

Bartosz Durys 229869 229869@edu.p.lodz.pl
Szymon Klewicki 229911 229911@edu.p.lodz.pl

Zadanie 4 - Algorytm mrówkowy

1. Cel

Celem zadania była implementacja algorytmu mrówkowego. Rozwiązanie powinno znajdować optymalną drogę odwiedzenia wszystkich atrakcji wesołym miasteczku. Należało również zbadać działanie programu dla różnych parametrów początkowych.

2. Działanie programu

Kod programu zawiera implementację algorytmu mrówkowego. Danymi wejściowymi są dwa pliki. Pierwszy z nich zawiera listę miejsc ze współrzędnymi w dwóch wymiarach. Drugi to parametry symulacji.

Dla każdej listy atrakcji i kombinacji ustawień wykonywana jest symulacja. Przebiega ona iteracyjnie w paru krokach:

1. Stworzenie kolonii mrówek.
2. Wybór początkowej atrakcji dla każdej z mrówek w kolonii.
3. Wybór następnych atrakcji.
4. Aktualizacja śladów feromonowych.
5. Wybór najlepszej mrówki.
6. Sprawdzanie warunków stopu algorytmu.

Warto dodać, że pierwsza atrakcja jest losowana. Przy następnych mamy sparametryzowane prawdopodobieństwo. Możemy wybrać kolejne miejsce też losowo lub selekcją ruletkową po obliczeniu prawdopodobieństwa każdej ścieżki. Do jej wyznaczenia wykorzystujemy:

- Intensywność pozostawionego feromonu na ścieżce.
- Odległość między miejscami.
- Parametr α - wpływ feromonu.
- Parametr β - wpływ odległości.

Pojedyncze prawdopodobieństwo można wyrazić równaniem:

$$p_{ij}^k = \frac{[f_{ij}(t)]^\alpha [\frac{1}{d_{ij}(t)}]^\beta}{\sum_{j_2 \in N_i^k} ([f_{ij_2}(t)]^\alpha [\frac{1}{d_{ij_2}(t)}]^\beta)}$$

Aktualizujemy ślady feromonowe w każdej iteracji jak wszystkie mrówki przejdą przez wszystkie atrakcje. Na początku musimy odparować dotychczasowe feromony przy użyciu parametru w konfiguracji. Następnie dodajemy feromony pozostawione przez mrówki w obecnej iteracji. Uwzględniamy tutaj całkowitą długość trasy przebytą przez każdą z mrówek:

$$f_{ij}(t+1) = (1-p)f_{ij}(t) + \sum_{k=1}^n \frac{1}{L_k}$$

Wybór najlepszej mrówki przy każdej iteracji potrzebny jest nam, żeby przedstawić jak wyglądały najkrótsze długości tras dla danego zestawu parametrów. Dodatkowo uwzględniamy też stagnację symulacji jako warunek stopu poza maksymalną liczbą iteracji.

3. Opis implementacji

W projekcie mamy dwa foldery *input* i *output*. W pierwszym umieszczamy pliki tekstowe ze współrzędnymi kolejnych atrakcji. Po przetworzeniu wszystkich symulacji otrzymujemy w drugim folderze statystyki w plikach JSON.

Do konfiguracji używamy też pliku *params.json*, w którym wpisujemy parametry:

- Liczba mrówek.
- Prawdopodobieństwo wyboru losowej drogi.
- Wpływ feromonu.
- Wpływ odległości.
- Maksymalna liczba iteracji - warunek stopu.
- Maksymalna liczba stagnacji - warunek stopu.
- Współczynnik wyparowania feromonów.

W projekcie stworzyliśmy parę klas:

- *Ant* - klasa mrówki. Posiada listy odwiedzonych miejsc i tych jeszcze do odwiedzenia. Jej metody oferują następujące funkcjonalności:
 - Wylosowanie następnego miejsca.
 - Obliczenie pokonanego dystansu.
 - Selekcja ruletkowa kolejnego miejsca na bazie obliczonych prawdopodobieństw.
 - Obliczenie prawdopodobieństwa wyboru pozostałych atrakcji.
- *Colony* - klasa kolonii mrówek. Tworzy mrówki, porusza nimi, aktualizuje feromony na ścieżce i wybiera najlepszą mrówkę.
- *StateMatrix* - klasa macierzy stanu. Obliczana jest macierz składająca się z długości dróg i intensywności feromonów. W trakcie symulacji aktualizowane są intensywności feromonów w obiekcie tej klasy.
- *FileManager* - klasa menadżera plików. Obsługuje wykorzystywane pliki i zapisuje statystyki symulacji.
- *SimStat* - klasa przechowująca statystyki symulacji.

Po uruchomieniu pliku *simulation.py* obserwujemy komunikaty z postępem każdej symulacji. Tworzone są w tym momencie pliki ze statystykami. Każdy z nich reprezentuje jedną przeprowadzoną symulację.

4. Metodyka

Mamy do przeanalizowania 6 plików reprezentujące inną listę atrakcji ze współrzędnymi. Uprościmy notację plików:

1. A-n32-k5 - A1.
2. A-n80-k10 - A2.
3. B-n31-k5 - B1.
4. B-n78-k10 - B2.
5. P-n16-k8 - P1.
6. P-n76-k5 - P2.

Dla każdego pliku stosujemy kombinację poszczególnych wartości parametrów:

- Liczba mrówek: 10, 30, 50.
- Prawdopodobieństwo wyboru losowej drogi: 0.3.
- Wpływ feromonu: 1, 2.
- Wpływ odległości: 1, 3.
- Maksymalna liczba iteracji: 1000.
- Maksymalna liczba stagnacji: 200.
- Współczynnik parowania feromonów: 0.1, 0.5.

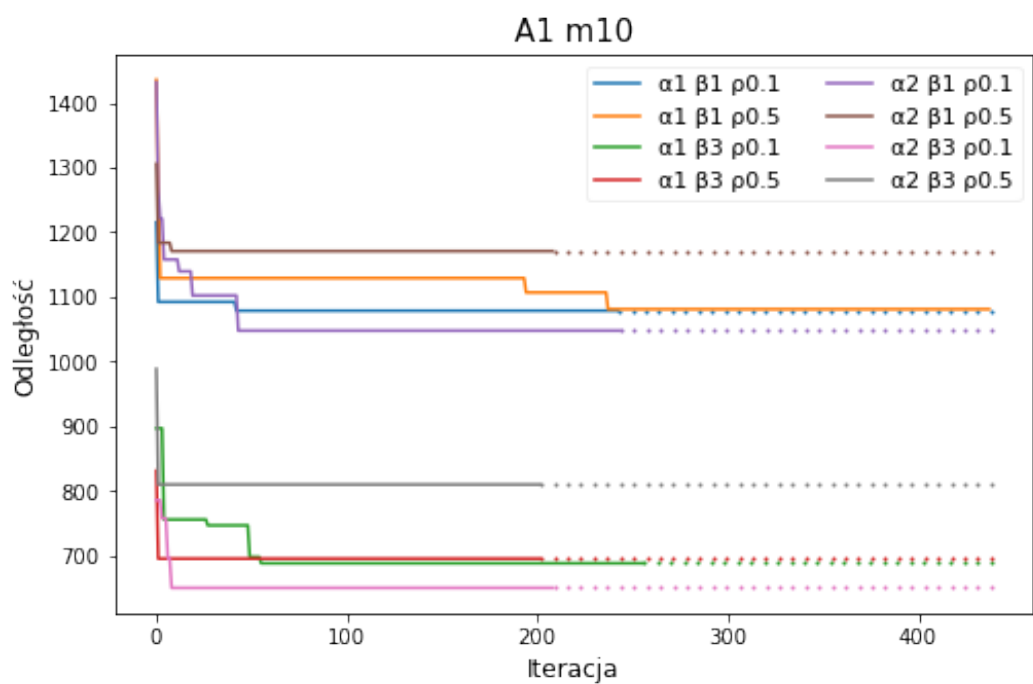
Jako że tylko cztery parametry mają więcej niż jedną możliwą wartość, to tylko je będziemy uwzględniać podczas analizy. Zastosujemy notację pojedynczych liter:

