

Sprawozdanie - Algorytmy inspirowane naturą w planowaniu trasy

Małgorzata Makiela, Kacper Duda, Katarzyna Żuchowska

15.06.2024

1 Wstęp

Algorytmy inspirowane naturą to algorytmy, które naśladują naturalne procesy w celu rozwiązywania skomplikowanych problemów obliczeniowych, od kilku lat cieszą się rosnącym zainteresowaniem. W naszej pracy skupiliśmy się na algorytmie ACO (ang. Ant Colony Optimization).

Ten rodzaj algorytmu został wykorzystany w naszym projekcie do wyznaczania optymalnej trasy między dwoma punktami na mapie, co ma zastosowanie w wielu dziedzinach, począwszy od logistyki i transportu, poprzez robotykę, aż po planowanie tras w sieciach telekomunikacyjnych lub sieciach dostarczania energii.

2 Metoda ACO

Algorytm ACO, znany również jako Optymalizacja Kolonii Mrówek, to metaheurystyka inspirowana naturalnym zachowaniem mrówek w ekosystemie. Mrówki, poszukując jedzenia, zaczynają od losowego wyboru drogi. Podczas powrotu do gniazda z pokarmem, każda mrówka zostawia na swojej ścieżce feromon - substancję, która służy do komunikacji i której ślad jest wyczuwalny przez inne mrówki.

Gdy inne mrówki napotkają na te ślady, zaczynają podążać najbardziej "zamarkowanymi" ścieżkami, jednocześnie zwiększając ich atrakcyjność poprzez dodanie więcej feromonu. W rezultacie, krótkie ścieżki są zazwyczaj mocno "zamarkowane", co sprawia, że stają się one atrakcyjniejsze dla innych mrówek.

Algorytm ACO wykorzystuje tę ideę, stosując analogiczny proces dla wyznaczania optymalnych tras. Poszczególne "mrówki" są w stanie poruszać się po własnych ścieżkach, zaczynając od losowo wybranego punktu. Podobnie jak prawdziwe mrówki, agentami zostawiają "feromon" na ścieżkach, które wydają

się najbardziej efektywne lub obiecujące.

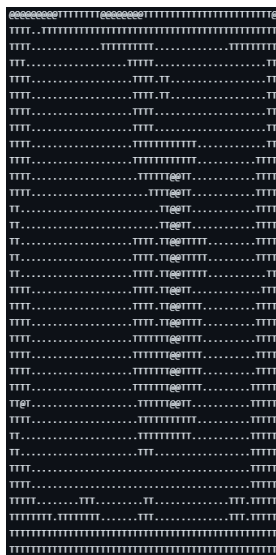
Główne cechy algorytmu ACO obejmują zdolność do samodzielnego uczenia się na podstawie doświadczeń oraz zdolność do adaptacji do zmieniających się warunków. Są one możliwe dzięki kluczowym elementom algorytmu, takim jak ewaporacja feromonu (czyli stopniowe zanikanie "atrakcyjności" ścieżki) oraz zdolność do eksploracji nowych ścieżek.

3 Dane wejściowe

W projekcie wykorzystaliśmy dane ze strony: <https://www.movingai.com/benchmarks/grids.html>.

Pobraliśmy z niej mapy, czyli modele terenu złożone z regularnej siatki komórek, gdzie każda z nich reprezentuje przeszkodę lub teren otwarty.

Dodatkowo do każdej mapy pobraliśmy scenariusz, czyli zestaw danych zawierające odległości między konkretnymi punktami na mapach.



Rysunek 1: Przykładowy wygląd mapy, znak kropki "." na mapie oznacza brak przeszkody, pozostałe oznaczają przeszkodę

0	den009d.map	50	34	7	10	7	13	3.00000000
0	den009d.map	50	34	6	30	5	30	1.00000000
0	den009d.map	50	34	16	16	15	13	3.41421356
0	den009d.map	50	34	13	22	11	22	2.00000000
0	den009d.map	50	34	30	29	30	29	0.00000000
0	den009d.map	50	34	38	15	37	16	1.41421356
0	den009d.map	50	34	28	3	27	4	2.00000000
0	den009d.map	50	34	43	17	41	16	2.41421356
0	den009d.map	50	34	16	30	17	27	3.41421356
0	den009d.map	50	34	4	27	5	27	1.00000000
1	den009d.map	50	34	13	3	7	4	6.41421356
1	den009d.map	50	34	10	8	17	6	7.82842712
1	den009d.map	50	34	35	11	38	6	6.24264069
1	den009d.map	50	34	27	2	34	2	7.00000000

Rysunek 2: Przykładowy wygląd scenariusza (kolejne kolumny oznaczają: id, nazwę, szerokość mapy, wysokość mapy, współrzędne punktu początkowego, współrzędne punktu końcowego, optymalną długość trasy

4 Sposób implementacji

Pierwszym etapem algorytmu jest przetworzenie mapy i stworzenie z niej siatki zer i jedynek, odpowiadającym przeszkodzie i jej braku.

