UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Nature-inspired metaheuristics

Jose M Sallan

Introduction to metaheuristics

Outline

1. Nature-inspired metaheuristics
2. Genetic algorithms
3. Swarm intelligence
4. Conclusions

1 Nature-inspired metaheuristics

2 Genetic algorithms

3 Swarm intelligence

4 Conclusions

Nature-inspired metaheuristics

Nature-inspired metaheuristics use strategies to optimize combinatorial problems imitating natural processes:

- **Evolutionary algorithms:** genetic algorithms, evolutionary programming, etc.
- **Swarm intelligence:** particle swarm optimization and ant colonies.

The most used natured-inspired metaheuristic is **genetic algorithms**.

1 Nature-inspired metaheuristics

2 Genetic algorithms

3 Swarm intelligence

4 Conclusions

Elements of a GA

Genetic algorithms (GA) tackle optimization problems mimicking the natural selection process. The elements of a GA are:

1. A **fitness function**.
2. A **population** of solutions.
3. The reproduction operators: **crossover** and **mutation**, compatible with the **encoding** of solutions.
4. A **selection rule** of population elements to breed a new generation.
5. A **stopping condition**: number of iterations, no better solution obtained in k iterations

GA in action

The steps of the GA metaheuristic are:

1. Define a **starting population** of candidate solutions.
2. Generate new candidate solutions using a **crossover operator**.
3. With a probability of mutation, modify the generated candidate solutions with the **mutation operator**.
4. Until the stopping condition is not met, start a new iteration from step 2.

Starting population

The **population** is a set of N candidate solutions. This set is transformed in each iteration, leading to a new generation.

As N increases, possibilities of exploration increase (and also computational cost).

The elements of the **starting population** are chosen randomly, although it can be **seeded** with "good" candidates (e.g., solutions obtained with constructive algorithms).

Crossover and mutation

The elements of the population of each iteration (generation) are obtained through crossover and mutation.

- A **crossover** operator creates a new solution combining chromosomes of two or more solutions. The inputs of crossover are selected with a **selection rule**.
- A **mutation** operator alters the chromosome of a solution, with a probability specified in the algorithm. It helps preserving genetic diversity.

Crossover and mutation operators are contingent upon how solutions are **encoded**.

Crossover operators

A crossover operator for binary encoding.

010101110011 101010101011

010101101011

Crossover operators

A crossover operator for permutative encoding.

$1, 4, 6, 9, 3, 2, 5, 7$ $\qquad$ $1, 9, 6, 3, 8, 7, 2, 5, 4$

$\Downarrow$

$1, 4, 6, 9, 3, 8, 7, 2, 5$

Mutation operators

A mutation operator for binary encoding.

$$010101110011 \quad \Rightarrow \quad 010110010011$$

Mutation operators

Mutation operators for permutative encoding.

Swap:

$$1, 4, 6, 9, 3, 8, 2, 7, 5 \quad \Rightarrow \quad 1, 4, 6, 7, 3, 8, 2, 9, 5$$

Insertion:

$$1, 4, 6, 9, 3, 8, 2, 7, 5 \quad \Rightarrow \quad 1, 4, 6, 7, 9, 3, 8, 2, 5$$

Selection rules

To imitate natural evolutionary processes, only good-fitting population elements are allowed as inputs of crossover. There are several **selection rules** available:

- **Stochastic universal sampling:** elements are selected at random (i.e., no selection).
- **Fitness proportionate selection:** the probability of choosing an element is proportional to its fitness.
- **Tournament selection:** picking the best-fitting individual from a subset of elements selected at random.

Complementarily, a **elitist strategy** can be adopted: maintaining the best(s) solution(s) of the prior generation in the new population.

GA specification

A concrete specification for a GA needs to specifiy multiple parameters:

- The fitness function.
- Population size.
- Possible seeding of starting population.
- Crossover and mutation operators.
- Selection rule for crossover.
- Probability of mutation.
- Possible adoption of elitist strategy.

The tuning of a GA must be carried out using a **experimental design**.

GA variants

Researchers have developed several variants of GAs:

1. **Memetic algorithms:** after each iteration, members of the resulting population can be refined with local search with a level of probability.
2. **Island model genetic algorithms:** GA is performed in multiple subpopulations (islands). Islands are seeded from time to time with candidate solutions from other islands (migration). Adequate for parallel computation.

