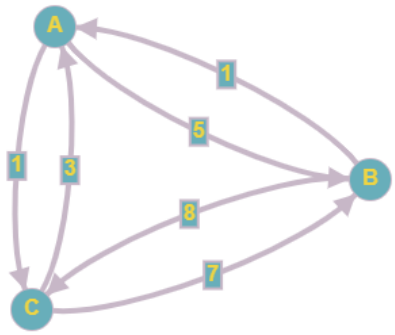
Strategie zespołowe - Projekt
Niesymetryczny, niestacjonarny problem komiwojażera i optymalizacja kolonią mrówek (ACO)

Magdalena Cisowska 160527
Jan Cybulski 160228

1 Wstęp i opis problemu

Problem komiwojażera (z ang. *travelling salesaman problem*, TSP) jest to zagadnienie optymalizacji, gdzie celem jest znalezienie minimalnej długości trasy Hamiltona dla pełnego cyklu grafu ważonego. Oznacza to, że w grafie, w którym wartość liczbowa na krawędzi reprezentuje koszt drogi pomiędzy tworzącymi ją węzłami, należy znaleźć najkrótszą pod względem kosztu ścieżkę. Warunkiem jest, aby każdy węzeł w grafie był odwiedzony tylko jeden raz. Problem ten jest analogiczny do sytuacji kuriera, który musi odwiedzić każde miasto, aby zostawić w nim paczkę, natomiast w jego interesie leży, aby odwiedzić każde miasto tylko jeden raz oraz zrobić to jak najmniejszym nakładem sił - np. zużywając najmniej paliwa, a więc pokonując łącznie najmniejszą możliwą liczbę kilometrów.

O TSP mówi się, że jest niesymetryczny, kiedy koszt drogi pomiędzy węzłami może być różny w zależności od kierunku - czy z A do B, czy z B do A. Wówczas każda droga ($A \rightarrow B$ oraz $B \rightarrow A$) reprezentowana jest jako osobna krawędź. Niestacjonarny TSP jest natomiast wtedy, kiedy wartości kosztów na krawędziach mogą zmieniać się w czasie.



Rysunek 1: Przykład asymetrycznego, pełnego grafu

Dużym problemem TSP jest bardzo duża złożoność obliczeniowa - dla n miast liczba kombinacji możliwych ścieżek wynosi $(n-1)!/2$, czyli dla 20 miast będzie to $6 * 10^{16}$ wariantów do sprawdzenia.

1.1 Optymalizacja kolonią mrówek

TSP jest akademickim zagadnieniem, dlatego zdążyło powstać wiele podejść, które pozwalają rozwiązać problem stosunkowo małym nakładem obliczeniowym. Jednym z nich jest optymalizacja kolonią mrówek (ACO). Jest to algorytm inteligencji roju, inspirowany zachowaniem niektórych gatunków mrówek. Powstał on w latach 1940-1950 za sprawą Pierre-Paul Grass'e, który zauważył, że pewne gatunki termitów reagują na tzw. istotne bodźce.

Mrówki, chcąc oznaczyć faworyzowaną ścieżkę, zostawiają na niej substancję zwaną feromonem. W ten sposób "komunikują" innym osobnikom, że podążanie ścieżką z największym stężeniem feromonu będzie korzystne - zaprowadzi je np. do źródła pożywienia. Taki pośredni sposób komunikacji pomiędzy osobnikami nazywany jest stygmergią - informacja przekazywana jest za pomocą modyfikacji środowiska.

Do rozwiązania problemu TSP symulowana jest grupa sztucznych mrówek, poruszających się po grafie, gdzie węzeł reprezentuje miasto, a krawędź reprezentuje połączenie między obydwoma miastami. Wprowadzamy zmienną zwaną feromonem, która to jest przypisana do każdej krawędzi (razem z kosztem drogi) i może być modyfikowana i odczytywana przez każdą mrówkę. ACO jest algorytmem iteracyjnym, gdzie w każdej iteracji przez graf przechodzi cała populacja mrówek. Zgodnie z założeniami TSP, każda z nich odwiedza każde miasto i robi to tylko raz. Podczas tworzenia rozwiązania, mrówka wybiera kolejne węzły opierając się na stochastycznym mechanizmie związanym z feromonem.