- Liczba mrówek: m
- Wpływ feromonu: α
- Wpływ odległości: β
- Współczynnik parowania feromonów: ρ

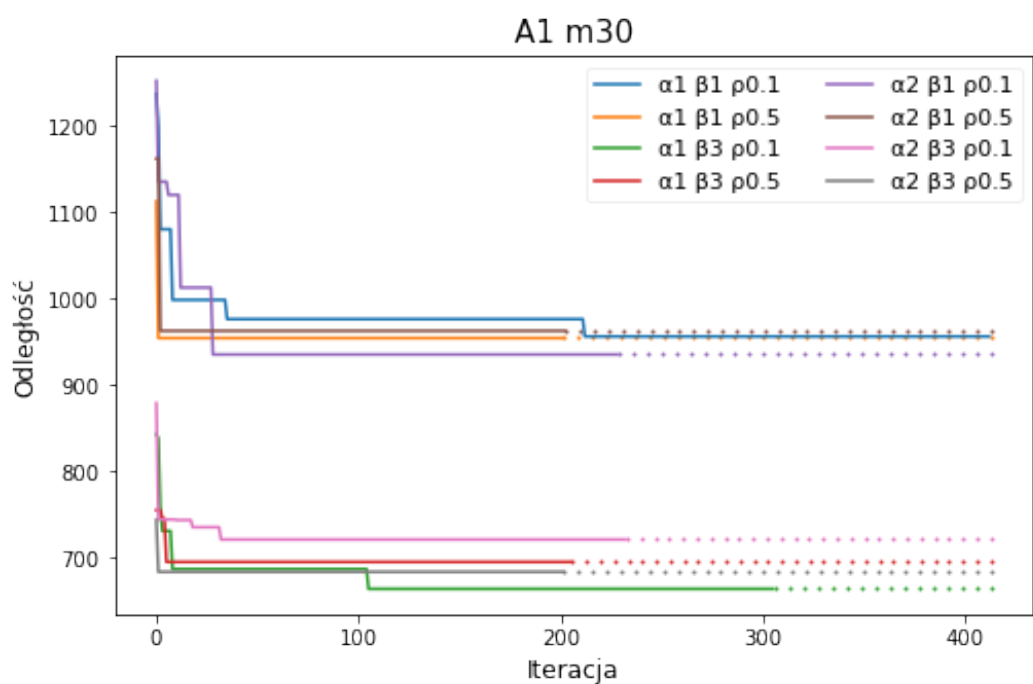
Także mamy $6 \cdot 24 = 144$ plików do przeanalizowania. Dlatego każdą kombinację pliku i liczby mrówek przedstawimy na osobnych wykresach. To nam da zaledwie 18 różnych wykresów - 3 dla każdego pliku. Na wykresach natomiast przedstawimy odległości pokonane przez najlepszą mrówkę w danej iteracji.

5. Wyniki

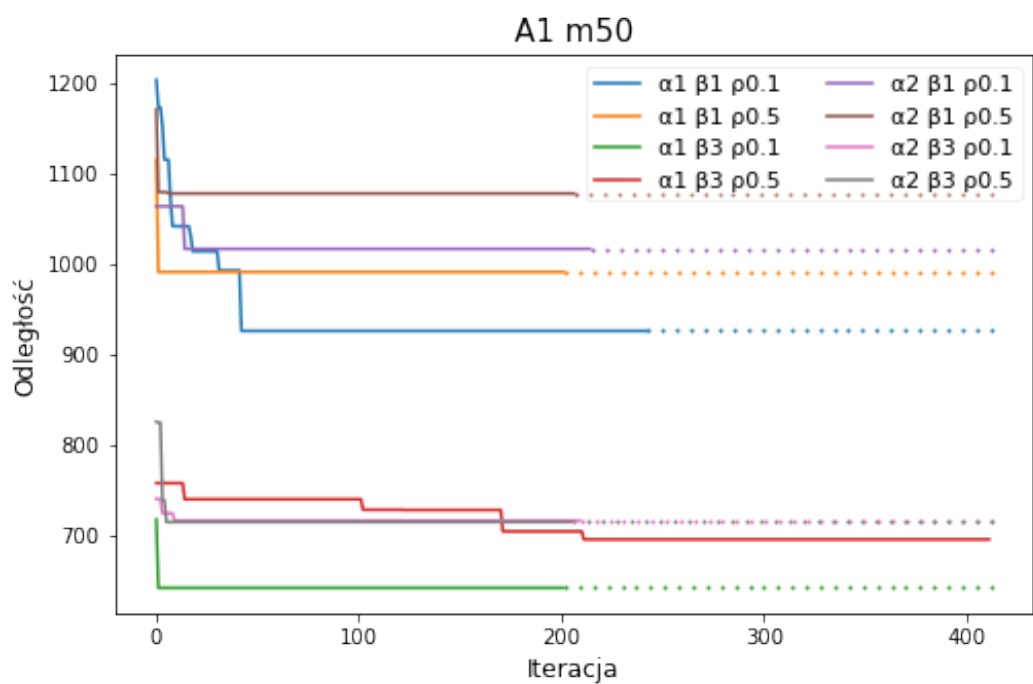
Tutaj zaprezentujemy nasze zmierzone wartości. Poziome linie kropek na wykresach to przedłużenie ostatniej iteracji symulacji, w której doszło do stagnacji.



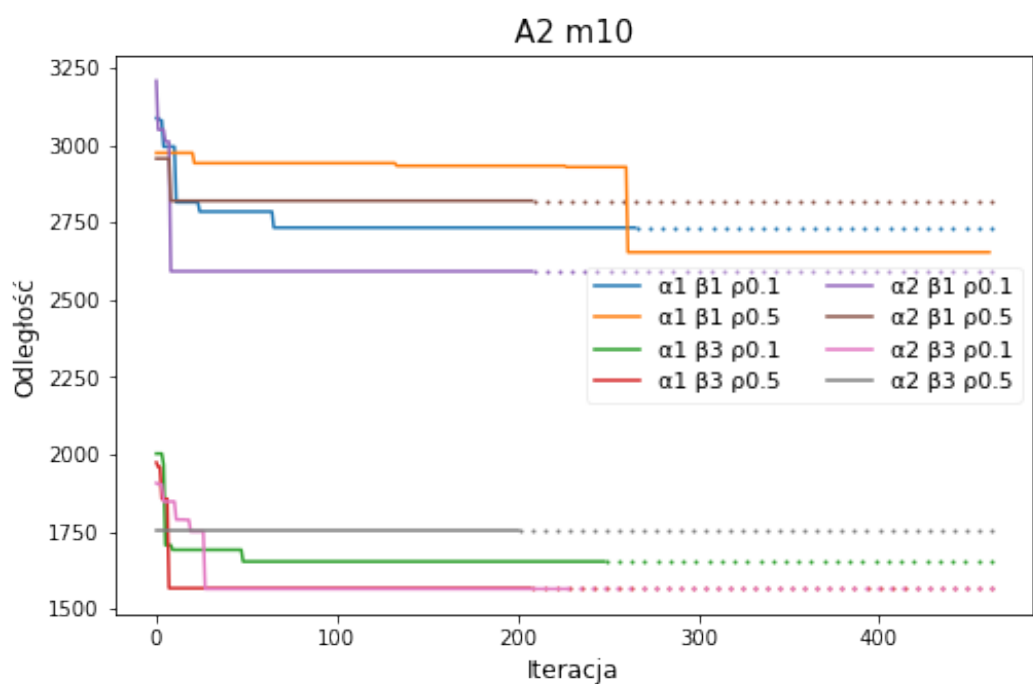
1.1. Najlepsze odległości w danej iteracji dla pliku A1 i 10 mrówek.



