

Ant Colony Optimization

Maciej Nemś

20 maja 2021

Zewnętrzny stymulant

W latach czterdziestych i pięćdziesiątych *XX* wieku francuski entomolog Pierre-Paul Grassé zaobserwował, że niektóre gatunki termitów mocno reagują na zewnętrzne bodźce w środowisku. Bodźce te były produkowane przez niektóre owady z kolonii, natomiast reagowała na nie cała kolonia. Ten sposób komunikacji został nazwany Stygmergią.

Zewnętrzny stymulant

W latach czterdziestych i pięćdziesiątych *XX* wieku francuski entomolog Pierre-Paul Grassé zaobserwował, że niektóre gatunki termitów mocno reagują na zewnętrzne bodźce w środowisku. Bodźce te były produkowane przez niektóre owady z kolonii, natomiast reagowała na nie cała kolonia. Ten sposób komunikacji został nazwany Stygmergią.

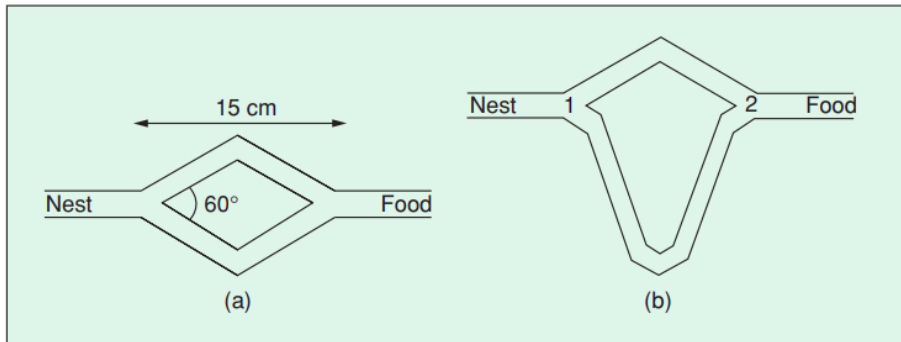
Stygmergia

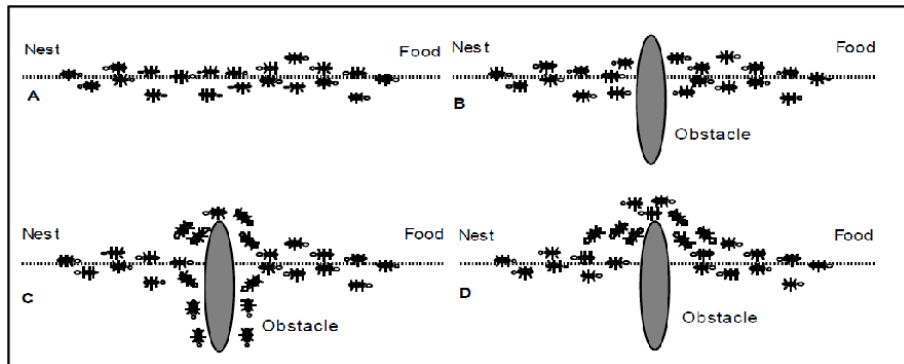
Charakteryzuje się dwiema cechami:

- jest to komunikacja poprzez zmianę środowiska
- zmiana jest lokalna, czyli mają do niej dostęp jedynie owady będące w pobliżu miejsca zmiany

Na początku mrówki wyruszają w poszukiwaniu pożywienia. Po znalezieniu celu, zostawiają feromony na drodze powrotnej do kolonii. W ten sposób mrówki, które znalazły krótszą ścieżkę częściej będą zostawiać feromony na ścieżce. Przez to więcej owadów będzie wybierało tę krótszą ścieżkę.

Na początku mrówki wyruszają w poszukiwaniu pożywienia. Po znalezieniu celu, zostawiają feromony na drodze powrotnej do kolonii. W ten sposób mrówki, które znalazły krótszą ścieżkę częściej będą zostawiać feromony na ścieżce. Przez to więcej owadów będzie wybierało tę krótszą ścieżkę. Jako prosty eksperyment przedstawiono poniższą sytuację. W przypadku (a) mrówki dzieliły się po równo na obie ścieżki. W przypadku (b) po pewnym czasie mrówki chodziły prawie jedynie krótszą ścieżką.