Pod koniec iteracji dokonywana jest modyfikacja wartości feromonu na krawędziach w oparciu o jakość rozwiązań utworzonych przez mrówki. Pod tym względem istnieje kilka odmian ACO, a w implementacji wykorzystano jedną z nich - algorytm Min-Max Ant System.

2 Algorytm Min-Max Ant System (MMAS)

Jest to algorytm udoskonalający metodę Ant System. Jedyną różnicą jest drobna zmiana w modyfikacji feromonu na końcu iteracji. Aktualizacja feromonów następuje tylko dla "najlepszej" mrówki - takiej, która przebyła najkrótszą trasę. Ponadto, wartość feromonu jest ograniczona. Modyfikacja odbywa się wg następującej formuły:

$$\tau_{ij} = [(1-\rho)\tau_{ij} + \Delta\tau_{ij}^{best}]_{\tau_{min}}^{\tau_{max}} \tag{1}$$

, gdzie:

- ρ - współczynnik wyparowywania [0-1],
- $\Delta\tau_{ij}^{best} = \frac{1}{L_{best}}$ jeżeli (i,j) należy do najlepszej trasy (inaczej 0). L_{best} to długość trasy najlepszej mrówki (względem danej iteracji lub najlepszej do tej pory).
- wartości graniczne τ_{min} i τ_{max} dobiera się eksperymentalnie.

Reszta wzorów pozostaje niezmienna względem algorytmu AS. Prawdopodobieństwo przejścia mrówki k do miasta j:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\Sigma \tau_{ij}^\alpha \eta_{ij}^\beta}, & \text{jeżeli } c_{ij} \in \mathbf{N}(s^p) \\ 0, & \text{w przeciwnym wypadku} \end{cases} \tag{2}$$

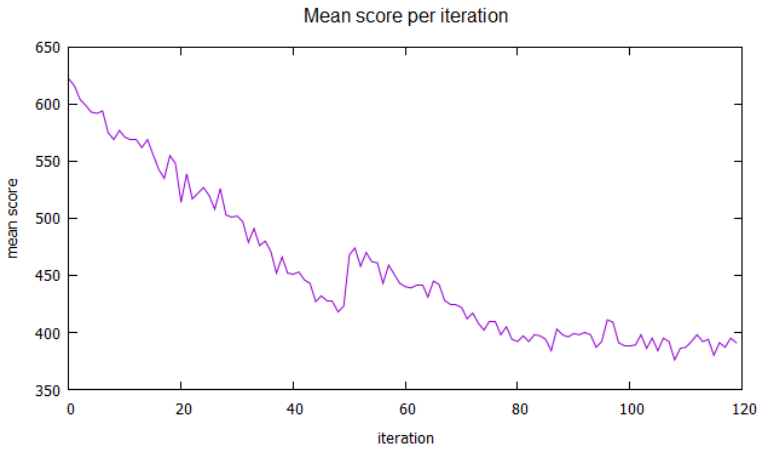
, gdzie parametry α, β sterują istotnością wartości feromonu τ względem wartości heurystycznej η (kosztu drogi), a $\mathbf{N}(s^p)$ to zbiór nieodwiedzonych jeszcze przez mrówkę k miast.

3 Implementacja i wyniki

Dokonano implementacji algorytmu MMAS. Graf tworzony jest poprzez utworzenie danej liczby węzłów i połączenie ich wszystkich między sobą, aby uzyskać pełen, niesymetryczny graf. Wartości drogi losowano z zadanego przedziału, natomiast początkowa wartość feromonu na krawędzi (i,j) wynosiła $1/liczba\ węzłów*droga_{ij}$. W iteracji 50. dokonywano zaburzenia wartości na losowo wybranych 20 węzłach grafu - koszt był zwiększany na losową wartość z nowego, wyższego przedziału. Symulowało to niestacjonarność problemu TSP.

