Bio-Inspired Algorithms

Ant Colony Optimization

Daniel Vaz
daniel.ramosvaz@esiee.fr

2026-03-19

In the News

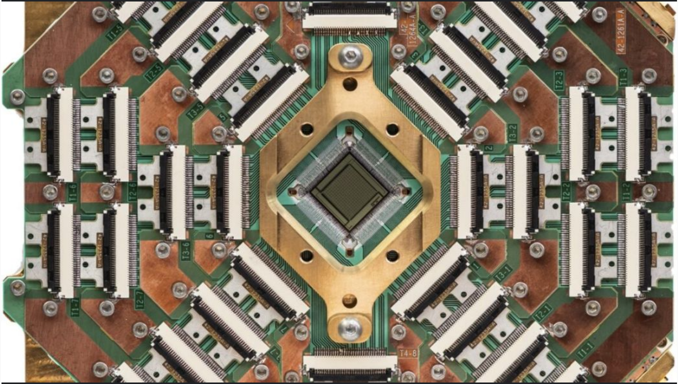
ars TECHNICA

HERE WE GO AGAIN

D-Wave quantum annealers solve problems classical algorithms struggle with

The latest claim of a clear quantum supremacy solves a useful problem.

JOHN TIMMER – 12 MAR 2025 21:36 | 21



Credit: D-Wave

https://bit.ly/41KTgLi

Recap

Bio-inspired algorithms:

- Inspired in nature
- Mostly for optimization problems
    - Find the solution with the best fitness
- NP-hard problems: as hard as any in NP
- We will use heuristic approaches
    - which use practical knowledge

Some simple heuristics:

- Random search: tests random sampled solutions
- Hill climbing: moves to better neighbors
- A*: uses heuristic to center shortest path search

Local search:

- Repeated local improvement
- Need solution representation and neighborhood
- Focused on exploitation, little or no exploration

Greedy local search:

- Choose neighbor that minimizes fitness
- Inefficient, but very general
- Gets trapped in local minima
- Examples: minimization, TSP, scheduling, maze solving
- Variations, optimization, and improvements

Local search with random restart

- Introduces exploration by restarting from random solutions
- Greatly improves results
  - Even for comparable number of iterations

Simulated annealing

- Accept bad moves with some probability
  - Depending on the solution and temperature
- Bad moves allow escaping from local optima
- Good quality and also quite customizable

Tabu search

- Keeps a list of forbidden solutions
- To prevent cycling, but also to guide the search

Types of approaches

Approaches can be:

- deterministic or stochastic
- based on a single solution or a population
- local or global

Algorithm: Random search

Types of approaches

Approaches can be:

- deterministic or stochastic
- based on a single solution or a population
- local or global

Algorithm: Hill climbing

Types of approaches

Approaches can be:

- deterministic or stochastic
- based on a single solution or a population
- local or global

Algorithm: Greedy local search

Types of approaches

Approaches can be:

- deterministic or stochastic
- based on a single solution or a population
- local or global

Algorithm: Simulated annealing

Types of approaches

Approaches can be:

- deterministic or stochastic
- based on a single solution or a population
- local or global

Algorithm: Ant colony optimization

Outline

Ant Colony Optimization

Ant Colony Optimization

Inspiration and Principle

Natural Inspiration

Inspired by ant foraging

- Mechanism of searching for food
- Once food is found:
    - Organized behavior
    - Lines of ants following paths
- How are these paths designed?
    - Complex philosophical question
    - Concept of emergence

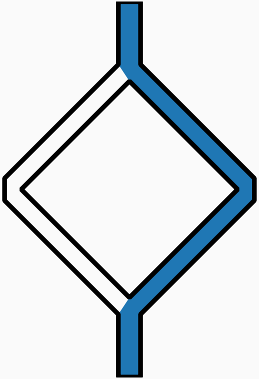
Double Bridge Experiment

Setup:

