DM812
METAHEURISTICS

Lecture 8

Ant Colony Optimization

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

Marco Chiarandini

Department of Mathematics and Computer Science
University of Southern Denmark, Odense, Denmark
<marco@imada.sdu.dk>

Outline

Outline

Swarm Intelligence

Definition: Swarm Intelligence

Swarm intelligence deals with systems composed of many individuals that coordinate using decentralized control and self-organization.

In particular, it focuses on the collective behaviors that emerges from the local interactions of the individuals with each other and with their environment and without the presence of a coordinator

Examples:

Natural swarm intelligence

- colonies of ants and termites
- schools of fish
- flocks of birds
- herds of land animals

Artificial swarm intelligence

- artificial life (boids)
- robotic systems
- computer programs for tackling optimization and data analysis problems.

Swarm Intelligence

Research goals in Swarm Intelligence:

- scientific
  modelling swarm intelligence systems to understand the mechanisms that allow coordination to arise from local individual-individual and individual-environment interactions
- engineering
  exploiting the understanding developed by the scientific stream in order to design systems that are able to solve problems of practical relevance

The Biological Inspiration

Double-bridge experiment [Goss, Aron, Deneubourg, Pasteels, 1989]

- If the experiment is repeated a number of times, it is observed that each of the two bridges is used in about 50% of the cases.
- About 100% the ants select the shorter bridge

Self-organization

Four basic ingredients:

1. Multiple interactions
2. Randomness
3. Positive feedback (reinforcement)
4. Negative feedback (evaporating, forgetting)

Communication is necessary

- Two types of communication:
  - **Direct**: antennation, trophallaxis (food or liquid exchange), mandibular contact, visual contact, chemical contact, etc.
  - **Indirect**: two individuals interact indirectly when one of them modifies the environment and the other responds to the new environment at a later time.
    This is called stigmergy and it happens through pheromone.

Stigmergy

- "The coordination of tasks and the regulation of constructions does not depend directly on the workers, but on the constructions themselves. The worker does not direct his work, but is guided by it. It is to this special form of stimulation that we give the name STIGMERGY (stigma, sting; ergon, work, product of labour = stimulating product of labour)." Grassé P. P., 1959

Stigmergy
Stimulation of workers
by the performance
they have achieved
Grassé P. P., 1959

Mathematical Model

[Goss et al. (1989)] developed a model of the observed behavior:

Assuming that at a given moment in time,

- m_1 ants have used the first bridge
- m_2 ants have used the second bridge,

The probability $\Pr[X=1]$ for an ant to choose the first bridge is:

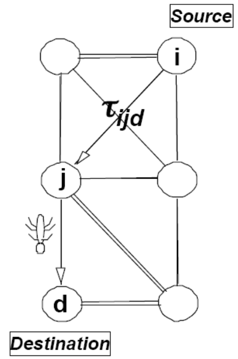
$$\Pr[X=1] = \frac{(m_1+k)^h}{(m_1+k)^h + (m_2+k)^h}$$

(parameters k and h are to be fitted to the experimental data)

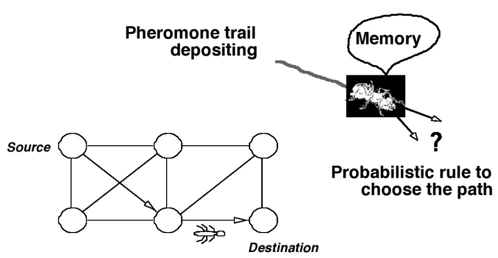
Why Does it Work?

Three important components:

- TIME: a shorter path receives pheromone quicker (this is often called: "differential length effect")
- QUALITY: a shorter path receives more pheromone
- COMBINATORICS: a shorter path receives pheromone more frequently because it is likely to have a lower number of decision points



From Real to Artificial Ants



Our Basic Design Choices

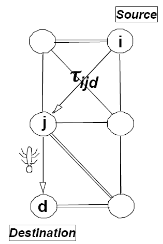
- Ants are given a memory of visited nodes
- Ants build solutions probabilistically (without updating pheromone trails)
- Ants deterministically retrace backward the forward path to update pheromone
- Ants deposit a quantity of pheromone function of the quality of the solution they generated