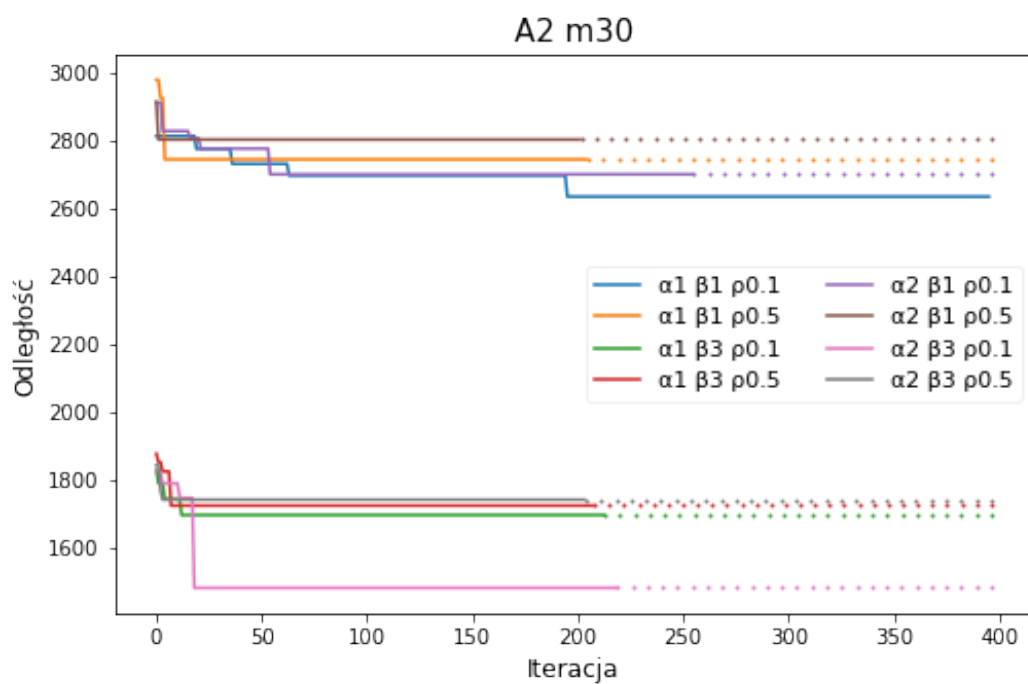
1.2. Najlepsze odległości w danej iteracji dla pliku A1 i 30 mrówek.



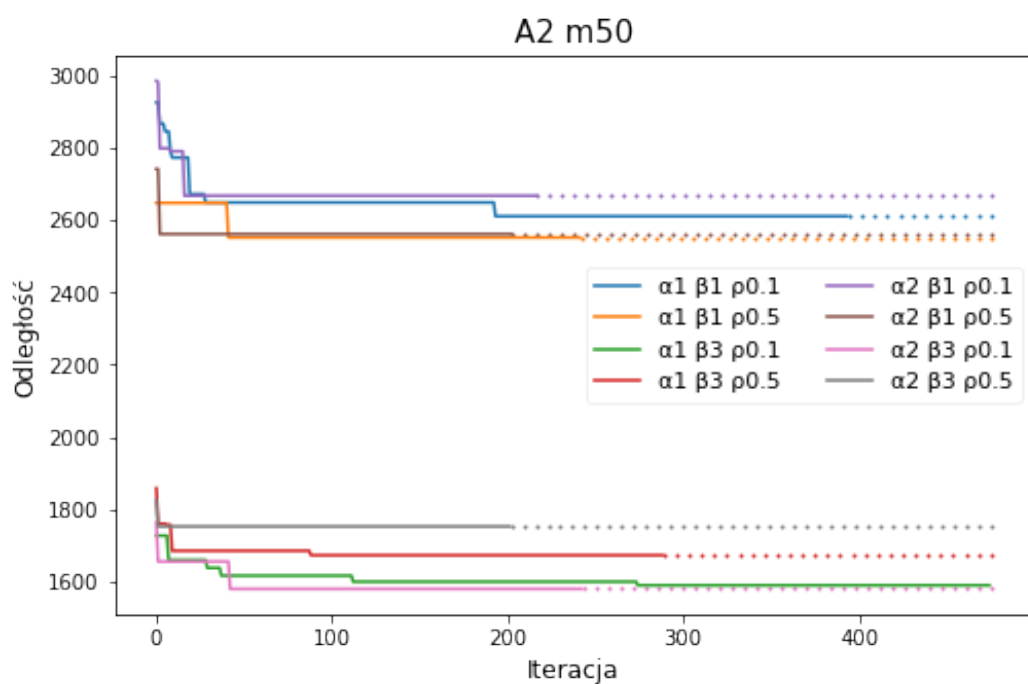
1.3. Najlepsze odległości w danej iteracji dla pliku A1 i 50 mrówek.



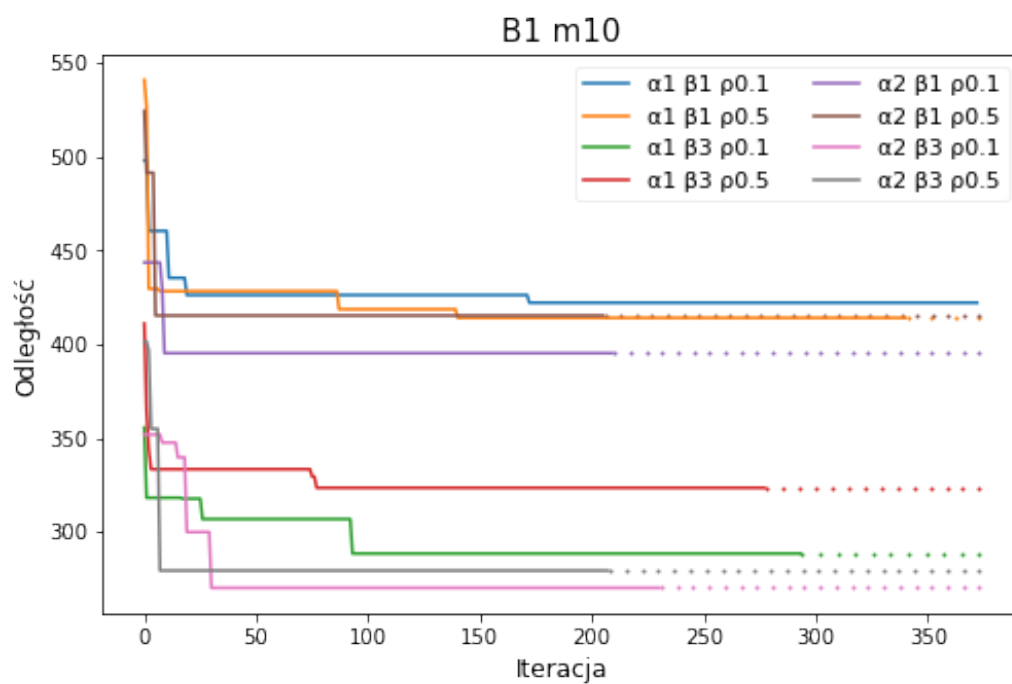
2.1. Najlepsze odległości w danej iteracji dla pliku A2 i 10 mrówek.