Swarm Intelligence

Zdecentralizowany zbiór wielu samodzielnych systemów. Zazwyczaj polega na wielu agentach, którzy wymieniają się lokalnie danymi między sobą oraz mogą wpływać na swoje środowisko.

Ant Colony Optimization

Jest to probabilistyczna metoda rozwiązywania problemów optymalizacyjnych. Jest inspirowana zachowaniem mrówek. Metoda opiera się na obliczeniach wielu agentów, którzy komunikują się ze sobą za pomocą modyfikowania środowiska (podobne do mrówek zostawiających feromony). Metoda ta została rozpowszechniona przez Marco Dorigo.

Model reprezentacji problemu optymalizacji kombinatorycznej

$$P = (\mathcal{S}, \Omega, f)$$

\mathcal{S} - przestrzeń szukania, zdefiniowana poprzez skończoną liczbę dyskretnych zmiennych decyzyjnych.

Ω - zbiór ograniczeń na zmiennych

f - funkcja celu $\mathcal{S} \rightarrow \mathbb{R}_0^+$, która ma zostać zminimalizowana

Ant Colony Optimization algorithm

inicjalizacja;

while *warunek ukończenia nie jest spełniony* **do**

 Skonstruuj rozwiązania mrówek;

 Wykonaj lokalne szukanie;

 Zaktualizuj feromony;

end

Mamy m mrówek. Na początku każda ma rozwiązanie $s_p = \emptyset$.

W każdym kroku mrówka powiększa swoje rozwiązanie o jakiś element

$c_i^j \in N(s_p)$

τ_{ij} - to feromony przypisane wyborowi danego elementu

$N(s_p)$ to zbiór elementów, które można dodać do rozwiązania s_p .

$\eta(\cdot)$ - funkcja przypisująca heurystyczną wartość elementom c_i^j .

α, β - parametry regulujące wpływ heurystycznej wartości elementu oraz wpływu szlaku feromonów.

Prawdopodobieństwo wyboru

$$p(c_i^j | s_p) = \frac{\tau_{ij}^\alpha \cdot [\eta(c_i^j)]^\beta}{\sum_{c_i^l \in N(s_p)} \tau_{il}^\alpha \cdot [\eta(c_i^l)]^\beta}, \quad \forall c_i^j \in N(s_p)$$

W tym kroku wybierane są najlepsze rozwiązania mrówek. Jest to coś, czego mrówki nie mogą zrobić osobno.

τ_{ij} - feromony dla wyboru c_i^j

ρ - współczynnik zanikania

$\Delta\tau_{ij}^k$ - jest to zmiana feromonów zostawiona przez mrówkę k .

Przykładowo (Q to stała, L_k , to koszt rozwiązania mrówki k):

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{jeśli } k \text{ używa } c_i^j \\ 0 & \text{wpp} \end{cases}$$

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_k^m \Delta\tau_{ij}^k$$

Ant System dla TSP

Rozwiązuje Traveling Salesman Problem. Jest to jeden z pierwszych i najprostszych zastosowań metaheurystyki ACO. Jest jednak istotnie gorszy od najlepszych znanych rozwiązań.

AntNet dla network routing

Rozwiązuje problem routing'u pakietów w sieci. Osiąga dobre wyniki w porównaniu do konkurencji. Jest to spowodowane tym, że *ACO* dobrze radzi sobie z adaptacją do dynamicznych zmian w zadaniu, czyli przykładowo do zmian w obciążeniu sieci podczas routing'u. Algorytm może cały czas działać na bieżąco i reagować na zmiany.

AS dla problemu TSP

Każda mrówka początkowo zaczyna w losowym miejscu w grafie.

AS dla problemu TSP

Każda mrówka początkowo zaczyna w losowym miejscu w grafie.

Mrówki posiadają pamięć, w której trzymają częściowe rozwiązanie.

Początkowo trzymają jedynie miasto startowe.

AS dla problemu TSP

Każda mrówka początkowo zaczyna w losowym miejscu w grafie.

Mrówki posiadają pamięć, w której trzymają częściowe rozwiązanie.

Początkowo trzymają jedynie miasto startowe.

Mrówki przemieszczają się od miasta do miasta, przechodząc jedynie przez nieodwiedzone miasta i zapisując w pamięci częściowe rozwiązania.

Każda mrówka początkowo zaczyna w losowym miejscu w grafie.

Mrówki posiadają pamięć, w której trzymają częściowe rozwiązanie.

