

Wrocław, 27.07.2001

Algorytmy genetyczne - projekt
Wydział Informatyki i Zarządzania, Rok studiów IV

Optymalizacja kolonii mrówek w zastosowaniu do problemu komiwojażera

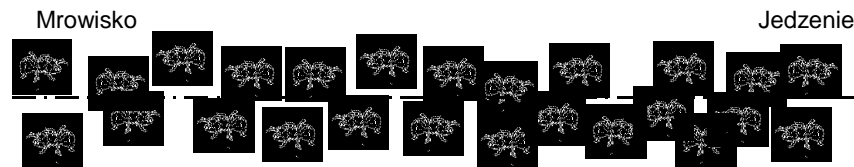
Autorzy dokumentu: *Tomasz Linda,*
Daniel Abrich

Prowadzący: *dr H. Kwaśnicka*

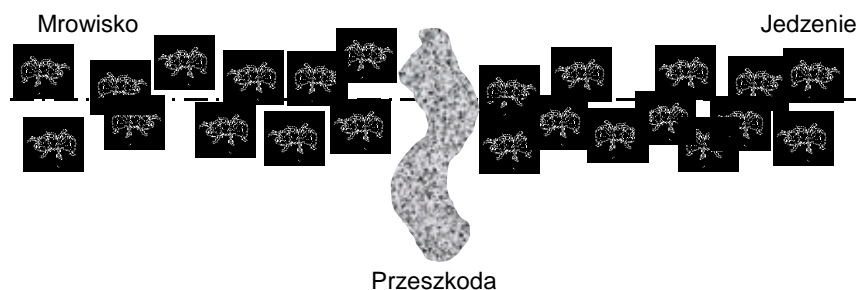
1 Wstęp

Algotymy mrówkowe powstały w wyniku obserwacji kolonii żywych mrówek. Mrówki są owadami społecznymi, żyjącymi w koloniach, a ich zachowania są skierowane bardziej na przeżycie społeczności jako całości niż pojedynczego osobnika. Owady społeczne przyciągnęły uwagę wielu naukowców ze względu na wysoki poziom strukturalizacji kolonii w porównaniu do prostoty pojedynczego osobnika. Interesującym i ważnym zachowaniem kolonii mrówek jest szukanie pożywienia, a w szczególności jak mrówki znajdują najkrótszą drogę między źródłem pożywienia a mrowiskiem. Poza tym ma ona zdolność przystosowywania się do zmian w środowisku, np. znalezienie nowej najkrótszej drogi w momencie gdy stara przestaje spełniać swoje zadanie ze względu na pojawienie się przeszkody.

Rozważmy poniższy rysunek, na którym mrówki poruszają się po prostej linii łączącej źródło pożywienia z mrowiskiem:

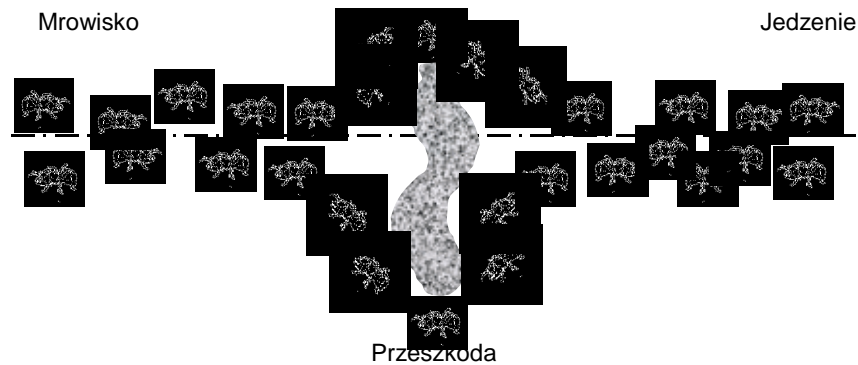


Podczas marszu od źródła pożywienia do mrowiska i z powrotem, mrówki zostawiają na podłożu substancję zwaną *feromonem*, formując w ten sposób szlak feromonowy. Mrówki wyczuwają feromony i gdy muszą wybrać drogę starają się wybrać ścieżkę o większej ich koncentracji. Dodatkowo szlak feromonowy umożliwia mrówkom znalezienie drogi powrotnej do mrowiska (lub do źródła pożywienia) oraz pozwala na znalezienie źródeł pożywienia znalezionych przez innych członków ich społeczności. Te elementarne zachowania mrówek pozwalają wyjaśnić, w jaki sposób mogą one znaleźć najkrótszą drogę łączącą ponownie ścieżkę, po tym jak nagłe pojawienie się niespodziewanej przeszkody ją przerwało:

