CS4000 Intelligent Systems

Metaheuristics

Dr. Santiago Conant
sconant@tec.mx

Content

- Optimization problems
- Metaheuristics concepts
- Swarm intelligence
  - Ant Systems
  - Bee systems
  - Particle Swarm Optimization
- Conclusions

Optimization problems

It is the problem of finding the **best solution** among all its feasible solutions.

Steps to solve an optimization problem:

1. Build a model of the problem:
   - Identify and express in mathematical terms the objective, variables, and constraints of the problem.
2. Determine the type of problem:
   - Continuous optimization vs discrete optimization
   - Unconstrained optimization vs constrained optimization
   - Optimization without objectives, with one objective or with multiple objectives
   - Deterministic optimization vs. stochastic optimization
3. Choose the solution method and apply it.
4. Analyze and process the results.

Search algorithms

Stochastic algorithms

Global optimization algorithms based on **stochastic local search** of neighborhoods from random initial solutions.

- Random search.
- Stochastic hill climbing.
- Iterated local search.
- Tabu search.
- …

Evolutionary algorithms

Computational methods inspired by the process and the mechanisms of **biological evolution**.

- Genetic algorithms.
- Genetic programming.
- Evolutionary strategies.
- Differential evolution.
- …

Search algorithms

Probabilistic algorithms

Algorithms that explore the problem space based on a **probabilistic model** of the candidate solutions.

- Population-based incremental learning.
- Bayesian optimization algorithms.
- Cross-entropy method.
- …

Physical algorithms

Stochastic optimization algorithms inspired by a **physical process** within nature.

- Simulated Annealing.
- Harmony search.
- Cultural algorithms.
- Memetic algorithms.
- …

Search algorithms

Swarm algorithms

Probabilistic algorithms based on **collective intelligence** emerging through the cooperation of a large number of homogeneous agents.

- Particle swarm optimization.
- Ant system.
- Bees algorithm.
- Bacterial foraging optimization.
- …

Immune algorithms

Computational methods inspired by the process and mechanisms of **biological immune systems**.

- Clonal selection algorithm.
- Negative selection algorithm.
- Artificial immune recognition system.
- Immune network algorithm.
- …

Heuristics for optimization

- Heuristics are strategies that **“guide" the search process** for an optimum.
- The objective is **efficiently explore** the search space in order to find optimal or near-optimal solutions.
- Heuristics are **problem-dependent techniques.**
- They are usually **adapted to one type of problem** and try to make the most of their particularities.
- They are often too greedy so **they are usually trapped in local optima** and fail to find the global optimum solution.

Metaheuristics for optimization

- The metaheuristics are strategies that **“guide" the search process.**
- The objective is **efficiently explore** the search space in order to find optimal or near-optimal solutions.
- The techniques that constitute metaheuristic algorithms range **from simple local search procedures to complex learning processes.**
- Metaheuristic algorithms are **approximate and usually non-deterministic.**
- They can incorporate **mechanisms to avoid being trapped** in confined areas of the search space

Properties of metaheuristics

- The basic concepts of metaheuristics allow an **abstract description**.
- Metaheuristics are **not specific to problems**.
- Metaheuristics can make use of specific domain knowledge in the form of **heuristics that are controlled by the top-level strategy**.
- Today's most advanced metaheuristics use the search experience (**built into some kind of memory**) to guide the search

