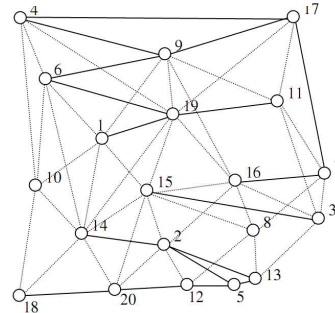


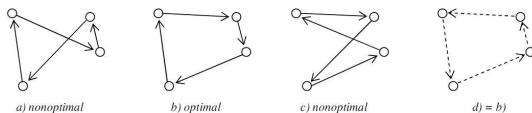


Ant Colony Optimisation

Problemy optymalizacji kombinatorycznej Problem komiwojażera (ang. TSP)



■ Np.: TSP - 4 miasta



liczba rozwiązań: $l = \frac{(N-1)!}{2}$

- np.: TSP-10 miast
- liczba rozwiązań: 181 440
- np.: TSP-50 miast
- liczba rozwiązań (około):
10 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000
000 000 000 000

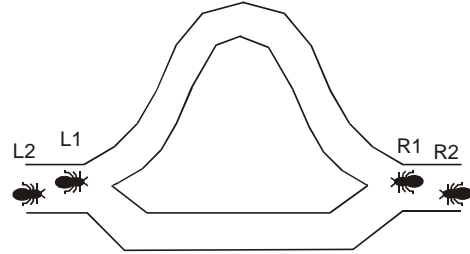
- Wiele zastosowań problemu komiwojażera: problemy w sieciach, marszrutyzacja i planowanie drogi, harmonogramowanie, itp...
- Problem TSP jest znany jako problem **NP – zupełny**; nie istnieją wielomianowe algorytmy rozwiązania problemu

- **problemy klasy P**: można je rozwiązać w czasie wielomianowym (lub trochę wolniej)
- **Problemy klasy NP**: w czasie wielomianowym można stwierdzić czy „zgadywane” rozwiązania są prawidłowe
- **Problemy NP–zupełne**

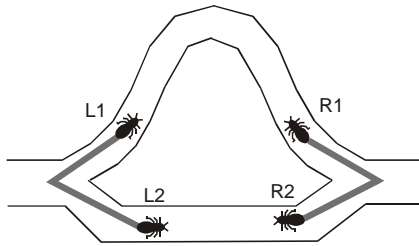
Inspiracja

- Społeczne zachowanie mrówek argentyńskich (*Iridomyrmex humilis*) jako inspiracja dla systemu wieloagentowego (ang. *multi-agent system, MAS*) (Marco Dorigo, 1995).
- Jeśli od gniazda mrówek do miejsca składowania żywności prowadzą dwie drogi – jedna krótsza, a druga dłuższa – to po pewnym czasie większość mrówek będzie korzystała z krótszej drogi.

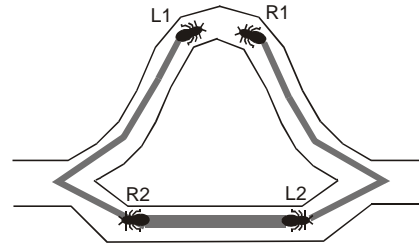
Eksperyment biologiczny



Eksperyment biologiczny



Eksperyment biologiczny



Eksperyment biologiczny

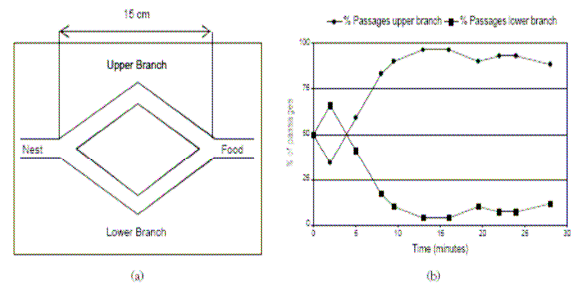
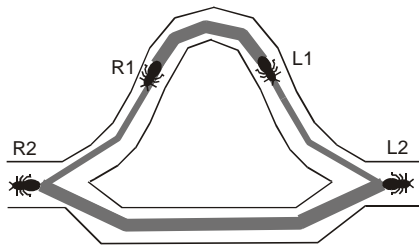


Figure 1. Single bridge experiment. (a) Experimental setup. (b) Results for a typical single trial, showing the percentage of passages on each of the two branches per unit of time as a function of time. Eventually, after an initial short transitory phase, the upper branch becomes the most used. After Deneubourg et al., 1990 [25].