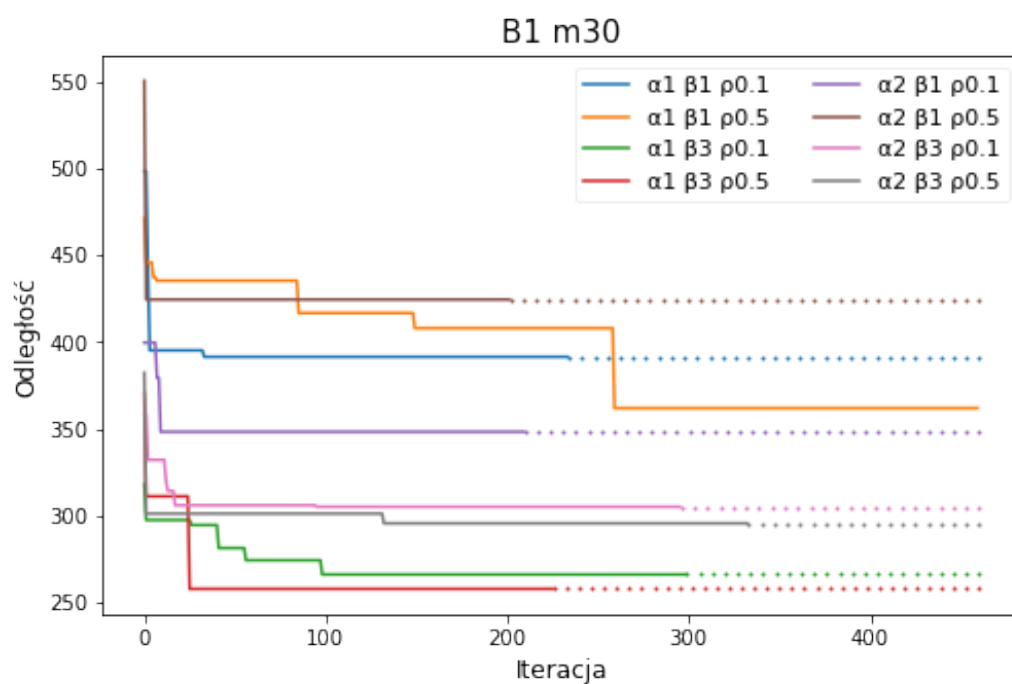
2.2. Najlepsze odległości w danej iteracji dla pliku A2 i 30 mrówek.



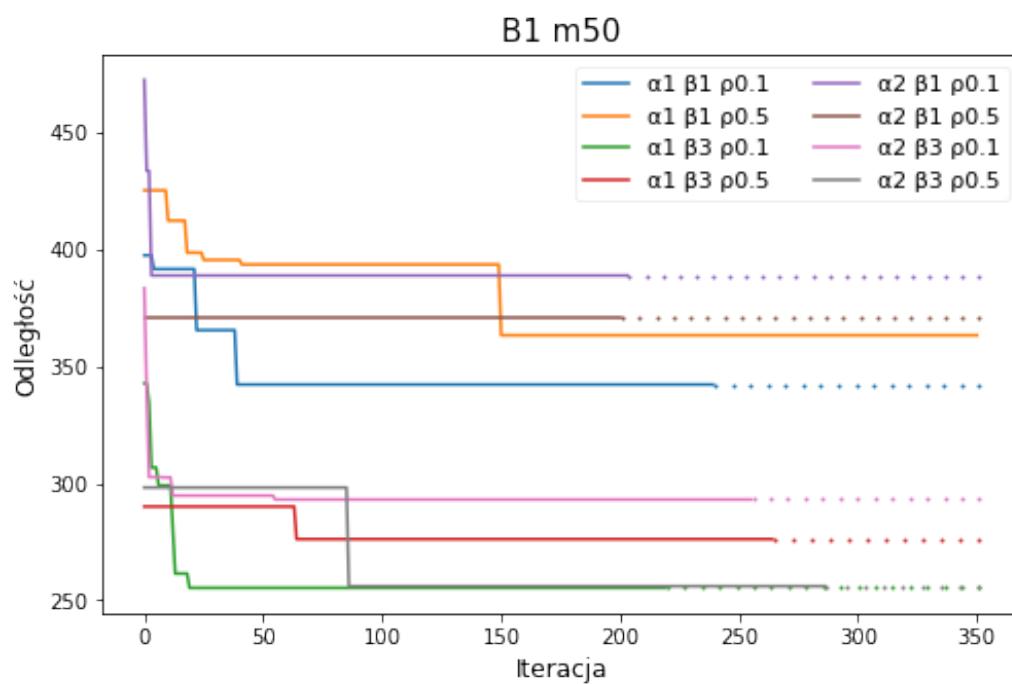
2.3. Najlepsze odległości w danej iteracji dla pliku A2 i 50 mrówek.



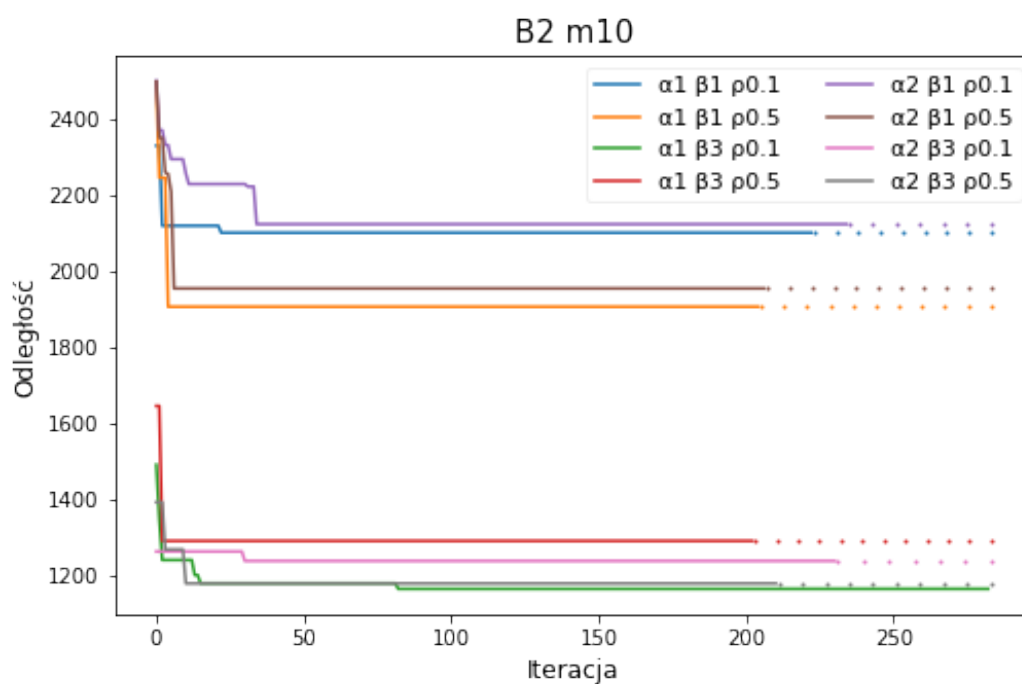
3.1. Najlepsze odległości w danej iteracji dla pliku B1 i 10 mrówek.



