

Inteligencja roju / Inteligencja rozproszona

Zbiór organizmów (nieskomplikowanych agentów) wchodzących ze sobą w interakcje

Proste zachowania osobników +

Mechanizm współdzielenia informacji (wiedzy o środowisku/problemie) =>

Złożone zachowanie całych zbiorowości (rojów)

Inteligencja roju / Inteligencja rozproszona

- Mrówki, ptaki, ławice ryb, pszczoły, osy, termity, świetliki, ...
 - Stosunkowo prymitywne osobniki
 - Bogate / złożone zachowanie roju
 - Zachowanie emergentne



Algorytmy mrówkowe

Na podstawie pracy Marco Dorigo, Gianni Di Caro

Jedna z nowszych dyscyplin CI

- Sieci neuronowe - 1943
- Algorytmy genetyczne / PE – lata 1960 / 1975
- Logika rozmyta – 1965
- Ant Colony Optimization – lata 60-te obserwacja zachowań termitów, ujęcie algorytmiczne - od 1991
- Particle Swarm Optimization – 1995 / 1997

Inspiracja

- Obserwacja kolonii mrówek, w szczególności metody porozumiewania się w celu dotarcia do pożywienia.
- Pojedyncza mrówka przemieszcza się „losowo”.
- Kolonia mrówek funkcjonuje w sposób zorganizowany - potrafi więcej niż suma prosta poszczególnych osobników – efekt synergii.
- Efekt globalny na bazie lokalnych interakcji.

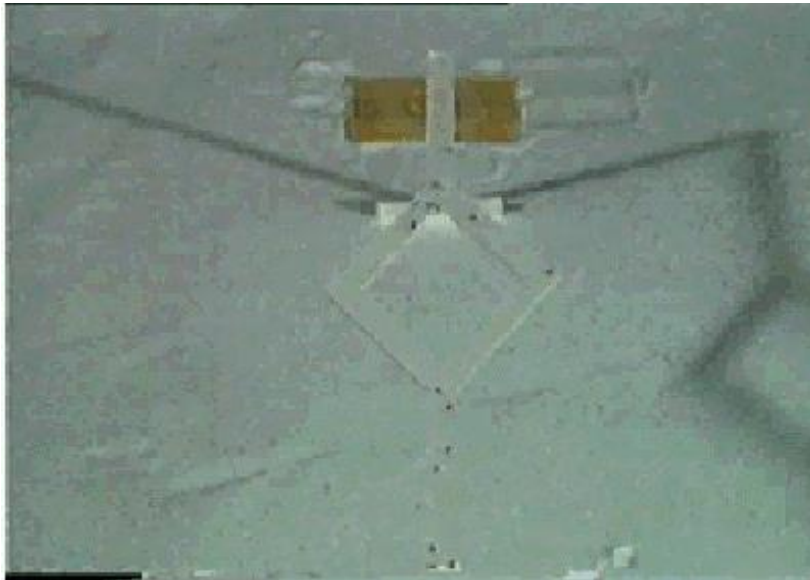
Inspiracja

- **Samorganizacja** o dużym poziomie złożoności odbywa się **bez wyraźnego lidera**.
- Modelowanie zachowania (społecznego) kolonii mrówek zorientowane na aspekt **samorganizacji** może być pomocne przy konstruowaniu modeli rozproszonych rozwiązujących problemy optymalizacyjne – tzw. **Swarm Optimization**.
- **Swarm Optimization** (głównie) **ACO & PSO**.
W mniejszym stopniu **BEES**, **flocks of BIRDS**, **schools of FISH**, **FIREFLIES**