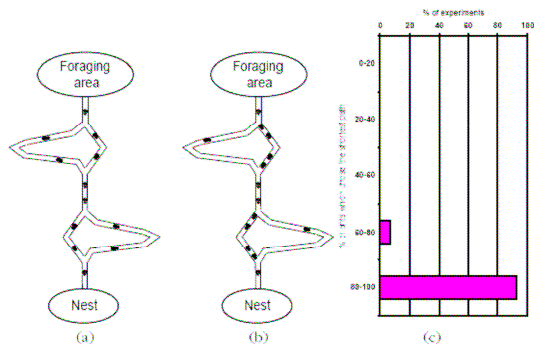


Figure 2. Double bridge experiment. (a) Ants start exploring the double bridge. (b) Eventually most of the ants choose the shortest path. (c) Distribution of the percentage of ants that selected the shorter path. After Goss et al. 1989 [60].

Algorytm mrówkowy (1)

- **Algorytm mrówkowy** (ang. Ant Colony Pptimization – ACO) został opracowany przez **Marco Dorigo** w jego pracy doktorskiej
- Algorytm mrówkowy jest techniką przeznaczoną głównie dla problemów wyszukiwania **najlepszych ścieżek w grafie**
- Inspiracja pochodzi ze świata mrówek, które potrafią znaleźć najkrótszą trasę między **mrowiskiem a pożywieniem**
- Na początku mrówki wędrując w stronę pożywienia wybierają trasę losowo, ale wracając do mrowiska pozostawiają na swojej trasie **śląd feromonowy**

Algorytm mrówkowy (2)

- Feromony stopniowo **parują**
- Na krótszej trasie parowanie jest stosunkowo **wolniejsze** niż na innych trasach, więc kolejne mrówki wybierają tą trasę **chętniej** niż inne trasy
- Kiedy zostanie znaleziona korzystna dla mrówek trasa, kolejne mrówki wybierają ją chętniej wzmacniając ślad feromonowy, jest to zjawisko **pozytywnego sprzężenia zwrotnego** - Atrakcyjna ścieżka staje się coraz bardziej atrakcyjna, a mniej atrakcyjna trasa traci na znaczeniu (sprzężenie zwrotne o dodatnim charakterze).

Algorytm mrówkowy został z powodzeniem użyty do problemu komiwojażera

- Tworzona jest populacja mrówek. Jej rozmiar jest jednym z parametrów algorytmu.
- Pojedyncza mrówka generuje swoją ścieżkę niezależnie od swoich towarzyszek.
- Dla każdej mrówki generowane jest losowo miasto, z którego ma rozpocząć wędrowkę.
- Mrówka porusza się po grafie, szukając sekwencji wierzchołków grafu tworzącej najkrótszą drogę od wierzchołka startowego do końcowego.

Algorytm mrówkowy (3)

- W systemie istnieje pośrednia interakcja między mrówkami oraz pewna forma gromadzenia doświadczeń i ich wykorzystywania w dalszych poszukiwaniach.
- W miarę upływu czasu mrówki wspólnie wypracowują zbiór najkrótszych ścieżek wiodących je do wyznaczonych celów (inteligencja zbiorowa).
- Działanie poszczególnych agentów w systemie wieloagentowym odpowiada działaniu pojedynczych mrówek poszukujących odpowiednich dla nich zasobów.
- Każdy z agentów (mrówek) realizuje identyczną strategię szukania najkrótszej drogi do celu.

