Ant Colony Optimization
Artificial Ants as a Computational Intelligence Technique

Mahindar A Kota Rohit Kumar Pranay Dhondi

April 19, 2015

Overview

Introduction - Biological Inspiration

- Ant colony optimization (ACO) belongs to the class of **swarm intelligence** that takes inspiration from the social behaviors of insects and of other animals.
- ACO takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving optimization problems.

Introduction - Biological Inspiration

- **Deneubourg and Goss [1992]** thoroughly investigated the pheromone laying and following behavior of ants. In the **double bridge experiment**, the nest of a colony of Argentine ants was connected to a food source by two bridges of different lengths.
- In such a setting, ants start to explore the surroundings of the nest and eventually reach the food source. Along their path between food source and nest, Argentine ants deposit pheromone. Initially, each ant randomly chooses one of the two bridges.

Introduction - Biological Inspiration

- Eventually, the ants choosing by chance the short bridge are the first to reach the nest. The short bridge receives, therefore, pheromone earlier than the long one and this fact increases the probability that further ants select it rather than the long one.
- This brings a further amount of pheromone on that bridge making it more attractive with the result that after some time the whole colony converges toward the use of the same bridge.

The Optimization Technique

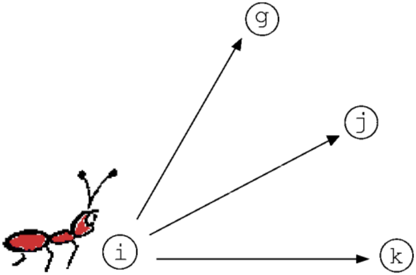

Figure : An ant in city i chooses the next city to visit via a stochastic mechanism: if j has not been previously visited, it can be selected with a probability that is proportional to the pheromone associated with edge (i, j).

The Optimization Technique

In ACO, a number of artificial ants build solutions to an optimization problem and exchange information on their quality via a communication scheme that is inspired of the one adopted by real ants.

Ant Colony Optimization Metaheuristic

- Ant colony optimization (ACO) has been formalized into a metaheuristic for combinatorial optimization
- A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications

Ant Colony Optimization Metaheuristic

- The model of a combinatorial optimization problem is used to define the pheromone model of ACO

A combinatorial optimization problem

A model $P = (\mathbf{S}, \mathbf{\Omega}, f)$ of a combinatorial optimization problem consists of:

- *a search space $\mathbf{S}$ defined over a finite set of discrete decision variables $X_i,\ i = 1, ..., n$;*
- *a set $\mathbf{\Omega}$ of constraints among the variables; and*
- *an objective function $f : \mathbf{S} \to \mathbb{R}_0^+$ to be minimized.*[2]

The generic variable X_i takes values in $\mathbf{D}_i = \{v_i^1, ..., v_i^{|\mathbf{D}_i|}\}$. A feasible solution $s \in \mathbf{S}$ is a complete assignment of values to variables that satisfies all constraints in $\mathbf{\Omega}$. A solution $s^* \in \mathbf{S}$ is called a global optimum if and only if: $f(s^*) \leq f(s)\ \forall s \in \mathbf{S}$.

Ant Colony Optimization Metaheuristic

- **Solution Component** : Every possible assignment of a value to a variable
- A pheromone value is assosciated with each possible solution component
- In ACO, an artificial ant builds a solution by traversing the fully connected construction graph G_C *(V, E)* , where V is a set of vertices and E is a set of edges
- This graph can be obtained from the set of solution components C in two ways: components may be represented either by vertices or by edges

Algorithm 1 The Ant Colony Optimization Metaheuristic

Set parameters, initialize pheromone trails
while termination condition not met **do**
  ConstructAntSolutions
  ApplyLocalSearch (optional)
  UpdatePheromones
end while

Ant Colony Optimization Metaheuristic

Construct Ant Solutions

- A set of **m** artificial ants constructs solutions from elements of a finite set of available solution components $\mathbf{C} = \{c_{ij}\}$, where $i = 1,...n$, $j = 1,...|D_i|$,
- A solution construction starts from an empty partial solution $s^p = \emptyset$
- At each construction step, the partial solution s_p is extended by adding a feasible solution component from the set $N(s_p) \subset C$, which is defined as the set of components that can be added to the current partial solution s_p without violating any of the constraints in Ω
- The choice of a solution component from $N(s_p)$ is guided by a stochastic mechanism, which is biased by the pheromone associated with each of the elements of $N(s_p)$

Apply Local Search

- After every step before updating the pheromone values, we can improve the solutions obtained by the ants through a local search