C. Blum and A.Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison,* ACM Computing Surveys (CSUR), 2003.

Swarm intelligence

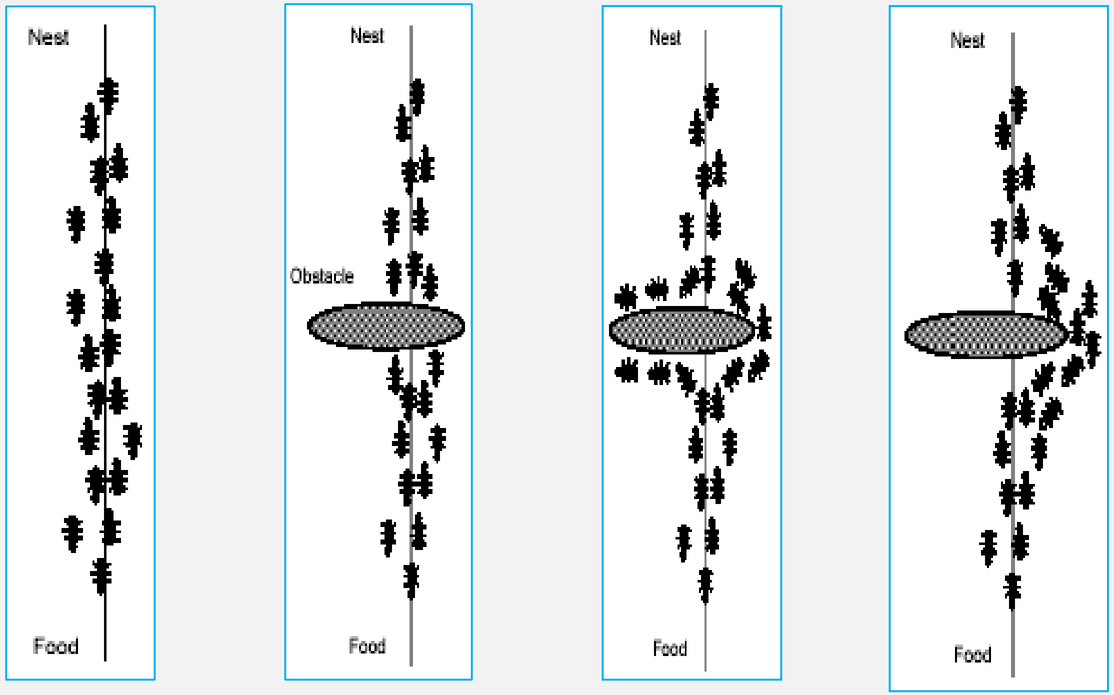
- **Collective system** capable of **performing difficult tasks** in dynamic and varied environments **without any external guidance or control** and **without central coordination.**
- **Achieves a collective performance** that can not normally be achieved by an individual acting alone.
- It is a natural model particularly **suitable for distributed problem solving.**
- Inherent characteristics of Swarm systems
  - Parallelism
  - Stochastic nature
  - Adaptability
  - Use positive feedback

Ant System (AS)

Natural behavior of an ant

Foragingmodes

- Wander mode
- Search mode
- Way back mode
- Attraction mode
- Tracking mode
- Transport mode



Ant Algorithms - (P.Koumoutsakos - based on notes L. Gamberdella (Www.idsia.ch)

Problems resolved with AS

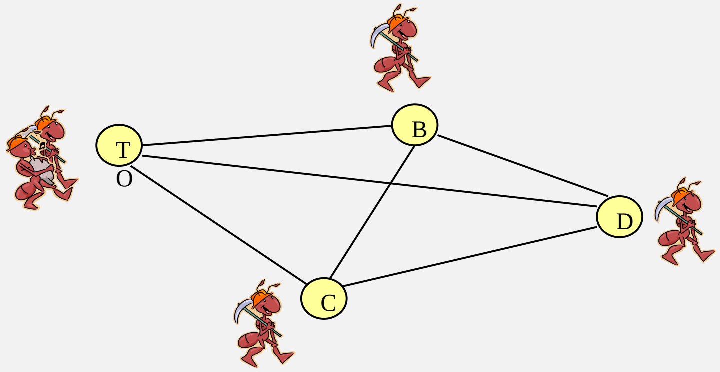
- Traveling salesman
- Quadratic assignment
- Scheduling tasks
- Vehicle routing
- Graph coloring
- Routing networks
- Constraint Satisfaction
- Multiple backpack
- ...

How to implement an AS?

- Ants: Simple agents.
- Move an ant: Choose the next component in the construction of the solution.
- Pheromone: $\Delta\tau_{i,j}^{k}$
- Memory: M_K or $Tabu_K$
- *Next move*: Use probability to move an ant.