Początkowo trzymają jedynie miasto startowe.

Mrówki przemieszczają się od miasta do miasta, przechodząc jedynie przez nieodwiedzone miasta i zapisując w pamięci częściowe rozwiązania.

Miasta wybieramy z prawdopodobieństwem:

$$p(d_{ij}|s_p) = \frac{\tau_{ij}^\alpha \cdot [1/d_{ij}]^\beta}{\sum_{d_{ij} \in N(s_p)} \tau_{il}^\alpha \cdot [1/d_{ij}]^\beta}$$

Następnie aktualizujemy feromony za pomocą:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_k^m \frac{1}{L_k}$$

Routing

Mamy graf, który reprezentuje sieć telekomunikacyjną. Celem AntNet jest znalezienie minimalnej ścieżki między każdą parą wierzchołków.

Dodatkowo wagi na krawędziach mogą się zmieniać wraz ze zmianą czasu zależnie od tego, ile danych jest przez nie przesyłanych.

Routing

Mamy graf, który reprezentuje sieć telekomunikacyjną. Celem AntNet jest znalezienie minimalnej ścieżki między każdą parą wierzchołków.

Dodatkowo wagi na krawędziach mogą się zmieniać wraz ze zmianą czasu zależnie od tego, ile danych jest przez nie przesyłanych.

Każda mrówka początkowo zaczyna w losowym miejscu w grafie i ma dotrzeć w losowe miejsce.

Routing

Mamy graf, który reprezentuje sieć telekomunikacyjną. Celem AntNet jest znalezienie minimalnej ścieżki między każdą parą wierzchołków.

Dodatkowo wagi na krawędziach mogą się zmieniać wraz ze zmianą czasu zależnie od tego, ile danych jest przez nie przesyłanych.

Każda mrówka początkowo zaczyna w losowym miejscu w grafie i ma dotrzeć w losowe miejsce.

Mrówki podobnie jak w AS chodzą po grafie i za pomocą feromonów wybierają przejścia z odpowiednią dystrybucją. Mrówka kończy chodzić w momencie dotarcia do celu.

Routing

Mamy graf, który reprezentuje sieć telekomunikacyjną. Celem AntNet jest znalezienie minimalnej ścieżki między każdą parą wierzchołków.

Dodatkowo wagi na krawędziach mogą się zmieniać wraz ze zmianą czasu zależnie od tego, ile danych jest przez nie przesyłanych.

Każda mrówka początkowo zaczyna w losowym miejscu w grafie i ma dotrzeć w losowe miejsce.

Mrówki podobnie jak w AS chodzą po grafie i za pomocą feromonów wybierają przejścia z odpowiednią dystrybucją. Mrówka kończy chodzić w momencie dotarcia do celu.

Tym razem zamiast τ_{ij} mamy τ_{ijd} , gdzie d oznacza cel danej mrówki.

Każda krawędź posiada heurystyczną wartość niezależną od celu mrówki.
Przykładowo:

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum_{l \in N_i} q_{il}}$$

gdzie q_{ij} , to liczba bitów czekających na przesłanie między i a j .

Każda krawędź posiada heurystyczną wartość niezależną od celu mrówki.
Przykładowo:

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum_{l \in N_i} q_{il}}$$

gdzie q_{ij} , to liczba bitów czekających na przesłanie między i a j .

Przemierzając się po grafie mrówki symulują opóźnienia, które miałyby przesyłane pakiety.

Każda krawędź posiada heurystyczną wartość niezależną od celu mrówki.
Przykładowo:

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum_{l \in N_i} q_{il}}$$

gdzie q_{ij} , to liczba bitów czekających na przesłanie między i a j .

Przemierzając się po grafie mrówki symulują opóźnienia, które miałyby przesyłane pakiety.

Do liczenia jakości ścieżki może więc zostać wykorzystane T_{sd} , czyli czas, jaki zajęło przedostanie się ze źródła do celu. Dodatkowo do obliczenia jakości trzeba wykorzystać obciążenie sieci.

Każda krawędź posiada heurystyczną wartość niezależną od celu mrówki.
Przykładowo:

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum_{l \in N_i} q_{il}}$$

gdzie q_{ij} , to liczba bitów czekających na przesłanie między i a j .

Przemierzając się po grafie mrówki symulują opóźnienia, które miałyby przesyłane pakiety.