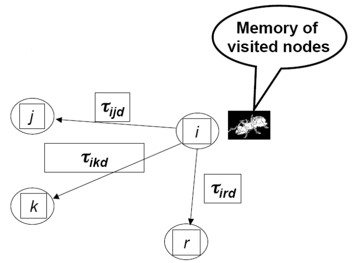
From Real to Artificial Ants

Using Pheromone and Memory to Choose the Next Node

For ant k:

$$p^k_{ijd}(t) = f\big(\tau_{ijd}(t)\big)$$



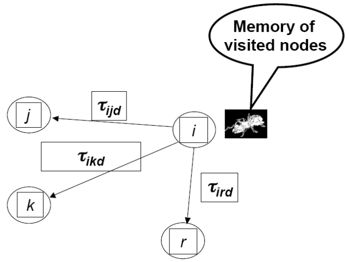


From Real to Artificial Ants

Ants' Probabilistic Transition Rule

For ant k:

$$p^k_{ijd}(t) = \frac{[\tau_{ijd}(t)]^\alpha}{\sum_{h \in J^k_i} [\tau_{ihd}(t)]^\alpha}$$



- τ_{ijd} is the amount of pheromone trail on edge (i, j, d)
- J^k_i is the set of feasible nodes ant k positioned on node i can move to

From Real to Artificial Ants

Ants' Pheromone Trail: Deposition and Evaporation

Evaporation:

$$\tau_{ijd}(t+1) \leftarrow (1-\rho) \cdot \tau_{ijd}(t)$$

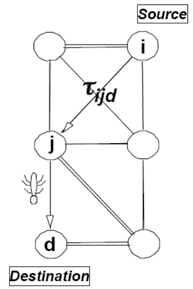
Deposition

$$\tau_{ijd}(t+1) \leftarrow \tau_{ijd}(t+1) + \Delta^k_{ijd}(t)$$

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

$$\Delta^k_{ijd}(t) \sim \text{quality}^k$$

eg: quality^k proportional to the inverse of the time it took ant k to build the path from i to d via j.

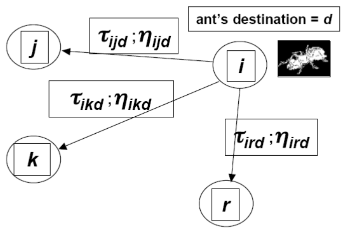


From Real to Artificial Ants

Using Pheromones and Heuristic to Choose the Next Node

For ant k

$$p^k_{ijd}(t) = f(\tau_{ijd}(t), \eta_{ijd}(t))$$

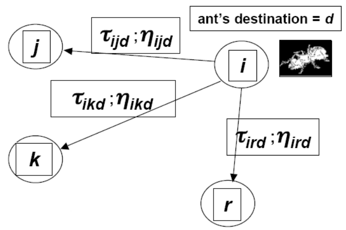


- τ_{ijd} is a value stored in a pheromone table
- η_{ijd} is a heuristic evaluation of link (i, j, d) which introduces problem specific information

From Real to Artificial Ants

Ants' Probabilistic Transition Rule (Revised)

$$p^k_{ijd}(t) = \frac{[\tau_{ijd}(t)]^\alpha \cdot [\eta_{ijd}(t)]^\beta}{\sum_{h \in J^k_i} [\tau_{ihd}(t)]^\alpha \cdot [\eta_{ijd}(t)]^\beta}$$



- τ_{ijd} is the amount of pheromone trail on edge (i, j, d)
- η_{ijd} is the heuristic evaluation of link (i, j, d)
- J^k_i is the set of feasible nodes ant k positioned on node i can move to

From Real to Artificial Ants

Simple Ant Colony Optimization Algorithm

1. Ants are launched at regular instants from each node to randomly chosen destinations
2. Ants build their paths probabilistically with a probability function of:
   - artificial pheromone values
   - heuristic values
3. Ants memorize visited nodes and costs incurred
4. Once reached their destination nodes, ants retrace their paths backwards, and update the pheromone trails
5. Repeat from 1.

The pheromone trail is the stigmergic variable

Artificial versus Real Ants:

Main Differences

Artificial ants:

- Live in a discrete world
- Deposit pheromone in a problem dependent way
- Can have extra capabilities:
  local search, lookahead, backtracking
- Exploit an internal state (memory)
- Deposit an amount of pheromone function of the solution quality
- Can use heuristics

