A biomimetic algorithm for combinatorial optimization problems

Ant Colony Optimization

Duanmu Mei
Gourgoulias Konstantinos

UMass Amherst

December 14, 2012

The basics : What is a combinatorial optimization problem?

Generally, we need to pick an object out of a finite set $\mathcal{S}$, so that it minimizes a certain 'cost' function $f : \mathcal{S} \to \mathbb{R}$.

The basics : What is a combinatorial optimization problem?

Generally, we need to pick an object out of a finite set $\mathcal{S}$, so that it minimizes a certain 'cost' function $f : \mathcal{S} \to \mathbb{R}$.

Problem : $\mathcal{S}$ is usually very large.

The basics : What is a combinatorial optimization problem?

Generally, we need to pick an object out of a finite set $\mathcal{S}$, so that it minimizes a certain 'cost' function $f : \mathcal{S} \to \mathbb{R}$.

Problem : $\mathcal{S}$ is usually very large.

Searching for the optimal solution is often very expensive!

The basics : What is a combinatorial optimization problem?

Generally, we need to pick an object out of a finite set $\mathcal{S}$, so that it minimizes a certain 'cost' function $f : \mathcal{S} \to \mathbb{R}$.

Problem : $\mathcal{S}$ is usually very large.

Searching for the optimal solution is often very expensive!

A famous example : Travelling salesman problem in Sweden

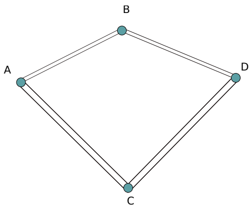
Solved by 96 dual core Intel Xeon processors at GeorgiaTech.

Running time : March 2003 to May 2004.

Figure: Full picture here.

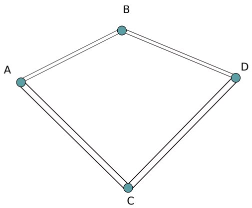
What do ants have to do with anything?

Example : Double bridge experiment



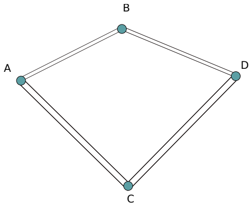
- N ants move from A to D and back.

Example : Double bridge experiment



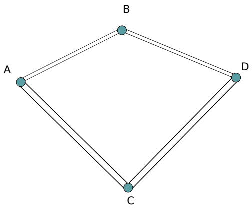
- N ants move from A to D and back.
- Rules :

Example : Double bridge experiment



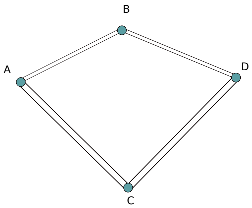
- N ants move from A to D and back.
- Rules :
    1. Once an ant picks a path, it can't return back until it reaches D.

Example : Double bridge experiment



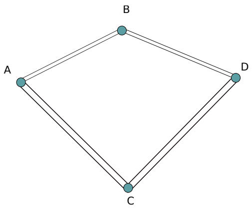
- N ants move from A to D and back.
- Rules :
  1. Once an ant picks a path, it can't return back until it reaches D.
  2. Once an ant reaches D, it has to return to A from the same path.

Example : Double bridge experiment



- N ants move from A to D and back.
- Rules :
  1. Once an ant picks a path, it can't return back until it reaches D.
  2. Once an ant reaches D, it has to return to A from the same path.
- Goal : To find the shorter route between the two paths.

Example : Double bridge experiment



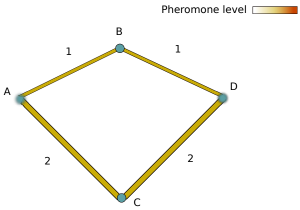
- N ants move from A to D and back.
- Rules :
  1. Once an ant picks a path, it can't return back until it reaches D.
  2. Once an ant reaches D, it has to return to A from the same path.
- Goal : To find the shorter route between the two paths.
- As in the video, the ants will choose between the two paths based on the "pheromone" level of each one.

Example : Double bridge experiment

Let's see what happens!

Example : Double bridge experiment

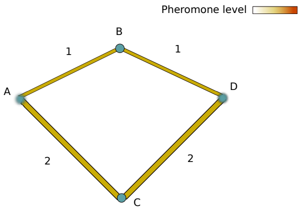
Let's see what happens!



Initially we have no information about which route is "better".

Example : Double bridge experiment

Let's see what happens!



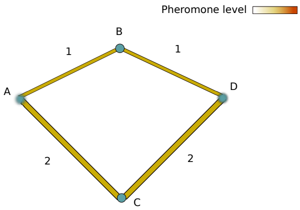
Initially we have no information about which route is "better".