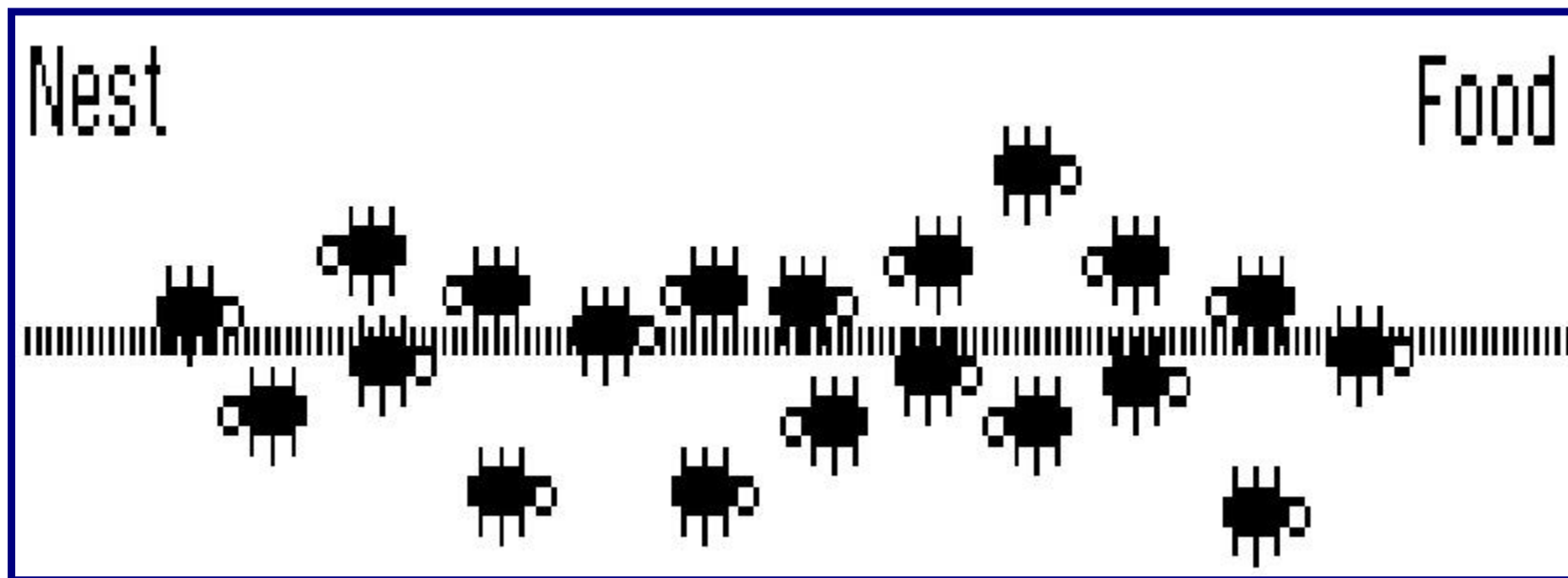
Historia



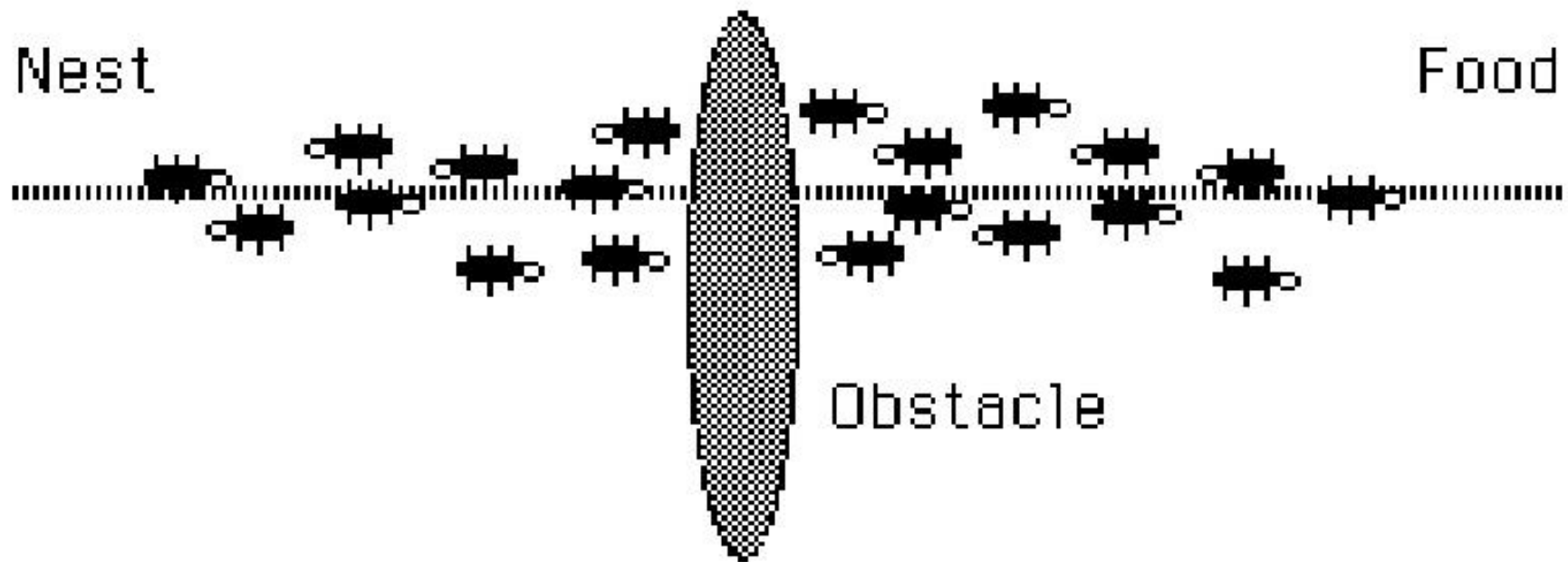
- Goss i in. 1989, Deneuborg i in. 1990, eksperymenty z mrówkami – *double bridge experiment*
- Dorigo i in. 1991, zastosowanie do problemu najkrótszej ścieżki
- Aktualnie: meta-heurystyka do rozwiązywania dyskretnych problemów optymalizacyjnych.