Outline

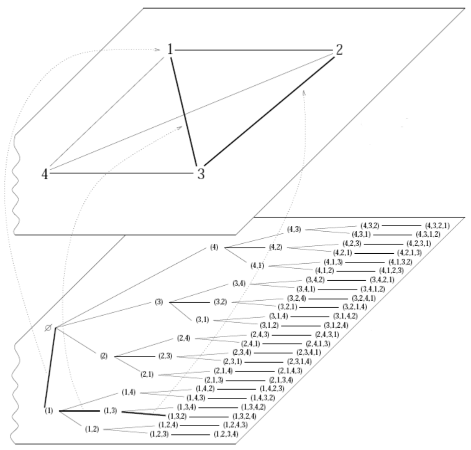
Ant Colony Optimization

The Metaheuristic

- The optimization problem is transformed into the problem of finding the best path on a weighted graph $G(V, E)$ called construction graph
- The artificial ants incrementally build solutions by moving on the construction graph.
- The solution construction process is
  - stochastic
  - biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants.
- All *pheromone trails* are initialized to the same value, τ_0.
- At each iteration, *pheromone trails* are updated by decreasing (*evaporation*) or increasing (*reinforcement*) some trail levels on the basis of the solutions produced by the ants

Ant Colony Optimization

Example: A simple ACO method for the TSP



- *Construction graph*
- To each edge ij in G associate
  - pheromone trails τ_{ij}
  - heuristic values $\eta_{ij} := \frac{1}{c_{ij}}$
- Initialize pheromones
- *Probabilistic construction:*

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}},$$

- *Update pheromone trail levels*

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \rho \cdot \text{Reward}$$

ACO Metaheuristic

- Population-based method in which artificial ants iteratively construct candidate solutions.
- Solution construction is probabilistically biased by pheromone trail information, heuristic information and partial candidate solution of each ant (memory).
- Pheromone trails are modified during the search process to reflect collective experience.

Ant Colony Optimization (ACO):
initialize pheromone trails
while termination criterion is not satisfied **do**
  generate population P of candidate solutions
   using subsidiary randomized constructive search
  apply subsidiary local search on P
  update pheromone trails based on P

Note

- In each cycle, each ant creates one candidate solution using a constructive search procedure.
- Ants build solutions by performing randomized walks on a construction graph $G = (V, E)$ where V are solution components and G is fully connected.
- All pheromone trails are initialized to the same value, τ_0.
- Pheromone update typically comprises uniform decrease of all trail levels (evaporation) and increase of some trail levels based on candidate solutions obtained from construction + local search.
- Subsidiary local search is (often) applied to individual candidate solutions.
- Termination criterion can include conditions on make-up of current population, *e.g.*, variation in solution quality or distance between individual candidate solutions.

Example: A simple ACO algorithm for the TSP (Revised)

- Search space and solution set: all Hamiltonian cycles in given graph G.
- Associate pheromone trails τ_{ij} with each edge (i, j) in G.
- Use heuristic values $\eta_{ij} := \frac{1}{c_{ij}}$
- Initialize all weights to a small value τ_0 $(\tau_0 = 1)$.
- *Constructive search:* Each ant starts with randomly chosen vertex and iteratively extends partial round trip π^k by selecting vertex not contained in π^k with probability

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}]^{\alpha} \cdot [\eta_{il}]^{\beta}}$$

α and β are parameters.

Example: A simple ACO algorithm for the TSP (2)

- *Subsidiary local search:* Perform iterative improvement based on standard 2-exchange neighborhood on each candidate solution in population (until local minimum is reached).
- *Update pheromone trail levels* according to

$$\tau_{ij} := (1-\rho)\cdot\tau_{ij} + \sum_{s\in sp'} \Delta_{ij}(s)$$

where $\Delta_{ij}(s) := 1/C^s$ if edge (i,j) is contained in the cycle represented by s', and 0 otherwise.

Motivation: Edges belonging to highest-quality candidate solutions and/or that have been used by many ants should be preferably used in subsequent constructions.

- *Termination:* After fixed number of cycles (= construction + local search phases).

Outline

ACO Variants

Variants of ACO ttested on the TSP

- Ant System AS (Dorigo et al., 1991)
- Elitist AS (EAS)(Dorigo et al., 1991; 1996)
  - The iteration best solution adds more pheromone
- Rank-Based AS (ASrank)(Bullnheimer et al., 1997; 1999)
  - Only best ranked ants can add pheromone
  - Pheromone added is proportional to rank
- Max-Min AS (MMAS)(Stützle & Hoos, 1997)
- Ant Colony System (ACS) (Gambardella & Dorigo, 1996; Dorigo & Gambardella, 1997)
- Approximate Nondeterministic Tree Search ANTS (Maniezzo 1999)
- Hypercube AS (Blum, Roli and Dorigo, 2001)