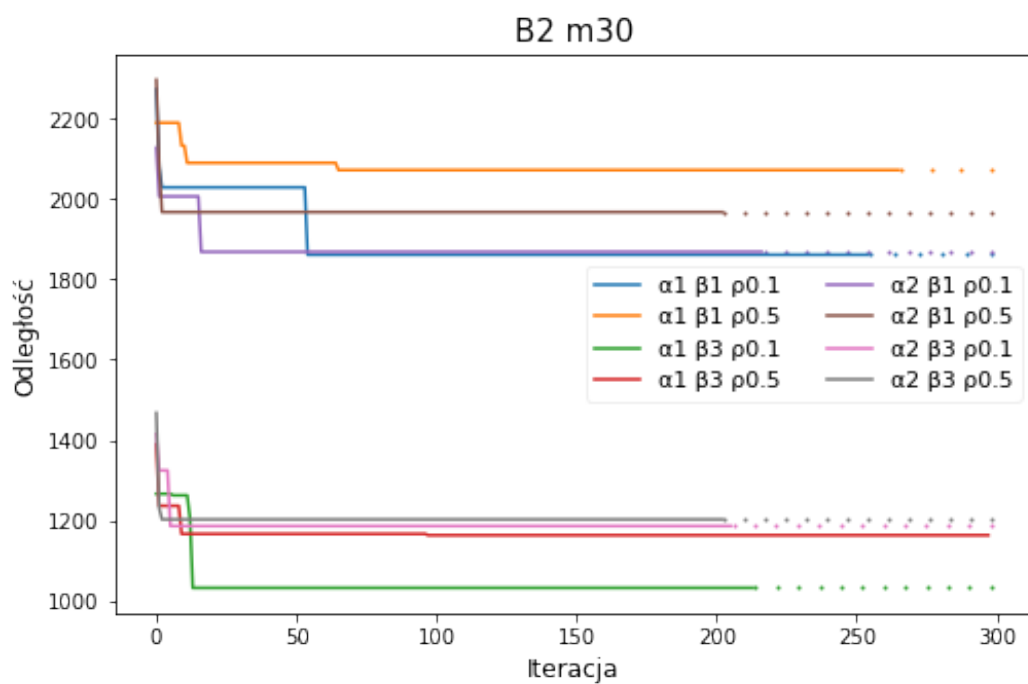
3.2. Najlepsze odległości w danej iteracji dla pliku B1 i 30 mrówek.



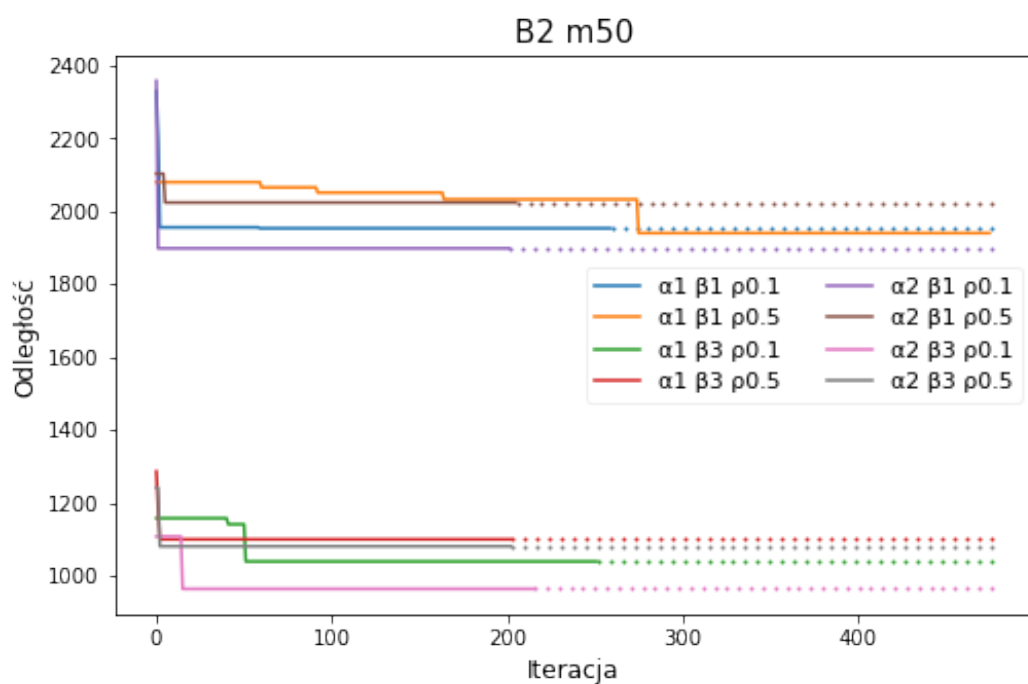
3.3. Najlepsze odległości w danej iteracji dla pliku B1 i 50 mrówek.



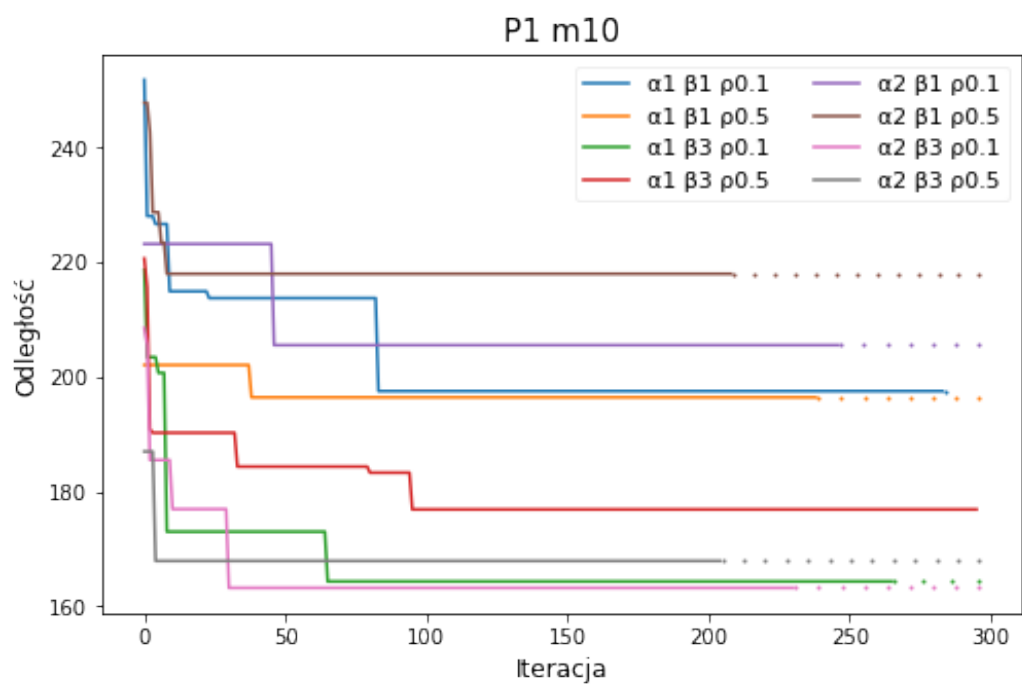
4.1. Najlepsze odległości w danej iteracji dla pliku B2 i 10 mrówek.