Na początku pozwolono mrówkom startować z dowolnych węzłów oraz nie sprecyzowano końcowego węzła. Przyjęto następujące wartości kluczowych parametrów:

- $\alpha = 0.5$,
- $\beta = 1$,
- liczba mrówek = 20,
- liczba węzłów grafu = 45,
- $\rho = 0.1$,
- liczba iteracji = 120.

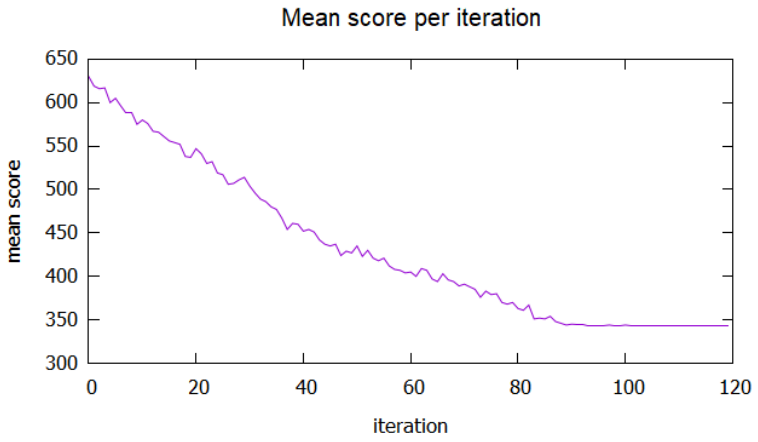


Rysunek 2: Zależność średniego kosztu przejścia grafu od liczby iteracji. Wszystkie mrówki startowały z losowo wybieranych węzłów. Widoczne jest wyraźne zaburzenie w iteracji 50

Zauważamy, że od około 80 iteracji wykres przestał opadać, pozostały jednak drobne fluktuacje wokół średniego kosztu przejścia równego 400. Wynika to z faktu, że mrówki wciąż startują z losowych węzłów, nie minimalizujemy w tym momencie konkretnej ścieżki łączącej węzeł startu i końca, a staramy się ją jedynie odnaleźć. Stąd wykres się nie stabilizuje. Ponadto, istotnym jest także fakt, że poszukujemy jedynie ścieżki Hamiltona, a nie cyklu. Cykl ma tę własność, że rozpoczyna i kończy się w tym samym węźle.

Algorytm okazał się odporny na niestacjonarność - chwilę po zaburzeniu wag w grafie średni koszt gwałtownie wzrósł - mrówki zaczęły zatem próbować innych ścieżek (z powodzeniem), co doprowadziło do ponownego opadania funkcji, a feromon na zmienionych krawędziach (niegdyś zachęcających) stopniowo i bezpowrotnie wyparował.

Następnym podejściem było narzucenie mrówkom kończenia w węźle, w którym zaczynały. Podobnie, jak poprzednio, startowaliśmy z losowych węzłów, zwiększono jednak dwukrotnie liczbę mrówek, aby dodatkowo ustabilizować wykres.



Rysunek 3: Zależność średniego kosztu przejścia grafu od liczby iteracji. Wszystkie mrówki startowały z losowych węzłów, po czym kończyły zawsze na tym, z którego zaczęły.