- Fork in the path
- Two paths with the same length

Results:

- All ants take the same path
  - In 90% of cases
- Symmetry breaking mechanism



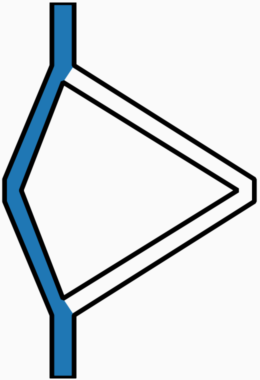
Double Bridge Experiment

Setup:

- Fork in the path
- Two paths with different length

Results:

- All ants take the shortest path
  - In 90% of cases
- Optimization of the path length



Optimizing Behavior

Single ant: follows any path

- Regardless of optimality
- Similar to random walk

What drives optimization?

- Complexity from interaction of simple agents
- Optimization as emergent phenomenon

Agent Interaction

How do the ants coordinate?

- Stigmergy: Communication through the environment
- Indirect form of communication

Mechanism elements:

- Environment
- Pheromones

Ant Mechanisms

Environment

- Serves as the information storage (memory)
- Dictates the tasks of the agents
- Allows for complexity to develop

Pheromones

- Ants deposit pheromones (marking)
    - Usually while carrying food
- Evaporation mechanism
    - Pheromones dissipate

Optimization using Ants

Example: finding a shortest path

Algorithm: simulate ants in an abstract environment

- Including movement, marking, evaporation
- Take as solution the path used by the ants

Optimization through feedback cycle

- Shortest paths are marked more often
  - As ants repeat the path faster
- Thus, they are picked more often
  - Which reinforces marking
  - And lets other paths evaporate
- Small differences are amplified
  - Symmetry breaking

ACO Meta-Heuristic

Principle

- Move virtual ants on virtual paths
    - Need to define what is a path
- Ants sense and deposit pheromones on paths
    - Paths with more pheromones are picked more often
- Global algorithm operated locally

Functioning

1. Each ant builds its own solution
    - In function of pheromones
2. Some (or all) ants deposit pheromones
    - Along their own solution

ACO Meta-Heuristic

Algorithm AntColonyOptimization

Input: Graph $G = (V, E)$, fitness f, population size m

Output: solution p^*

$\tau := \textsc{InitializePheromones}(G)$
$p^* := \textsc{RandomSolution}(G)$
while not $\textsc{Finished}$ **do**
    $P := \textsc{ConstructSolutions}(G, \tau, m)$
    $\textsc{SolutionImprovement}(G, f, P)$ *// optional*
    $\textsc{GlobalUpdatePheromones}(G, f, \tau, P)$
    $p^* := \textsc{Best}(P, p^*, f)$
end while
return p^*

construct solutions for each ant.

Ant Colony Optimization

Ant System

Ant System

Ant System: first ant colony optimization algorithm

- Initially an algorithm for TSP
- So we keep TSP in mind as an example

Solution Construction

- Every ant starts from a random vertex
- Probabilistic rule for next vertex selection
- Continue until complete tour, without repetition

Pheromone Update

- Every ant deposits pheromones
  - Inversely proportional to solution quality
- Global update after all the solutions are constructed

Ant Movement

Solution construction: Path

- An ant starts at the source
- It moves randomly until it reaches the target
- Once it does, the path is our solution

How to move randomly?

τ : tau

- Each edge uv has some quantity τ_{uv} of pheromones
- At each point, pick next vertex as a random neighbor
    - With probabilities proportional to τ_{uv}

Ant Movement (Ant System)

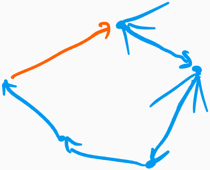
From u, move along uv with probability

$$p_{uv} = \frac{\tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}}{\sum_{w} \tau_{uw}^{\alpha} \cdot \eta_{uw}^{\beta}}$$

↳ scaling sum of probs to 1.