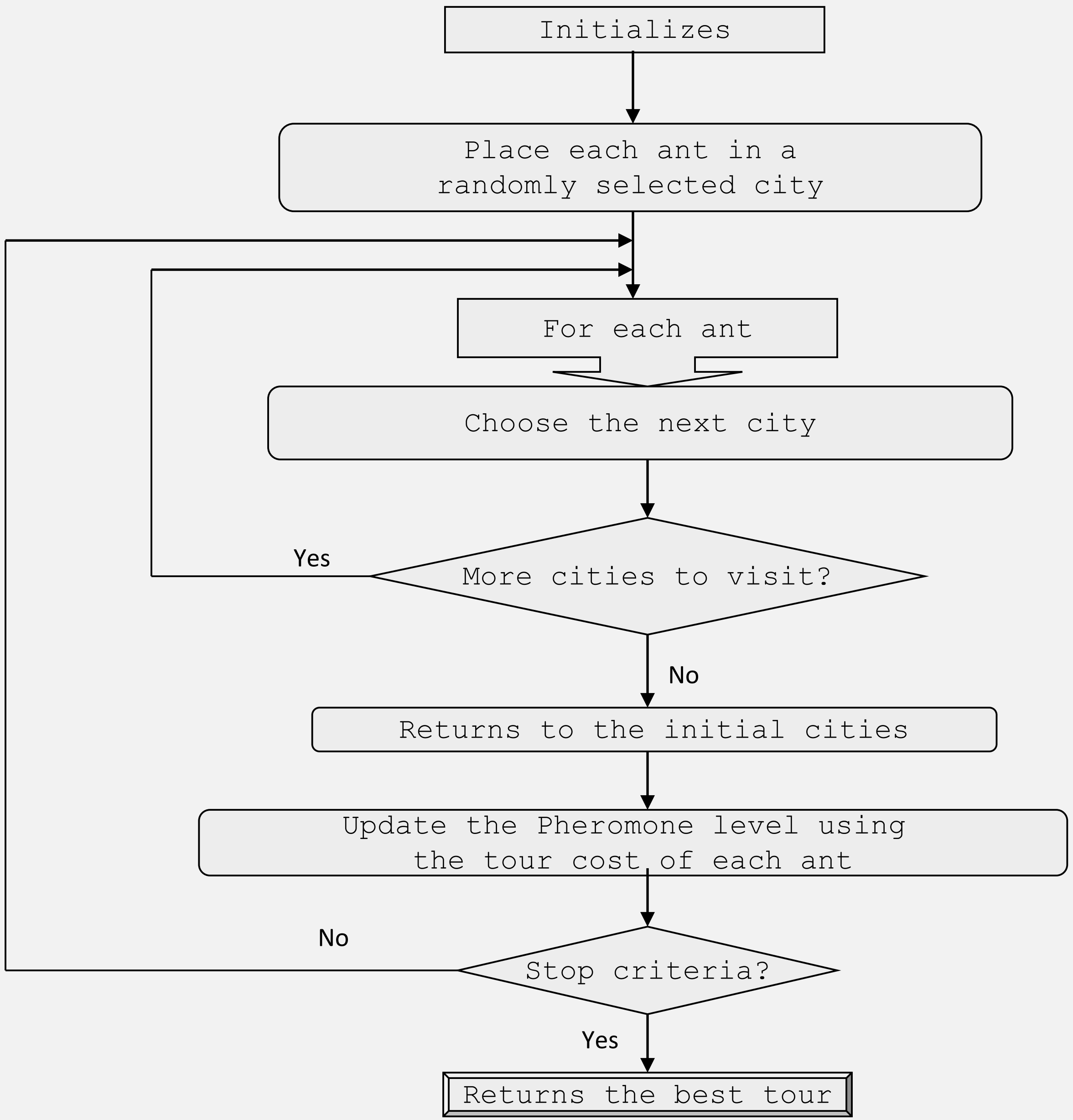
Example: Ant System for the Traveling Salesman Problem (TSP)

- Given a network of N cities, find a minimum total distance tour that visits each city only once.
- Graph (N, E), where:
  - *N* = cities / nodes,
  - E = arcs
  - d_{ij}= the cost of going to the city i to city j (edge weight)



- An ant moves from one city to the next with some transition probability.
- Each arc is assigned a static value returned by a heuristic function η_{ij} based on the cost of the arc.
- Each arc of the graph is augmented with a pheromone trace τ_{ij} deposited by the ants.
- Pheromone is dynamic and is learned at runtime.