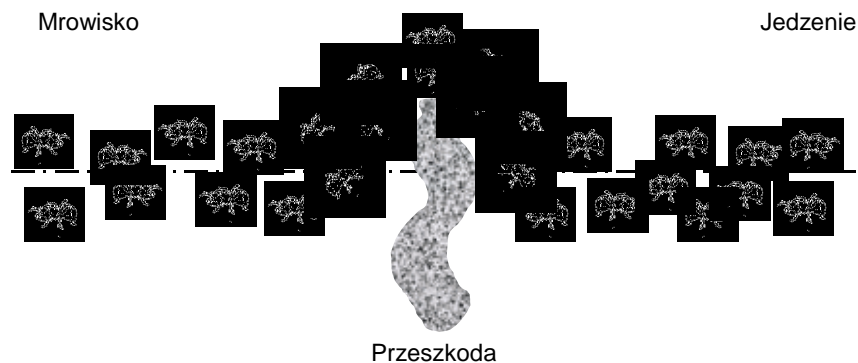


W momencie pojawienia się przeszkody, mrówki, które są tuż przed nią nie mogą kontynuować marszu szlakiem feromonowym, dlatego muszą wybrać pomiędzy skrętem w prawo lub w lewo. Z probabilistycznego punktu widzenia można by się spodziewać, że mrówki podziela

się na dwie równe grupy, z których każda podąży w innym kierunku. Z analogiczną sytuacją będziemy mieli do czynienia po drugiej stronie przeszkody:



Mrówki, które przypadkowo wybiorą krótszą ścieżkę naokoło przeszkody dużo szybciej połączą rozerwany szlak feromonowy niż pozostałe mrówki. W związku z tym krótsza droga otrzyma znacznie większą ilość feromonów w jednostce czasu co spowoduje, że więcej mrówek będzie wybierać krótszą drogę. Dzięki temu procesowi dodatniego sprzężenia (autokatalizacji) w krótkim czasie dojdzie do sytuacji, w której wszystkie mrówki będą wybierać krótszą drogę:



Widać, że proces opisany powyżej jest pewnym rodzajem rozproszonego mechanizmu optymalizacji, w który każda pojedyncza mrówka ma bardzo mały wkład. Interesujące jest, że chociaż każda z pojedynczych mrówek jest zdolna do rozwiązania problemu (np. znalezienia drogi pomiędzy mrowiskiem a źródłem pożywienia), to tylko zespół mrówek – kolonia – przejawia zachowanie „dążące do znalezienia najkrótszej drogi”. Zauważmy, że mrówki potrafią przejawiać to specyficzne zachowanie dzięki prostej formie pośredniej komunikacji poprzez zostawianie feromonów, znanej jako stygmergia (*stigmergy*).

1.1 Optymalizacja kolonii mrówek (Ant Colony Optimisation)

Optymalizacja kolonii mrówek (Ant Colony Optimisation –ACO) jest próbą stworzenia algorytmów komputerowych wzorujących się na naturze, a konkretnie na zachowaniach kolonii mrówek.

W algorytmach ACO kolonia sztucznych mrówek skończonej wielkości kolektywnie szuka dobrej jakości rozwiązań rozważanego problemu. Każda mrówka znajduje rozwiązanie, lub jakiś jego fragment, zaczynając od stanu początkowego zależnego od problemu. Podczas szukania własnego rozwiązania każda mrówka zbiera informacje o charakterystyce problemu, a także o własnych dotychczasowych osiągnięciach. Na bazie tych informacji mrówki modyfikują reprezentację problemu postrzeganą przez inne mrówki. Mrówki działają jednocześnie a zarazem niezależnie od siebie, wykazując cechy pracy zespołowej. Oczywiście nie komunikują się one bezpośrednio, a tylko mechanizmowi stygmergii.