Ant Movement (Ant System)

TSP:



From u, move along uv with probability

$$p_{uv} = \frac{\tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}}{\sum_w \tau_{uw}^{\alpha} \cdot \eta_{uw}^{\beta}}$$

Parameters

- τ_{uv}: Amount of pheromones on uv
- eta η_{uv}: natural interest of uv (e.g. length)
  - Maximizing by default, use $\beta < 0$ for minimizing
- α: importance of pheromone
  - Typically 1 to 3; ignored if 0
- β: importance of natural interest
  - Typically 2 to 5; ignored if 0

Pheromone update (Ant System)

Update: done after every round by

- Evaporation of existing
- Deposit of new pheromones

Evaporation

- By exponential decay: $\tau_{uv} \to (1 - \rho)\tau_{uv}$
- ρ is the evaporation rate

Deposit

- As a function of the number (and quality) of paths through uv
- Add $1/f(s)$ for every ant s traversing the edge

Algorithm AntSystem

Input: Graph $G = (V, E)$, path-fitness f, population size m

Set $\tau_{uv} := 0$ for all $uv \in E$

while not FINISHED **do**

construct

  for $i = 1$ to m **do** // *Generate m solutions*

    Generate random single-vertex path $p_i = (v_i)$

    while $f(p_i) = +\infty$ **do**

      $u := \textsc{Last}(p_i)$, $X := V \setminus V(p_i)$

      Add a random edge uv, $v \in X \cup \{v_i\}$ w.p. $\sim \tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}$

    end while

  end for

evaporation

  Set $\tau := (1 - \rho)\tau$

  for $i = 1$ to m **do**

deposit

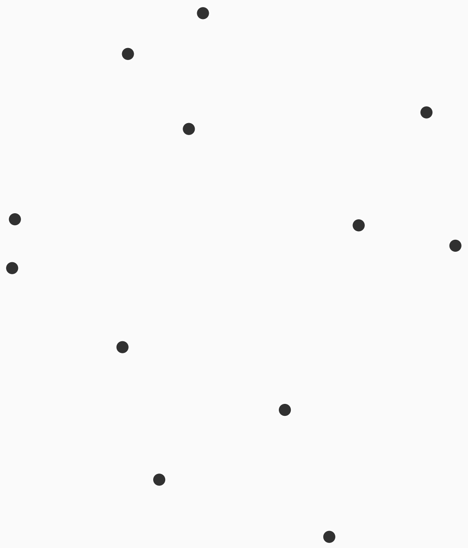
    for $uv \in E(p_i)$ **do** Set $\tau_{uv} := \tau_{uv} + 1/f(p_i)$