AS algorithm for TSP



Computation of transition probability

Transition probability so that ant k goes from city i to j while building its tour.

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{k \in permitido_k} [\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & \text{if } \ j \in \text{permitted}_{\text{k}} \\ 0 & \text{on the contrary} \end{cases}$$

where:

$\tau_{ij}(t)$ = **pheromone trail** : a type of global information

$\eta_{ij} = \dfrac{1}{d_{ij}}$ = **visibility** : heuristic desirability of choosing the city j when in city i.

$\text{permitted}_{\text{k}}$ = **memory** : Set of cities that have not been visited by ant k

α y β = **search parameters**

Pheromone trail in AS

After completing a tour, each ant leaves some pheromones for each arc it has used, depending on how well the ant has performed.

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij}$$

$where:$

ρ is a parameter of pheromone consumption

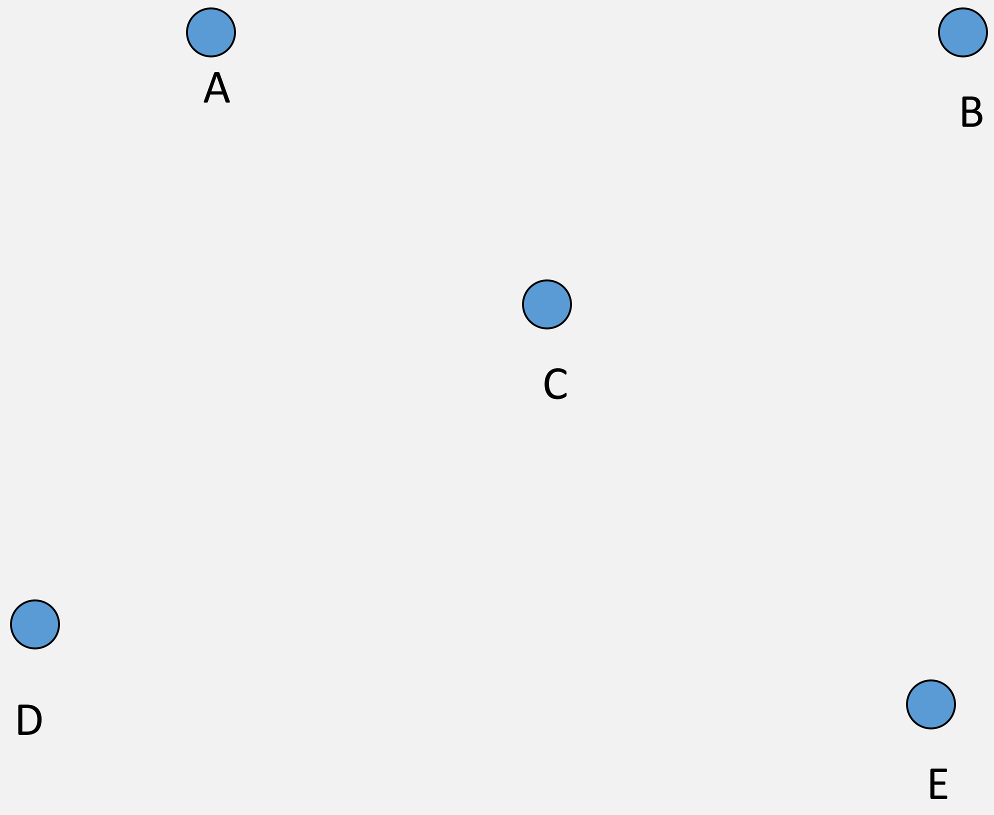
$\Delta\tau_{ij} = \Delta\tau_{ij}^{1} + \Delta\tau_{ij}^{2} + \ldots + \Delta\tau_{ij}^{m}$, for the m ants