- Aby nie wracać do już odwiedzonych wierzchołków, mrówka jest wyposażona w pamięć, w której przechowuje listę takich wierzchołków. Na starcie lista jest pusta, a po dojściu do celu zawiera wierzchołki w kolejności ich odwiedzenia.
- Mrówka rusza z miasta startowego i dochodząc do każdego kolejnego miasta pozostawia na krawędzi grafu feromon.
- Przy wyborze dalszej drogi mrówka nie bierze pod uwagę wierzchołków z listy miejsc już odwiedzonych, kieruje się natomiast feromonem pozostawionym przez te mrówki, które szły przed nią.
- Wybór kolejnego odcinka drogi jest losowy, ale zgodny z zasadą – im więcej feromonu na danej krawędzi grafu, tym większe prawdopodobieństwo wyboru tej drogi przez mrówkę.

- Algorytm ma iteracyjny charakter – po zakończeniu bieżącej iteracji każda mrówka czeka na wyznaczenie nowego miasta początkowego, skąd w następnej iteracji ponownie wyruszy, by przemierzać swoją trasę.
- Istotny jest efekt gromadzenia się feromonu na krawędziach grafu w miarę upływu czasu (kolejnych iteracji).
- W miarę upływu czasu pozostawiony feromon paruje, przez co zmniejsza się jego ilość na ścieżkach.

Reprezentacja danych

		<i>i</i>			
			d_{ij}		
<i>j</i>		f_{ij}			

d_{ij} – odległość między miastami i oraz j
 f_{ij} – poziom feromonu na drodze między miastami i oraz j

Taka reprezentacja dotyczy symetrycznego problemu TSP – tzn. z jednakową odległością między miastami i oraz j w obu kierunkach

- Reprezentacja danych – żadna informacja, która mogłaby być trwale dostępna w kolejnych iteracjach, nie jest gromadzona w mrówkach.
- Rola mrówek sprowadza się do generowania trasy na podstawie ilości feromonu złożonej na poszczególnych krawędziach grafu, a następnie do złożenia swojej porcji feromonu na kolejno przebytych krawędziach.
- Najważniejszym elementem algorytmu przechowującym informacje o rozwiązaniu jest więc struktura zawierająca wartości poziomu feromonu na poszczególnych krawędziach.

- Prawdopodobieństwo wybrania drogi przez mrówkę k znajdującą się w mieście i do miasta j w iteracji t

$$p_{ij}^k(t) = \frac{[f_{ij}(t)]^\alpha \cdot \left[\frac{1}{d_{ij}(t)}\right]^\beta}{\sum_{l \in N_i^k} \left([f_{il}(t)]^\alpha \cdot \left[\frac{1}{d_{il}(t)}\right]^\beta\right)}$$

- α, β – parametry określające odpowiednio: wpływ pozostawionego feromonu i odległości między miastami i oraz j na wyliczone prawdopodobieństwo
- N_i^k – zbiór miast, których mrówka k jeszcze nie odwiedziła, a do których prowadzi droga z miasta i

- Aktualizacja poziomu feromonu – uwzględnia zarówno odparowanie feromonu (I składnik), jak i złożenie feromonu przez mrówkę k (II składnik)

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

- ρ – współczynnik odparowania
- $t, t+1$ – dwie sąsiednie iteracje
- n – liczba mrówek w populacji
- L_k – całkowita długość trasy przebytej przez mrówkę k

Algorytm (zastosowany do problemu komiwojażera):

początek

ustalenie początkowego poziomu feromonu na krawędziach grafu;

powtarzaj

odparowanie części feromonu ze wszystkich krawędzi;

wylosowanie miast początkowych i wygenerowanie ścieżek dla wszystkich mrówek na podstawie aktualnego poziomu feromonu na krawędziach grafu;

naniesienie nowego feromonu na krawędzie w ilościach wynikających z długości ścieżek znalezionych przez mrówki (im ścieżka dłuższa, tym mniej feromonu);

opcjonalnie – operacje dodatkowe, które nie mogą być wykonane przez pojedyncze mrówki, np. złożenie dodatkowego feromonu na krawędziach, aby wyeliminować przesadną dominację jednej ścieżki nad innymi;

az do (warunek zakończenia);

koniec.