Przez rozwiązanie problemu rozumiemy ścieżkę o najniższym koszcie (najkrótszą) prowadzącą przez stany problemu, w zgodzie z jego ograniczeniami. Złożoność każdej z mrówek jest taka, że nawet pojedyncza mrówka jest w stanie znaleźć rozwiązanie (prawdopodobnie kiepskiej jakości). Rozwiązania dobrej jakości są jednak wynikiem globalnej kooperacji pomiędzy wszystkimi agentami w kolonii jednocześnie szukającymi różnych rozwiązań.

Zgodnie z przyjętą definicją sąsiedztwa (zależnego od problemu) każda mrówka szuka rozwiązania przesuując się poprzez sekwencję sąsiednich stanów. Przesunięcia są wybierane na bazie stochastycznej polityki przejść między stanami. Mają na nią wpływ:

- prywatne informacje mrówki (jej wewnętrzny stan lub pamięć),
- szlak feromonowy oraz informacje lokalne zależne od problemu.

Stan wewnętrzny mrówki zawiera informacje o jej historii. Może być używany do przechowywania przydatnych informacji pozwalających na oszacowanie jakości znalezionej rozwiązania, lub też wpływu każdego z wykonanych przesunięć na jego jakość. Może on także odgrywać fundamentalną rolę w ocenianiu możliwości zrealizowania znalezionej rozwiązania. W przypadku niektórych problemów, przeważnie w optymalizacji kombinatorycznej, pewne przesunięcia osiągalne dla mrówki w danym stanie mogą ją zaprowadzić w ślepą uliczkę. Można tego uniknąć wykorzystując pamięć mrówki. Zatem mrówki mogą znajdować „wykonalne” rozwiązania bazując tylko na wiedzy o aktualnym stanie oraz o efektach działań, które mogą być w tym stanie wykonane.

Decyzja kiedy i jak dużo mrówka powinna wypuścić feromonu do środowiska zależy od problemu i od projektu implementacji. Mrówki mogą zostawiać feromon podczas szukania rozwiązania lub po jego znalezieniu wracając przez wcześniej odwiedzone stany, lub w obu przypadkach.

Wyżej wymienione szlaki feromonowe oraz wartości heurystyczne definiują *tablice decyzyjne mrówek*, tj. probabilistyczne tablice używane przez politykę decyzyjną mrówek w celu

skierowania poszukiwań w kierunku najbardziej interesujących regionów przestrzeni poszukiwań. Stochastyczny składnik polityki wyboru przesunięcia i wcześniej wspomniany mechanizm parowania feromonu, zapobiega gwałtownemu dryfowaniu wszystkich mrówek w najbardziej obiecującym kierunku. Poziom stochastyczności oraz częstość odświeżania szlaku feromonowego wyznacza równowagę pomiędzy odkrywaniem nowych stanów w przestrzeni poszukiwań, a eksploataowaniem nagromadzonej wiedzy.

W momencie, gdy mrówka kończy swoje zadanie polegające na znalezieniu rozwiązania i zostawieniu informacji feromonowej ginie, czyli jest usuwana z systemu.

1.2 Zastosowanie ACO do rozwiązania problemu komiwojażera (Traveling salesman problem)

Pierwszym problemem jaki próbowano rozwiązać przy pomocy algorytmów ACO był problem komiwojażera. Jego definicja jest następująca. Mamy graf zupełny $G=(N,E)$, gdzie zbiór N węzłów reprezentuje miasta, a zbiór krawędzi E połączenia między nimi. Załóżmy, że d_{ij} jest długością krawędzi $(i,j) \in E$, czyli odległością między miastami i i j , gdzie $i,j \in N$. Problem polega na znalezieniu najkrótszego cyklu Hamiltona w grafie G .

1.2.1 System mrówkowy (*Ant System* – AS)

System mrówkowy był pierwszym algorytmem ACO. Jest on o tyle ważny, że stał się prototypem dla wielu innych algorytmów mrówkowych, które znalazły wiele interesujących i udanych zastosowań.

W AS sztuczne mrówki znajdują rozwiązanie (drogę) poruszając się po grafie problemu z jednego miasta do drugiego. Algorytm uruchamia t_{max} iteracji. Podczas każdej z nich m mrówek szuka dróg wykonując n kroków, w których podejmowana jest probabilistyczna decyzja. W praktyce, kiedy w węźle i mrówka wybiera jako węzeł docelowy węzeł j , to krawędź (i,j) jest dodawana do aktualnie budowanej drogi. Te kroki są powtarzane, aż mrówka skończy budować swoją drogę.