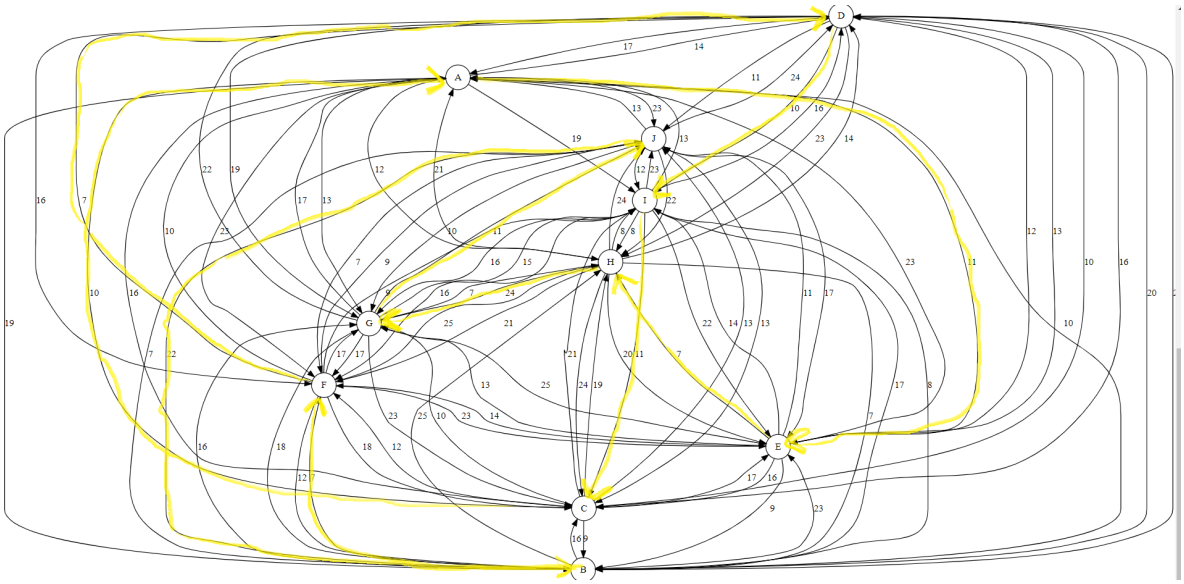
Dla tego wariantu, poszukiwanie cyklu zamiast ścieżki Hamiltona powoduje, że po 90 iteracjach osiągamy optymalny koszt przejścia cyklu - świadczy o tym całkowite wypłaszczenie funkcji, a więc wszystkie mrówki wybierają tę samą sekwencję miast.

3.1 Przykładowe rozwiązanie zadania TSP

Aby zwizualizować rozwiązanie pewnego niesymetrycznego zadania TSP, zmniejszono liczbę węzłów do 10 i przeprowadzono symulację. Optymalny cykl Hamiltona miał koszt wynoszący 77 i wyglądał on następująco:

$$E \to H \to G \to J \to B \to F \to D \to I \to C \to A \to E \tag{3}$$

Optymalny cykl przedstawia poniższy rysunek.



Rysunek 4: Przykład optymalnego cyklu Hamiltona dla 10 elementowego niesymetrycznego grafu, uzyskanego na drodze symulacji algorytmu MMAS.

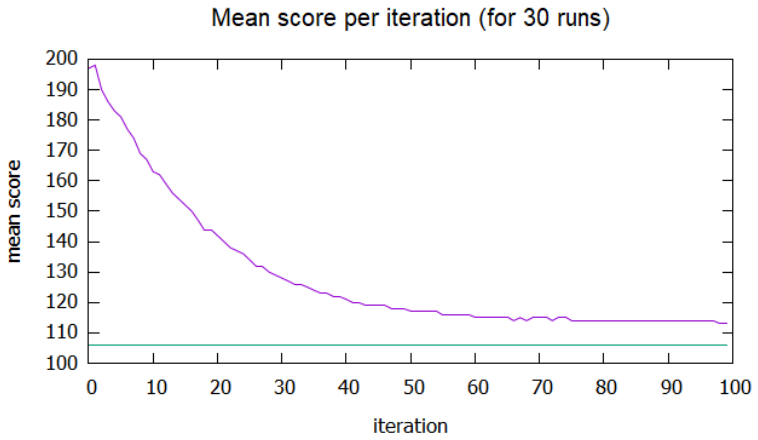
3.2 Zachowanie algorytmu dla 30 przebiegów

Algorytm uruchomiono 30 razy dla tego samego grafu. Liczył on 15 węzłów. Wykreślono następnie funkcję średniego kosztu uśrednionego dla 30 przebiegów. Podczas pracy algorytmu można było zauważyć, że za każdym razem zbiegał on w inny sposób do rozwiązania, ponadto, rozwiązania wyznaczone w poszczególnych uruchomieniach były względem siebie różne. Najmniejszy koszt cyklu Hamiltona wynosił 106 i wyglądał następująco:

$$H \to N \to B \to A \to D \to M \to E \to L \to I \to G \to K \to J \to C \to F \to H \tag{4}$$

