Ant Colony Optimization and Particle Swarm Optimization

– welcome to the zoo

Thomas Stidsen
thst@man.dtu.dk

DTU-Management
Technical University of Denmark

Outline

- Ant Colony Optimization (ACO)
- Particle Swarm Optimization (PSO)

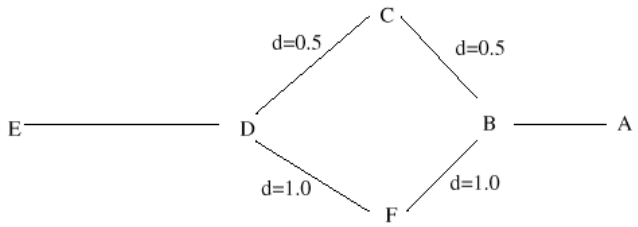
The power of many ...

- One ant is stupid
- But, many ants are smart collectively.

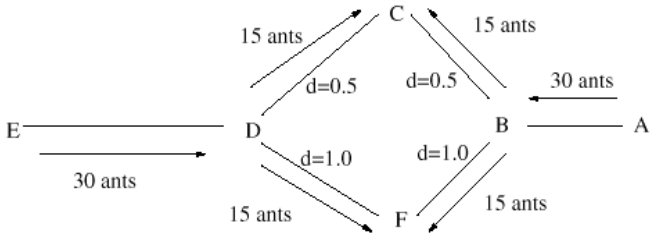
What is Ant Optimization?

- “Swarm intelligence is a property of systems of non-intelligent robots exhibiting collectively intelligent behaviour”.
- Characteristics:
  - distributed, no central control or data source.
  - ability to change environment.
  - perception of environment, i.e. sensing.

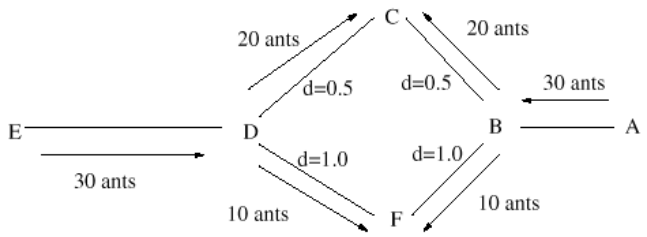
Pheromone Trails



Pheromone Trails Continued



Pheromone Trails Continued Again



The Travelling Salesant

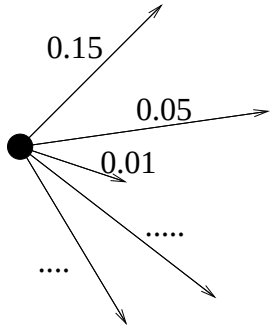
- We have given a graph $G = (N, A)$. d_{ij} is the distance between i and j.
- Before we start we have k ants in each city.

The actions of an ant

- Choose the next city to go to. The choise is a function of the distance to the city and the amount of pheromone.
- Cities already chosen can not be visited.
- When finished: Put out a trail of pheromone on the TSP-tour.

The pheromone decision

When the salesant has to decide which next city to visit:



Choose the next city

- The simple one:
  $p_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{k \in j} \tau_{ij}(t)}$

Generally the simple method has convergence problems as we do not consider the distance d_{ij}.

- Define the visibility n_{ij} : $n_{ij} = \frac{1}{d_{ij}}$.
- Now define $p_{ij}(t)$ as:
  $p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[n_{ij}]^{\beta}}{\sum_{k \in j}[\tau_{ij}(t)]^{\alpha}[n_{ij}]^{\beta}}$ if transition is allowed, 0 else.

Be aware of

α and β controls the relative importance of trail and visibility. Note that if α = 0 one would have a random greedy algorithm with multiple starting points.

- High values of and will lead to stagnation, that is, early convergence.
- Low values of and might lead to weak convergence.

In papers and values from 0:5 to 5 are often used. Experiments are needed to set them appropriately.

Update of pheromone

Basicly we update the pheromone trail after each TSP iteration:
$\tau_{ij}(t+1) = \gamma\tau_{ij}(t) + \Delta\tau_{ij}(t)$
The factor $0 < \gamma < 1$ is the "evaporation" factor. This ensures a certain "dynamism" is the system.

Update of pheromon continued

How should we set $\Delta\tau_{ij}(t)$?

- Arcs not used by the ant does not get any pheromone.
- $\Delta\tau_{ij}(t) = \sum_k \Delta\tau_{ij}^k(t)$

Update of pheromone continued again

- $\Delta\tau_{ij}^k(t) = c$ where c is a constant.
- Use the arc length dij of the ant k:$\Delta\tau_{ij}^k(t) = \frac{Q}{d_{ij}}$ where Q is a constant.
- Use the tour length T_k of the ant k: $\Delta\tau_{ij}^k(t) = \frac{Q}{T_k}$ where Q is a constant.
- In an elitist strategy we put more emphasis on the pheromone from the "good" ants.

Good values for γ generally tends to be from 0.5 to 0.9.

Stopping criterion

- Run a fixed number of iterations, or better: For a fixed amount of time !
- Run until stagnation occurs (all ants travel the same tour).
- Run until convergence has occurred (a soft version of stagnation).

ACO - Biological Similarities

- Colony of cooperating individuals.
- An (artificial) pheromone track.
- A sequence of local moves.
- A stochastic decision policy using **local** information and **no** lookahead.

The ACO algorithm

Initialise pheromone values
repeat
  Fora ant $k \in \{1, ..., m\}$
    construct k'th ant solution
  Endfor
  For all pheromone values
    decrease pheromone (evaporation)
  Endfor
  For all good solutions
    increase pheromone (intensification)
  Endfor
until time

ACO - Biological Differences

- Their moves consists of transitions from state to state.
- Artificial ants has an internal state (= memory).
- Artificial ants deposit an amount of pheromone that is a function of the quality of the solution found.
- Artificial ants' timing inpheromone laying is problem dependent and often does not reflect real ants' behaviour.
- *Extra capabilities* are often added to enhance algorithm performance.

Different Names

- Ant Colonies.
- Ant Optimization.
- Ant Systems.
- Collective Intelligence.

Particle Swarm Optimization

- Mimic a swarm of particles, insects ...
- Gradually the swarm zooms in on the optimal solution

The PSO algorithm

Initialise location and velocity of each particle
repeat
    For each particle
        evaluate objective function for each particle
    For each particle
        update best solution
    update best global solution
    For each particle
        update the velocity
        compute the new locations of the articles
until finished()

The parameters are then

- x_{id}: The "new" positions ...
- v_{id}: The "velocity" of the particle (the new position of the particle ...)
- p_{id}: The "individual" bests of the particles
- p_{gd}: The "global" best of all particles

The Updates

- $v_{id} = w \cdot v_{id} + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (p_{gd} - x_{id})$
- $x_{id} = x_{id} + v_{id}$

The effect is that the good solutions (particle positions) "attracts" the particles ...

Solved Problems

- PSO is mostly used for continuous optimization (it seems)
- There is an example in the book that

To my big surprise, there are actually articles which describe PSO for different combinatorial optimization problems ...

Comments

- For continuous problems, it is very hard to compare with gradient methods. PSO could be used if the functions are not differentiable ...
- I doubt that it is reasonable for integer problems ...
- There are a number of suggestions for improvements in the book.