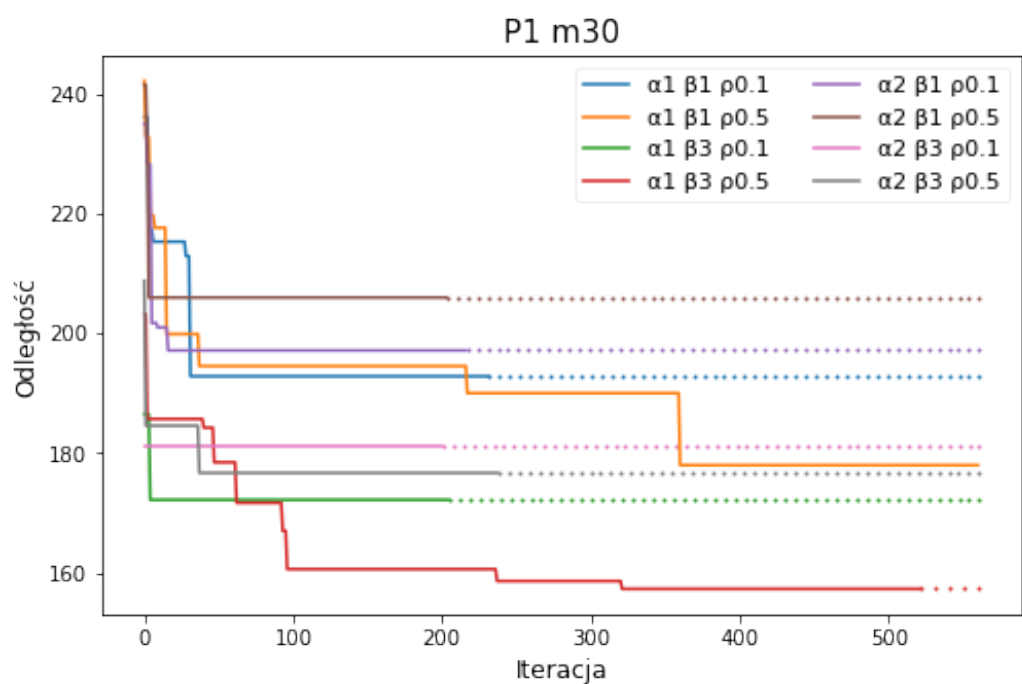
4.2. Najlepsze odległości w danej iteracji dla pliku B2 i 30 mrówek.



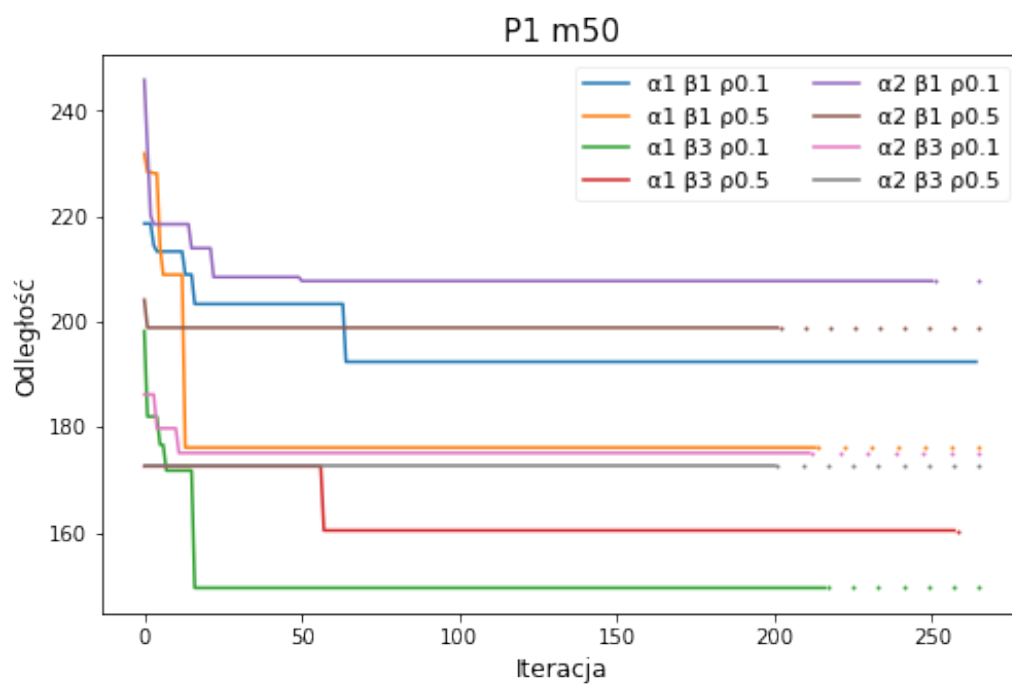
4.3. Najlepsze odległości w danej iteracji dla pliku B2 i 50 mrówek.



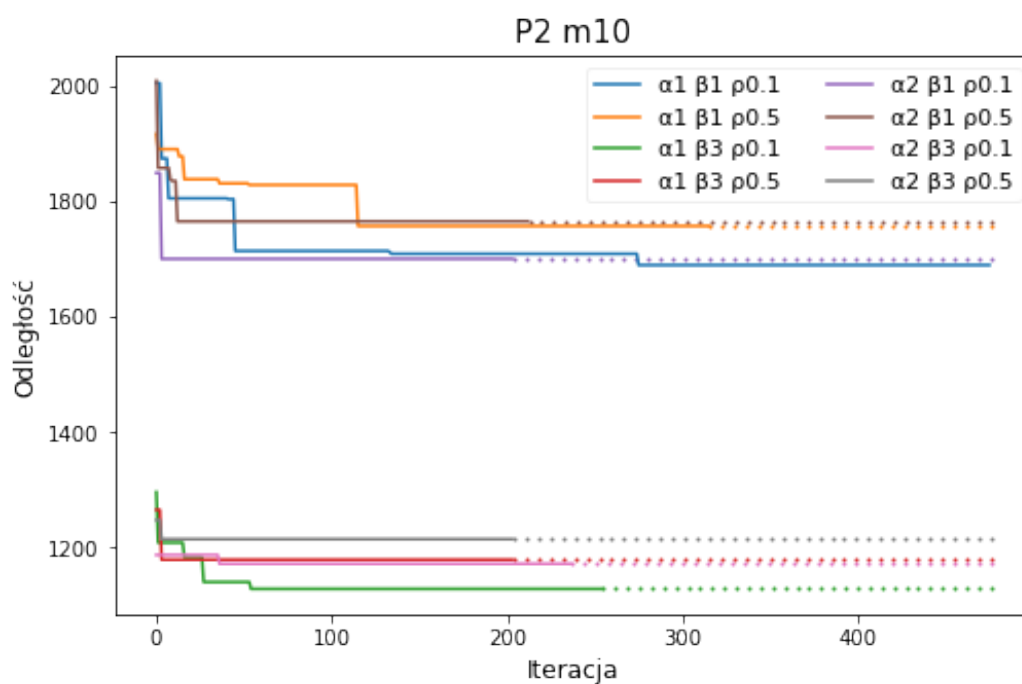
5.1. Najlepsze odległości w danej iteracji dla pliku P1 i 10 mrówek.