Wymyślono trzy rodzaje algorytmu AS różniące się sposobem w jaki uaktualniany jest szlak feromonowy. Te algorytmy to:

- gęstościowy (*ant-density*), w którym mrówki zostawiają stałą ilość feromonu podczas budowania drogi,
- ilościowy (*ant-quantity*), w którym mrówki zostawiają ilość feromonu odwrotnie proporcjonalną do długości wybranej krawędzi podczas budowania drogi,
- cykliczny (*ant-cycle*), w którym mrówki zostawiają feromon dopiero w momencie, gdy zbudują całą drogę.

Eksperymenty wykazały, że wydajność algorytmu cyklicznego była znacznie wyższa, niż pozostałych dwóch, więc przestaniemy się nimi zajmować, a określenie AS będziemy odnosić do algorytmu cyklicznego.

Jak już wcześniej powiedzieliśmy, w AS każda z mrówek zostawia feromon na odpowiednich zmiennych feromonowych związanych z odwiedzionymi krawędziami, aby uczynić je bardziej atrakcyjnymi dla przyszłych mrówek. Następnie mrówki giną. W AS nie występują akcje demonów, a parowanie feromonów odbywa się na przemian z aktywnością mrówek.

Ilość feromonu $\tau_{ij}(t)$ związanego z krawędzią (i,j) ma za zadanie reprezentować atrakcyjność podążenia z miasta i do miasta j tą krawędzią. Informacja feromonowa zmienia się, odzwierciedlając doświadczenia zdobyte przez mrówki podczas rozwiązywania problemu. Ilość feromonu zostawianego przez mrówki jest proporcjonalna do jakości znalezionej drogi: Im krótsza droga tym większa ilość feromonu pozostawiona na krawędziach wchodzących w jej skład. Dzięki temu poszukiwania posuwają się w dobrych kierunkach. Natomiast dzięki parowaniu feromonu unikamy stagnacji, tj. sytuacji, w której wszystkie mrówki podążają tą samą drogą.

Pamięć (stan wewnętrzny) mrówek zawiera odwiedzone miasta i jest zwana listą tabu (*tabu list*). Pamięć definiuje dla każdej mrówki k zbiór miast, które mrówka będąca w mieście i ma jeszcze do odwiedzenia. Dzięki wykorzystaniu pamięci mrówka k jest w stanie znaleźć dobre rozwiązanie poprzez wygenerowanie grafu przestrzeni stanów (w TSP odpowiada to odwiedzeniu każdego miasta dokładnie raz). Dzięki pamięci mrówka może ponadto zostawić feromon na odwiedzonych krawędziach po znalezieniu rozwiązania.

Tablica decyzyjna mrówek $A_i = [a_j^i(t)]_{j \in N_i}$ w węźle i powstaje z lokalnych wartości szlaku feromonowego oraz lokalnych wartości heurystycznych:

$$a_{ij}(t) = \frac{\tau_{ij}(t)[\eta_{ij}]^\beta}{\sum_{l \in N_i} \tau_{il}(t)[\eta_{il}]^\beta} \quad \forall j \in N_i,$$

gdzie: $\tau_{ij}(t)$ – ilość feromonu na krawędzi (i,j) w chwili t ,

$\eta_{ij} = 1/d_{ij}$ – heurystyczna wartość przesunięcia z miasta i do miasta j ,

N_i – zbiór sąsiednich miast do miasta i ,

β - współczynnik odpowiedzialny za wagę wartości heurystycznych

Prawdopodobieństwo z jakim mrówka k zdecyduje się na przejście z miasta i do miasta $j \in N_i^k$ podczas budowania drogi, w t -tej iteracji algorytmu wyraża się wzorem:

$$p_{ij}^k(t) = \frac{a_{ij}(t)}{\sum_{l \in N_i^k} a_{il}(t)}$$

gdzie: $N_i^k \subseteq N_i$ – zbiór węzłów w sąsiedztwie węzła i , których mrówka k jeszcze nie odwiedziła (węzły w N_i^k są wybrane ze zbioru N_i na bazie pamięci mrówki $k - M^k$).

