POLITECHNIKA BYDGOSKA
Wydział Telekomunikacji, Informatyki i Elektrotechniki

Politechnika Bydgoska im. Jana i Jędrzeja Śniadeckich

Wydział Telekomunikacji, Informatyki i Elektrotechniki

Porównanie algorytmów optymalizacyjnych: Firefly vs Whale

Projekt

Przedmiot: **Sztuczna inteligencja**

Imię i nazwisko: Dominik Pszczółkowski, Mateusz Zakrent
Nr albumu: 122404, 122457
Prowadzący: dr hab. inż. Tomasz Talaśka, prof. PBŚ
Data oddania: 9.06.2025

Bydgoszcz, 9.06.2025

Spis treści

1 Cel zadania

Celem zadania było porównanie skuteczności dwóch metaheurystycznych algorytmów optymalizacyjnych – świetlika (Firefly Algorithm) oraz wieloryba (Whale Optimization Algorithm) – w kontekście problemu komiwojażera (TSP).

2 Rozwiązanie

Implementację wykonano w języku Python, używając środowiska PyCharm. Najpierw wygenerowano zestaw 25 punktów, a następnie zastosowano oba algorytmy do znalezienia najkrótszej możliwej ścieżki.

Oba algorytmy testowano na tych samych danych wejściowych, by zapewnić równe warunki porównania. Parametry można modyfikować w pliku `main.py`. Wynik działania programu pojawia się w konsoli (długość ścieżki) oraz w folderze `Dane` jako animowany plik `.gif`.

Przykładowe uruchomienie programu:

```
Załadowano punkty z pliku points.npy, shape: (25, 2)
[WOA]      Najlepszy dystans: 111.07796176240291
[WOA]      Liczba iteracji w historii: 20
✅ GIF zapisany jako dane/whale_0.20_111_20_40000.gif
```

3 Struktura projektu

- `Firefly.py` – algorytm świetlika
- `Whale.py` – algorytm wieloryba
- `Main.py` – plik główny
- `Plot_utils.py` – generowanie animacji (.gif)
- `Route_gen.py` – generowanie punktów
- `Points.npy` – zapisane punkty
- `Tsp_utils.py` – obliczenia tras

4 Algorytm świetlika

Algorytm świetlika to metaheurystyka inspirowana naturalnym zachowaniem świetlików, szczególnie ich zdolnością do emitowania światła w celu przyciągania partnerów.
W kontekście optymalizacji, każdy świetlik reprezentuje potencjalne rozwiązanie, a jego "jasność"odpowiada jakości tego rozwiązania.

Główne założenia algorytmu są następujące:

- Świetliki są uniseksualne — każdy może przyciągać innych.
- Jaśniejsze świetliki przyciągają ciemniejsze; przyciąganie maleje wraz z odległością.
- Jeśli nie ma jaśniejszych, świetlik porusza się losowo.

Wzór aktualizacji pozycji świetlika:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \beta \exp\left[-\gamma r_{ij}^2\right] (\mathbf{x}_j^t - \mathbf{x}_i^t) + \alpha_t \boldsymbol{\epsilon}_t$$

- $\mathbf{x}_i^t$ – pozycja świetlika i w iteracji t
- β_0 – początkowa atrakcyjność
- γ – współczynnik tłumienia światła
- r_{ij} – odległość między świetlikami i i j
- α – współczynnik losowości
- ϵ – wektor losowy

5 Algorytm wieloryba

Jest to metaheurystyka inspirowana strategią polowania humbaków, znaną jako "bubble-net feeding". W tej technice wieloryby tworzą spiralne bąbelki, aby otoczyć i schwytać swoją zdobycz.
WOA modeluje to zachowanie, aby efektywnie przeszukiwać przestrzeń rozwiązań w problemach optymalizacyjnych.

Algorytm WOA składa się z trzech głównych mechanizmów:

1. **Otaczanie zdobyczy:** agent aktualizuje pozycję względem najlepszego rozwiązania.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right|$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}$$

   - $\vec{X}^*(t)$ to pozycja najlepszego rozwiązania w iteracji t
   - $\vec{X}(t)$ to obecna pozycja wieloryba
   - $\vec{A}$ i $\vec{C}$ to wektory współczynników

   Wektory te obliczane są w następujący sposób:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \quad \text{oraz} \quad \vec{C} = 2 \cdot \vec{r}$$

   - Składowe wektora $\vec{a}$ liniowo maleją od 2 do 0 w trakcie iteracji.
   - r_1 oraz r_2 to losowe wektory z zakresu $[0, 1]$.