  end for

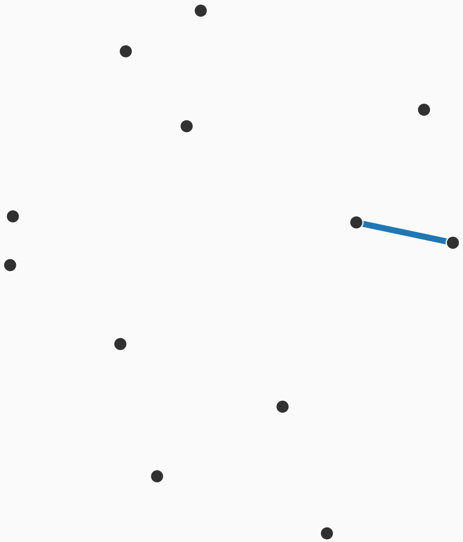
end while

return p_i with smallest $f(p_i)$

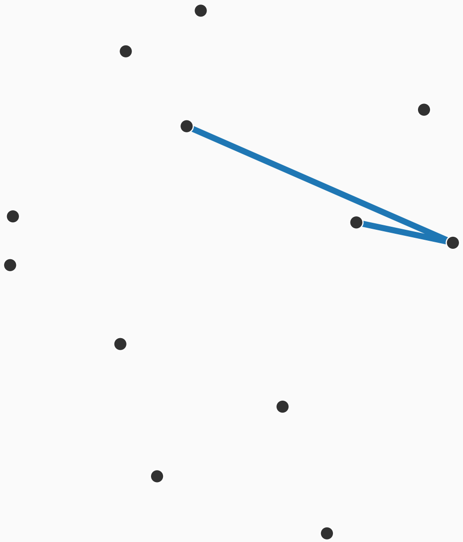
Ant System



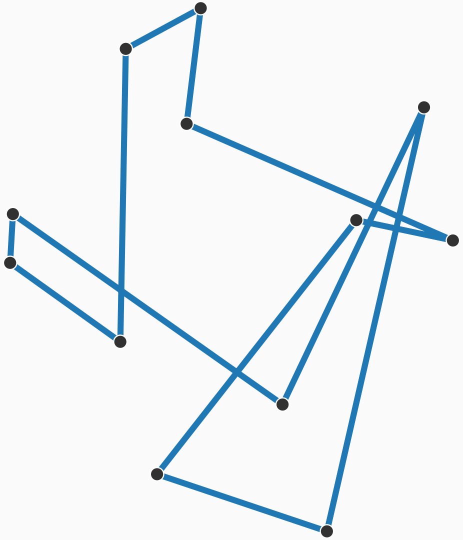
Ant System



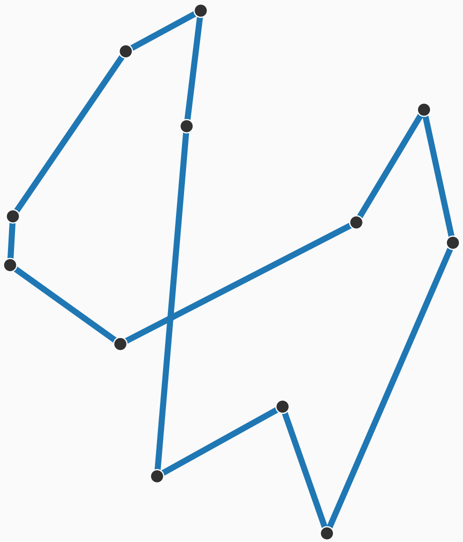
Ant System



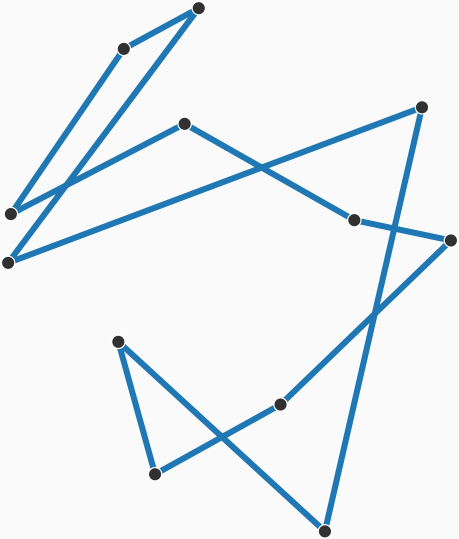
Ant System



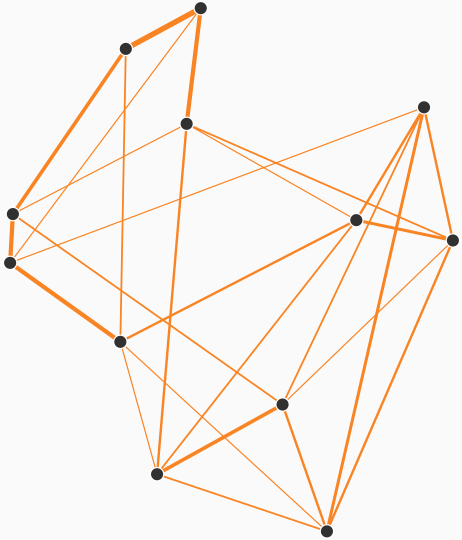
Ant System



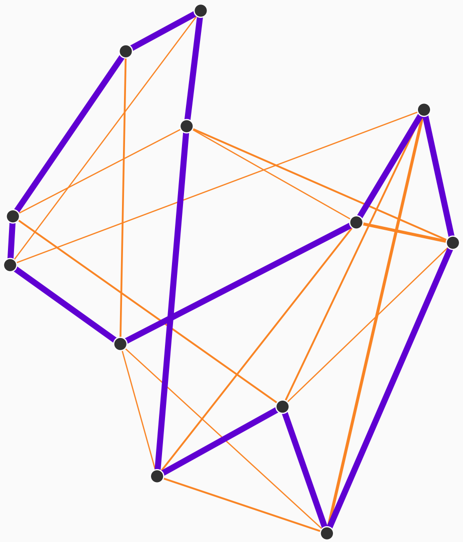
Ant System



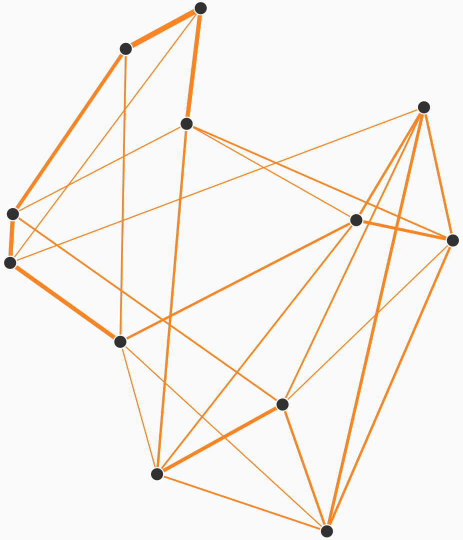
Ant System



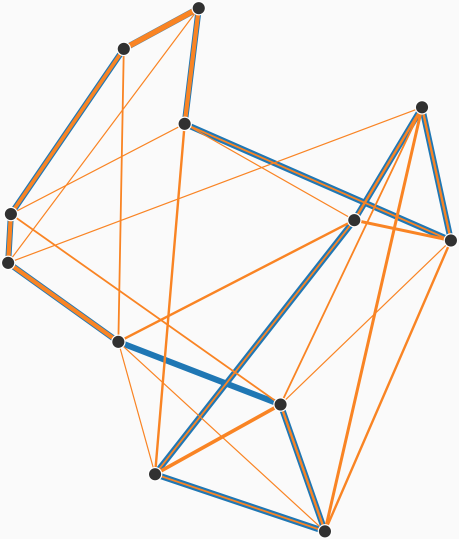
Ant System



Ant System



Ant System



Ant System

Advantages

- Simple algorithm that finds good solutions
- Incentivized research to develop improvements

Limitations

- Slow convergence on large instances
- Limited exploration capability
- Not state-of-the-art

Ant Colony Optimization

Advanced Algorithms

Elitist Strategy: boosts global optimum

- Similar to ant system
- Adds a strong extra weight to the global best
  - Found since the beginning of the algorithm
  - Deposit of $\gamma/f(s_{\text{best}})$

AS-rank:

- Allows only r ants to deposit pheromones
- The global best with weight r (add $+r/f(s_{\text{best}})$)
- The i-th best in the round with weight $r - i$

Max-Min Ant System (MMAS)

Pheromone limits

- Pheromone value between τ_{min} and τ_{max}
- Pheromone values initialized to the upper limit

Rank-1 pheromone update

- Only global best is updated
- Updated with weight 1, that is, add $+1/f(s_{\text{best}})$
- Sometimes, iteration-best is used instead

Other improvements:

- Trail reinitialization
- Local search before pheromone update

Ant Colony System (ACS)

