P–Metaheuristics

Swarm intelligence (Ch 3.6) and Conclusions of P-heuristics (Ch 3.9)

Joen Dahlberg

Communications and Transport Systems
Department of Science and Technology
Linköping University

LINKÖPING UNIVERSITY

Outline

1 Swarm Intelligence
  - Short Introduction
  - Ant Colony
  - Particle Swarm
2 Conclusions of P-heuristics

LINKÖPING UNIVERSITY

Introduction

Algorithms inspired from the collective behaviour of species such as ants, bees, wasps, termite, fish, and birds are referred as swarm intelligence algorithms.

The main characteristics of swarm–intelligence–based algorithms are particles are simple, they cooperate by an indirect communication medium, and do movements in the decision space.

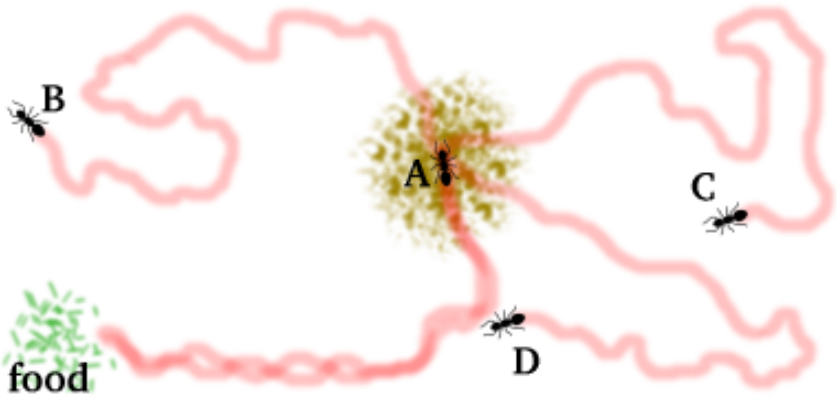
Among the most successful swarm intelligence inspired optimization algorithms are ant colony optimization and particle swarm optimization.

Ant Colony Optimization (ACO)

Traditionally, ACO have been applied to combinatorial optimization problems, e.g. scheduling, routing and assignment problem. It is inspired by ants scouting for food.

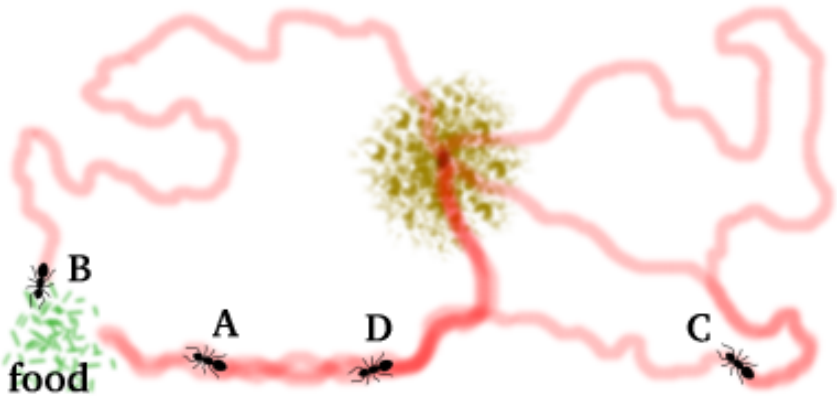
Ant Colony Optimization (ACO)

Traditionally, ACO have been applied to combinatorial optimization problems, e.g. scheduling, routing and assignment problem. It is inspired by ants scouting for food.