Ant System

- Initialization:

$$\tau_{ij} = \tau_o = \frac{m}{C^{NN}}$$

Motivation: sligthly more than what evaporates

- Construction: m ants in m randomly chosen cities

$$p_{ij} = \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l\in\mathcal{N}_i^k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta}, \qquad \alpha \text{ and } \beta \text{ parameters}$$

- Update

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} \qquad \text{to all the edges}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta_{ij}^k \qquad \text{to the edges visited by the ants, } \Delta_{ij}^k = \frac{1}{C^k}$$

Elitist Ant System

- Update

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} \qquad \text{to all the edges}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta_{ij}^{k} + e\cdot\Delta_{ij}^{bs} \qquad \text{to the edges visited by the ants}$$

$$\Delta_{ij}^{bs} = \begin{cases} \frac{1}{C^{bs}} & (ij) \text{ in tour } k,\ bs \text{ best-so-far} \\ 0 & \text{otherwise} \end{cases}$$

Rank-based Ant System

- Update: only $w-1$ best ranked ants + the best-so-far solution deposit pheromone:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} \qquad \text{to all the edges}$$

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{w-1} \Delta_{ij}^{k} + w\cdot\Delta_{ij}^{bs} \qquad \text{to the edges visited by the ants}$$

$$\Delta_{ij}^{k} = \frac{1}{C^{k}}$$

$$\Delta_{ij}^{bs} = \frac{1}{C^{bs}}$$

$\mathcal{MAX}$-$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS)

Peculiarities in pheromone management:

- Update

  $$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} \qquad \text{to all the edges}$$

  $$\tau_{ij} \leftarrow \tau_{ij} + \Delta_{ij}^{bs} \qquad \text{only to the edges visited by the best ant}$$

  Meaning of best alternates during the search between:
  - best-so-far
  - iteration best
- bounded values τ_{min} and τ_{max}
- $\tau_{max} = \frac{1}{\rho C^*}$ and $\tau_{min} = \frac{\tau_{max}}{a}$
- Reinitialization of τ if:
  - stagnation occurs
  - idle iterations

Results obtained are better than AS, EAS, and ASrank, and of similar quality to ACS's

Ant Colony System (ACS)

Three main ideas:

- Different state transition rule

$$j = \begin{cases} \arg\max_{l\in N_i^k}\{\tau_{il}\eta_{il}^{\beta}\} & \text{if } q \le q_0 \\ p_{ij} = \frac{[\tau_{ij}]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{l\in\mathcal{N}_i^k}[\tau_{il}]^{\alpha}\cdot[\eta_{il}]^{\beta}} & \text{otherwise} \end{cases}$$

- Global pheromone update

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \rho\Delta_{ij}^{bs}(s)$$

  to only (ij) in best-so-far tour (O(n) complexity)
- Local pheromone update: happens during tour construction to avoid other ants to make the same choices:

$$\tau_{ij} \leftarrow (1-\epsilon)\cdot\tau_{ij} + \epsilon\tau_0 \qquad \epsilon = 0.1, \tau_0 = \frac{1}{nC^{NN}}$$

  Parallel construction preferred to sequential construction

Approximate Nondeterministic Tree Search

- Use of lower bound to compute heuristic value
  - Add an arc to the current partial solution and estimate LB of complete solution
- Different solution construction rule

$$p_{ij}^k = \frac{\alpha\tau_{ij} + (1-\alpha)\eta_{ij}}{\sum\limits_{l \in N_i^k} \alpha\tau_{il} + (1-\alpha)\eta_{il}}$$

- Different pheromone trail update rule

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{i=1}^{k} \Delta_{ij}^k \qquad \Delta_{ij}^k = \begin{cases} \theta(1 - \frac{C^k - LB}{L_{avg} - LB} & \text{if } (ij) \text{ in belongs to } T \\ 0 & \text{otherwise} \end{cases}$$

Ant Colony Optimization
The Metaheuristic
ACO Variants
Analysis
Application Examples

Strongly Invariant ACO

Considers instances which are equivalent up to a linear transformation of units.

The siACO is an algorithm that enjoy the property of that its internal state at each iteration is the same on equivalent instances.

For AS:

- Use heuristic values $\eta_{ij} := \frac{C^{NN}}{n \cdot c_{ij}}$
- Update according to

$$\tau_{ij} := (1-\rho) \cdot \tau_{ij} + \sum_{s \in sp'} \Delta_{ij}(s)$$

  where $\Delta_{ij}(s) := \frac{C^{NN}}{m \cdot C^s}$
  if edge (i,j) is contained in the cycle represented by s', and 0 otherwise.