$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{Q}{L_k} & si\,(i,\,j) \in \text{tour of ant k} \\ 0 & \text{on the contrary} \end{cases}$$

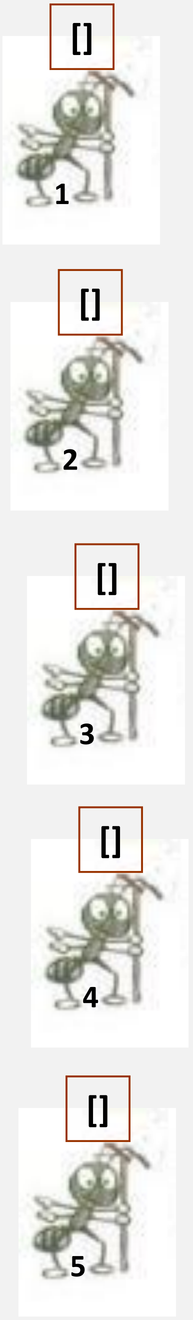
Q is a parameter of contribution of pheromone, and

L_k is the length of the tour of ant k

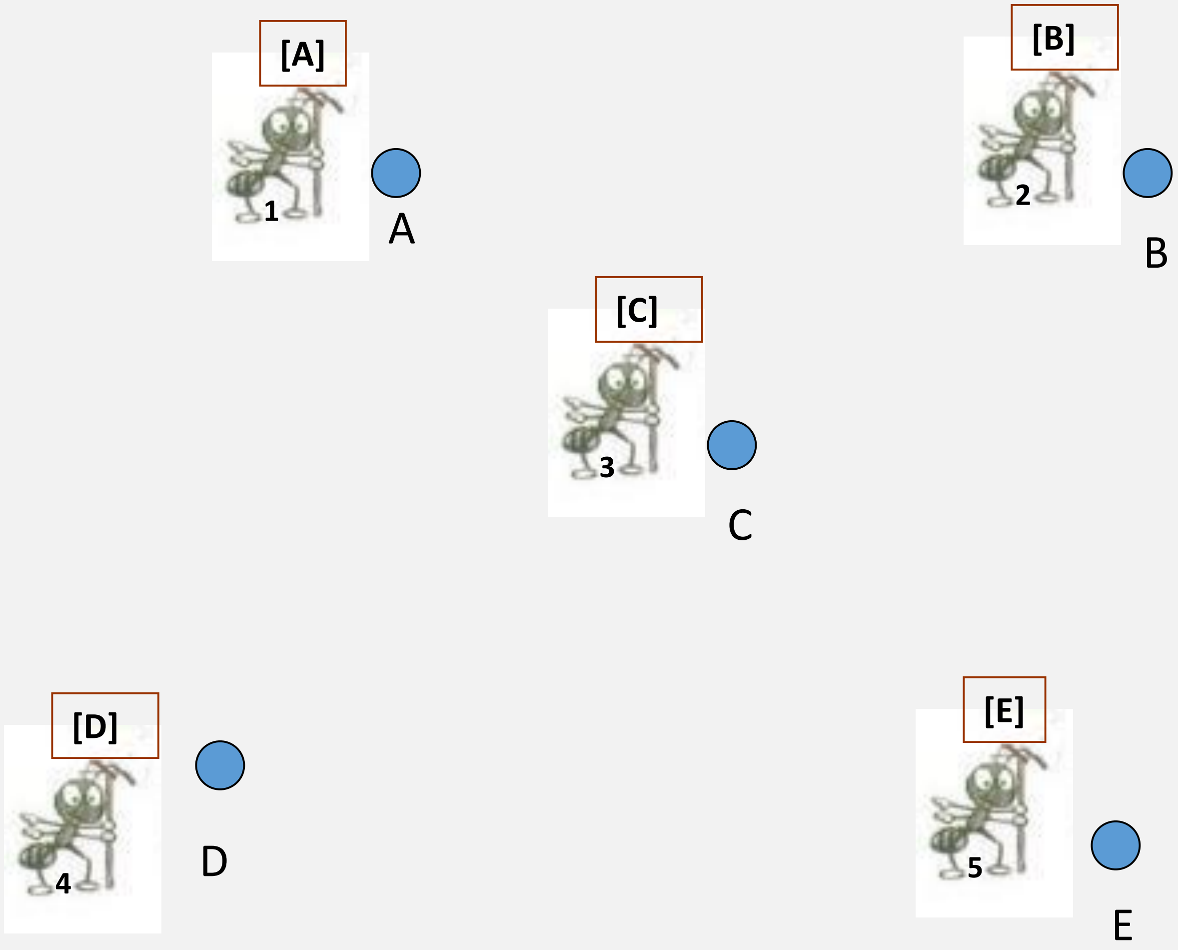
An example of simple TSP



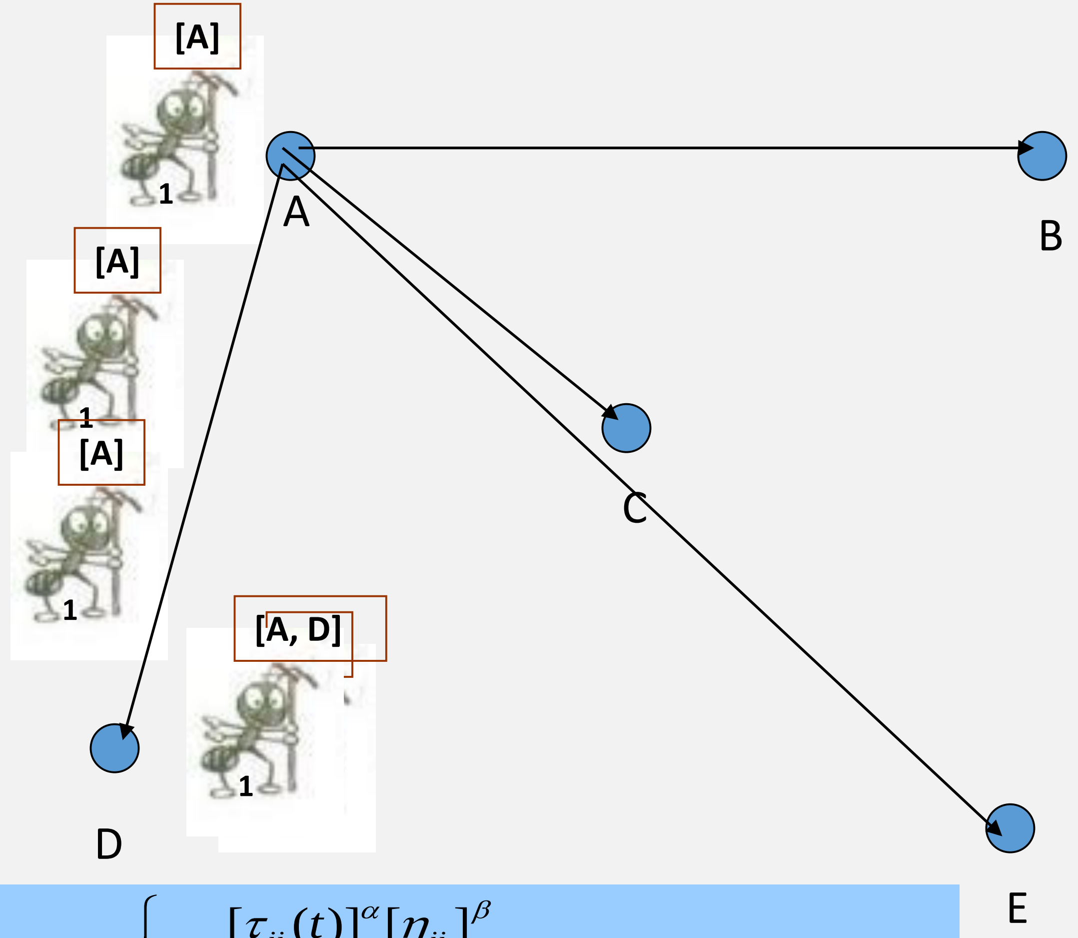
$d_{AB} = 100$ $d_{BC} = 60$ ... $d_{FROM} = 150$