2. **Spiralny ruch:** modelowanie spiralnego pływania wokół celu.

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t)$$

$$\vec{D}' = \left| \vec{X}^*(t) - \vec{X}(t) \right|$$

   - $\vec{D}'$ to odległość między wielorybem a zdobyczą
   - b to stała definiująca kształt spirali
   - l to losowa liczba z zakresu $[-1, 1]$

3. **Losowe poszukiwanie:** gdy $|A| > 1$, agent przeszukuje przestrzeń globalnie.

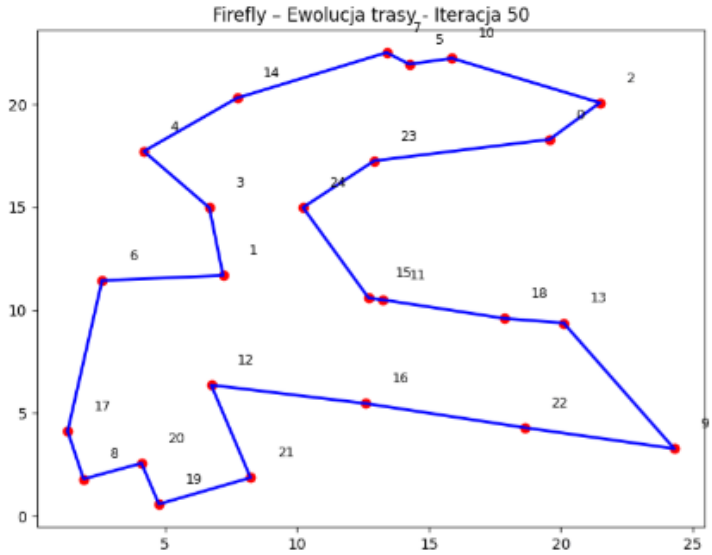
$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}(t) \right|$$

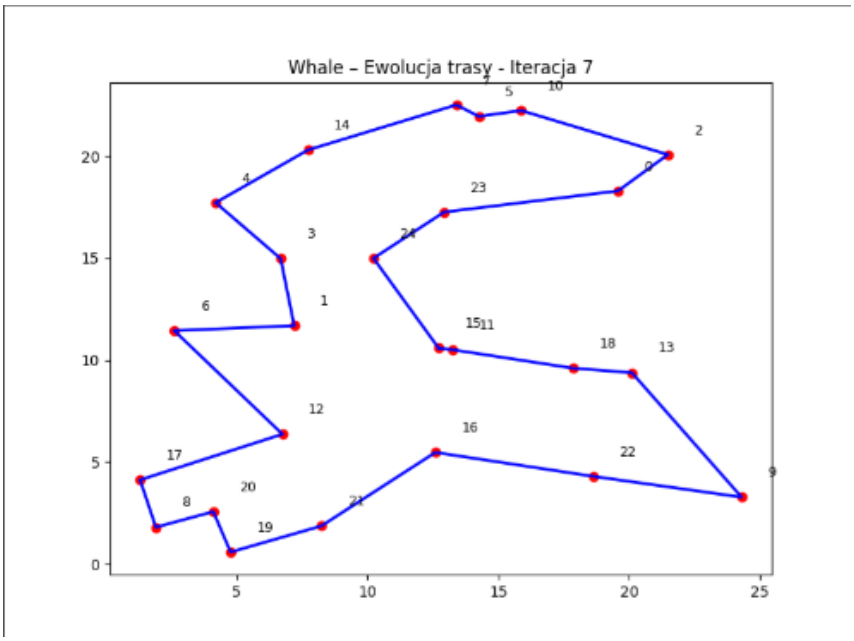
$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D}$$

   - $\vec{X}_{\text{rand}}$ to losowo wybrana pozycja z obecnej populacji.

6 Porównanie wyników

Najlepszy wynik dla tego zestawu punktów, jaki udało się osiągnąć, to dystans o wartości 104. Udało się go osiągnąć obu algorytmom, jednak świetlikowi wychodzi to o wiele łatwiej.