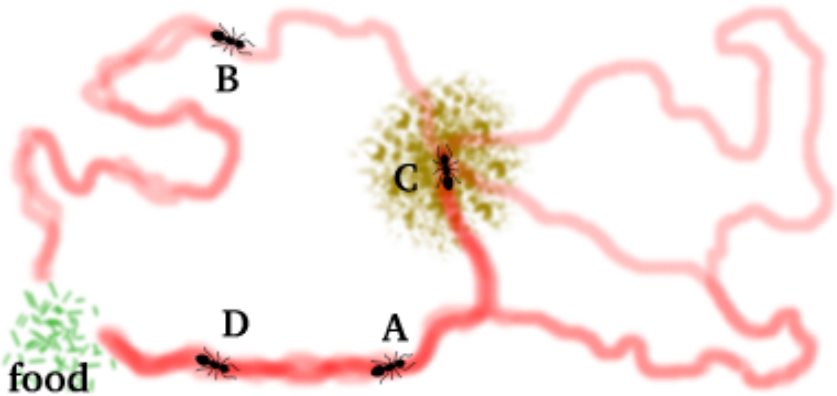
Ant Colony Optimization (ACO)

Traditionally, ACO have been applied to combinatorial optimization problems, e.g. scheduling, routing and assignment problem. It is inspired by ants scouting for food.



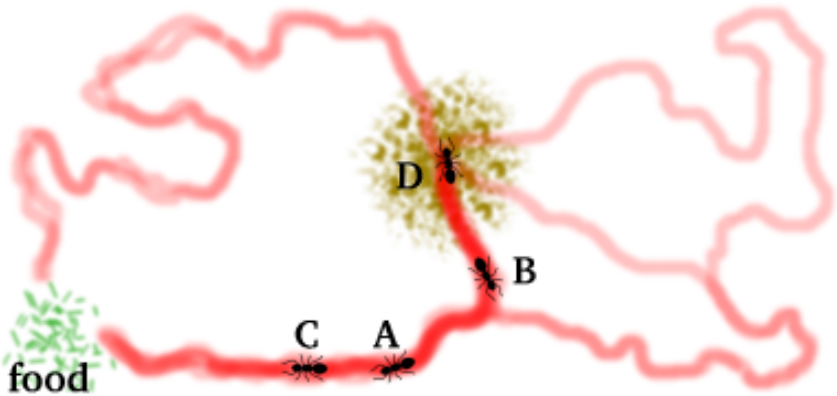
Ant Colony Optimization (ACO)

Traditionally, ACO have been applied to combinatorial optimization problems, e.g. scheduling, routing and assignment problem. It is inspired by ants scouting for food.



Ant Colony Optimization (ACO)

Traditionally, ACO have been applied to combinatorial optimization problems, e.g. scheduling, routing and assignment problem. It is inspired by ants scouting for food.



Ant Colony Optimization (ACO)

Traditionally, ACO have been applied to combinatorial optimization problems, e.g. scheduling, routing and assignment problem. It is inspired by ants scouting for food.

Movie: Ant leaving trail

Inspired parameters

The algorithm uses ants and pheromone trails. The pheromone trail leads the ants to good solutions. The pheromone trails decreases over time (evaporation), but the ants adds new pheromones based on their findings (reinforcements).

Translation to heuristics

Ants

They are stochastic greedy procedures that construct solutions in a probabilistic manner by adding solution components until a complete solution is derived. The problem is usually a decision problem where ants walk on edges between nodes.

Pheromone trails

Each edge in the network is given a probabilistic value, τ_{ij}, that an ant will choose that particular edge. The collection of τ–values is the pheromone trail(s).

Translation to heuristics

Evaporation

In the evaporation phase, the probabilistic values are reduced proportionally to the current value, $\tau_{ij} := (1 - \rho)\tau_{ij}$, $\forall(i, j) \in E$, where $\rho \in]0, 1]$. Intensification and diversification of solutions can be adjust based on the choice of ρ.

Reinforcement

There are three different strategies for increasing the probabilistic values. Either at each step of the ant's walk, when a complete solution is found, or when all ants found complete solutions.

Standard algorithm

Algorithm 3.12 Template of the ACO.

```
Initialize the pheromone trails ;
Repeat
   For each ant Do
      Solution construction using the pheromone trail ;
      Update the pheromone trails:
         Evaporation ;
         Reinforcement ;
Until Stopping criteria
Output: Best solution found or a set of solutions.
```

Example 3.18 ACO for the TSP

Algorithm 3.13 Ant colony algorithm for the TSP problem (ACO–TSP).