1 Nature-inspired metaheuristics

2 Genetic algorithms

3 Swarm intelligence

4 Conclusions

Swarm intelligence

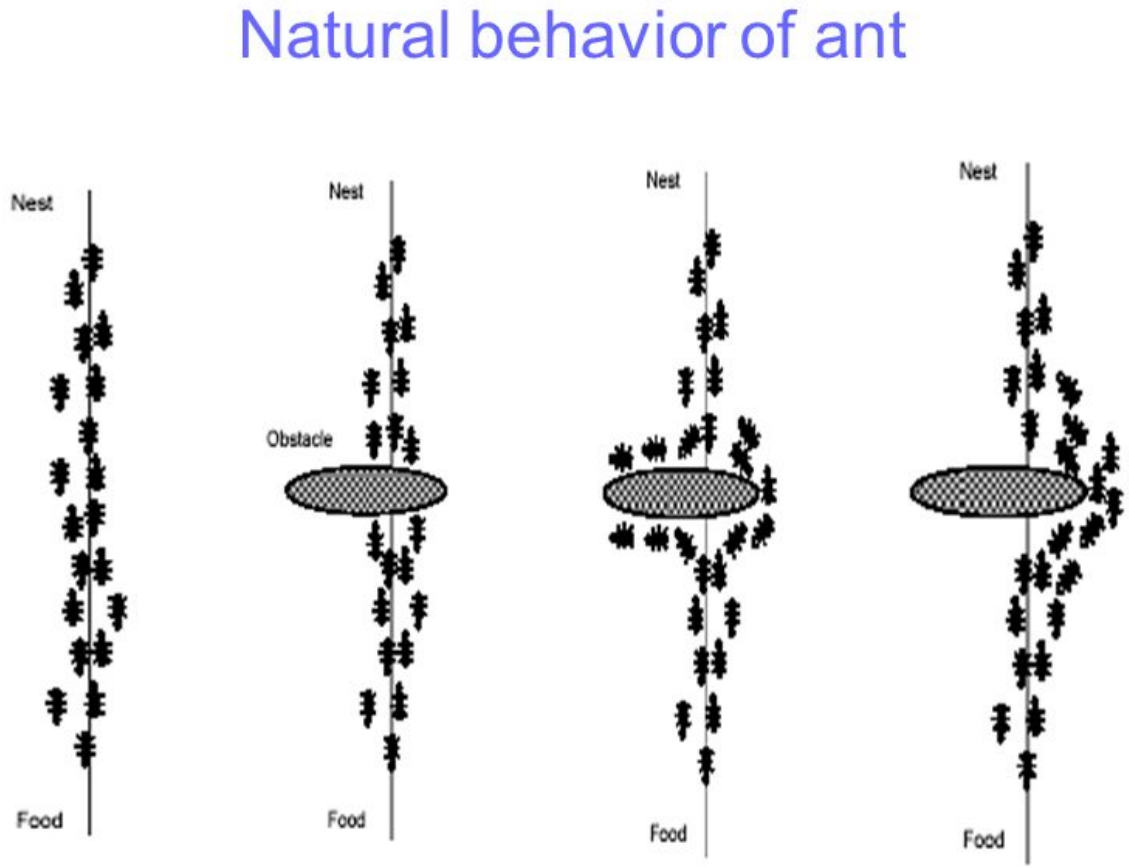
Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. We can use artificial SI systems to define heuristics for optimization problems.

There main SI metaheuristics for optimization are:

- **Ant colony** optimization (ACO)
- **Particle swarm** optimization (PSO)