Ant Colony Optimization Metaheuristic

Apply Local Search

- After every step before updating the pheromone values, we can improve the solutions obtained by the ants through a local search

Update Pheromones

- In this step we increase the pheromone values associated with promising solutions and decrease those that are associated with bad ones
- Generally pheromone values assosciated are decreased through *pheromone evaporation*

Main ACO Algorithms

Ant System

- It is the first ACO algorithm proposed in the literature
- **Main Characteristic:** At each iteration, the pheromone values are updated by all the m ants that have built a solution in the iteration

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \ ,$$

- ρ is the evaporation rate, m is the number of ants and $\Delta\tau_{ij}^{k}$ is the pheromone laid on edge (i,j) by ant k

Main ACO Algorithms

Ant System

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

- Q is a constant and L_k is the length of the tour constructed by k

Main ACO Algorithms

Constructing Ant Solutions in Ant System

- In the construction of a solution, ants select the next city to be visited through a stochastic mechanism.
- When ant k is in city i and has so far constructed the partial solution s^p,the probability of going to city j is given by

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{c_{il} \in \mathbf{N}(s^p)} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} & \text{if } c_{ij} \in \mathbf{N}(s^p), \\ 0 & \text{otherwise,} \end{cases}$$

- $\mathbf{N}(s^p)$ is the set of feasible components satisfying the constraints. The parameters α and β control the relative importance of the pheromone versus the heuristic information η_{ij} , which is given by

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

- In the above equation d_{ij} is the distance between cities i and j

Main ACO Algorithms

Ant Colony System

- This is an variant of Ant System algorithm
- Main contribution of ACS is the introduction of local pheromone update in addition to the pheromone update performed at the end of the construction process
- The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed:

$$\tau_{ij} = (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \ ,$$

- where $\phi \in [0, 1]$ is the pheromone decay coefficient and τ_0 is the initial value of pheromone

Main ACO Algorithms

Ant Colony System

- Main goal of the local update is to diversify the search performed by subsequent ants during an iteration, by decreasing the pheromone concentration on the traversed edges, ants encourage subsequent ants to choose other edges and, hence, to produce different solutions
- Offline update performed in ACS is

$$\tau_{ij} \leftarrow \begin{cases} (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij} & \text{if } (i,j) \text{ belongs to best tour,} \\ \tau_{ij} & \text{otherwise.} \end{cases}$$

Travelling Salesman Problem (TSP)

Introduction

- Travelling salesman problem (TSP) consists of finding the shortest route in complete weighted graph G with n nodes and $n(n-1)$ edges, so that the start node and the end node are identical and all other nodes in this tour are visited exactly once
- Some applications of TSP are
  - Shortest of costumer servicing route
  - Planning bus lines
  - and many more ......

Problem Formulation

- Construct C, the matrix of shortest distances between nodes in G
- The TSP can be formulated in the category programming binary , where variables are equal to 0 or 1 i.e $x_{ij} = 1$ if the path from node i to j is realised in TSP
- The TSP problem thus can be represented as

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Travelling Salesman Problem (TSP)

Problem Formulation (Constraints)

- The above formulation is subject to below constraints

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,2,\ldots n \;\; i \neq j$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,2,\ldots n \;\; i \neq j$$

$$u_i - u_j + n x_{ij} \leq n - 1 \quad i,j = 2,3,\ldots n \quad i \neq j$$

$$x_{ij} \in \{0,1\} \qquad i,j = 1,2,\ldots n \quad i \neq j$$

- u_i indicates the order in which the nodes are to be travelled

Travelling Salesman Problem (TSP)

- Moving of virtual ant depends on the amount of pheromone on the graph edges. The probability p_{ik} of transition of a virtual ant from the node i to the node k is given by formula

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{c_{il} \in \mathbf{N}(s^p)} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} & \text{if } c_{ij} \in \mathbf{N}(s^p), \\ 0 & \text{otherwise,} \end{cases}$$

Travelling Salesman Problem (TSP)

- Every ant has its own memory, which is used for saving information about travelled path (for example about travelled nodes). This memory can also serve to ensure constraints or to evaluate the solution.
- Artificial ants use the same reverse path as the path to the food resource based on his internal memory. Ant puts the pheromone on the edges of reverse path depending upon the quality of solution found so far.
- The evaporation helps to find the shortest path and provide that no other path will be assessed as the shortest.