Initialize the pheromone information ;
Repeat
  For each ant **Do**
    Solution construction using the pheromone trails:
      $S = \{1, 2, \ldots, n\}$ /* Set of potentially selected cities */
      Random selection of the initial city i ;
      Repeat
        Select new city j with probability $p_{ij} = \frac{\tau_{ij}^{\alpha} \times \eta_{ij}^{\beta}}{\sum_{k \in S} \tau_{ik}^{\alpha} \times \eta_{ik}^{\beta}}$;
        $S = S - \{j\}$; $i = j$;
      Until $S = \emptyset$
  End For
  Update the pheromone trail:
    For $i, j \in [1, n]$ **Do**
      $\tau_{ij} = (1 - \rho)\tau_{ij}$ /* Evaporation */ ;
    For $i \in [1, n]$ **Do**
      $\tau_{i\pi(i)} = \tau_{i\pi(i)} + \Delta$ /* π: best found solution */ ;
Until Stopping criteria
Output: Best solution found or a set of solutions.

Example 3.18 ACO for the TSP

τ_{ij} is the standard pheromone.
η_{ij} is a problem-dependent pheromone, in this case distance^{-1} ($\frac{1}{d_{ij}}$).
α and β are parameters representing the relative influence of the two pheromones.
$\Delta = \frac{1}{f(\pi)}$, where π is the obtained tour* of the ant.

* in the algorithm π is the best found tour, why? wrong?

Particle Swarm Optimization (PSO)

Originally, PSO has been successfully designed for continuous optimization problems. It mimic the social behaviour of e.g. bird flocking and fish schooling.