Do liczenia jakości ścieżki może więc zostać wykorzystane T_{sd} , czyli czas, jaki zajęło przedostanie się ze źródła do celu. Dodatkowo do obliczenia jakości trzeba wykorzystać obciążenie sieci.

Na koniec mrówka wraca się po odwiedzonych wierzchołkach i aktualizuje feromony na krawędziach.

Rozpoznawanie krawędzi



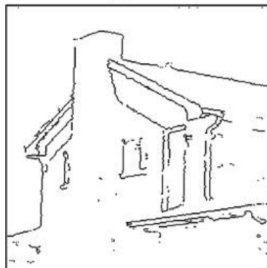
a1



a2



a3



Jako graf jest traktowany cały obrazek. Pojedynczy wierzchołek, to pojedynczy piksel. Mrówki poruszają się piksel po pikselu.

Jako graf jest traktowany cały obrazek. Pojedynczy wierzchołek, to pojedynczy piksel. Mrówki poruszają się piksel po pikselu.

Mrówki są rozmieszczane na losowych pikselach obrazka.

Jako graf jest traktowany cały obrazek. Pojedynczy wierzchołek, to pojedynczy piksel. Mrówki poruszają się piksel po pikselu.

Mrówki są rozmieszczane na losowych pikselach obrazka.

Podobnie jak w poprzednich algorytmach mrówki wybierają następny wierzchołek grafu za pomocą feromonu oraz heurystycznej wartości. Dla danego piksela c :

$$\begin{array}{ccc} lg & g & pg \\ lc & c & pc \\ ld & d & pd \end{array}$$





heurystyczna wartość jest przeliczana jako

$$\frac{\max(|lg - pd|, |pg - ld|, |g - d|, |lc - pc|)}{I_{max}}$$

gdzie wartości, to natężenie czarnego koloru w pikselu (obraz jest biało-czarny).

<i>Problem type</i>	<i>Problem name</i>	<i>Authors</i>	<i>Year</i>
Routing	Traveling salesman	Dorigo et al.	1991, 1996
		Dorigo & Gambardella	1997
		Stützle & Hoos	1997, 2000
	TSP with time windows	López Ibáñez et al.	2009
	Sequential ordering	Gambardella & Dorigo	2000
	Vehicle routing	Gambardella et al.	1999
		Reimann et al.	2004
		Favoretto et al.	2007
		Fuellerer et al.	2009
		Multicasting	Hernández & Blum
Assignment	Quadratic assignment	Maniezzo	1999
		Stützle & Hoos	2000
	Frequency assignment	Maniezzo & Carbonaro	2000
	Course timetabling	Socha et al.	2002,2003
	Graph coloring	Costa & Hertz	1997
Scheduling	Project scheduling	Merkle et al.	2002
		den Besten et al.	2000
	Weighted tardiness	Merkle & Middendorf	2000
		Stützle	1997
	Flow shop	Rajendran, Ziegler	2004
		Blum	2005
	Open shop	Blum	2005
Car sequencing	Solnon	2008	
Subset	Set covering	Lessing et al.	2004
	<i>l</i> -cardinality trees	Blum & Blesa	2005
	Multiple knapsack	Leguizamón & Michalewicz	1999
	Maximum clique	Solnon, Fenet	2006
Machine learning	Classification rules	Parpinelli et al.	2002
		Martens et al.	2006
		Otero et al.	2008
	Bayesian networks	Campos, Fernández-Luna	2002
	Neural networks	Socha, Blum	2007
Bioinformatics	Protein folding	Shmygelska & Hoos	2005
	Docking	Korb et al.	2006
	DNA Sequencing	Blum et al.	2008
	Haplotype Inference	Benedettini et al.	2008

ANTS International Conference on Swarm Intelligence

-  M. Dorigo, M. Birattari, and T. Stützle.
Ant colony optimization.
Computational Intelligence Magazine, IEEE, 1:28--39, 12 2006.
-  M. Dorigo and C. Blum.
Ant colony optimization theory: A survey.
Theoretical Computer Science, 344(2):243--278, 2005.
-  M. Dorigo and T. Stützle.
Ant Colony Optimization: Overview and Recent Advances, volume 146, pages 227--263.
09 2010.
-  H. Nezamabadi-pour, S. Saryazdi, and E. Rashedi.
Edge detection using ant algorithms.
Soft Comput., 10:623--628, 05 2006.