Droga do pożywienia

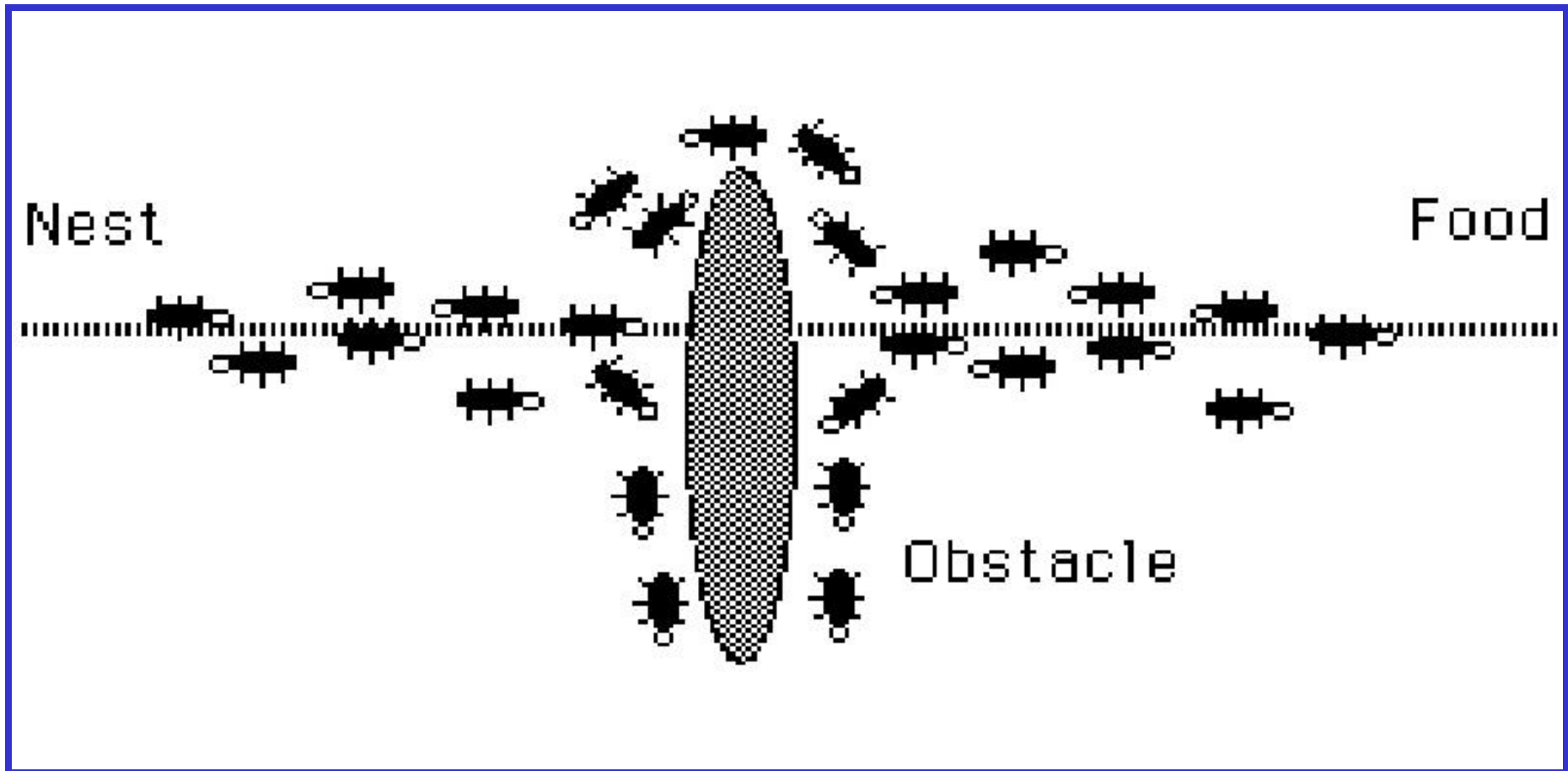


Przeszkoda

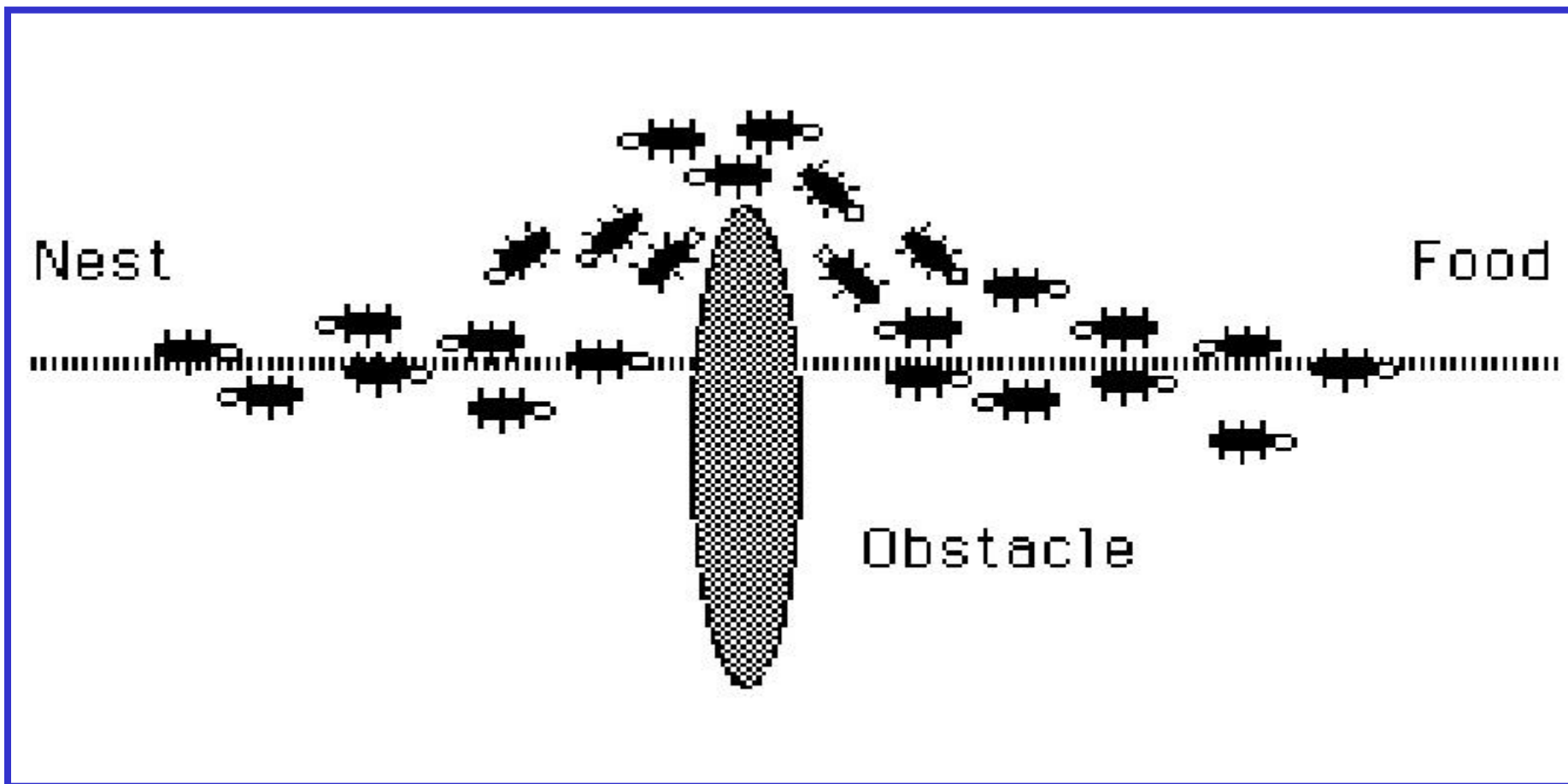


Krótsza ścieżka „zagęszcza się”