Simulation : Traveling Salesman Problem

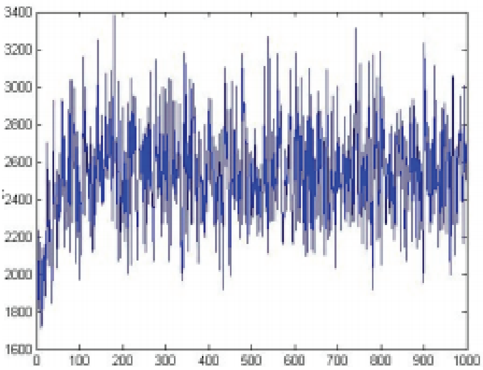

Figure 1 Running of simulation for *m*=100

Figure : Number of ants equal to 100, Best tour length obtained : 1713, Deviation: 17%, Optimum being 1453

Simulation : Traveling Salesman Problem

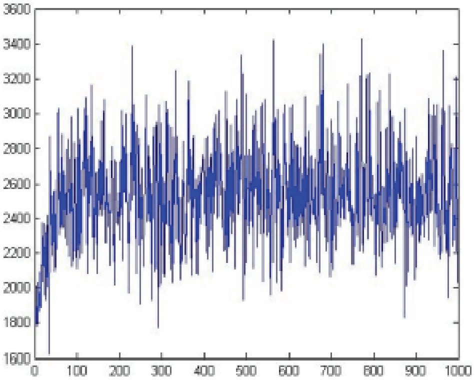

Figure 2 Running of simulation for *m*=1000

Figure : Number of ants equal to 1000, Best tour length obtained : 1621, Deviation: 11.5%, Optimum being 1453

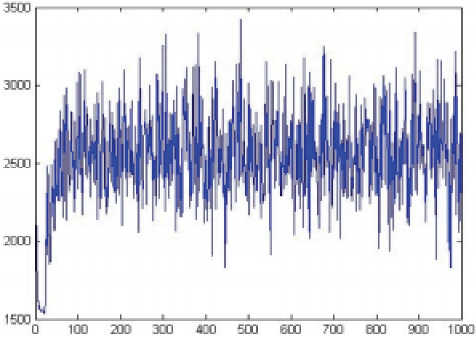

Figure 3 Running of simulation for *m*=5000

Figure : Number of ants equal to 5000, Best tour length obtained : 1532, Deviation: 5.4%, Optimum being 1453

Simulation : Traveling Salesman Problem

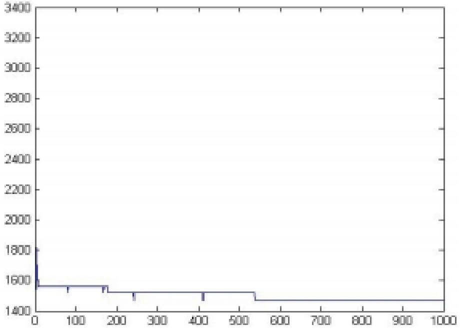

Figure 4 Running of simulation for *m*=10000

Figure : Number of ants equal to 10000, Best tour length obtained : 1465, Deviation: 0.8%, Optimum being 1453

Simulation : Traveling Salesman Problem

Simulation with different no. of ants, optimal solution being 1453

Iterations	ants	tour length	best iteration	deviation(%)
1000	100	1713	10	17
1000	1000	1612	34	11.5
1000	5000	1532	21	5.4
1000	10000	1465	242	0.8

Conclusions

- Behavior of ants can be used to solve many optimization problems
- We formalized the behavior of ants into a pheromone model, a metaheuristic. Natural phenomena like evaporation and deposition of pheromones can be observed in the model
- We have seen an application of this model to solve the *Travelling Salesman Problem* and its experimental results
- The great advantage over the use of exact methods is that ACO algorithm provides relatively good results by a comparatively low number of iterations, and is therefore able to find an acceptable solution (about 0.8% deviation) in a comparatively short time, so it is useable for solving problems occurring in practical applications.

References

Ivan Brezina Jr. and Zuzana Ciskova (2011)
Solving the Travelling Salesman Problem Using the Ant Colony Optimization
Management Information Systems, Vol. 6 (2011), No. 4, pp. 010-014

Marco Dorigo, Mauro Birattari, and Thomas Stutzle (2006)
Artificial Ants as a Computational Intelligence Technique
ISSN 1781-3794

Marco Dorigo, Luca Maria Gambardella (1997)
Ant colonies for the traveling salesman problem
Accepted for publication in BioSystems, 1997. In press

http://www.aco-metaheuristic.org/

http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms

Thank you for listening

Any Questions?