Rank-1 pheromone update with weight ρ

Pseudo-random moves

- Each neighbor v gets a value $\omega_v = \tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}$
- With probability q_0 choose neighbor with max ω_v
- Otherwise, use normal move in ant system
- q_0 usually close to 1
  - Favors exploitation over exploration

Pheromone consumption

- Ants "eat" pheromone when traversing edges

Ant Colony Optimization

Applications

$$p_{uv} = \frac{\tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}}{\sum_{uw} \tau_{uw}^{\alpha} \cdot \eta_{uv}^{\beta}}$$

Solution

- Each solution is a tour visiting every vertex
- Components: edges
- Natural interest: $\eta_{uv} := \ell_{uv}$, $\beta < 0$

↳ shorter edges are better

$$\beta = -1 \quad \eta_{uv}^{\beta} = \frac{1}{\eta_{uv}}$$

TSP

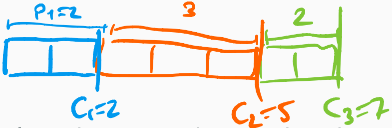
Solution

- Each solution is a tour visiting every vertex
- Components: edges
- Natural interest: $\eta_{uv} := \ell_{uv}$, $\beta < 0$

Solution construction

- Ant starts at a random starting vertex
- Memory: path, visited vertices
- Advance to an unvisited vertex
- After all vertices visited, complete the tour
  - By moving to the starting vertex

Single-Machine Tardiness Scheduling



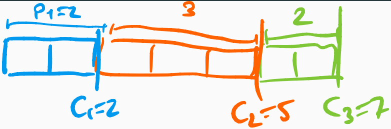
Problem

- n jobs to process, each with length p_i, weight w_i, due date d_i
- Schedule jobs sequentially with completion times C_i
- Minimize $\sum_i w_i \max(C_i - d_i, 0)$

$d_2=4$ $C_2-d_2=1$

$d_1=5$ $C_1-d_1=-3$

Single-Machine Tardiness Scheduling



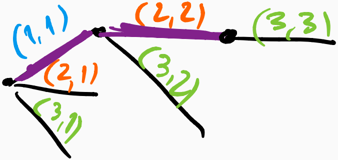
Problem

- n jobs to process, each with length p_i, weight w_i, due date d_i
- Schedule jobs sequentially with completion times C_i
- Minimize $\sum_i w_i \max(C_i - d_i, 0)$

Solution: Assignment

- Components: pairs (i, j), job i assigned to position j
- Natural interest: due date $\eta_{ij} := d_i$, $\beta < 0$
  - Other option: modified due date $\eta_{ij} := \max(C + p_j, d_j)$

(1,1) (2,2) (3,3)



Single-Machine Tardiness Scheduling

Problem

- n jobs to process, each with length p_i, weight w_i, due date d_i
- Schedule jobs sequentially with completion times C_i
- Minimize $\sum_i w_i \max(C_i - d_i, 0)$

Solution: Assignment

- Components: pairs (i, j), job i assigned to position j
- Natural interest: due date $\eta_{ij} := d_i$, $\beta < 0$
  - Other option: modified due date $\eta_{ij} := \max(C + p_j, d_j)$

Solution construction: in order of increasing j

Set Cover

$n=5$ $\{1,2,4\}$, $\{3,4\}$, $\{2,5\}$, $\{1,2,5\}$.
1 1 1 1

Problem

- m subsets: $S_i \subseteq [n]$ with associated cost c_i
- Find minimum-cost collection of subsets such that union is $[n]$

Set Cover

Problem

- m subsets: $S_i \subseteq [n]$ with associated cost c_i
- Find minimum-cost collection of subsets such that union is $[n]$

Solution: collection of subsets S_i

- Components: indices i (of subsets S_i)
- Natural interest: $\eta_i := |S_i'|/c_i$, *new* elements covered per euro
  - Other options: $\eta_i := 1/c_i$, $\eta_i := |S_i|/c_i$
  - $\eta_i := 0$ if subset does not cover any new elements