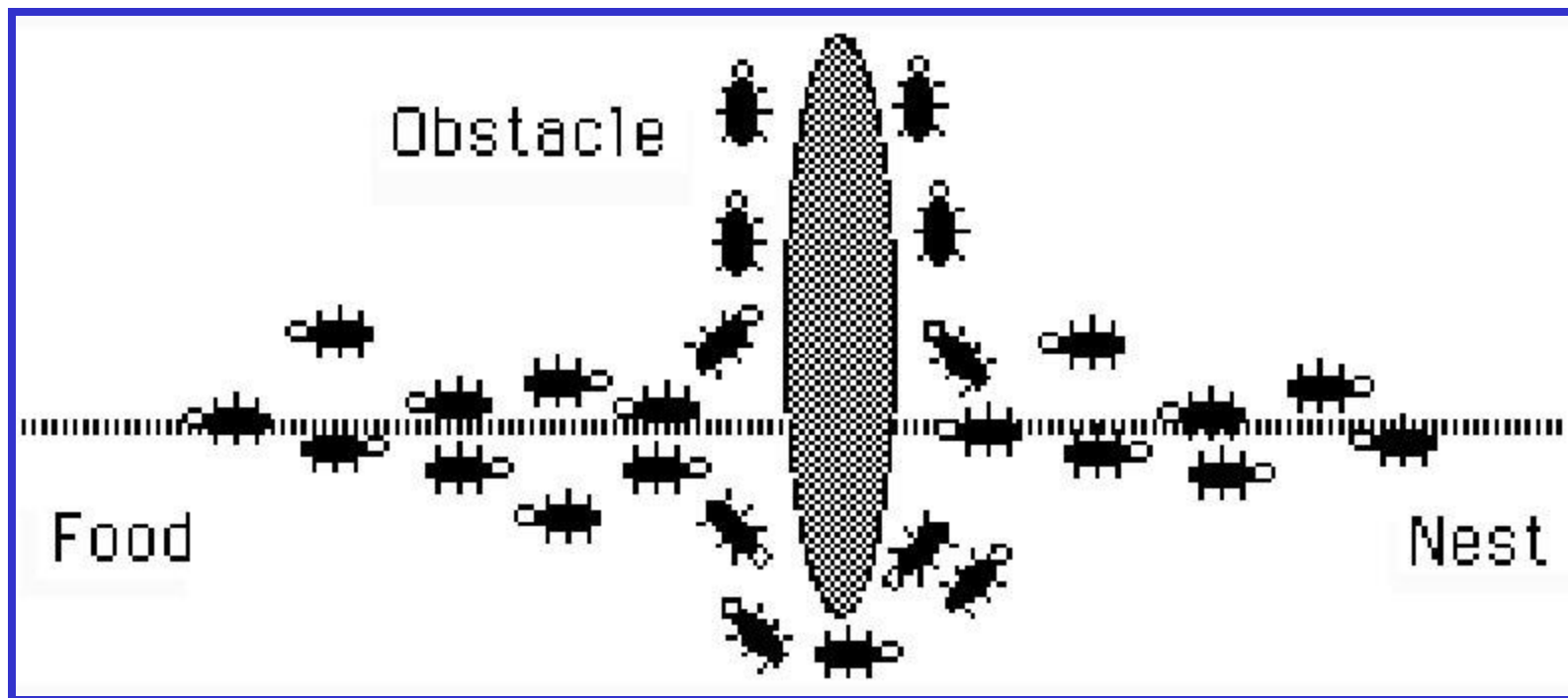
Dla uproszczenia: początkowo $p_{Ai}=p_{Bi}=0.5$



