Ant Colony Optimization

Luca Maria Gambardella, IDSIA, 2005

Ants do not directly communicate. The basic principle is stigmergy, a particular kind of indirect communication based on environmental modification.

- Stimulation of workers by the performance they have achieved (Grassé P. P., 1959)
- Foraging behavior: searching for food by parallel exploration of the environment

Insects, Social Insects, and Ants

- 10^18 living insects (rough estimate)
- ~2% of all insects are social
- Social insects are:
  - All ants
  - All termites
  - Some bees
  - Some wasps
- 50% of all social insects are ants
- Avg weight of one ant between 1 and 5 mg
- Tot weight ants ~ Tot weight humans

How Do Ants Coordinate their Activities?

- Foraging ant colonies can synergistically find shortest paths in distributed/dynamic environments:
  - While moving back and forth between nest and food ants mark their path by pheromone laying
  - Step-by-step routing decisions are biased by the local intensity of pheromone field (stigmergy)
  - Pheromone is the colony’s collective and distributed memory: it encodes the collectively learned quality of local routing choices toward destination target


How Ants Find Food

Social insects, following simple, individual rules, accomplish complex colony activities through flexibility, robustness and self-organization.

Ants Foraging Behavior

Shortest paths: an emerging behavior from stigmergy

Pheromone Trail Following

Ants and termites follow pheromone trails

Simple Bridge Experiment

Goss et al., 1989, Deneubourg et al., 1990

% ants in upper and lower branches

Asymmetric Bridge Experiment

Goss et al., 1989, Deneubourg et al., 1990

Reverse-engineering of ant colony mechanisms: Ant Colony Optimization (ACO) metaheuristic:

- Combinatorial optimization
- Adaptive routing

Multiple autonomous/concurrent agents (ants): solution construction as sequential decision process:

- Model: a network of decision points where the quality of the choices is expressed by pheromone variables
- Building Solutions: constructing a path in the network according to a stochastic decision policy
- Use of solution outcomes to iteratively update pheromone (generalized policy iteration based on Monte Carlo sampling)
- No explicit solutions representation. The collectively learned knowledge is distributed in the pheromone


ACO

- ACO algorithms are multi-agent systems that exploit artificial stigmergy for the solution of combinatorial optimization problems.
- Artificial ants live in a discrete world. They construct solutions making stochastic transition from state to state.
- They deposit artificial pheromone to modify some aspects of their environment (search space). Pheromone is used to dynamically store past history of the colony.
- Artificial Ants are sometime “augmented” with extra capabilities like local optimization or backtracking

Similarities with Real Ants

- Colony of simple cooperative individuals.
  - an artificial pheromone trail is used for local stigmergetic communication
  - a sequence of local moves to find shortest path
  - a stochastic construction policy (exploration and exploitation) based on local information
**Differences with real ants**

- Artificial ants use a discrete world
- Artificial ants have internal state and memory
- The deposited pheromone is proportional to the quality of the solution (some real ants have a similar behavior)
- Extra capabilities (lookahead, local optimization, backtracking)

**Travelling Salesman Problem (TSP)**

Problem: given N cities, and a distance function $d$ between cities, find a tour that:

1. goes through every city once and only once
2. minimizes the total distance

- Problem is NP-complete
- Classical combinatorial optimization problem to test algorithms

*First ACO application, Ant System, Dorigo et al. 1992*

**Search Space**

Discrete Graph

To each edge is associated a static value returned by an heuristic function $\eta(r,s)$ based on the edge-cost

Each edge of the graph is augmented with a pheromone trail $\tau(r,s)$ deposited by ants. Pheromone is dynamic and it is learned at run-time

**ACS: Ant Colony System for TSP**

**ACS State Transition rule**

Next city is chosen between the not visited cities according to a probabilistic rule

**Exploitation**: the best edge is chosen

**Exploration**: one of the edge in proportion to its value

**ACS state transition rule: formulae**

$$p(k)(r,s) = \begin{cases} \frac{\tau(r,s) \cdot \eta(r,s)}{\sum_{u \in J^k(r)} \tau(r,u) \cdot \eta(r,u)} & \text{if } \not \in J^k(r) \\ \text{otherwise} & \text{(Exploration)} \end{cases}$$

where

- $\tau$ is a stochastic variable distributed as follows:
  $$\tau(r,s) = \frac{[\tau(r,s)]^\varphi \cdot [\eta(r,s)]^{1-\varphi}}{\sum_{r \in R} [\tau(r,s)]^\varphi \cdot [\eta(r,s)]^{1-\varphi}}$$
- $\varphi$ is the inverse of the distance
- $J^k(r)$ is the set of cities still to be visited by ant k positioned on city r
- $\tau$ and $\eta$ are parameters
ACS state transition rule: example

next state:
with probability $q_0$ exploitation
with probability $(1-q_0)$ biased exploration.

