Optimization Based on the Swarm Intelligence

Zdeněk Krupka, Vladimír Řeřucha

Abstract:

In this paper the term swarm intelligence is explained and corresponding metaheuristics ant colony optimization and particle swarm optimization are described.

Keywords:

Swarm intelligence, ant colony optimization, particle swarm optimization

1 Introduction

Swarm intelligence is usually defined as any attempt to design algorithms or distributed problem–solving devices inspired by collective behavior of social insect colonies and other animal societies [1]. By other words the swarm intelligence (SI) can be described as the exhibition of collective intelligence by groups of simple agents. The basic premise of the swarm intelligence is presented in the computerized modeling of artificial agents modeled after biological entries that are not programmed with intentional goals individually and yet exhibit problem solving abilities as a collective behavior. Two main concepts of swarm intelligence are stigmergy and allelomimesis. Heard stigmergy meaning communication thought the environment and allelomimesis meaning an individual's reaction its neighbor.

From the system theory point of view the characteristic features of swarm intelligence are: no central control, simple rules for each individual, emergent phenomena and self-organization.

Swarm intelligence – metaheuristic is an arbitrary problem solving strategy which falls under the previous definition. There exist two different metaheuristic: ant colony optimization (ACO) and particle swarm optimization (PSO).

2 Ant colony optimization

Ant colony optimization is metaheuristic for difficult combinatorial optimization problem modeled after the stigmernetic communication of ants finding shortest paths to food sources. This optimization method uses the virtual antslaying out virtual pheromone in the problem states they visit. Every virtual ant can be represented as constructive procedure which solves the problem enters thought a problem graph which is created by nodes and edges. As in the nature the virtual ants communicate indirectly and the solution to the problem emerges by the cooperation of the colony.

Next will be shown how the modeling of ants behavior can be used for the solution of travelling salesmen problem: ants are given a list of cities (nodes) they have already visited, every ant starts at random city (node), follows trail probabilistically without updating pheromone; and after completing the tour ant deposits pheromone depending on quality of solution. The shorter the tour the more pheromone is placed ants work simultaneously and new ants are created as needed to keep the population on desired level. The search is finished when a short enough tour is found or when the determined time has elapsed.

Fore realization of described procedure it is necessary to determine [2]:

- the probability of ant k at city *i* to visit city *j*

$$P_{i,k}(j) = \frac{\tau_{ij}}{\sum \tau_{ic}}, c \notin J_i^k, \tag{1}$$

where J_i^k is list of visited cities and τ_{ij} is the amount of pheromone on edge between two cities *i* and *j*;

- pheromone $\Delta\tau_{ij}^k$ which ant k deposit at the end of each tour on links

$$\Delta\tau_{ij}^k = \prec \begin{matrix} 0 \ldots when\ (i,j)\ has\ been\ used \\ \frac{Q}{L^k} \ldots otherwise \end{matrix}, \tag{2}$$

where Q is fixed amount of pheromone and L^k is length of tour of ant *k*;

-the pheromone for all ants

$$\Delta\tau_{ij} = \sum_{k=1}^{n} \Delta\tau_{ij}^k, \tag{3}$$

- the pheromone for all links

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij}, \tag{4}$$

where ρ is the evaporation coefficient.

Philosophy for solution of traveling salesmen problem by means of ant colony optimization can be represented by following algorithm expressed in pseudo code, see for instance [2].

```
function ACO-TSP()
  nant = number of ants()
  nnode = number of nodes()
  place ants on nodes
```

```
repeat
      loop for k from 1 to nant do
            loop for step from 1 to nnode do
                  choose next node according to probability phi
            end
      end
      update pheromone trails
until some tour good enough or enough time has elapsed
return best tour found so far
```

Note that presented algorithm is nonparallel version of ant colony optimization, but generally the ants-like in nature-can work simultaneously allowing parallel implementation.

3 Particle swarm optimization

Particle swarm optimization is a metaheuristic for the optimization of continuous functions. It utilized the knowledge about the collective behavior – for instance – of swarm of birds. Metaphorically say the virtual birds start in random position with random velocity, there are flying through the space, remember the best position that they have seen, communicate good positions to each other and adjust their own position and velocity based on the good position.

From the computer point of view is the behavior of virtual birds modeled by the particles in solution space that have position and velocity. Particles are moving through the space and are evaluated according to some fitness criterion after each iteration. Over the time, particles within their communication grouping which have better fitness values. A large number of members that make up the particle swarm make the technique resilient against getting stuck in local optima.

For implementation of previous procedure it is necessary update the velocity and position of each particle i through the following formulas at each iteration, see for instance [3]:

$$v_i(t+1) = v_i(t) + c_1 r_1 [x_{bi}(t) - x_i(t)] + c_2 r_2 [x_{gi}(t) - x_i(t)], \tag{5}$$

$$x_i(t+1) = x_i(t) + v_i(t), \tag{6}$$

where $v_i(t)$ and $x_i(t)$ are the actual values of particle velocity and position, $x_{bi}(t)$ is the best particle has seen, $x_{gi}(t)$ is the global best seen by the swarm, r_1, r_2 are random values in the range <0,1> and c_1, c_2 are the learning factors which express how much the particle is directed towards good position and their values are in the interval from one to two. Note, that when a particle takes part of the population as its topological neighbors, then $x_{gi}(t)$ can be replaced by the local best value $x_{li}(t)$. Corresponding algorithm can be express in the following pseudo code, see for instance [4].

```
For each particle
  Initialize particle
End
Do
  For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness in history
              Set actual value as the new xb
        End
  End
  Choose particle with the best fitness value of all the particles as xg
  For each particle
        Calculate particle velocity
        Update particle position
  End
While the maximum iterations or minimum error criteria is not attained
```

4 Conclusion

With ant colony optimization and particle swarm optimization two simple algorithms have been created which can solve difficult computation problems efficiently, while being easy to understand.

In military can be swarm intelligence utilized for solution of traveling salesmen problem for surveillance missions and for applying swarm behavior to unmanned combat aerial vehicle missions.

Because exist a wide variety of swarm behavior in nature, there is a great chance will be see more algorithms and systems modeled after social insects and other social animals

Reference

[1] Bonabeau, E., Dorigo, M., Theraulaz, G.: *Swarm intelligence : From natural to artificial systems*, 1st. ed. Oxford, N. York: Oxford University Press, 1999, ISBN 13: 9780195131598

[2] http://www.dcs.bbk.ac.uk/~sven/aiin06/ainn/html

[3] http://wikipedia.org/wiki/ant_colony_optimization

[4] http://www.swarmintelligence.org/tutorials.php

[5] Wolpert, D. H., Tumer, K., : An *introduction to collective intelligence, in Hanbook of agent technology*, 1st. ed., V.Bradshaw, AAAI/MIT Press, 2001

[6] Honovar, V., Uhr, L.: *Artifical Intelligence and neural networks*, 1st. ed., St. Louis: Academic Press, 1994, ISBN 0123550556