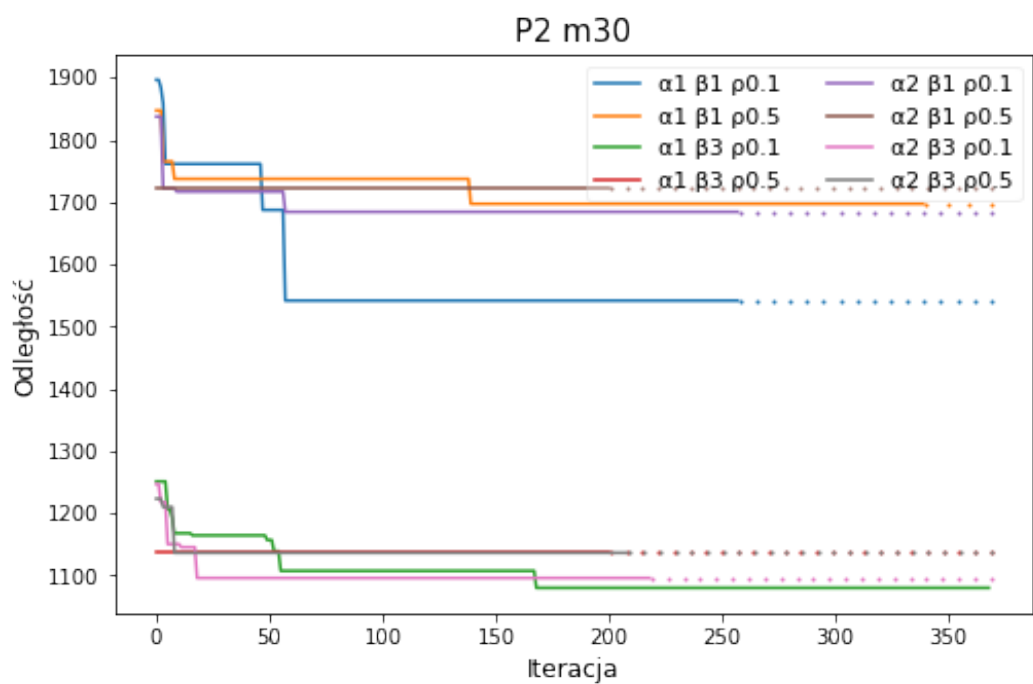
5.2. Najlepsze odległości w danej iteracji dla pliku P1 i 30 mrówek.



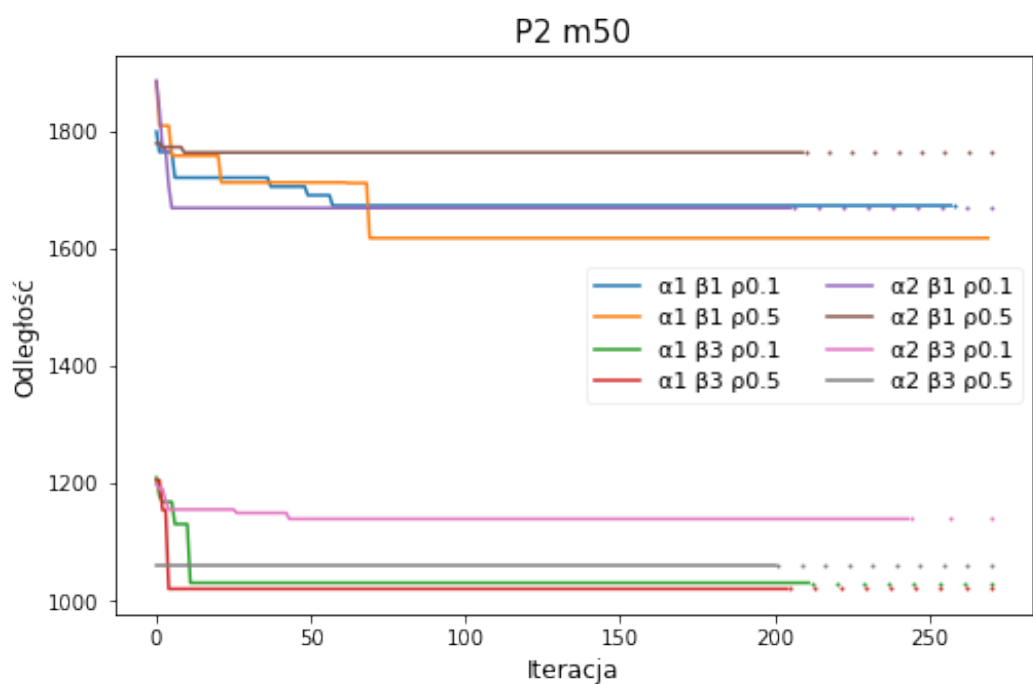
5.3. Najlepsze odległości w danej iteracji dla pliku P1 i 50 mrówek.



6.1. Najlepsze odległości w danej iteracji dla pliku P2 i 10 mrówek.



6.2. Najlepsze odległości w danej iteracji dla pliku P2 i 30 mrówek.



6.3. Najlepsze odległości w danej iteracji dla pliku P2 i 50 mrówek.

6. Analiza

Przy naszej analizie skupimy się na ocenie wpływu każdego z wyszczególnionych parametrów. Każdy kolejny akapit poświęcimy jednemu parametrowi w kolejności:

1. Liczba mrówek: m
2. Wpływ feromonu: α
3. Wpływ odległości: β
4. Współczynnik parowania feromonów: ρ

Rozpoczynamy od wpływu liczby mrówek. Spójrzmy na wykresy dla różnych wartości tego parametru. Dla przykładu wykresy 6.x. Możemy stwierdzić, że dla większych wartości potrzebujemy mniejszej liczby iteracji do znalezienia dobrego rozwiązania. Osiągamy przy tym też lepsze rezultaty niż przy symulacjach z mniejszą kolonią. Najlepsze odległości końcowe na wykresie 6.1. sięgają wartości bliskiej 1100. Na następnym jest lepiej. Przy 6.3 mają już lekko ponad 1000 jednostek.

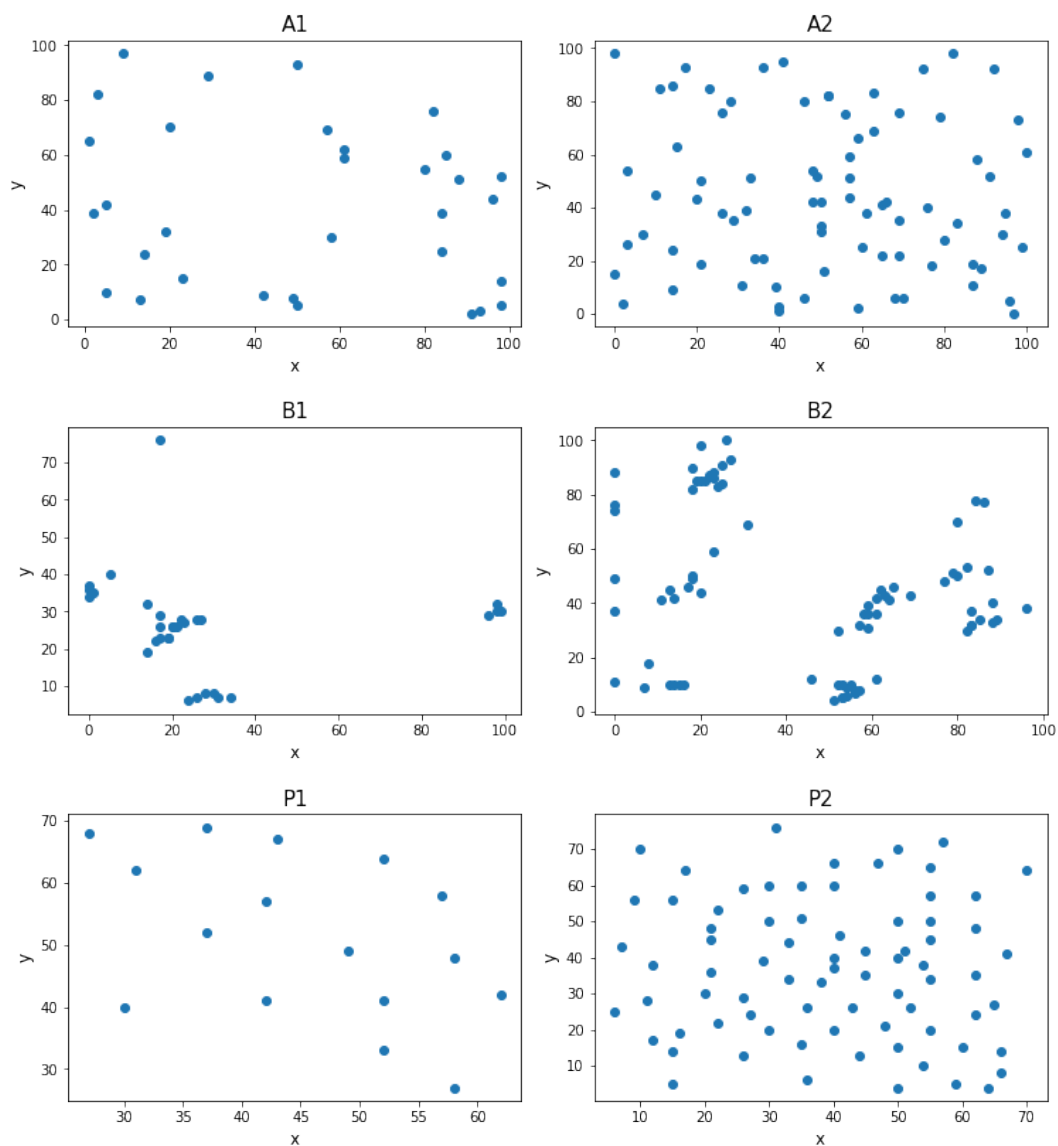
Parametr wpływu feromonu. Jest to ustawienie trudne do interpretowania i wykresy nie przedstawiają jednoznacznego wniosku. Mimo to, inny parametr - współczynnik parowania feromonów - pokazuje, że intensywność feromonów wpływa na działanie algorytmu. Do rozstrzygnięcia tej kwestii są potrzebne dalsze eksperymenty.