ACS local trail updating

If an edge $(r,s)$ is visited by an ant

$$\tau(r,s) = (1-p) \tau(r,s) + p \Delta \tau(r,s)$$

with $\Delta \tau(r,s) = \tau_0$.

ACS global trail updating

At the end of each iteration, the best ant so far, is allowed to reinforce its tour by depositing additional pheromone proportional to the length of the tour

$$\tau(r,s) = (1-\alpha) \tau(r,s) + \alpha \cdot \Delta \tau(r,s)_{\text{Global}}$$

where

$$\Delta \tau(r,s)_{\text{Global}} = \frac{1}{L_{\text{best}}}$$

Best solutions structures emerge step by step from the computation

Among the state of the art algorithms for TSP and ATSP problems

Pheromone is useful?

Effectiveness of distributed pheromone learning

Best tour length as a function of elapsed CPU time (avg on 100 runs)
Comparison of ACS with other heuristics on random TSPs

<table>
<thead>
<tr>
<th>Problem name</th>
<th>ACS (average)</th>
<th>SA (average)</th>
<th>E (average)</th>
<th>SOM (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>City set 1</td>
<td>5.88</td>
<td>5.88</td>
<td>5.98</td>
<td>6.06</td>
</tr>
<tr>
<td>City set 2</td>
<td>5.98</td>
<td>6.01</td>
<td>6.03</td>
<td>6.25</td>
</tr>
<tr>
<td>City set 3</td>
<td>5.58</td>
<td></td>
<td>5.70</td>
<td>5.83</td>
</tr>
<tr>
<td>City set 4</td>
<td>5.74</td>
<td>5.61</td>
<td>5.68</td>
<td>5.87</td>
</tr>
<tr>
<td>City set 5</td>
<td>5.18</td>
<td>6.33</td>
<td>6.49</td>
<td>6.70</td>
</tr>
</tbody>
</table>

Comparison of ACS with other natural algorithms on geometric TSPs

<table>
<thead>
<tr>
<th>Problem name</th>
<th>ACS</th>
<th>GA</th>
<th>E</th>
<th>SOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>City set 1</td>
<td>5.88</td>
<td>5.88</td>
<td>6.06</td>
<td>6.06</td>
</tr>
<tr>
<td>City set 2</td>
<td>5.98</td>
<td>6.01</td>
<td>6.03</td>
<td>6.25</td>
</tr>
<tr>
<td>City set 3</td>
<td>5.58</td>
<td></td>
<td>5.70</td>
<td>5.83</td>
</tr>
<tr>
<td>City set 4</td>
<td>5.74</td>
<td>5.61</td>
<td>5.68</td>
<td>5.87</td>
</tr>
<tr>
<td>City set 5</td>
<td>5.18</td>
<td>6.33</td>
<td>6.49</td>
<td>6.70</td>
</tr>
</tbody>
</table>

ACS Extension

Current wisdom says that a very good strategy for the approximate solution of combinatorial optimization problems is the coupling of:

- a constructive heuristic,
- a local search.

The problem is to find good couplings:

ACO (and other derived algorithms) seems (as shown by experimental evidence) to provide such a good coupling.

ACS plus local search

Loop

- Randomly position m agents on n cities
- For step = 1 to n
  - For ant = 1 to m
  - Apply the state transition rule
  - Apply local search
  - Each solution is optimized by a problem specific heuristic
- Apply global trail updating rule using the best optimized solution
- Until End_condition

Local Search

A 2-exchange always inverts a path.
Results obtained by ACS-3-opt on TSP problems taken from the First International Contest on Evolutionary Optimization, IEEE-EC 96, May 20-22, 1996, Nagoya, Japan

ACS-3-opt applied to TSP

<table>
<thead>
<tr>
<th>Problem name</th>
<th>ACS-3-opt</th>
<th>GA+local</th>
<th>Average</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(170-city problem)</td>
<td>20.81%</td>
<td>29.10%</td>
<td>27.718</td>
<td>0.11%</td>
</tr>
<tr>
<td>(100-city problem)</td>
<td>19.12%</td>
<td>23.43%</td>
<td>27.693</td>
<td>0.03%</td>
</tr>
<tr>
<td>(70-city problem)</td>
<td>14.42%</td>
<td>21.43%</td>
<td>27.804</td>
<td>0.18%</td>
</tr>
<tr>
<td>(48-city problem)</td>
<td>9.51%</td>
<td>11.36%</td>
<td>27.908</td>
<td>0.33%</td>
</tr>
<tr>
<td>(318-city problem)</td>
<td>8.51%</td>
<td>10.25%</td>
<td>28.312</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