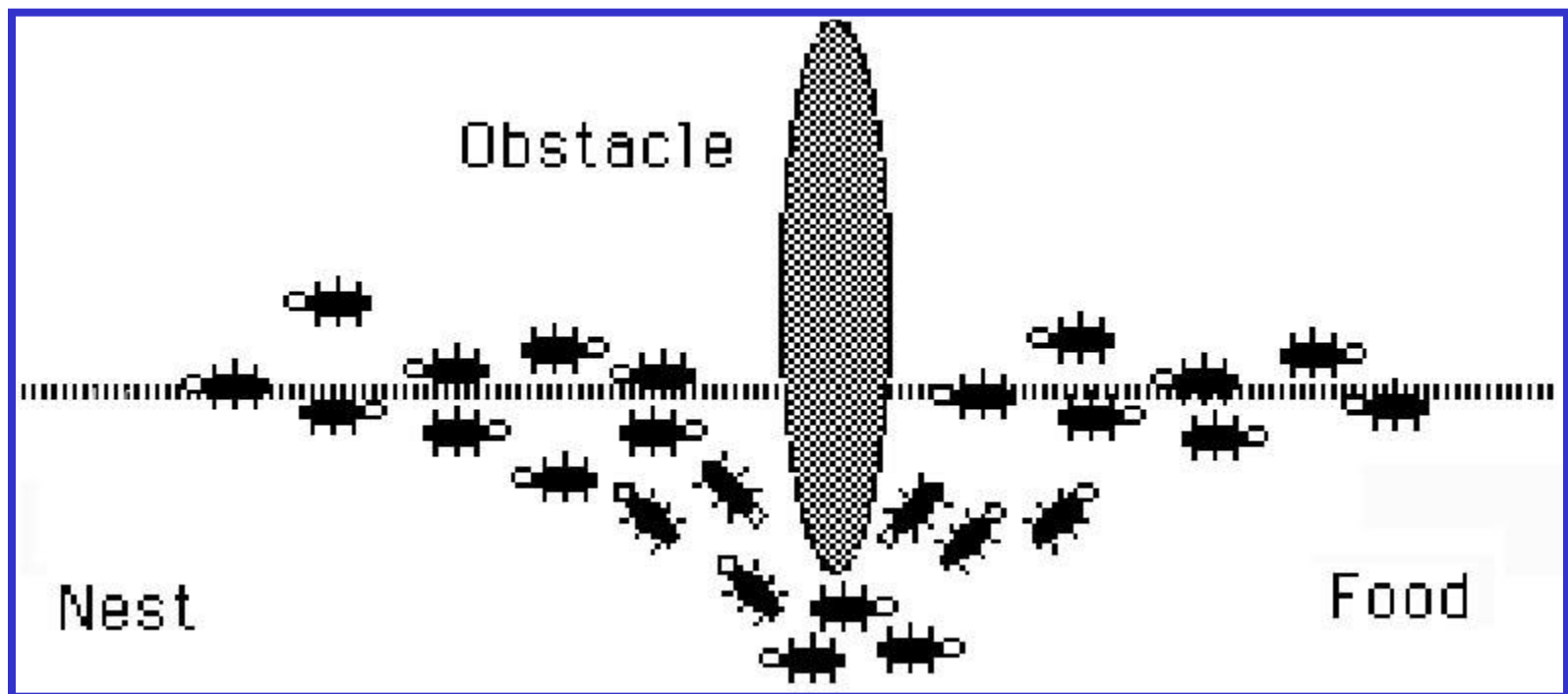
Nowa ścieżka do pożywienia



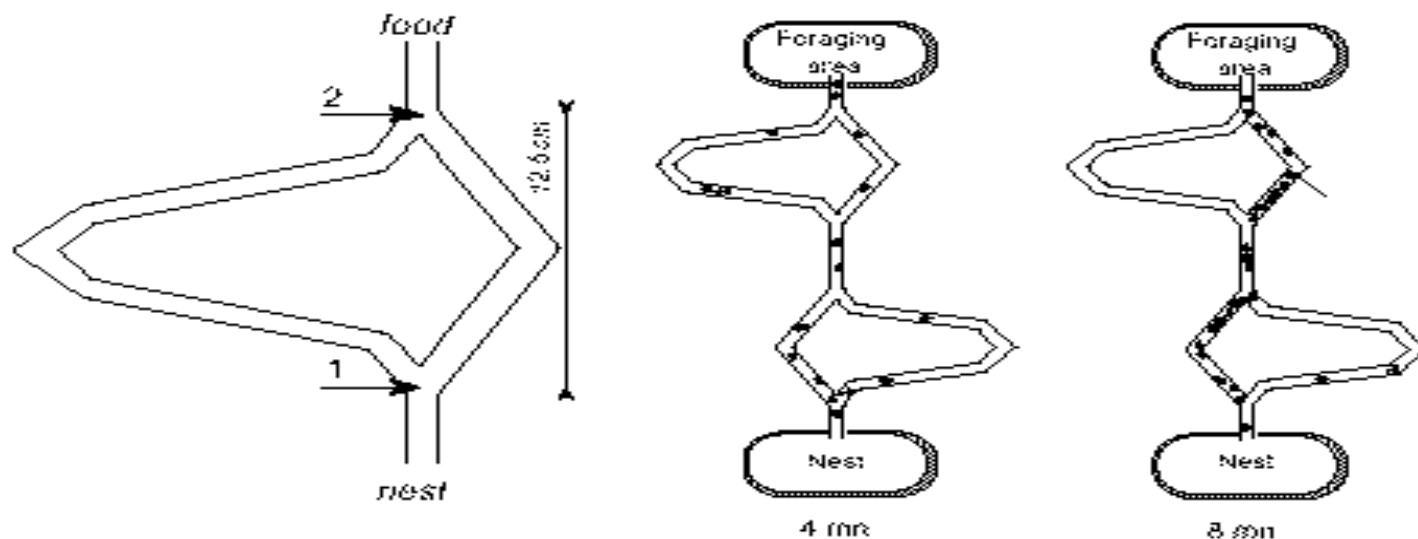
Adaptacja do zmian środowiska



Adaptacja do zmian środowiska



Double bridge experiment – znajdowanie najkrótszej ścieżki



- W drodze do pożywienia oraz w drodze powrotnej, zaznaczają ścieżkę (osadzają feromon).
- Mrówki wybierają ścieżki na podstawie stężenia feromonu.
- Po pewnym czasie najkrótsza ścieżka wybierana jest z największym prawdopodobieństwem.

Problem TSP

Definicja teoriografowa

Złożoność

Algorytm 2-OPT, 3-OPT, k-OPT

Warianty praktyczne TSP

ACO w problemie TSP

Każda mrówka samodzielnie (ale **nie niezależnie**) poszukuje rozwiązania.

Pełna iteracja składa się z następujących kroków:

- [1] Umieść losowo mrówki w miastach.
- [2] Iteracyjnie, dla każdej mrówki:
 - [2a] Wybierz miasto jeszcze nie odwiedzone, aż do momentu gdy droga jest kompletna (rozwiązanie syntaktycznie poprawne).
- [3] Uaktualnij wielkość feromonu na przebytej drodze.
- [4] Zastosuj schemat parowania feromonu.

ACO w problemie TSP

Resetuj feromon

For i=0 to Liczba_Iteracji

{

 Stwórz/resetuj mrówki

 Dla każdej mrówki:

 {

Znajdź rozwiązanie

 Jeśli jest lepsze od najlepszego, to zapamiętaj

 }

Wyparuj część feromonu

Dla każdej mrówki:

 Rozłóż feromon proporcjonalnie do znalezionej rozwiązania

}

Zwróć najlepsze rozwiązanie

ACO w problemie TSP - **wybór miasta**

- Wiele mrówek poszukujących jednocześnie rozwiązania ($m=n$, zwykle *każda mrówka w innym mieście*).
- W każdym kroku każda mrówka wybiera kolejny etap drogi (kolejne miasto).
- Przejście z miasta i do j zależy od:
 1. Listy nie odwiedzonych (przez daną mrówkę) miast (możliwości wyboru kontynuacji).
 2. Odległości pomiędzy miastami. $1/d_{ij}$ reprezentuje lokalną informację – heurystyczną „chęć” do odwiedzenia miasta j o ile mrówka znajduje się w mieście i .
 3. Wartości feromonu $\tau_{ij}(t)$; dla każdej krawędzi reprezentuje „nabytą chęć” odwiedzenia miasta j będąc w mieście i .

ACO w problemie TSP - **wybór miasta**

- Reguła przejścia: prawdopodobieństwo przejścia mrówki k z miasta i do miasta j :

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{\lambda \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}; \quad \eta_{ij} = \frac{1}{d_{ij}}$$

- α i β są odpowiednio dobranymi parametrami.

ACO w problemie TSP - **wybór miasta**

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{\lambda \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}$$

- $\alpha = 0$: algorytm zachłanny,
- $\beta = 0$: natychmiastowy (losowy) wybór drogi, która może nie być optymalna.
- Konieczne jest odpowiednie zrównoważenie tych tendencji.

ACO w problemie TSP – aktualizacja feromonu

- Po znalezieniu rozwiązania mrówki osadzają feromon.
- Wartość uaktualniania ścieżki feromonowej:

Jeżeli $T^k(t)$ jest ścieżką (rozwiązaniem) mrówki k w iteracji t ,
 $L^k(t)$ jest długością tego rozwiązania, Q jest parametrem.

$$\Delta\tau_{ij}^k = Q / L^k(t) \quad \text{gdy } (i, j) \in T^k(t); 0 \quad \text{w.p.p.}$$

- Aktualizacja feromonu (uwzględniająca parowanie):

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

lub

$$\tau_{ij}(t) = (1 - \rho)(\tau_{ij}(t) + \Delta\tau_{ij}(t))$$

ACO w problemie TSP - *Podsumowanie*

- **Idea bazowa:**
 - Budowanie (globalnego) rozwiązania na podstawie rozwiązań lokalnych agentów, przechowując w pamięci dobre rozwiązania częściowe.
 - W celu uniknięcia przedwczesnej zbieżności zastosowanie schematu parowania feromonu.