Ant colony optimization

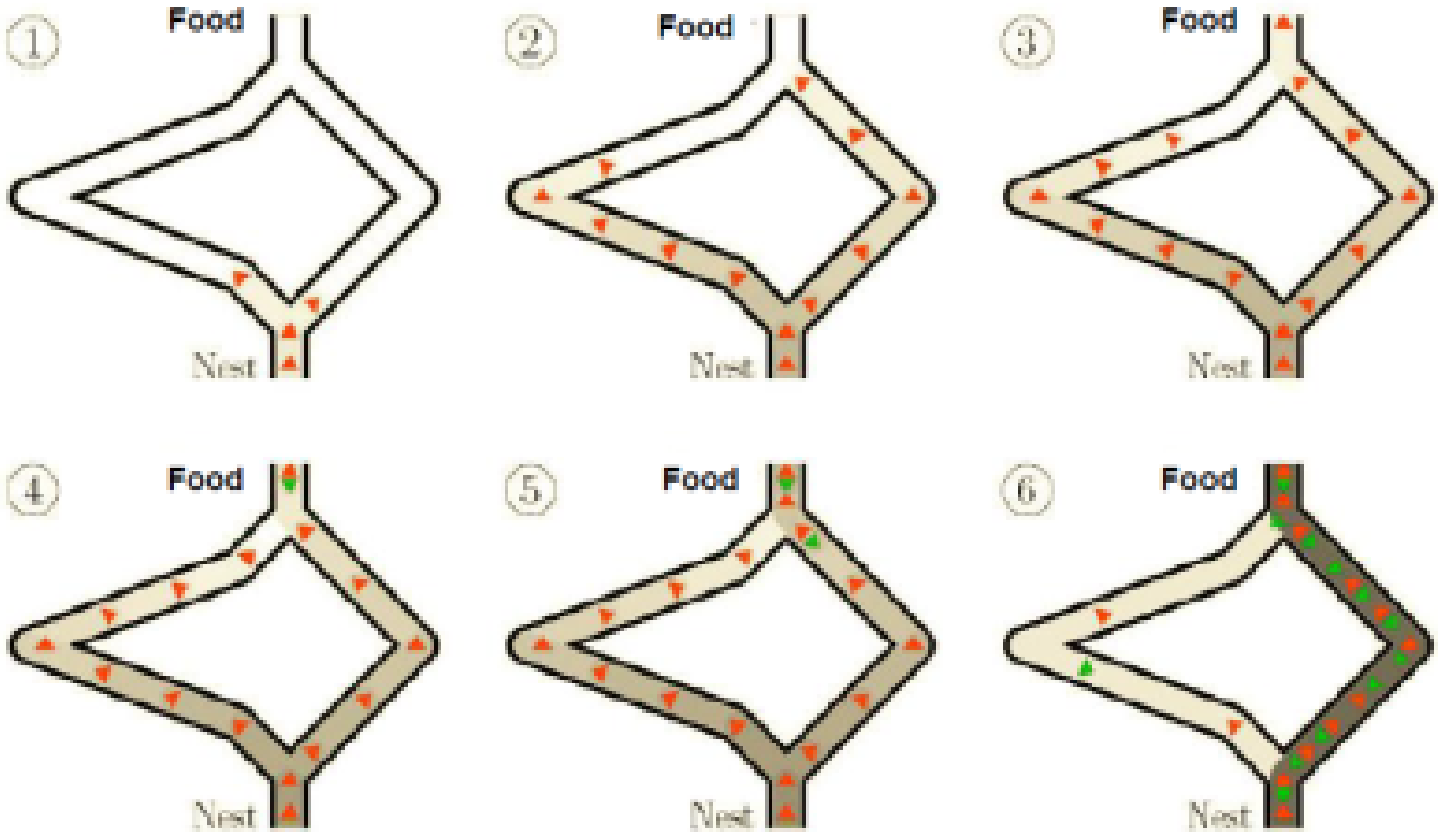
Ant colony optimization is a metaheuristic modeled on the actions of an ant colony.



Ant Algorithms – (P.Koumoutsakos – based on notes L. Gamberdella (www.idsia.ch)

Ant colony optimization

Natural ants use **pheromones** to track shortest paths from food to nest.



Source: https://www.studyblue.com/notes/note/n/ant-colony-optimization/deck/18671530

Ant colony optimization

Ant colony optimization is useful in problems that deal with finding **better paths through graphs**.

Artificial 'ants' (simulation agents) locate optimal solutions by moving through a parameter space representing all possible solutions.

The simulated 'ants' record their positions and the quality of their solutions (similar as natural ants do with pheromones), so that in later simulation iterations more ants locate better solutions.

Particle swarm optimization

Particle swarm optimization (PSO) mimics the behaviour of natural flocks of animals (bees, birds) to solve optimization problems.

PSO deals with problems in which the optimum is a point in a n-dimensional search space.

Candidate solutions, called **particles**, move to a **position** in the solution space with a defined **velocity** in each iteration. This velocity is guided by the best solution found and swarm's current best position.

1 Nature-inspired metaheuristics

2 Genetic algorithms

3 Swarm intelligence

4 Conclusions

Conclusions

Many optimization problems can be solved through nature-inspired metaheuristics.

Genetic algorithms and its variants can be applied to many optimization problems. They often represent a 'brute force' approach to problem solving (high computational cost). GA implementation requires tuning the multiple parameters of the algorithm.

Ant colony optimization can be applied to problems consisting in finding better paths through graphs (e.g., routing problems) or that can be transformed into a problem of this kind (e.g., job shop).
Particle swarm optimization is more suitable for problems with continuous decision variables, although it can be adapted for solving combinatorial problems.