Can be extended to other ACO versions and to other problems: QAP and Scheduling

Ant Colony Optimization
The Metaheuristic
ACO Variants
Analysis
Application Examples

Outline

Ant Colony Optimization
The Metaheuristic
ACO Variants
Analysis
Application Examples
Theoretical
Experimental

Analytical studies

- [Gutjahr, *Future Generation Computer Systems*, 2000, *Information Processing Letters* 2002] and [Stützle and Dorigo, *IEEE Trans. on Evolutionary Computation*, 2002] have proved convergence with prob 1 to the optimal solution of different versions of ACO
- Runtime analysis of Different MMAS ACO algorithms on Unimodal Functions and Plateaus [Neumann, Sudholt and Witt, *Swarm Intelligence*, 2009]
- [Meuleau and Dorigo, *Artificial Life Journal*, 2002] have shown that there are strong relations between ACO and stochastic gradient descent in the space of pheromone trails, which converges to a local optimum with prob 1
- [Zlochin et al. *TR*, 2001] have shown the tight relationship between ACO and estimation of distribution algorithms
- Studies on pheromone dynamics [Merkle and Middendorf, *Evolutionary Computation*, 2002]

Things to check

- Parameter Tuning
- Synergy
- Pheromone Development
- Strength of local search (exploitation vs exploration)
- Heuristic Information (linked to parameter β)
  Results show that with $\beta = 0$ local search can still be enough
- Lamarkian vs Darwinian Pheromone Updates
- Run Time impact

Parameter tuning

Our experimental study of the various ACO algorithms for the TSP has identified parameter settings that result in good performance. For the parameters that are common to almost all the ACO algorithms, good settings (if no local search is applied) are given in the following table.

ACO algorithm	α	β	ρ	m	τ_0
AS	1	2 to 5	0.5	n	m/C^{nn}
EAS	1	2 to 5	0.5	n	$(e+m)/\rho C^{nn}$
AS_{rank}	1	2 to 5	0.1	n	$0.5r(r-1)/\rho C^{nn}$
$\mathcal{MM}AS$	1	2 to 5	0.02	n	$1/\rho C^{nn}$
ACS	—	2 to 5	0.1	10	$1/nC^{nn}$

Here, n is the number of cities in a TSP instance. All variants of AS also require some additional parameters. Good values for these parameters are:

EAS: The parameter e is set to $e = n$.

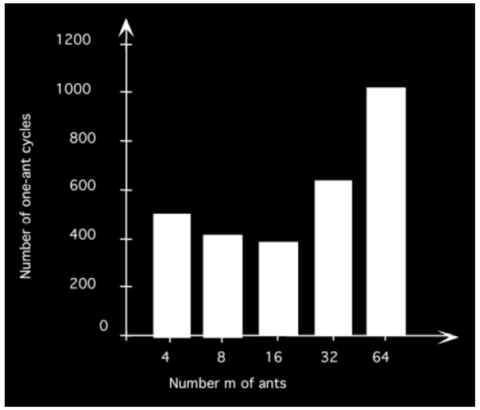
AS_{rank}: The number of ants that deposit pheromones is $w = 6$.

$\mathcal{MM}AS$: The pheromone trail limits are $\tau_{max} = 1/\rho C^{bs}$ and $\tau_{min} = \tau_{max}(1 - \sqrt[n]{0.05})/((avg - 1) \cdot \sqrt[n]{0.05})$, *where avg* is the average number of different choices available to an ant at each step while constructing a solution (for a justification of these values see Stützle & Hoos (2000). When applied to small TSP instances with up to 200 cities, good results are obtained by using always the iteration-best pheromone update rule, while on larger instances it becomes increasingly important to alternate between the iteration-best and the best-so-far pheromone update rules.

ACS: In the local pheromone trail update rule: $\xi = 0.1$. In the pseudorandom proportional action choice rule: $q_0 = 0.9$.

It should be clear that in individual instances, different settings may result in much better performance. However, these parameters were found to yield reasonable performance over a significant set of TSP instances.

How Many Ants?



Number of tours generated to find the optimal solution as a function of the number m of ants used

Pheromone development

http://tandem.informatik.uni-leipzig.de/~merkle/ACO/gifs/l-r-eval-lateness.html

Outline