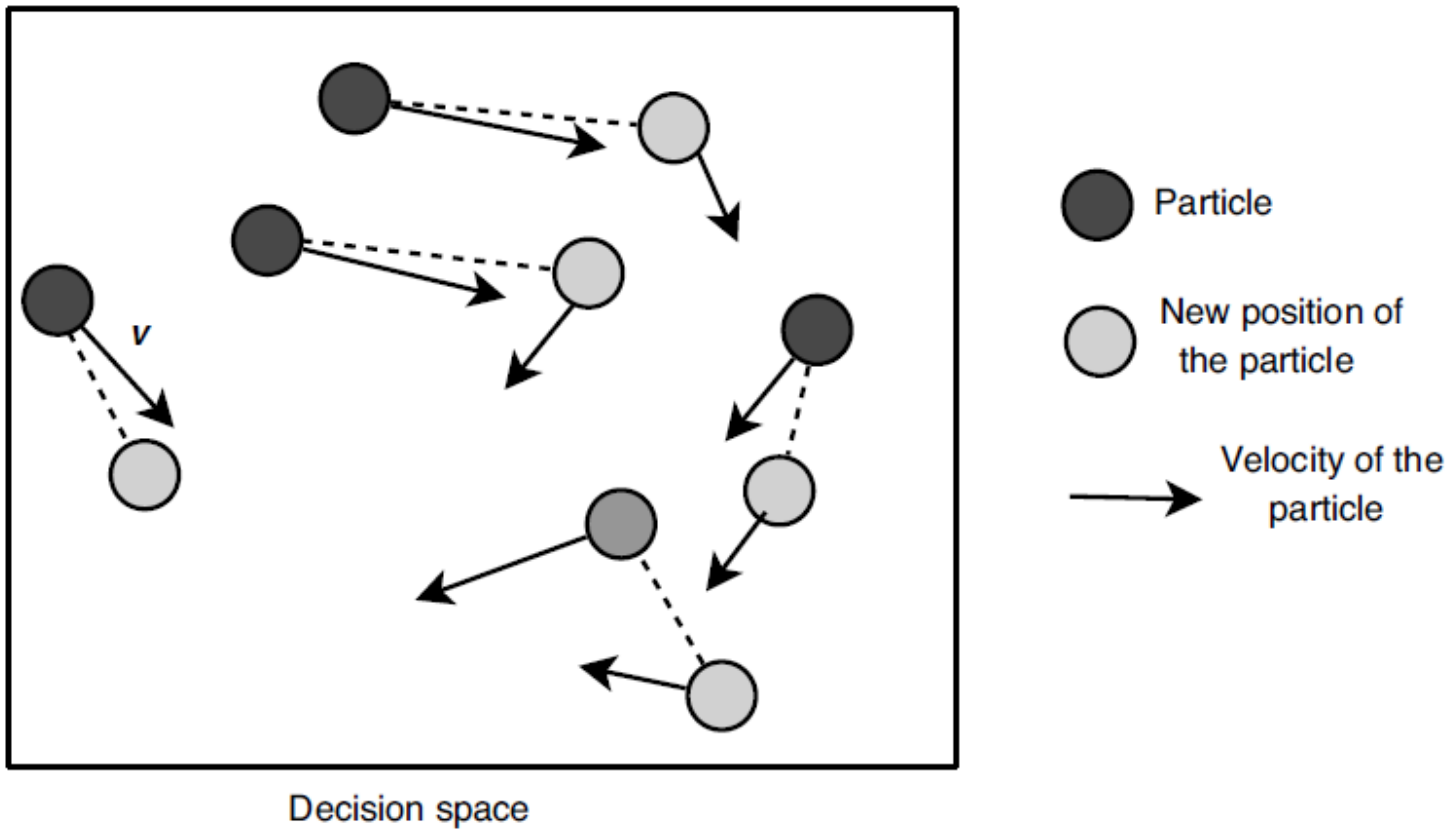
Movie: Flock of Starlings

Inspired parameters

Each particle represents a (complete) solution, it's current position, $x_i(t)$, in the decision space. The particle has a velocity, $v_i(t)$, which defines the moving direction and length of the move at every step of the particle. The particle have a defined neighbourhood, it's social influence. The velocity is changed based on the personal best solution, $pbest_i$ (p_i), and the locally (amongst the particles in the neighbourhood) best found solution, $lbest$ (p_l). The particle may also have an x–fitness and/or p–fitness value(s).

The neighbourhood can range from all particles, a global influence, the $gbest$ Method, to a subset of particles, down to zero distance, no influence at all.

Inspired parameters



Algorithm

Algorithm 3.14 Template of the particle swarm optimization algorithm.

Random initialization of the whole swarm ;
Repeat
  Evaluate $f(x_i)$;
  For all particles i
    Update velocities:
      $v_i(t) = v_i(t-1) + \rho_1 \times (p_i - x_i(t-1)) + \rho_2 \times (p_g - x_i(t-1))$;
    Move to the new position: $x_i(t) = x_i(t-1) + v_i(t)$;
    If $f(x_i) < f(pbest_i)$ **Then** $pbest_i = x_i$;
    If $f(x_i) < f(gbest)$ **Then** $gbest = x_i$;
    Update(x_i, v_i) ;
  EndFor
Until Stopping criteria

Algorithm

$v_i(t) = v_i(t-1) + \rho_1 \cdot (p_i - x_i(t-1)) + \rho_2 \cdot (p_{g/l} - x_i(t-1))$

Adjustment towards individual best solution scaled with a random variable $\rho_1 \in [0, 1]$.

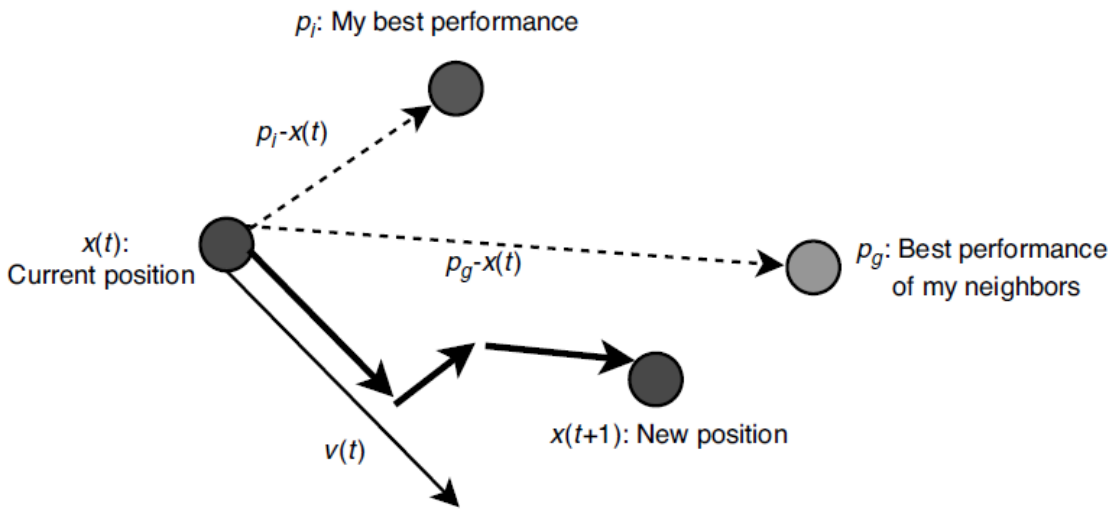
Adjustment towards global/local best solution scaled with a random variable $\rho_2 \in [0, 1]$.