N/2 of ants pick A-B.

The rest pick the other one.

Example : Double bridge experiment

Let's see what happens!



Initially we have no information about which route is "better".

N/2 of ants pick A-B.

The rest pick the other one.

After the return trip, we calculate the pheromone they deposited at each path.

Example : Double bridge experiment

Pheromone Update:

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/L_k & \text{if ant k used edge (i,j) in its tour.} \\ 0 & \text{otherwise} \end{cases}$$

L_k is the length of the tour constructed by ant k, $k = 1, \ldots, N$.

Let $0 < p < 1$ be the evaporation rate of the pheromones.

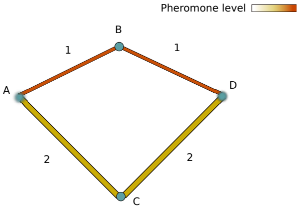
Pheromone Update:

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/L_k & \text{if ant k used edge (i,j) in its tour.} \\ 0 & \text{otherwise} \end{cases}$$

L_k is the length of the tour constructed by ant k, $k = 1, \ldots, N$.

Let $0 < p < 1$ be the evaporation rate of the pheromones.

- Path A-B-D, $L_k = 4$. So $\tau_{AB} = (1-p)\epsilon + \dfrac{N}{8} = \tau_{BD}$



Example : Double bridge experiment

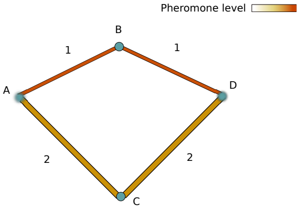
Pheromone Update:

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/L_k & \text{if ant k used edge (i,j) in its tour.} \\ 0 & \text{otherwise} \end{cases}$$

L_k is the length of the tour constructed by ant k, $k = 1, \ldots, N$.

Let $0 < p < 1$ be the evaporation rate of the pheromones.

- Path A-B-D, $L_k = 4$. So
  $\tau_{AB} = (1-p)\epsilon + \dfrac{N}{8} = \tau_{BD}$
- Path A-C-D, $L_k = 8$. So
  $\tau_{AC} = (1-p)\epsilon + \dfrac{N}{16} = \tau_{CD}$



Example : Double bridge experiment

$$P_{AB}^{k} = \frac{\tau_{AB}}{\tau_{AB} + \tau_{AC}} = \frac{\frac{1}{2}\epsilon + \frac{N}{8}}{\frac{1}{2}\epsilon + \frac{N}{8} + \frac{1}{2}\epsilon + \frac{N}{16}} = \frac{\frac{1}{2}\epsilon + \frac{N}{8}}{\epsilon + \frac{3N}{16}}$$

Example : Double bridge experiment

$$P_{AB}^{k} = \frac{\tau_{AB}}{\tau_{AB} + \tau_{AC}} = \frac{\frac{1}{2}\epsilon + \frac{N}{8}}{\frac{1}{2}\epsilon + \frac{N}{8} + \frac{1}{2}\epsilon + \frac{N}{16}} = \frac{\frac{1}{2}\epsilon + \frac{N}{8}}{\epsilon + \frac{3N}{16}}$$

$$P_{AC}^{k} = \frac{\tau_{AC}}{\tau_{AB} + \tau_{AC}} = \frac{\frac{1}{2}\epsilon + \frac{N}{16}}{\frac{1}{2}\epsilon + \frac{N}{8} + \frac{1}{2}\epsilon + \frac{N}{16}} = \frac{\frac{1}{2}\epsilon + \frac{N}{16}}{\epsilon + \frac{3N}{16}}$$

Ant System Algorithm

Step 1: Initialize Pheromone Construct Initial pheromone matrix, for each entry $\tau_{ij} = \epsilon$ which represent the pheromone between two nodes i and j.

Ant System Algorithm

Step 1: Initialize Pheromone Construct Initial pheromone matrix, for each entry $\tau_{ij} = \epsilon$ which represent the pheromone between two nodes i and j.

Step 2: Update pheromone. $\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$

Here, $\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant k used edge (i,j) in its tour.} \\ 0 & \text{otherwise} \end{cases}$

Ant System Algorithm

Step 1: Initialize Pheromone Construct Initial pheromone matrix, for each entry $\tau_{ij} = \epsilon$ which represent the pheromone between two nodes i and j.