Post-processing

- Remove redundant subsets
- 3-flip local search

Dynamic Network Routing

Problem: Shortest path in telecommunications network

- Very difficult when edge costs vary randomly over time
- Models the real-world conditions of the internet

Principle

- Solutions adapt to changing costs
  - By using pheromone deposition and evaporation
- Encode how to get from any source to any target

AntNet

Specialized algorithm for network routing

Ants

- Each ant has a different source s and a random target t
- Solution is an s-t-path
- Movement similar to ant system

Pheromones

- Associate with edge-target pairs: $\tau_{uvt} \in [0, 1]$
- An ant at u going to t uses the values τ_{uvt} for all v

AntNet

Specialized algorithm for network routing

Ants

- Each ant has a different source s and a random target t
- Solution is an s-t-path
- Movement similar to ant system

Pheromones

- Associate with edge-target pairs: $\tau_{uvt} \in [0, 1]$
- An ant at u going to t uses the values τ_{uvt} for all v

Process

- Ants are special packets in the network, with the same delays
- Given priority on the way back, for pheromone update

AntNet

Execution

- Simulated in a virtual network, with computers are vertices
  - With real data
- Special type of ant packets implemented
  - Online measurement of fitness

Pheromones

- For each edge and each target
- Each computer stores pheromones for outgoing edges
- Ants use the pheromones for right target

Solution

- The information we obtain is the pheromones
- To send a normal packet:
  - Simulate an ant at every computer

Other Applications

Applications in many NP-hard problems

Methods

- Local search used often
- Mostly Max-Min Ant System and Ant Colony System

Fine tuning

- Appropriate natural interest heuristic
- Informed construction mechanism
- Specialized local search methods

Ant Colony Optimization

Application Principles

Solutions and Pheromones

Solutions and Components

- Have a big influence on the performance of the algorithm
- Usually the best approach is natural
- For example:
  - Sequence or permutation of items (AntNet, TSP, Set Cover)
  - Sequence using position-job pairs (Scheduling)

Pheromones

- A good pheromone model brings good results
- Simple approach: pheromones per solution component
- Could also be more complex (e.g. AntNet)

Balancing Exploration and Exploitation

Good balance is usually achieved through pheromone management

- Pheromones induce a probability distribution for the solutions
- Greatly influences the exploration of the search space
- Parameters α, β also have an important role

Exploitation

- Encouraged by elitist or rank update strategies
- Also by focused moves such ACS pseudo-random moves

Exploration

- Mechanisms are introduced to counter-balance
- ACS: Consumption of pheromones on move
- MMAS: Lower limit and reinitialization

Enhancement through Local Search

Ant colony optimization and local search are complementary

- ACO obtains random, unrefined solutions of good quality
- Local search refines solutions, but needs a good start

Experimental results

- Suggest that the combination can yield excellent results
- Reduces variability of local search

Natural Interest Heuristic

Natural interest heuristic: allows to exploit problem knowledge

- Known in advance (common for NP-hard problems)
- Adapted to a specific setting
- Dynamic information, if the instance changes with time

Methods

- Pre-computed information (distances in TSP)
- Dependent on partial solution (set cover, scheduling)
  - Higher running time, but better solutions
- Even experimentally obtained (AntNet)
  - Useful in dynamic real-world settings

Conclusion

Summary

Ant Colony Optimization

- Meta-heuristic based on a population of ants
- Ants move according to pheromones
- Pheromones evaporate and are deposited by ants
- Creates differentiation by quantity of the pheromones

Approaches

- Ant system, elitist and ranked strategies, MMAS and ACS

Application Principles

- Applications: TSP, scheduling, set cover, network routing
- Different applications require different adjustments
  - Solutions and pheromones, balance, local search, heuristic