Następnie zostaje stworzony obiekt ACO, który przyjmuje jako parametry, - siatkę, wcześniej utworzoną,

- liczbę mrówek,
- współczynnik alpha - odpowiada za czułość feromonów,
- współczynnik beta - odpowiada za czułość geometrii,
- liczbę iteracji.

Prawdopodobieństwo wybrania następnego węzła (sąsiada) na trasie przez "mrówkę" jest opisywane przez wzór

$$result = \tau^\alpha * \eta^\beta \quad (1)$$

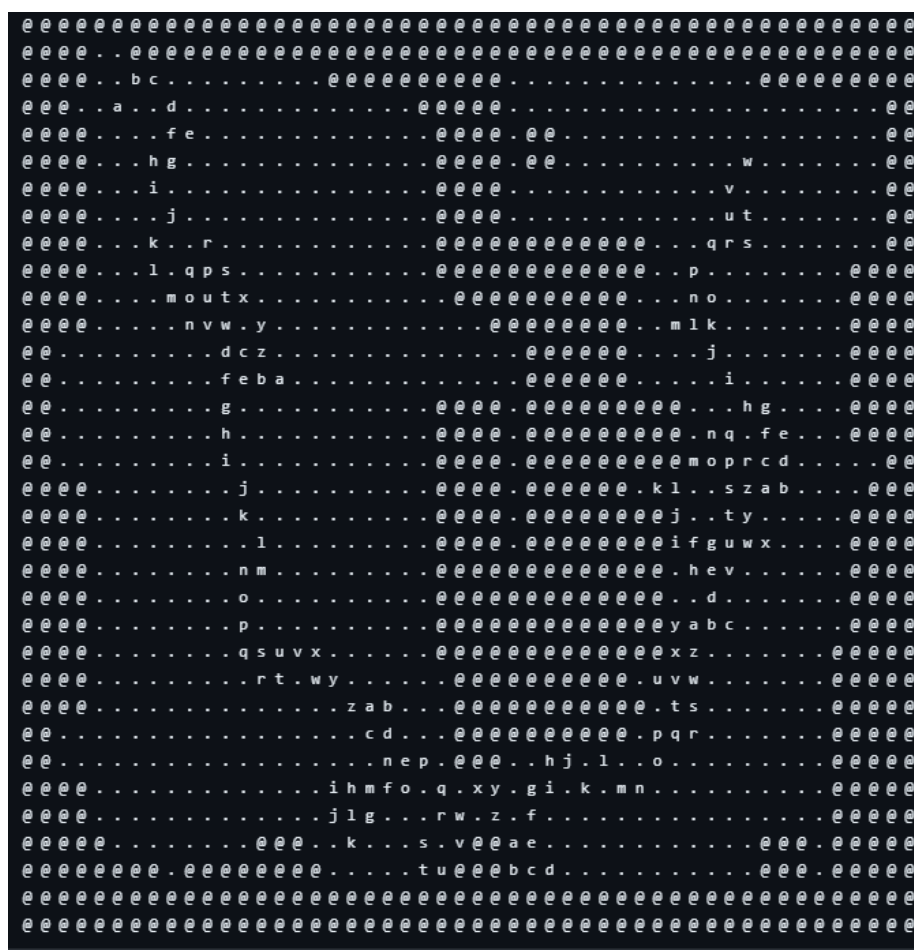
gdzie:

- alpha - odpowiada za wpływ śladu feromonu na wybór następnej lokalizacji przez "mrówkę". Jego wysoka sugeruje, że "mrówka" bardziej prawdopodobnie wybierze drogę z wyższym stężeniem feromonu.
- beta - kontroluje wpływ heurystyki na wybór następnej lokalizacji przez "mrówkę". Jego wysoka wartość wskazuje, że "mrówki" będą wybierać kierunek "prosto do celu", natomiast mała jego wartość skutkuje większą losowością.
- tau - stężenie feromonu na danej ścieżce. Im wyższe stężenie, tym większe prawdopodobieństwo, że "mrówka" wybierze tę ścieżkę.