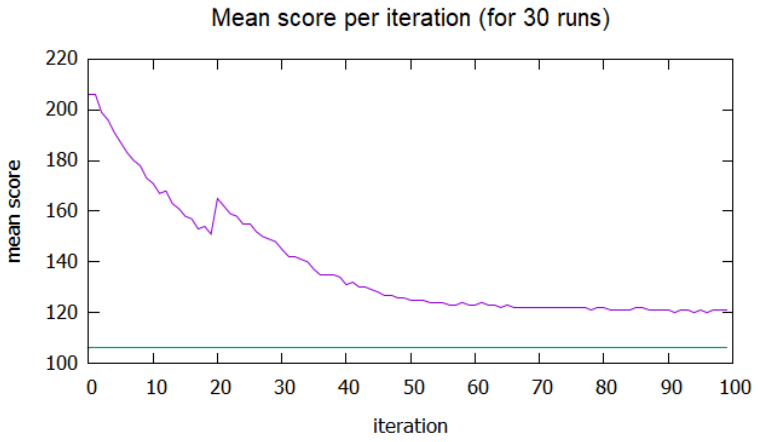
Znalazło się także rozwiązanie o koszcie 113 i wyglądało ono inaczej:

$$G \to E \to D \to M \to O \to L \to I \to F \to J \to H \to N \to B \to A \to C \to G \tag{5}$$



Rysunek 5: Uśrednione średnie koszty przejścia dla 30 uruchomień algorytmu. Liczba miast wynosiła 15, najlepsze utworzone rozwiązanie miało koszt 106 - zaznaczone zielonym przebiegiem. Zauważamy eksponencjalny kształt wykresu.

Dla problemu niestacjonarnego zauważamy nieco inny przebieg wykresu. Tym razem ponownie najkrótsza znaleziona trasa wynosiła 106, jednak zauważamy, że przebieg w miejscu wypłaszczenia oddalił się od tej wartości.



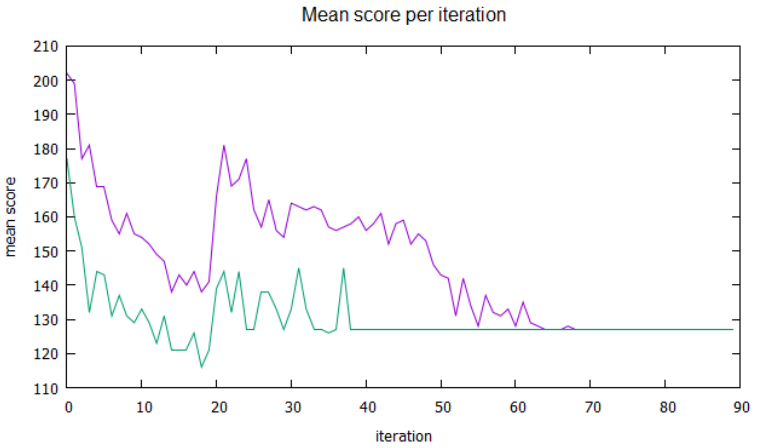
Rysunek 6: Uśrednione średnie koszty przejścia dla 30 uruchomień algorytmu. Liczba miast wynosiła 15, najlepsze utworzone rozwiązanie ponownie miało koszt 106 - zaznaczone zielonym przebiegiem. Zauważamy wpływ niestacjonarności

Przyczyny takiego zachowania mogą być różne. Wydaje się, że dzieje się tak za sprawą natury działania algorytmów ACO. Decyzje podejmowane przez mrówki opierają się na stochastycznym mechanizmie - w każdym węźle o przejściu decydują prawdopodobieństwa - bardzo często niezerowe dla więcej niż jednego węzła. Powoduje to, że dla stałego grafu, nie dla każdego uruchomienia udaje się algorytmowi zbiec do globalnego rozwiązania.

Dobrą praktyką wydaje się zatem wykonanie szeregu symulacji - np. 30 - a następnie ocena wygenerowanych rozwiązań i wybranie najlepszego.