Po tym jak wszystkie mrówki znajdą swoje drogi następuje parowanie feromonu na wszystkich krawędziach, a następnie każda mrówka k zostawia swoją dawkę feromonu $\Delta\tau_{ij}^k(t)$ na odwiedzonych przez siebie krawędziach.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{q_3}{L^k(t)}, & \text{gd } (i, j) \in T^k(t) \\ 0, & \text{gd } (i, j) \notin T^k(t) \end{cases},$$

gdzie: $T^k(t)$ – droga znaleziona przez mrówkę k w iteracji t

$L^k(t)$ – długość znalezionej drogi

q_3 – współczynnik dawki zostawianego feromonu.

Powyższe równanie dobrze ilustruje, że wartość $\Delta\tau_{ij}^k(t)$ zależy od tego, jak dobrze spisała się mrówka: im krótsza znaleziona droga, tym większa ilość zostawionego feromonu.

W praktyce proces parowania i zostawiania feromonu przez mrówki są zaimplementowane przy pomocy następującej reguły stosowanej dla wszystkich krawędzi:

$$\tau_{ij}(t) \leftarrow \gamma\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

gdzie: $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$,

m – liczba mrówek w każdej iteracji,

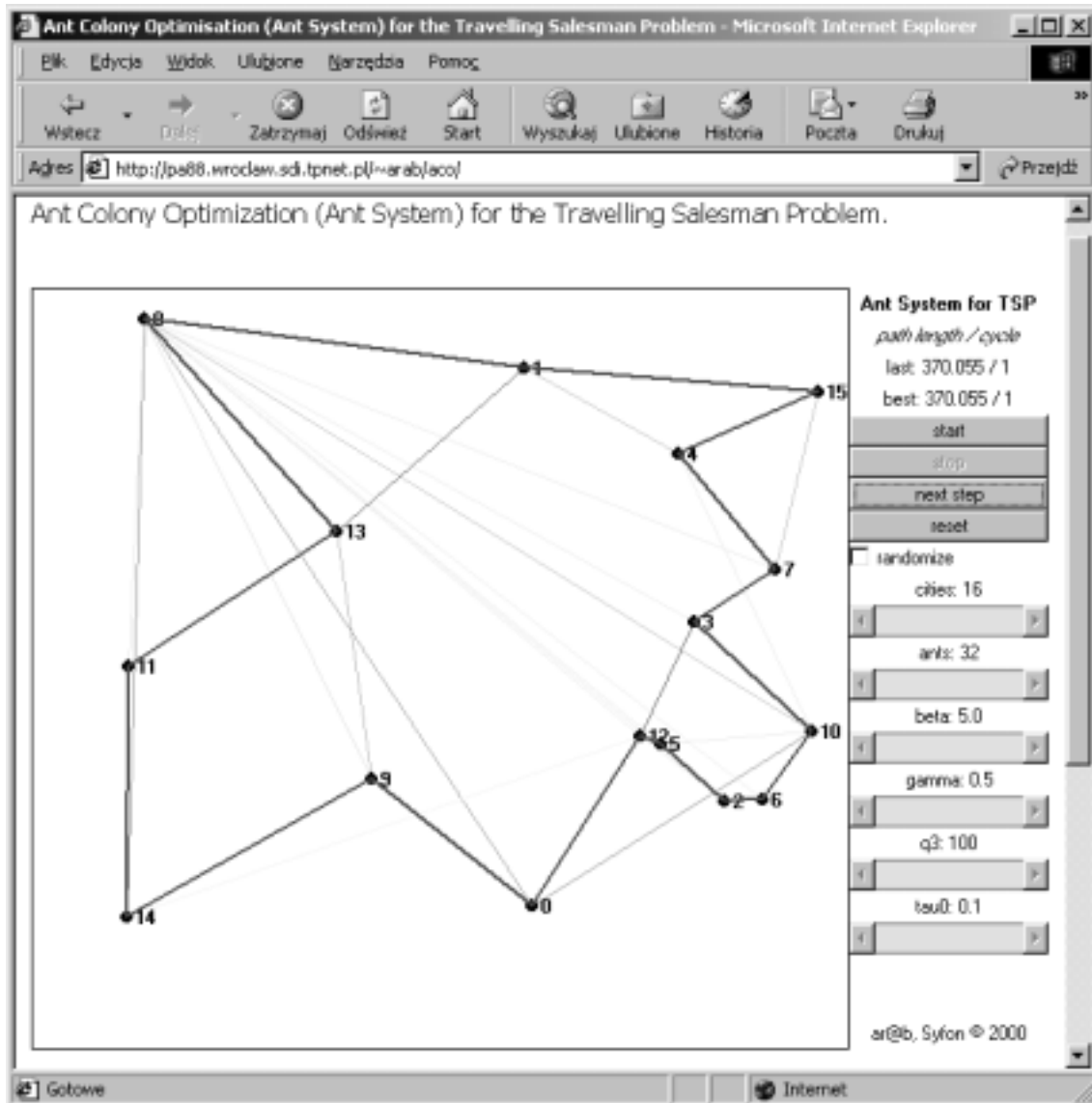
$\gamma \in (0,1]$ – współczynnik parowania feromonu.