Wpływ odległości. Dla większości wykresów widzimy klarowny podział na dwie grupy. Każda z nich ma inną wartość tego parametru. Lepsze wyniki osiąga ta z większą wagą ustawienia. Najbardziej przedstawiają to wykresy 2.x. Stopień tego może potwierdzić to, że automatycznie została umieszczona legenda między tymi grupami.

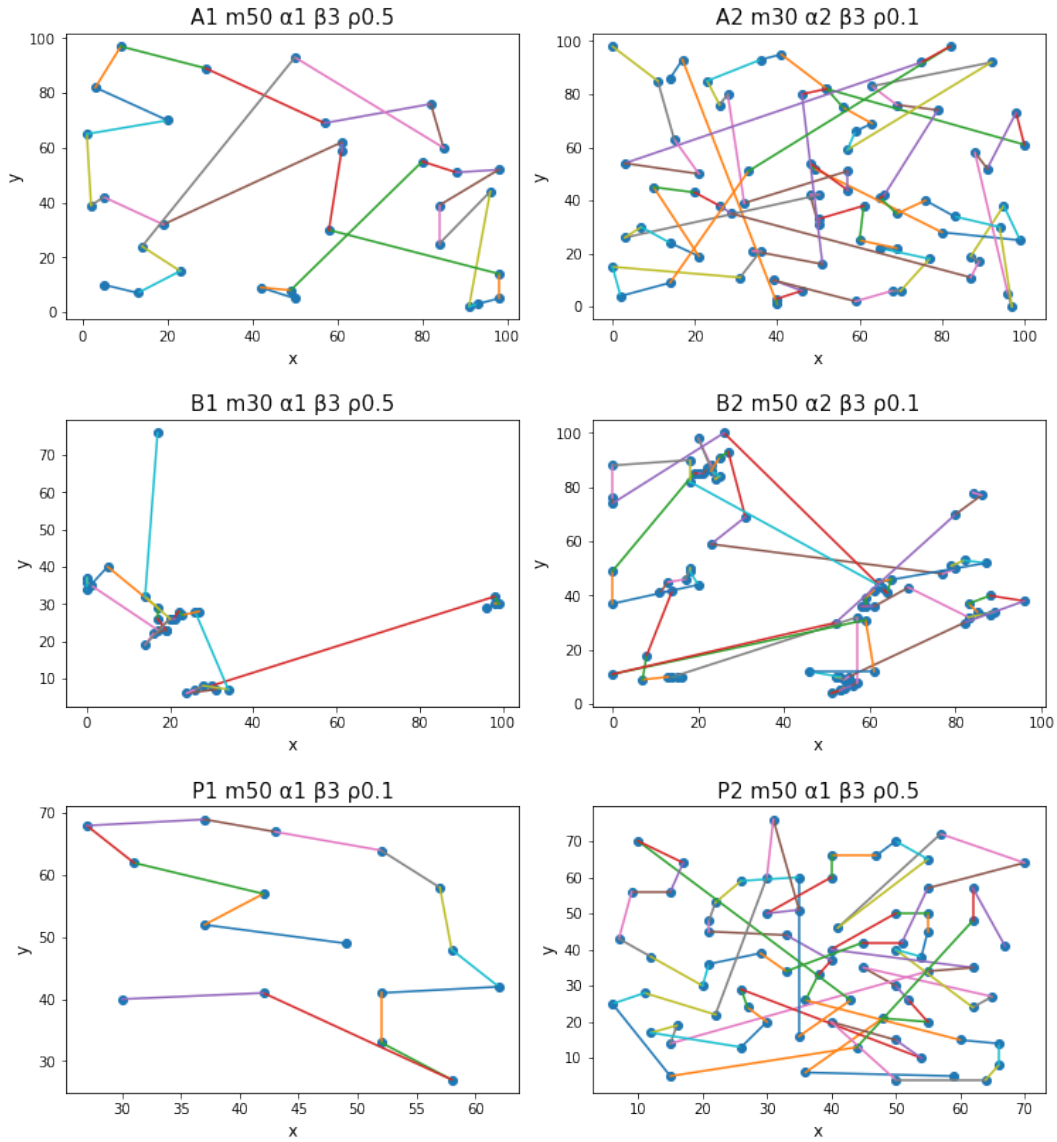
Został współczynnik parowania feromonów. Dla niego zauważamy lekką negatywną tendencję dla większych wartości. Taka sytuacja jest widoczna na przykład przy wykresach 1.x. po połączeniu w pary konfiguracji różniących się tylko tym parametrem. Prawdopodobnie intensywność feromonów zmniejsza się zbyt szybko w stosunku do uzupełniania jej przez mrówki.

7. Rysowanie drogi

Przedstawimy jeszcze jak wyglądało umiejscowienie atrakcji w poszczególnych plikach oraz wybrane najlepsze drogi.



7.1. Położenie atrakcji dla poszczególnych plików.



7.2. Wybrane drogi uzyskane w symulacji.

8. Wnioski

Po wykonaniu implementacji możemy stwierdzić, że algorytm nadaje się do szybkiego znajdowania w miarę optymalnej drogi. Narysowane parę z nich potwierdza taki stan rzeczy.

Ustaliliśmy, że większa liczba mrówek sprzyja lepszym rozwiązaniom dla mniejszej liczby iteracji. Wyższy współczynnik wpływu odległości między atrakcjami znacząco polepszył wyniki. Trudno to samo oznajmić w przypadku wagi feromonów. Do tego potrzebne są dalsze badania. Natomiast szybkie parowanie feromonów może wpłynąć negatywnie na otrzymanie lepszego rozwiązania.