ACO w problemie TSP - *Implementacja*

- **Kluczowe zagadnienia:**

- Liczba iteracji
- Liczba mrówek
- Początkowa wartość feromonu na krawędziach
- Współczynnik parowania feromonu (ρ)
- Priorytet feromonów (α)
- Priorytet heurystyki (β)

ACO w problemie TSP

- Działa praktycznie bezbłędnie dla problemów do 30 miast
- Dla większych problemów wymaga „dokręcenia śrubek”
- Modyfikacje:
- Mrówki rozkładające feromony:
 - Wszystkie / kilka najlepszych / najlepsza
- Moment rozkładania feromonów:
 - Lokalnie (po każdym przejściu na nowy wierzchołek)
 - Globalnie (po znalezieniu rozwiązania)

ACO w problemie TSP

- Ilość rozkładanego feromonu:
 - Stała (zwykle dodatkowo, przy podejściu elitystycznym)
 - Proporcjonalna do jakości rozwiązania
- Sposób wyboru następnego wierzchołka:
 - Ruletka
 - Pseudo-ruletka:
 - Losujemy φ
 - Jeśli $\varphi \leq \varphi^*$, to wybieramy najlepszego kandydata
 - Jeśli $\varphi > \varphi^*$, to przeprowadzamy „klasyczną” ruletkę

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{\lambda \in J_i^k} [\tau_{i\lambda}(t)]^\alpha \cdot [\eta_{i\lambda}]^\beta};$$

$$\eta_{ij} = \frac{1}{d_{ij}}$$

Elitist Ant System

- Dorigo, 1992
- Mocne wzmacnianie najlepszego rozwiązania
- Po położeniu feromonu przez wszystkie mrówki następuje dodatkowe zwiększenie feromonu na krawędziach należących do najlepszego rozwiązania:

$$e \cdot Q_{\text{best}}$$

e – ilość elitarnych mrówek;

Q_{best} – ilość feromonu na ścieżce najlepszej

DEMO

- Standard – 19 cities - 212,66
- Elitist – 19 cities
- Standard – 50 cities – 361,63
- Elitist – 50 cities

Rank AS

- Bullnheimer, Hartl and Strauss, 1997
- σ mrówek zostawia feromony proporcjonalnie do swojej rangi μ :

$$(\sigma - \mu) \cdot Q_{\mu}$$

Q_{μ} długość rozwiązania o numerze μ w hierarchii, $\mu=0, \dots, \sigma-1$

➔ Najlepsze rozwiązanie jest wzmacniane: $\sigma \cdot Q_{\text{best}}$

Min-Max AS

- Stützle & Hoos, 1997
- Ograniczony zakres feromonów

$$\tau_{ij} \in [\tau_{min}, \tau_{max}]$$

- Tylko najlepsza mrówka w danej iteracji zostawia feromony.
- Początkowa wartość feromonów = τ_{max}

ACS w problemie TSP

- Dalsze modyfikacje w stosunku do AS:
- Nowa reguła uaktualniania feromonu:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t)$$

- Wykorzystanie listy „najbliższych” miast. Wybór spoza listy możliwy jest dopiero po jej wyczerpaniu.
- Połączenie z metodami lokalnego przeszukiwania (2-OPT, 3-OPT).

ACS w problemie TSP

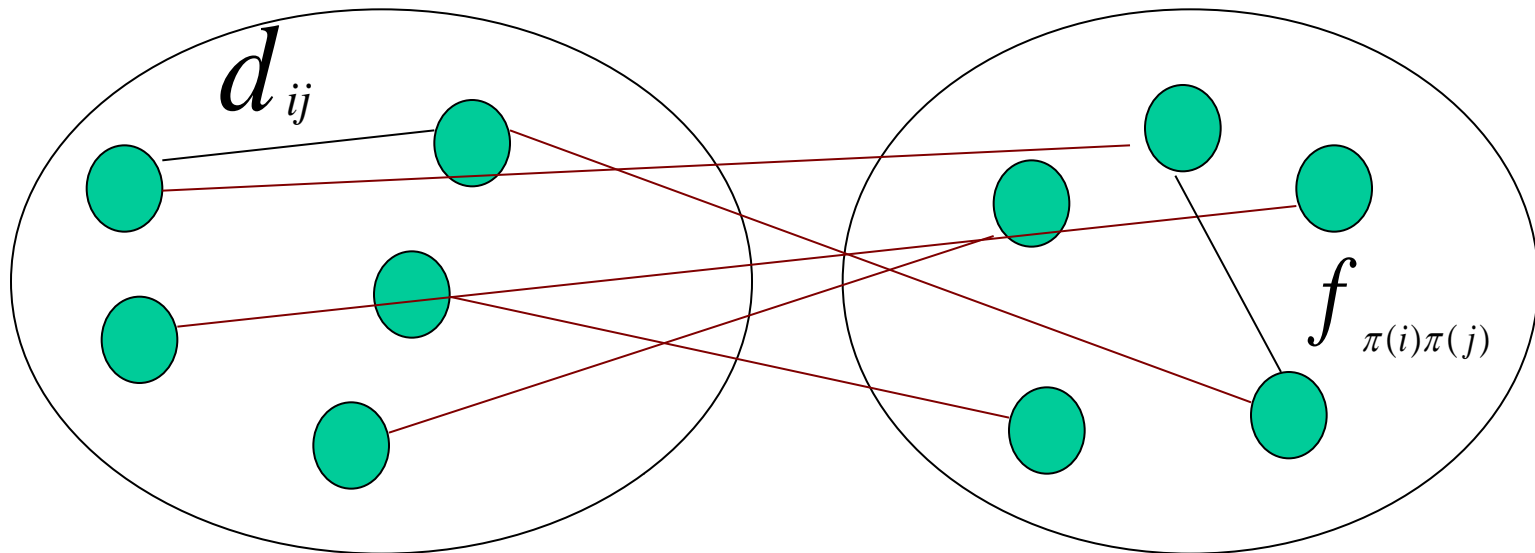
- ACS – metoda (średnio) porównywalna do GA, SA, EP dla problemów o wielkości od 50 do 100 miast.
- Dla większych problemów wyniki nieco słabsze, ale kompensowane szybkością metody