Algorithm

$v_i(t) = v_i(t-1) + \rho_1 \cdot (p_i - x_i(t-1)) + \rho_2 \cdot (p_{g/l} - x_i(t-1))$

Adjustment towards individual best solution scaled with a random variable $\rho_1 \in [0, 1]$.

Adjustment towards global/local best solution scaled with a random variable $\rho_2 \in [0, 1]$.



An overlook

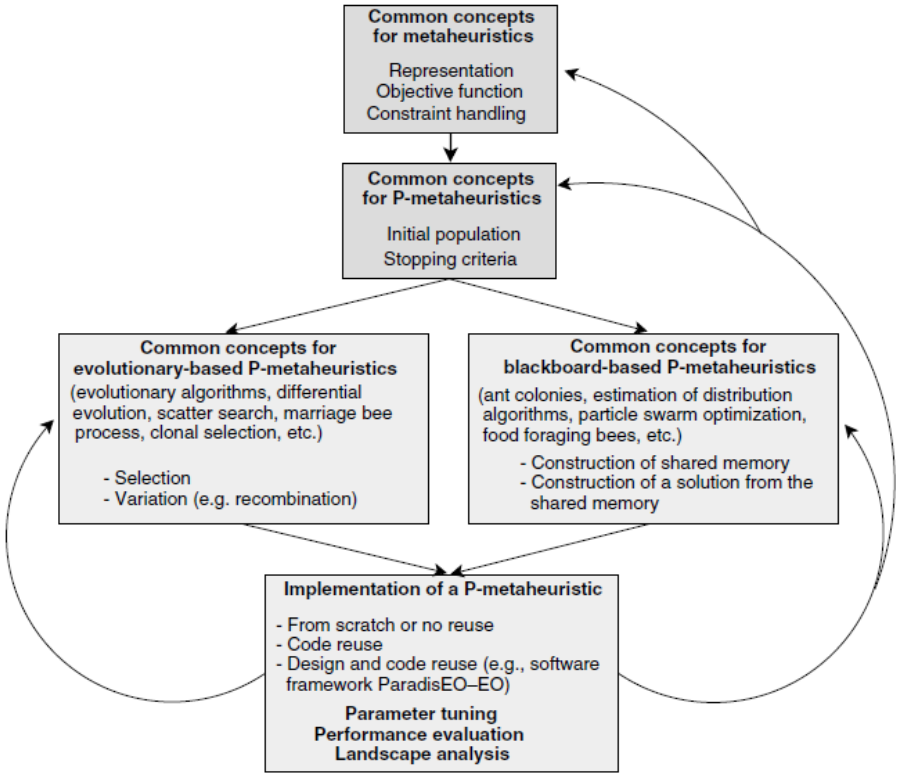


FIGURE 3.63 Development process of P-metaheuristics.

Final remark

P–metaheuristics and especially nature-inspired algorithms (e.g., evolutionary algorithms, immune systems, particle swarm, bee colony, and ant colony) are well suited to solve complex optimization problems dealing with uncertainty and dynamicity. Indeed, in solving this type of problems, one must focus on a more efficient diversification of the search to react promptly to changes of the landscape or generate more robust solutions.

Thank you for listening!

www.liu.se

LINKÖPING UNIVERSITY