2 Cel projektu

Celem projektu było napisanie przykładowej implementacji wraz z wizualizacją jednego z algorytmów optymalizacji kolonii mrówek (konkretnie systemu mrówkowego) w zastosowaniu do rozwiązania problemu komiwojażera, a następnie przeprowadzenie przy jego pomocy szeregu badań, prowadzące do znalezienia optymalnych parametrów startowych.

3 Realizacja

Program napisaliśmy jako w środowisku Java 1.1.7 w formie appletu i umieściliśmy na stronie internetowej pod adresem: <http://pa88.wroclaw.sdi.tpnet.pl/~arab/aco>. Wygląda on następująco:



Program nasz pozwala na symulowanie zachowania algorytmu (AS). Można dzięki niemu dowolnie dobrać parametry startowe algorytmu, a następnie śledzić kolejne fazy jego działania. Program może pracować w dwóch trybach: krokowym (klawisz „next step”) lub ciągłym (klawisze „start”/„stop”). Dodatkowo, w obu trybach pracy, w dowolnym momencie można przesunąć każde z miast i obserwować wpływ takiego przesunięcia na działanie algorytmu. Czerwony kolor oznacza najlepszą dotychczas znaną drogę (jej długość i cykl, w którym została znaleziona można zobaczyć w prawym górnym rogu ekranu), natomiast niebieski kolor symbolizuje natężenie feromonu na pozostałych krawędziach (przeskalowane względem najwyższej wartości feromonu).

4 Wnioski

Przy pomocy naszego programu przeprowadziliśmy szereg badań, po których doszliśmy do wniosku, że optymalnymi parametrami powodującymi znajdowanie faktycznie najkrótszej drogi w większości przypadków są: $\beta = 5$, $\gamma = 0.5$, $q_3 = 100$ oraz $\tau_0 = 0.1$.

Algorytm zachowuje się bardzo ładnie i w dużej części przypadków znajduje prawidłowe rozwiązanie w relatywnie krótkim czasie. Jednakże nie należy zapominać, że jest to algorytm heurystyczny i nie zawsze jest w stanie dojść do optymalnego rozwiązania. Widać to bardzo dobrze gdy poprzysuwamy sobie troszeczkę jedno z miast, tak aby spowodować, że poprzednio znaleziona droga nie będzie już najkrótszą. W takiej sytuacji algorytm najprawdopodobniej nie będzie w stanie znaleźć najlepszego rozwiązania. Jeżeli jednak uruchomimy cały algorytm od początku przy nowym ustawieniu miast to najprawdopodobniej znajdzie on pożądane rozwiązanie bez większych trudności.

Zadaniem napisanego przez nas programu nie jest udowodnienie nieomyślności przedstawionego algorytmu, a raczej pokazanie fenomenalnej zdolności algorytmów mrówkowych do przystosowywania się do szybko zmieniających się warunków. Doskonale można to zaobserwować uruchamiając program w trybie ciągłym (klawisz „start”), a następnie zmieniając położenie miast podczas jego pracy. Poza pewnymi wyjątkami (takimi jak opisany w poprzednim akapicie) algorytm zachowuje się zgodnie z założeniami.



Przeprowadzanie badań nad algorytmem przy pomocy naszego programu przysporzyło nam wiele radości. Polecamy go wszystkim chętnym jako rozrywkę na długie, zimowe wieczory.

Autorzy

Tomasz Linda

Daniel Abrich

Program można znaleźć pod adresem: <http://kosmos.kn.pl/~arab/aco/>