Wyniki – duże zbiory miast (TSPLIB)

Problem name	ACS-3-opt average (length)	ACS-3-opt average (sec)	ACS-3-opt % error	STSP-GA average (length)	STSP-GA average (sec)	STSP-GA % error
d198 (198-city problem)	15,781.7	238	0.01 %	15,780	253	0.00 %
lin318 (318-city problem)	42,029	537	0.00 %	42,029	2,054	0.00 %
alt532 (532-city problem)	27,718.2	810	0.11 %	27,693.7	11,780	0.03 %
rat783 (783-city problem)	8,837.9	1,280	0.36 %	8,807.3	21,210	0.01 %

Quadratic Assignment Problem (best available heuristic)

Przyporządkowanie 1-1 pomiędzy elementami zbiorów (np. lokalizacje i zasoby) minimalizujące określoną funkcję kosztu. (**NPH**)



$$\sum_{i,j=1}^n d_{ij} \cdot f_{\pi(i)\pi(j)}$$

Inne problemy

- Problem plecakowy
- Problem kolorowania grafu
- ...

➔ Dyskretne Problemy Optymalizacyjne

- Problemy dynamiczne!
 - Routing w sieciach teleinformatycznych
 - DCVRP

CVRP

- Capacitated Vehicle Routing Problem (rozwinięcie TSP)
- m klientów z określonym zapotrzebowaniem
- jeden centralny magazyn
- n ciężarówek z ograniczoną pojemnością Q

CEL:

Dostarczyć towar wszystkim klientom minimalizując sumaryczny koszt całej trasy

CVRP

- $G=(V,E)$; $V=\{v_0, v_1, \dots, v_n\}$
- $E=\{(v_i, v_j) \mid v_i, v_j \in V, i < j\}$
- v_0 – *centralny magazyn*,
- q_i – *zapotrzebowanie klienta w lokalizacji v_i ; ($q_i \ll Q$)*

$$\sum_{i=1}^n \sum_{j=1}^n d_{ij} \sum_{k=1}^n x_{ij}^k$$

$$x_{ij}^k = \begin{cases} 1, & (v_i, v_j) \text{ jest fragmentem drogi } k\text{-tej} \\ 0, & \text{w p.p.} \end{cases}$$

CVRP

- Rozwiązanie analogiczne do rozwiązania TSP

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{\lambda \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}; \quad \eta_{ij} = \frac{1}{d_{ij}}$$

- Elityzm
- Rangi
- MinMix

Środowisko dynamiczne (DVRP)

- Zmiany strukturalne:
 - dodawanie / usuwanie wierzchołków
- Zmiany parametryzacji problemu:
 - zmiana zapotrzebowania
 - **zmiana wagi krawędzi**

Zmiany strukturalne, (+/- v)

- Modyfikacje algorytmu wyboru wierzchołka (krawędzi)
 - większa liczba mrówek
 - zmiana algorytmu parowania feromonu
 - większy nacisk na heurystykę (eksploracja) w stosunku do „śladu feromonowego” (eksploatacja)
 - inne modyfikacje (?)
 - metoda UCT (?)

Zmiany parametryczne – wagi krawędzi

→ CVRP_{wTJ}

- Symulacja korka – różne „funkcje korka”
- Rozłożony feromon utrudnia szukanie nowych dróg
- Mechanizm „wyrównywania szans” – zmiana rozkładu feromonu ale z zachowaniem monotoniczności
 - Słabe krawędzie - mała zmiana
 - Silne krawędzie - duża zmiana

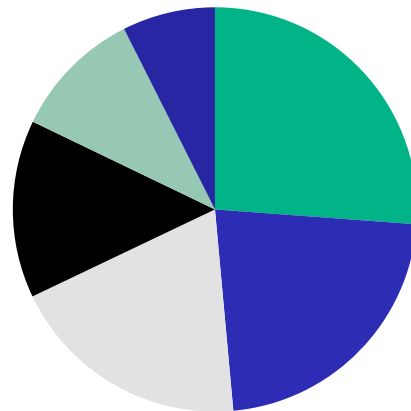
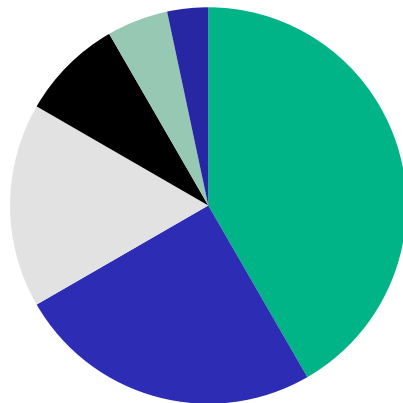
PODEJŚCIE 1 - Wygładzanie feromonu

$$f_{ij} = f_0 \cdot \left(1 + \log_p \frac{f_{ij}}{f_0} \right)$$

f_{ij} – ilość feromonu na wygładzanej krawędzi

f_0 – minimalna ilość feromonu

p - podstawa logarytmu



Wygładzanie feromonu

Różnice między krawędziami są zmniejszone

→ Porządek jest zachowany

W dalszym ciągu najmocniejsza krawędź ma największe prawdopodobieństwo wyboru

Prawdopodobieństwa wyboru krawędzi zbliżyły się do siebie

→ Większa szansa wyboru objazdów

DEMO

Wygładzanie globalne vs. lokalne

- Dla dużego grafu wygładzenie globalne powoduje istotną utratę informacji
- Potrzebny jest mechanizm lokalnego wygładzania
 - zasięg
 - sąsiedztwo?
- Maksymalna odległość od korka
 - wygładzane są tylko te krawędzie, których odległość od korka jest mniejsza od $[s * MaxDist]$, gdzie $s \in [0,1]$
 - $s = 0$ – brak wygładzania
 - $s = 1$ – globalne wygładzanie

Inne pomysły

$$f_{ij} = f_0 \cdot F\left(\frac{f_{ij}}{f_0}\right)$$

- Inne funkcje F
- Różne funkcje korka
 - trapez
 - stała
 - trójkąt
 - Gaussowska

PODEJŚCIE 2

(badania zrealizowane wspólnie z Maciejem Świechowskim)

Testing is performed via repeated run of the method for each **benchmark set**

- **N** repeats with various traffic conditions

Traffic is characterized by 3 distributions:

- Probability of encounter **$p \in \{0.02, 0.05, 0.15\}$** //for each edge
- Length in steps **$TTL := \text{uniform}\langle \text{int} \rangle(2,5)$**
- Intensity **$I := \text{uniform}\langle \text{int} \rangle(10,20)$**

Methods have no information of the traffic before it occurs, but have access to the probability distributions used to generate the traffic.