- eta - wartość funkcji heurystycznej, która jest odwrotnością odległości pomiędzy dwoma punktami. Im większa jej wartość, tym większe prawdopodobieństwo, że "mrówka" wybierze daną ścieżkę.

Następnie na tak przygotowanym obiekcie można uruchomić algorytm szukania trasy pomiędzy dwoma punktami podanymi w postaci (y,x).

Dodatkowo trasę przebyta przez mrówki została obrazowana za pomocą kolejnych liter alfabetu (po literze "z" następuje ponownie litera "a").



Rysunek 3: Trasa przebyta przez mrówki od punktu (5,3) do punktu (40, 5), dla mapy z rysunku 1

W ostatnim etapie algorytm oblicza długość ścieżki dodając 1 do długości gdy kolejny węzeł na trasie jest obok poprzedniego lub $\sqrt{2}$, gdy jest on na ukos od poprzedniego.

5 Wyniki

Wyniki przedstawiliśmy jako stosunek długości ścieżki otrzymanej do długości ścieżki optymalnej (ze scenariusza).

$$result = length/length_{scen} \quad (2)$$

Dla mapy z rysunku 1 i 170 scenariuszy dla tej mapy (częściowo na rysunku 2) obliczyliśmy średni wynik. Trasy w scenariuszach miały rosnącą długość.

$$average_result = 2.009364988836624 \quad (3)$$

Parametry, które przyjęliśmy do obliczeń to:

- liczba mrówek - 10,
- alpha - 0.5,
- beta - 1.0,
- evaporaton - 0.5,
- liczba iteracji - 100.

Dla krótszych tras (o długościach mniejszych lub równych 10) wyniki ocylowały w okolicach 1. Natomiast dla dłuższych wyniki sięgały 6,05.

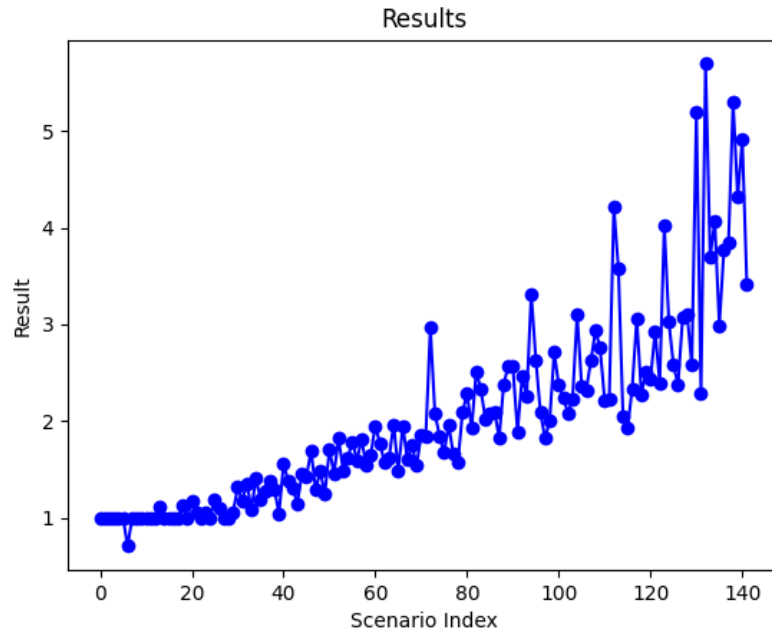
Aby sprawdzić wpływ współczynników alpha i beta na działanie algorytmu wybraliśmy jedną trasę na której uruchomiliśmy algorytm dla różnych ich wartości. Wybór jednej trasy i wąskiego przedziału wartości współczynników był spowodowany długim czasem obliczeń. Wyniki zostały obliczone według wzoru 2 i pokazane na wykresach.

6 Wnioski

Po analizie wyników naszego projektu dotyczącego wykorzystania algorytmu Ant Colony Optimization (ACO) do planowania trasy, możemy wyciągnąć następujące wnioski:

Jednym z głównych wyzwań związanym ze stosowaniem algorytmu ACO jest jego wymagająca złożoność obliczeniowa. Dla większych map i większej liczby mrówek algorytm może być wolny i niewydajny.

Algorytm ACO okazał się być mniej skuteczny dla długich tras, aczkolwiek przy np. zwiększeniu ilości mrówek mogłoby okazać się równie skuteczny. Nie mogliśmy tego jednak sprawić przez długi czas obliczeń.

