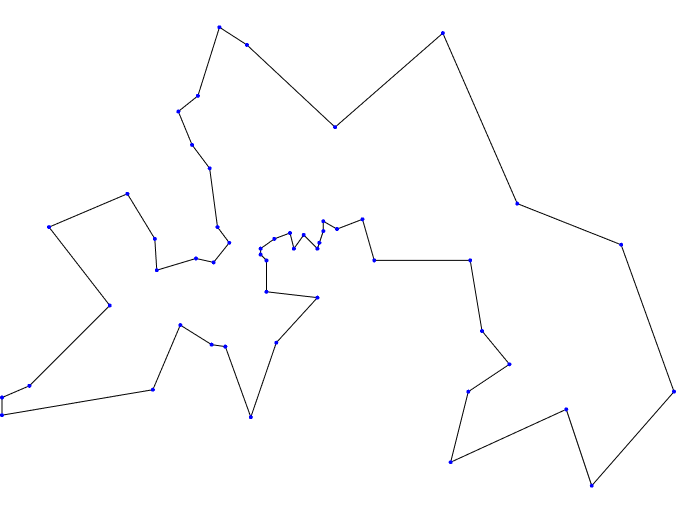
1. kolonia  
26545

2. kolonia  
9671

3. kolonia  
9015

6. kolonia  
8238

8. kolonia  
7906

476. kolonia  
7544

# 4. Wykresy i testy

**4.1.** Postanowiliśmy przetestować zaimplementowany przez nas algorytm na kilku płaszczyznach. Rozpoczniemy prezentację wyników testów od odniesienia do 5 instancji, z którymi pracowaliśmy przez cały semestr. W przypadku trzech z nich zanotowaliśmy poprawę względem wyników grupy ogłoszonych przed terminem oddania, dla jednej uzyskaliśmy wynik gorszy, a dla instancji berlin52 udało nam się uzyskać wartość optymalną.

|  |  |  |  |
| --- | --- | --- | --- |
| **nazwa instancji** | **wynik z pliku** | **nasz wynik** | **różnica** |
| berlin52 | 7544 | 7544 | 0 |
| bier127 | 119018 | 118768 | -250 |
| tsp250 | 12883 | 12686 | -197 |
| tsp500 | 89946 | 88592 | -1354 |
| tsp1000 | 25339 | 26283 | 944 |

\*Pomijając ograniczenie czasowe dla instancji tsp1000 udało nam się uzyskać wynik: **25509** w czasie 35 minut.

**4.2.** W kolejnej sekcji zaprezentujemy bardziej szczegółowo wyniki testów przeprowadzonych na instancjach z biblioteki instancji. ACO best przedstawia zawsze najlepszy wynik z próby 10 wykonań, a ACO avg średni wynik z tej samej ilości wykonań. OPT to najlepsze znane rozwiązanie. Na każde pojedyncze wykonanie nakładamy limit 3 minut. Ze względu na duże rozbieżności w długościach cykli otrzymanych dla poszczególnych instancji pogrupowaliśmy je tak, by wykresy były możliwie jak najbardziej czytelne.

Instancje o najkrótszych cyklach:

Instancje o cyklach "średniej" długości:

Instancje o najdłuższych cyklach:

Osobno przedstawimy strukturę wyników dla instancji berlin52, ponieważ długość jej cyklu nie przystaje do żadnej z grup. Ponadto możemy w ten sposób przekazać jeszcze dokładniejszą informację o otrzymanych wynikach.

Dla instancji pochodzących z biblioteki chcemy także zaprezentować zbiorcze porównanie jakości wyników. Jako miarę tej jakości przyjęliśmy wartość błędu naszego wyniku liczoną według wzoru: (nasz\_wynik - najlepszy\_znany)/najlepszy\_znany\*100%. Instancje uszeregowane są rosnąco według liczby miast.

**4.3.** Dokonaliśmy także testów dla zbioru "pseudolosowych" i losowych instancji. Postanowiliśmy więc w zestawieniu wyników dla nich uzyskanych odnieść się do wartości uzyskanej przez algorytm zachłanny. Instancje tsp250, tsp500 i tsp1000 to te, z którymi musieli zmierzyć się wszyscy. Tsp100 i tsp350 to instancje losowe wygenerowane przez nasz generator.