Step 2: Update pheromone. $\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$

Here, $\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant k used edge (i,j) in its tour.} \\ 0 & \text{otherwise} \end{cases}$

Step 3: Update probability to choose path from i to j.

$$P_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{C_{il} N(s^p)} \tau_{il}^{\alpha}\eta_{il}^{\beta}} & \text{if} C_{ij} \in N(s^p). \\ 0 & \text{otherwise} \end{cases}$$

Ant System Algorithm

Step 1: Initialize Pheromone Construct Initial pheromone matrix, for each entry $\tau_{ij} = \epsilon$ which represent the pheromone between two nodes i and j.

Step 2: Update pheromone. $\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$

Here, $\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant k used edge (i,j) in its tour.} \\ 0 & \text{otherwise} \end{cases}$

Step 3: Update probability to choose path from i to j.

$$P_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{C_{il} N(s^p)} \tau_{il}^{\alpha}\eta_{il}^{\beta}} & \text{if} C_{ij} \in N(s^p). \\ 0 & \text{otherwise} \end{cases}$$

Here $\eta_{ij} = \frac{1}{d_{ij}}$, where d_{ij} is the distance between nodes i and j. $N(s^p)$ is the set of feasible components; that is, edges (i,l) where l is a node not yet visited by the ant k.

convergence of Ant Colony Optimization Algorithm

Due to pheromone evaporation, the maximum possible pheromone level τ_{max} is bounded asymptotically.

For any τ_{ij}, we have $\lim_{t \to \infty} \tau_{ij}(t) \leqslant \tau_{max} = \frac{1}{p} \cdot g(s^*)$

convergence of Ant Colony Optimization Algorithm

Due to pheromone evaporation, the maximum possible pheromone level τ_{max} is bounded asymptotically.

For any τ_{ij}, we have $\lim_{t \to \infty} \tau_{ij}(t) \leqslant \tau_{max} = \frac{1}{p} \cdot g(s^*)$

Theorem : Let $P^*(t)$ be the probability that the algorithm finds an optimal solution at least once within the first t iterations. Then, for an arbitrary choice of a small $\epsilon > 0$ and for a sufficiently large , it holds that $P^*(t) \geqslant 1 - \epsilon$
and asymptotically $\lim_{t \to \infty} P^*(t) = 1$.

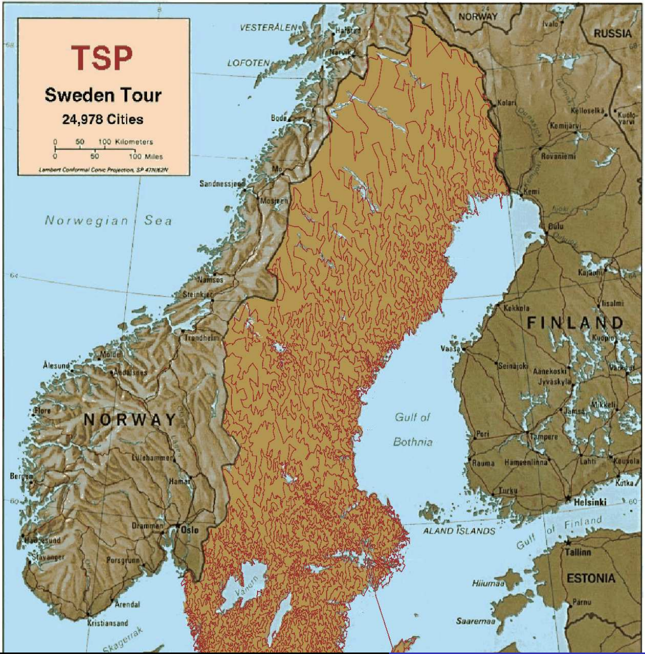
Applet

Link

Thank you!

References:

[Dorigo, 2002] Dorigo M., Stutzle T.,
A short convergence proof for a class of Ant Colony optimization algorithms,


TSP
Sweden Tour
24,978 Cities
0 50 100 Kilometers
0 50 100 Miles
Lambert Conformal Conic Projection, SP 47N/62N
Norwegian Sea
NORWAY
FINLAND
RUSSIA
ESTONIA
Gulf of Bothnia
Gulf of Finland
ALAND ISLANDS
VESTERÅLEN
LOFOTEN
Bodø
Sandnessjøen
Namsos
Steinkjer
Trondheim
Ålesund
Molde
Florø
Bergen
Lillehammer
Hamar
Drammen
Oslo
Haugesund
Stavanger
Porsgrunn
Arendal
Kristiansand
Skagerrak
Kolari
Rovaniemi
Kemi
Oulu
Kajaani
Kokkola
Vaasa
Seinäjoki
Jyväskylä
Pori
Tampere
Rauma
Turku
Helsinki
Lahti
Kotka
Tallinn
Hiiumaa
Saaremaa
Pärnu

back