3.3 Minimalny a średni koszt

Poniższy wykres przedstawia średni oraz minimalny koszt w każdej iteracji. Najmniejszy koszt charakteryzuje się pewnymi fluktuacjami, jednak ustabilizowuje się na tej samej wartości co średni koszt. Ponadto, jest on zawsze mniejszy (lub niewiększy) niż koszt średni.



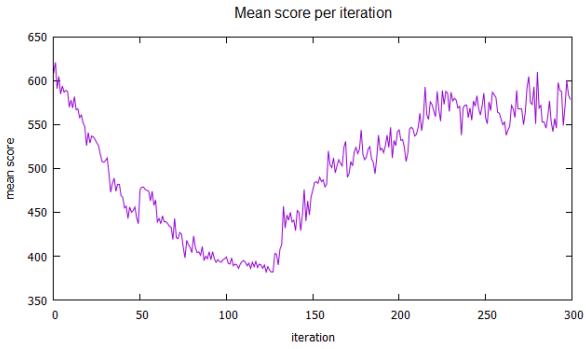
Rysunek 7: Średni (fioletowy) oraz minimalny (zielony) koszt dla każdej iteracji. Symulację przeprowadzono dla 15 miast oraz 20 mrówek. Graf zaburzono w 20 iteracji.

4 Podsumowanie

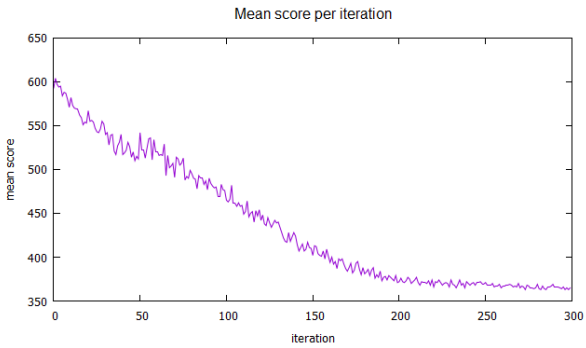
Implementacja metody MMAS, należącej do grupy metod optymalizacji kolonią mrówek, przebiegła pomyślnie. Udało się znacząco zmniejszyć koszt przejścia grafu, co pozwala przypuszczać, że odnajdywany cykl Hamiltona jest optymalny.

Podczas symulacji problematyczne okazało się generowanie grafu. Dla ilości miast większej niż 7 zadanie to jest bardzo żmudne do wykonania ręcznie, nie wspominając już o ilości miast używanej w symulacjach wynoszącej 45. Rozwiązaniem okazało się losowe generowanie wag krawędzi, przez co kolejne symulacje różniły się od siebie względem znalezionych cykli. Zaburzenia również generowane było losowo - dla 20 losowo wybranych krawędzi grafu. Powodowało to, że nie zawsze zaburzenie można było zaobserwować na wykresie. Jedynie wtedy, kiedy zmianie uległy krawędzie z dużym natężeniem feromonu, używane w danej chwili przez mrówki.

W opisywanych przykładach liczba iteracji nie przekraczała 150. Przy zbyt wielu iteracjach algorytmu można było zaobserwować nagły wzrost średniego kosztu drogi mrówek. Wynikało to ze znacznego nagromadzenia się feromonu na krawędziach, a gdy pojawiały się tam wartości rozumiane przez komputer jako nieskończoność, algorytm przestawał prawidłowo się zachowywać i wybierane krawędzie nie były optymalne. Aby zapobiec takiemu zachowaniu mamy do dyspozycji parametry takie, jak liczba iteracji, a także współczynnik wyparowywania ρ, co pokazują poniższe rysunki.



Rysunek 8: Średni koszt dla 300 iteracji algorytmu MMAS oraz $\rho = 0.1$



Rysunek 9: Średni koszt dla 300 iteracji algorytmu MMAS oraz $\rho = 0.7$. Większa wartość wyparowywania powoduje wolniejszą zbieżność algorytmu.

Podsumowując, algorytmy ACO okazują się bardzo szybkim narzędziem do rozwiązywania problemu komiwojażera. Implementacja nie przysparza problemów, a strojenie parametrów jest stosunkowo łatwe.