Porównanie najlepszego wyniku z wynikiem algorytmu zachłannego:

Porównanie średniego wyniku (z próby 10 uruchomień) z wynikiem algorytmu zachłannego:

Ponownie postanowiliśmy poza bezwzględnymi wartościami wyników przyjąć pewną wspólną miarę pozwalającą porównać jakość wyników pomiędzy poszczególnymi instancjami o różnych wielkościach. W przypadku instancji "losowych" jest to wymyślony przez nas "wskaźnik poprawy" obliczony zgodnie z wzorem: (wynik\_zachłanny - nasz\_wynik)/wynik zachłanny \* 100%. Instancje są uszeregowane rosnąco według liczby miast.

**4.4.** Aby jeszcze lepiej zobrazować działanie zaimplementowanego przez nas algorytmu przedstawimy w ostatniej sekcji wpływ parametrów 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. Przyjęliśmy więc jego stałą wartość ρ = 0.1

Przeprowadziliśmy wstępną selekcję zakresu wartości, ostatecznie wybierając do testów następujące:

β ϵ {2, 3, 4, 5, 6}

φ ϵ {0.74, 0.8, 0.86, 0.92, 0.98}

q0 ϵ {0.5, 0.6, 0.7, 0.8, 0.9}

Instancja, którą wybraliśmy to **bier127**.

Ponieważ testowaliśmy parametry jako 3-elementowe kombinacje, wynik testu jest w pewnym sensie funkcją trzech zmiennych. Dla wygody przedstawimy więc wyniki w tabeli, wyróżniając najlepsze. Wynik średni jest w przypadku tego testu średnią próby 5 uruchomień. Aby zoptymalizować czas testu zmniejszyliśmy liczbę kolonii, dlatego wyniki odbiegają w większości od zakresu otrzymanego w poprzednich zestawieniach. Uszeregowane są od najlepszych według średniej długości cyklu.