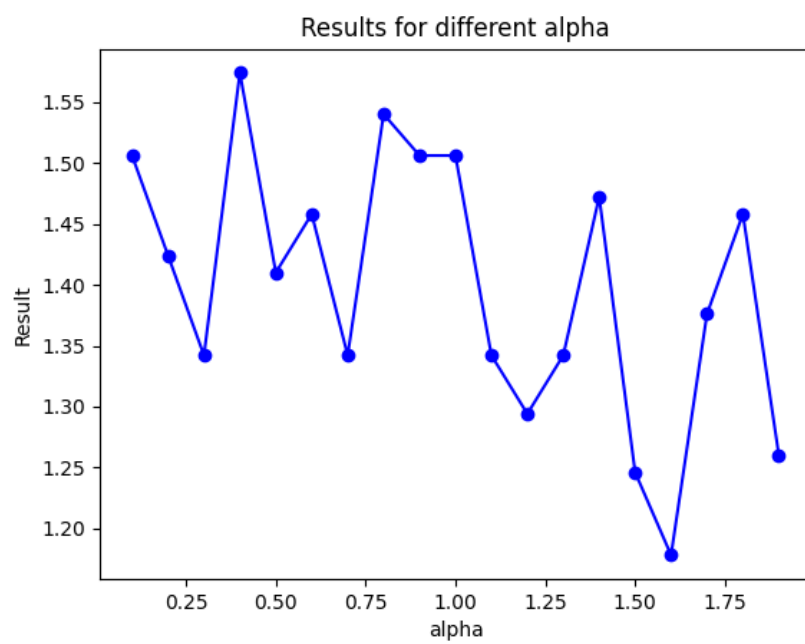


Rysunek 4: Wyniki dla kolejnych scenariuszy (długości tras w kolejnych scenariuszach są posortowane rosnąco)

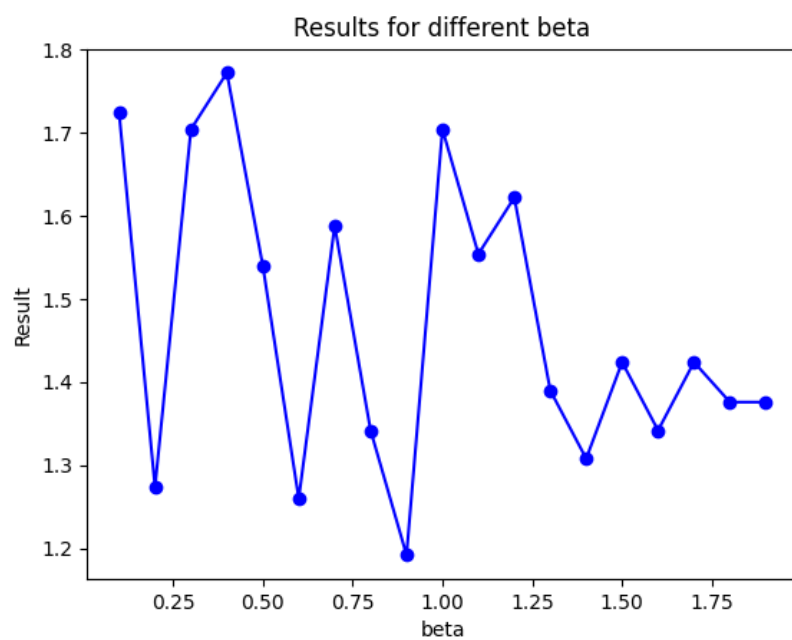
Współczynnik "alpha", odpowiadający za wpływ śladu feromonowego na wybór tras przez mrówki, nie miał znaczącego wpływu na ogólne wyniki algorytmu.

W przypadku współczynnika "beta", który kontroluje wpływ informacji heurystycznej na wybór trasy przez mrówkę, obserwowaliśmy stabilność i lekko poprawioną wydajność algorytmu dla wartości powyżej 1.5.

Podsumowując, pomimo pewnych ograniczeń, algorytm ACO pokazał obiecujące wyniki w kontekście planowania trasy. Warto jednak zauważyć, że skuteczność i użyteczność algorytmu ACO mogą znacznie zależeć od specyficznego zastosowania i parametryzacji.



Rysunek 5: Wyniki dla różnych warości współczynnika alpha



Rysunek 6: Wyniki dla różnych wartości współczynnika β