Modifications to the ant algorithm

Inspired by the classical algorithm to solve a TSP problem.

Each ant finds a complete solution

- but the main simulation takes only one drive-through from each route of the best solution

Initialization at the start:

- static initial solution produces initial pheromone deposits (**Clarke & Wright**)

Initialization in each step of the main simulation:

- clear the best-found solution and reset the pheromone to the initial values (!)

Initialization before each ant iteration:

- synchronize the complete state (vehicles locations, capacities left, unvisited cities, traffic conditions) with the current main simulation state.

The ant algorithm

Roulette selection (probability = 0.95, otherwise: greedy choice):

- Chose among the legal locations (unvisited customer, not exceeded vehicle capacity):

$$\text{probability} - p_{ij} = \frac{\tau_{ij}^{\alpha} * \eta_{ij}^{\beta}}{\sum (\tau_{ij}^{\alpha} * \eta_{ij}^{\beta})}, \eta_{ij} = \left(\frac{BASE}{d_{ij}} \right)^2$$

d_{ij} - current cost of the edge, $\alpha=2$ $\beta = 3$

BASE: the length of the initial (static solution)

The pheromone values are updated after a completed iteration:

$$\tau_{ij}(t+1) = Conf(0.1 * \tau(t)_{ij} + \sum_{a \in Ants} (\delta_{ij} * Q_a)), \quad \delta_{ij} \in \{0,1,10\}$$

0 – nie należy do żadnego rozwiązania

1 – należy ale nie do najlepszego (w tej iteracji)

10 – należy do najlepszego rozwiązania (w tej iteracji)

Conf obcina do wartości $[tau(min), tau(max)]$

Instance	P	Static (σ)	Ants (σ)	UCT (σ)
P19	0.02	388.9 (214.9)	281.2 (46.9)	244.9 (10.9)
P19	0.05	612.0 (213.3)	311.8 (93.1)	269.3 (23.1)
P19	0.15	1278.0 (358.9)	391.2 (155.9)	340.9 (61.3)
P45	0.02	1007.7 (326.2)	607.6 (53.9)	601.0 (20.7)
P45	0.05	1759.6 (411.9)	682.0 (74.4)	646.8 (35.7)
P45	0.15	3299.5 (733.3)	949.7 (281.7)	781.8 (54.3)
E51	0.02	989.2 (240.6)	614.1 (40.0)	615.6 (22.2)
E51	0.05	1571.6 (386.3)	650.1 (50.4)	667.1 (40.5)
E51	0.15	3509.7 (824.7)	789.9 (174.8)	845.4 (70.2)
A54	0.02	1939.2 (542.7)	1338.7 (84.0)	1254.9 (41.3)
A54	0.05	3072.4 (887.7)	1456.4 (286.0)	1347.5 (77.9)
A54	0.15	6275.0 (1441.9)	1829.0 (519.4)	1647.6 (139.2)
A69	0.02	2005.7 (531.8)	1395.9 (96.7)	1265.2 (43.3)
A69	0.05	3235.4 (644.1)	1538.1 (294.5)	1377.3 (103.0)
A69	0.15	6631.7 (1437.4)	2096.4 (588.4)	1731.8 (173.1)
E76	0.02	1318.7 (295.5)	746.0 (54.4)	779.0 (32.6)
E76	0.05	2130.0 (460.4)	826.7 (159.9)	834.7 (39.8)
E76	0.15	4536.7 (838.0)	1037.2 (264.3)	1085.6 (68.9)
A80	0.02	2774.1 (625.2)	1907.1 (146.7)	1929.3 (58.4)
A80	0.05	4100.6 (830.1)	2003.3 (442.6)	2063.8 (121.9)
A80	0.15	9066.5 (1437.4)	3161.8 (829.6)	2588.4 (141.7)
P101	0.02	1436.5 (262.6)	846.8 (69.2)	815.0 (25.4)
P101	0.05	2552.0 (547.3)	893.8 (127.3)	891.6 (38.2)
P101	0.15	5419.6 (801.5)	1375.0 (322.2)	1204.0 (81.9)
C150D	0.02	1883.0 (368.8)	1297.0 (75.5)	1202.0 (39.1)
C150D	0.05	3099.1 (587.5)	1392.5 (193.3)	1318.7 (53.5)
C150D	0.15	6766.7 (840.1)	1987.5 (492.4)	1810.2 (110.6)
Tai150b	0.02	4994.7 (1165.5)	4367.5 (515.0)	3021.5 (100.7)
Tai150b	0.05	8751.9 (1936.7)	4834.3 (836.4)	3270.1 (176.3)
Tai150b	0.15	18104.0 (2581.2)	7081.4 (1715.2)	4442.5 (285.5)
Best result count		0 (0)	8 (0)	22 (30)

The comparison of the **UCT method**, **Ant Colony Optimization** and **no optimization**.

The number of iterations of each ant per step:

- **200** for $n < 69$
- **100** for $n = 69$
- **75** for $n > 69$

Podsumowanie DCVRP

- Parametry systemu (liczba mrówek, elityzm?, rangi?, ...)
- Parowanie feromonu
- Eksploracja vs. eksploatacja
- Adaptacja
- Zmiany strukturalne
- Zmiany parametryczne
- Różne funkcje korka
- Zasięg lokalnych modyfikacji (wygładzania)
- ...
- Algorytm bazowy jest elastyczny ze swej natury

Cechy algorytmów mrówkowych

- Łatwość redefiniowania celu (adaptacja)
- Odporność na uszkodzenia poszczególnych elementów (agentów)
- Decentralizacja
- Samoorganizacja

Podsumowanie

- ACO (ACS) jest obiecującą heurystyką, szczególnie w przypadku zmienności struktury środowiska i/lub dynamicznie generowanych dodatkowych ograniczeń
- Może być stosowana jedynie do określonej grupy „problemów grafowych”
- Stosunkowo mała podbudowa teoretyczna.

Pytania?