Jak widać na wykresach są to potencjalnie realistyczne najlepsze trasy, intuicyjne dla ludzkiego oka.

```
Załadowano punkty z pliku points.npy, shape: (25, 2)
[Firefly] Najlepszy dystans: 109.58450508781213
[Firefly] Liczba iteracji w historii: 50
✅ GIF zapisany jako dane/firefly_0.4,1,0.1_109_50_300.gif
[Firefly] Najlepszy dystans: 104.05081225577364
[Firefly] Liczba iteracji w historii: 50
✅ GIF zapisany jako dane/firefly_0.4,1,0.1_104_50_300.gif
[Firefly] Najlepszy dystans: 107.16063947666999
[Firefly] Liczba iteracji w historii: 50
✅ GIF zapisany jako dane/firefly_0.4,1,0.1_107_50_300.gif
Firefly Done
[WOA]      Najlepszy dystans: 198.26781013148508
[WOA]      Liczba iteracji w historii: 50
✅ GIF zapisany jako dane/whale_0.19_198_50_300.gif
[WOA]      Najlepszy dystans: 189.57363232673745
[WOA]      Liczba iteracji w historii: 50
✅ GIF zapisany jako dane/whale_0.18_189_50_300.gif
[WOA]      Najlepszy dystans: 179.16046673904327
[WOA]      Liczba iteracji w historii: 50
✅ GIF zapisany jako dane/whale_0.18_179_50_300.gif
Whale Done
```

Przy ustawieniu ilości agentów optymalnej dla świetlika, wieloryb osiąga o wiele gorsze wyniki.

```
Załadowano punkty z pliku points.npy, shape: (25, 2)
[WOA]      Najlepszy dystans: 113.49303603250516
[WOA]      Liczba iteracji w historii: 15
✅ GIF zapisany jako dane/whale_0.17_113_15_40000.gif
[WOA]      Najlepszy dystans: 115.01465576951742
[WOA]      Liczba iteracji w historii: 15
✅ GIF zapisany jako dane/whale_0.25_115_15_40000.gif
[WOA]      Najlepszy dystans: 131.07941786880218
[WOA]      Liczba iteracji w historii: 15
✅ GIF zapisany jako dane/whale_0.17_131_15_40000.gif
[WOA]      Najlepszy dystans: 119.99046399188057
[WOA]      Liczba iteracji w historii: 15
✅ GIF zapisany jako dane/whale_0.21_119_15_40000.gif
[WOA]      Najlepszy dystans: 110.29181575563602
[WOA]      Liczba iteracji w historii: 15
✅ GIF zapisany jako dane/whale_0.17_110_15_40000.gif
Whale Done
```

Wieloryb z bardziej dostosowanymi parametrami.

Świetlik osiąga lepsze wyniki przy małej ilości agentów (200-300 daje wyniki prawie idealne), wymaga 50-60 iteracji do pełnej interpretacji trasy.
Dzieje się to kosztem jednak złożoności obliczeniowej, czyli dłuższym obliczeniom względem algorytmu wieloryba.

Wieloryb jest prostszym algorytmem, który bardzo szybko znajduje trasy, gdyż jest to głównie uzależnione od parametrów startowych.
Z tego względu wymaga jedynie 10-20 iteracji do pełnych obliczeń. Natomiast agentów im więcej tym lepiej: przy 20000 – 40000 (ze sporą losowością) osiąga wyniki o około 10-15% gorsze od świetlika.
Warto zaznaczyć że nadal przy tak ogromnej ilości agentów wykonuje się dużo szybciej.

6.1 Optymalne parametry

Firefly:

- Iteracje: 60
- Agenci: 300
- $\alpha = 0.4$, $\beta = 1$, $\gamma = 0.1$

Whale:

- Iteracje: 20
- Agenci: 20,000
- $b = 0.2$

Są to optymalne parametry dla obu algorytmów, jakie udało nam się znaleźć dla wygenerowanego zestawu punktów.

7 Wnioski

Algorytm Świetlika jest bardziej skomplikowany obliczeniowo, ale osiąga lepsze wyniki, jeżeli mamy na nie potrzebę i czas. Alternatywnie algorytm wieloryba jest bardzo szybki do potencjalnego zastosowania w sytuacjach gdzie zależy nam jedynie na zbliżonych wynikach.

8 Źródła

- Firefly Algorithm: `https://en.wikipedia.org/wiki/Firefly_algorithm`
- Whale Optimization Algorithm: `https://en.wikiversity.org/wiki/Whale_Optimization_Algorithm`