ACS-3-opt applied to ATSP

<table>
<thead>
<tr>
<th>Problem name</th>
<th>ACS-3-opt</th>
<th>GA+local</th>
<th>Average</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(170-city problem)</td>
<td>20.81%</td>
<td>29.10%</td>
<td>27.718</td>
<td>0.11%</td>
</tr>
<tr>
<td>(100-city problem)</td>
<td>19.12%</td>
<td>23.43%</td>
<td>27.693</td>
<td>0.03%</td>
</tr>
<tr>
<td>(70-city problem)</td>
<td>14.42%</td>
<td>21.43%</td>
<td>27.804</td>
<td>0.18%</td>
</tr>
<tr>
<td>(48-city problem)</td>
<td>9.51%</td>
<td>11.36%</td>
<td>27.908</td>
<td>0.33%</td>
</tr>
<tr>
<td>(318-city problem)</td>
<td>8.51%</td>
<td>10.25%</td>
<td>28.312</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Results obtained by ACS-3-opt on ATSP problems taken from the First International Contest on Evolutionary Optimization, IEEE-EC 96, May 20-22, 1996, Nagoya, Japan

Comparison of ACS-3-opt and GA+local search on TSPs

<table>
<thead>
<tr>
<th>Problem name</th>
<th>ACS-3-opt</th>
<th>GA+local</th>
<th>Average</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(170-city problem)</td>
<td>20.81%</td>
<td>29.10%</td>
<td>27.718</td>
<td>0.11%</td>
</tr>
<tr>
<td>(100-city problem)</td>
<td>19.12%</td>
<td>23.43%</td>
<td>27.693</td>
<td>0.03%</td>
</tr>
<tr>
<td>(70-city problem)</td>
<td>14.42%</td>
<td>21.43%</td>
<td>27.804</td>
<td>0.18%</td>
</tr>
<tr>
<td>(48-city problem)</td>
<td>9.51%</td>
<td>11.36%</td>
<td>27.908</td>
<td>0.33%</td>
</tr>
<tr>
<td>(318-city problem)</td>
<td>8.51%</td>
<td>10.25%</td>
<td>28.312</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Comparison of ACS-3-opt and GA+local search on ATSPs

<table>
<thead>
<tr>
<th>Problem name</th>
<th>ACS-3-opt</th>
<th>GA+local</th>
<th>Average</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(170-city problem)</td>
<td>20.81%</td>
<td>29.10%</td>
<td>27.718</td>
<td>0.11%</td>
</tr>
<tr>
<td>(100-city problem)</td>
<td>19.12%</td>
<td>23.43%</td>
<td>27.693</td>
<td>0.03%</td>
</tr>
<tr>
<td>(70-city problem)</td>
<td>14.42%</td>
<td>21.43%</td>
<td>27.804</td>
<td>0.18%</td>
</tr>
<tr>
<td>(48-city problem)</td>
<td>9.51%</td>
<td>11.36%</td>
<td>27.908</td>
<td>0.33%</td>
</tr>
<tr>
<td>(318-city problem)</td>
<td>8.51%</td>
<td>10.25%</td>
<td>28.312</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

Sequential Ordering Problem

It consists of finding a minimum weight Hamiltonian path on a directed graph subject to multiple precedence constraints among nodes.

SOP models real-world problems like production planning, single-vehicle pickup and delivery, and transportation problems.
Sequential Ordering Problem

- Escudero (1988)
- General ATSP Problem
  - Precedence Constrained ATSP Polytope (Balas, Fischetti, Pulleyblank, 1995).
  - Branch and Cut (Ascheuer, 1996)
- Maximum Partial Order/Arbitrary Insertion GA (Chen and Smith, 1996)
- Pick-Up and Delivery
  - Lexicographic search with labeling Procedure (Savelsbergh, 1990).

HAS-SOP: Hybrid Ant System for SOP

- Costructive phase based on ACS
- Trail updating as ACS
- New local search strategy based on a combination between lexicographic search and a new labeling procedure.
- New data structure to drive the search
- First in literature that uses a local search edge-exchange strategy to directly handle multiple constraints without any increase in computational time.

Ants for SOP

- Each ant iteratively starts from node 0 and adds new nodes until all nodes have been visited and node n is reached.
- When in node i, an ant chooses probabilistically the next node j from the set $F_i$ of feasible nodes.
- $F_i$ contains all the nodes j still to be visited and such that all nodes that have to precede j, according to precedence constraints, have already been inserted in the sequence.