| Nr | q0 | β | φ | Najlepszy | Średnia |  | Nr | q0 | β | φ | Najlepszy | Średnia |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0.5 | 6 | 0.74 | 119047 | 119491,4 |  | 51 | 0.6 | 4 | 0.98 | 119977 | 120671,6 |
| 2 | 0.6 | 3 | 0.92 | 118818 | 119551,6 |  | 52 | 0.8 | 3 | 0.86 | 118688 | 120689 |
| 3 | 0.5 | 6 | 0.98 | 119513 | 119617 |  | 53 | 0.5 | 4 | 0.74 | 119725 | 120695,2 |
| 4 | 0.7 | 3 | 0.92 | 119045 | 119696,6 |  | 54 | 0.6 | 3 | 0.98 | 119765 | 120708 |
| 5 | 0.5 | 5 | 0.86 | 118942 | 119873,8 |  | 55 | 0.5 | 3 | 0.98 | 120136 | 120760,2 |
| 6 | 0.5 | 4 | 0.92 | 119438 | 119892,6 |  | 56 | 0.5 | 3 | 0.8 | 119823 | 120772 |
| 7 | 0.8 | 5 | 0.92 | 119111 | 119947 |  | 57 | 0.8 | 6 | 0.86 | 119553 | 120834,4 |
| 8 | 0.7 | 4 | 0.8 | 119705 | 120073,4 |  | 58 | 0.8 | 6 | 0.74 | 119687 | 120843,2 |
| 9 | 0.5 | 6 | 0.86 | 119654 | 120131,4 |  | 59 | 0.6 | 6 | 0.74 | 119638 | 120854,8 |
| 10 | 0.5 | 6 | 0.92 | 119665 | 120135 |  | 60 | 0.7 | 6 | 0.92 | 118740 | 120857,6 |
| 11 | 0.9 | 4 | 0.92 | 119115 | 120142,6 |  | 61 | 0.8 | 3 | 0.98 | 120147 | 120859,8 |
| 12 | 0.6 | 5 | 0.74 | 119233 | 120166 |  | 62 | 0.9 | 4 | 0.8 | 119194 | 120868 |
| 13 | 0.6 | 4 | 0.92 | 119169 | 120167,2 |  | 63 | 0.8 | 6 | 0.92 | 119584 | 120868,4 |
| 14 | 0.7 | 3 | 0.74 | 119255 | 120189,4 |  | 64 | 0.9 | 5 | 0.98 | 119158 | 120878 |
| 15 | 0.6 | 6 | 0.98 | 119752 | 120198,8 |  | 65 | 0.6 | 3 | 0.8 | 119822 | 120880 |
| 16 | 0.9 | 3 | 0.98 | 119344 | 120217,6 |  | 66 | 0.5 | 3 | 0.74 | 119338 | 120882,8 |
| 17 | 0.9 | 6 | 0.92 | 119132 | 120227,4 |  | 67 | 0.7 | 5 | 0.8 | 119811 | 120895,8 |
| 18 | 0.7 | 5 | 0.98 | 119368 | 120235,4 |  | 68 | 0.8 | 4 | 0.98 | 119504 | 120906,4 |
| 19 | 0.5 | 4 | 0.8 | 119681 | 120245 |  | 69 | 0.6 | 5 | 0.86 | 119954 | 120914,4 |
| 20 | 0.6 | 5 | 0.8 | 119175 | 120252,4 |  | 70 | 0.9 | 4 | 0.98 | 119591 | 120929,4 |
| 21 | 0.6 | 6 | 0.92 | 119605 | 120256 |  | 71 | 0.7 | 5 | 0.86 | 119033 | 120930,4 |
| 22 | 0.7 | 5 | 0.74 | 119130 | 120273,8 |  | 72 | 0.9 | 6 | 0.86 | 120118 | 120950,4 |
| 23 | 0.5 | 5 | 0.8 | 119589 | 120315,6 |  | 73 | 0.5 | 5 | 0.92 | 119106 | 120979 |
| 24 | 0.5 | 5 | 0.98 | 119928 | 120340,6 |  | 74 | 0.9 | 6 | 0.98 | 119936 | 120982,2 |
| 25 | 0.8 | 4 | 0.74 | 119141 | 120349,6 |  | 75 | 0.7 | 4 | 0.74 | 119584 | 120996,8 |
| 26 | 0.9 | 5 | 0.86 | 119610 | 120355 |  | 76 | 0.6 | 4 | 0.86 | 119882 | 121011,2 |
| 27 | 0.8 | 6 | 0.8 | 119377 | 120356,2 |  | 77 | 0.6 | 3 | 0.74 | 119522 | 121017 |
| 28 | 0.7 | 4 | 0.86 | 119894 | 120357,6 |  | 78 | 0.8 | 5 | 0.8 | 120041 | 121020,4 |
| 29 | 0.5 | 5 | 0.74 | 119666 | 120366 |  | 79 | 0.7 | 6 | 0.98 | 120490 | 121027 |
| 30 | 0.6 | 5 | 0.92 | 119262 | 120378,4 |  | 80 | 0.5 | 6 | 0.8 | 120295 | 121033,4 |
| 31 | 0.9 | 3 | 0.86 | 119719 | 120388,6 |  | 81 | 0.7 | 4 | 0.98 | 119342 | 121033,6 |
| 32 | 0.6 | 6 | 0.8 | 119584 | 120398 |  | 82 | 0.9 | 3 | 0.92 | 119675 | 121176,4 |
| 33 | 0.6 | 5 | 0.98 | 119345 | 120434 |  | 83 | 0.6 | 4 | 0.74 | 119433 | 121198 |
| 34 | 0.8 | 5 | 0.86 | 120254 | 120451 |  | 84 | 0.9 | 3 | 0.74 | 119939 | 121248,6 |
| 35 | 0.7 | 6 | 0.8 | 120107 | 120465,4 |  | 85 | 0.5 | 3 | 0.92 | 120767 | 121308 |
| 36 | 0.7 | 3 | 0.86 | 119924 | 120468,6 |  | 86 | 0.8 | 3 | 0.8 | 120049 | 121330,6 |
| 37 | 0.8 | 5 | 0.74 | 119819 | 120470 |  | 87 | 0.9 | 4 | 0.86 | 120432 | 121331,4 |
| 38 | 0.9 | 5 | 0.8 | 119831 | 120470 |  | 88 | 0.8 | 3 | 0.92 | 119654 | 121447,4 |
| 39 | 0.8 | 5 | 0.98 | 119716 | 120481,2 |  | 89 | 0.8 | 4 | 0.86 | 120173 | 121484,8 |
| 40 | 0.9 | 6 | 0.8 | 119955 | 120488,6 |  | 90 | 0.5 | 2 | 0.8 | 120059 | 121494,2 |
| 41 | 0.6 | 4 | 0.8 | 120092 | 120509,6 |  | 91 | 0.9 | 5 | 0.74 | 120450 | 121523 |
| 42 | 0.7 | 4 | 0.92 | 119638 | 120521 |  | 92 | 0.8 | 4 | 0.8 | 119396 | 121540,4 |
| 43 | 0.9 | 5 | 0.92 | 119230 | 120535,8 |  | 93 | 0.7 | 6 | 0.74 | 119568 | 121592,6 |
| 44 | 0.5 | 4 | 0.86 | 119716 | 120554,6 |  | 94 | 0.9 | 4 | 0.74 | 120250 | 121617,2 |
| 45 | 0.6 | 6 | 0.86 | 119542 | 120562,8 |  | 95 | 0.5 | 2 | 0.92 | 119781 | 121681,6 |
| 46 | 0.5 | 4 | 0.98 | 119446 | 120598,8 |  | 96 | 0.9 | 2 | 0.92 | 120594 | 121717,2 |
| 47 | 0.7 | 6 | 0.86 | 120019 | 120616,4 |  | 97 | 0.7 | 3 | 0.98 | 119553 | 121773,4 |
| 48 | 0.8 | 6 | 0.98 | 119297 | 120623 |  | 98 | 0.8 | 2 | 0.74 | 120571 | 121890,2 |
| 49 | 0.7 | 5 | 0.92 | 119838 | 120629 |  | 99 | 0.8 | 2 | 0.92 | 120926 | 121981,6 |
| 50 | 0.8 | 4 | 0.92 | 119706 | 120639,4 |  | 100 | 0.5 | 3 | 0.86 | 119747 | 122051 |
| 101 | 0.7 | 2 | 0.74 | 119230 | 122073,2 |  | 114 | 0.7 | 2 | 0.98 | 119817 | 122699,8 |
| 102 | 0.9 | 2 | 0.8 | 120150 | 122085,2 |  | 115 | 0.6 | 2 | 0.74 | 120729 | 122723,8 |
| 103 | 0.9 | 6 | 0.74 | 120197 | 122095,2 |  | 116 | 0.7 | 3 | 0.8 | 120229 | 122900 |
| 104 | 0.9 | 2 | 0.86 | 119762 | 122146,8 |  | 117 | 0.8 | 2 | 0.98 | 120975 | 122960,4 |
| 105 | 0.5 | 2 | 0.86 | 120125 | 122213 |  | 118 | 0.7 | 2 | 0.8 | 120782 | 123145,8 |
| 106 | 0.9 | 2 | 0.98 | 120815 | 122319,2 |  | 119 | 0.6 | 2 | 0.86 | 120189 | 123169,2 |
| 107 | 0.6 | 3 | 0.86 | 119474 | 122398,4 |  | 120 | 0.8 | 2 | 0.86 | 121477 | 123361,8 |
| 108 | 0.6 | 2 | 0.98 | 121209 | 122442 |  | 121 | 0.6 | 2 | 0.92 | 121339 | 123645,4 |
| 109 | 0.8 | 3 | 0.74 | 119783 | 122455,8 |  | 122 | 0.5 | 2 | 0.98 | 121935 | 123793,8 |
| 110 | 0.7 | 2 | 0.92 | 121338 | 122471,6 |  | 123 | 0.9 | 2 | 0.74 | 121402 | 124083,4 |
| 111 | 0.7 | 2 | 0.86 | 120139 | 122496,2 |  | 124 | 0.9 | 3 | 0.8 | 120650 | 124594 |
| 112 | 0.6 | 2 | 0.8 | 120505 | 122538,6 |  | 125 | 0.8 | 2 | 0.8 | 122317 | 124983 |
| 113 | 0.5 | 2 | 0.74 | 120346 | 122661,8 |  |  |  |  |  |  |  |