Local Search

- Local search strategy to directly handle multiple constraints without any increase in computational time.

HAS-SOP

We tested and compare our algorithms on a set of problems in TSPLIB using a SUN Ultra SPARC 1 (167MHz).
The best-known results for many test problems from TSPLIB has been improved by using HAS-SOP.
MACS-VRPTW: some details

Unfeasible solutions are repaired by insertion procedures

Feasible solutions are improved with local search procedures

Benchmark problems

With Time Windows (TSPLIB)
56 problems (Solomon, 1987) of six different types
(C1,C2,R1,R2,RC1,RC2).
Each data set contains between eight to twelve 100-node problems.
- C = clustered customers with easy TW.
- R = customers location generated uniformly randomly over a square.
- RC = a combination of randomly placed and clustered customers.
- Sets of type 1 have narrow time windows and small vehicle capacity.
- Sets of type 2 have large time windows and large vehicle capacity.

### The Routing Problem

- The practical goal of routing algorithms is to build routing tables

- Routing is difficult because costs are dynamic
- Adaptive routing is difficult because changes in the control policy determine changes in the costs and vice versa

### AntNet: The Algorithm

- Ants are launched at regular instants from each node to randomly chosen destinations
- Ants are routed probabilistically with a probability function of:
  1. some artificial pheromone values, and
  2. some heuristic values, maintained on the nodes
- Ants memorize visited nodes and elapsed times
- Once reached their destination nodes, ants retrace their paths backwards, and update the routing tables

AntNet is distributed and not synchronized.

### Ants' Pheromone Trail Depositing

\[
\tau_{ij}^k(t+1) \leftarrow (1 - \rho) \tau_{ij}^k(t) + \Delta \tau_{ij}^k(t)
\]

where the (i,j)'s are the links visited by ant \( k \), and

\[
\Delta \tau_{ij}^k(t) = \text{quality}^k
\]

where \( \text{quality}^k \) is set proportional to the inverse of the time it took ant \( k \) to build the path from \( i \) to \( j \) via \( j \).

### AntNet: Experimental setup

- Realistic simulator (though not industrial)
- Many topologies
- Many traffic patterns
- Comparison with many state-of-the-art algorithms (Open Shortest Path First, SPF, Adaptive Bellman-Ford, Grouting, Predictive Grouting)

- Performance measures:
  - Throughput (bit/sec) measures the quantity of service, and average packet delay (sec) measures the quality of service

Japanese NTT net, American NSF net
AntNet: Some Results (1)

From Di Caro and Dorigo, 1999, Journal of Artificial Intelligence Research

Increasing UP traffic
UP traffic increased by reducing the mean session inter arrival time

Luca Maria Gambardella, IDSIA, 2005

AntNet: Adaptiveness

From Di Caro and Dorigo, 1999, Journal of Artificial Intelligence Research

NSF net NTT net
Data averaged over a 5 seconds sliding window

Luca Maria Gambardella, IDSIA, 2005

The ACO Metaheuristic

Dorigo, Di Caro & Gambardella, Artificial Life 1999

• Ant Colony System and AntNet have been extended so that they can be applied to any shortest path problem on graphs
• The resulting extension is called Ant Colony Optimization metaheuristic
• Currently two major application classes:
  – Routing in telecommunications networks
  – NP-hard combinatorial optimization problems

Luca Maria Gambardella, IDSIA, 2005

The ACO-metaheuristic

procedure ACO-metaheuristic()
  while (not-termination-criterion)
    schedule subprocedures
    generate-ants()
    evaporate-pheromone() [Optional]
    execute-daemon-actions() [Optional]
    end schedule subprocedures
  end while
end procedure

These are problem specific actions, like local search


Luca Maria Gambardella, IDSIA, 2005

From research to Applications

Dyvoil: Dynamic fleet optimization for fuel distribution, Pina Petroli SA, Grancia, CH

Customers
Ask for fuel delivery at home (house heating)
Multiple time windows
Combined delivery (e.g. 2 families)
Stochastic quantity
Accessibility restrictions

Luca Maria Gambardella, IDSIA, 2005
Every point of the network has a constant service time

Objective:
Maximization of the average tour efficiency.
This should implicitly have as a side effect the minimization of the number of tours and of the total km.
Ant Colony Optimization Major Publications

- Nature
- Scientific American
- The New York Times
- Harvard Business Review
- Scheduling

Luca Maria Gambardella, IDSIA, 2005