Postanowiliśmy także pokazać osobno wpływ parametru β przy stałych wartościach φ i q0. Testy także zostały wykonane dla instancji bier127.

# 5. Wnioski

Zauważyliśmy, że największy wpływ na jakość otrzymywanych wyników ma liczba kolonii (iteracji). Nie wynika to bezpośrednio z danych przedstawionych na wykresach ponieważ wykresy były przeważnie wykonywane dla maksymalnej liczby kolonii jaka mieściła się w zadanym czasie 3 minut. Potwierdza to jednak otrzymanie znacznie lepszego wyniku dla instancji tsp1000 po przekroczeniu limitu czasowego oraz pogorszenie wyników dla instancji bier127 w teście parametrów (gdzie liczba kolonii została zmniejszona ośmiokrotnie. W przypadku innych parametrów, czy to wybierając optymalne na bazie kilku testów i ich obserwacji czy poprzez sprawdzenie wielu kombinacji zauważyliśmy, że wybór "parametrów optymalnych" oznacza w przypadku naszego algorytmu optymalizację średniego wyniku lub optymalizację wyniku najlepszego, który nie zawsze jest powtarzalny. Dzieje się tak dlatego, ponieważ działanie naszego algorytmu nie jest w pełni niezależne od czynników losowych. Mimo to jesteśmy zadowoleni z większości otrzymanych wyników.

# 6. 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