Choose a random city for each ant

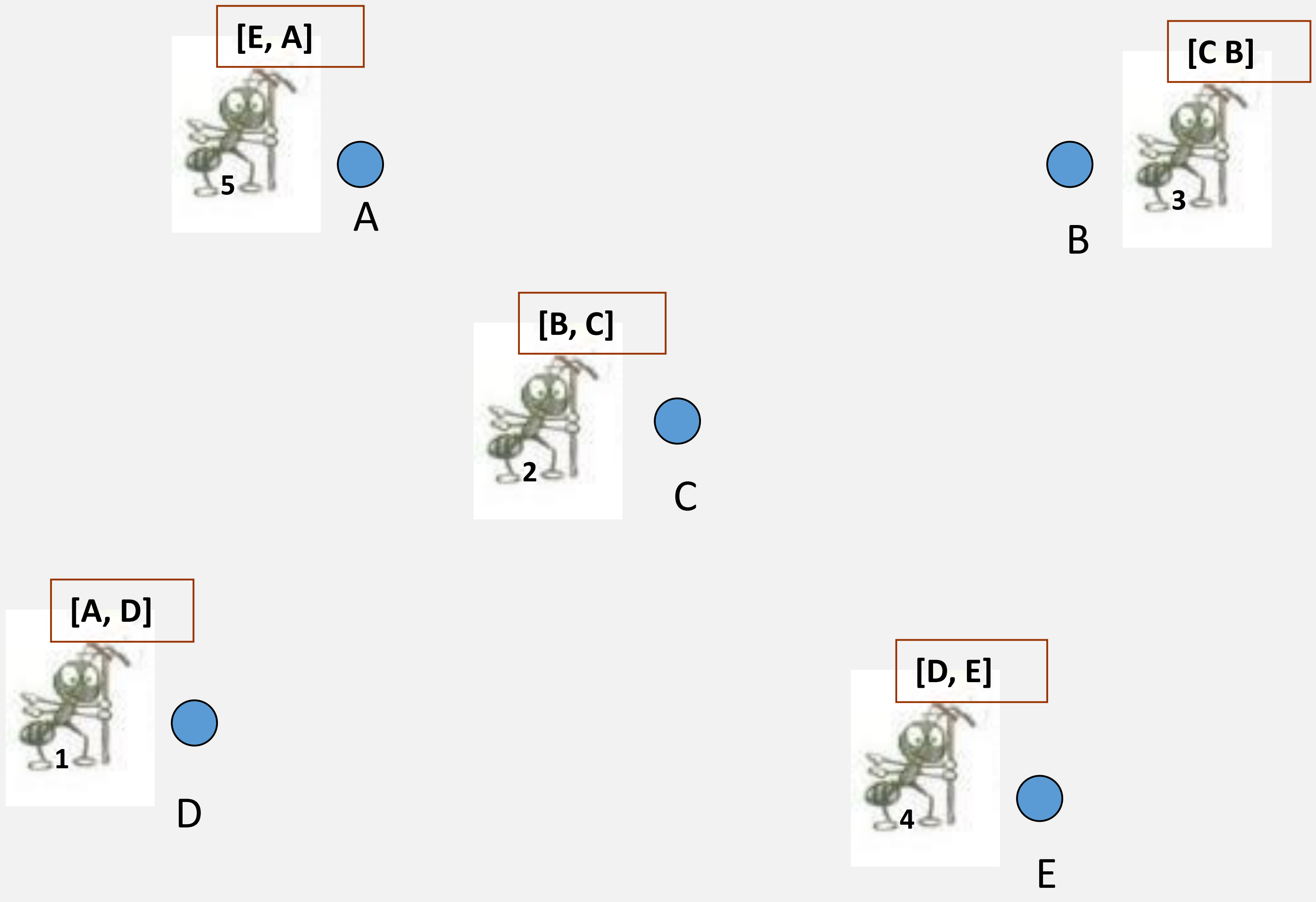


Selects the next city for each ant = **roulette wheel**

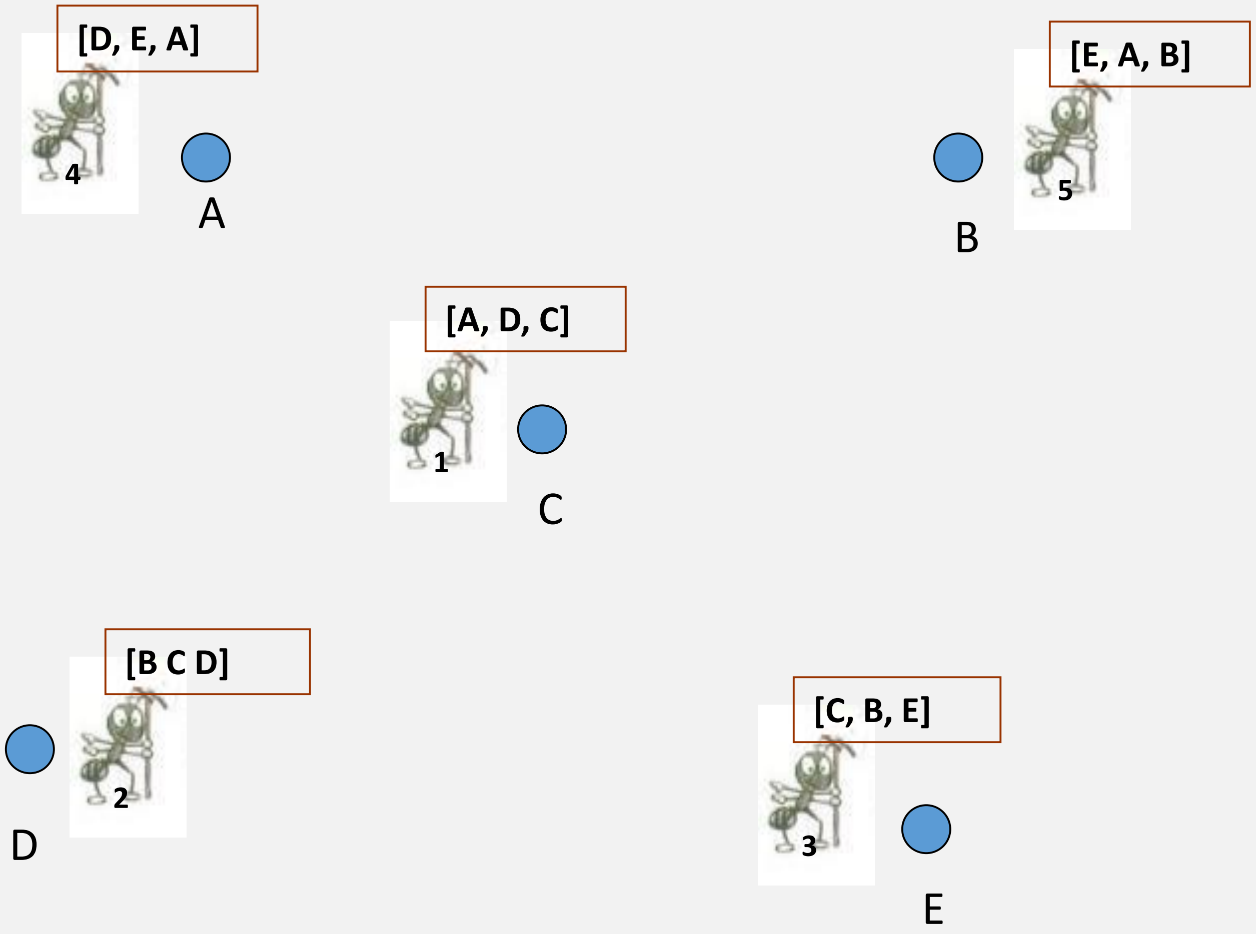


$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{k \in permitted_k} [\tau_{ik}(t)]^{\alpha}[\eta_{ik}]^{\beta}} & \text{si } j \in \text{permitted}_k \\ 0 & \text{on the contrary} \end{cases}$$

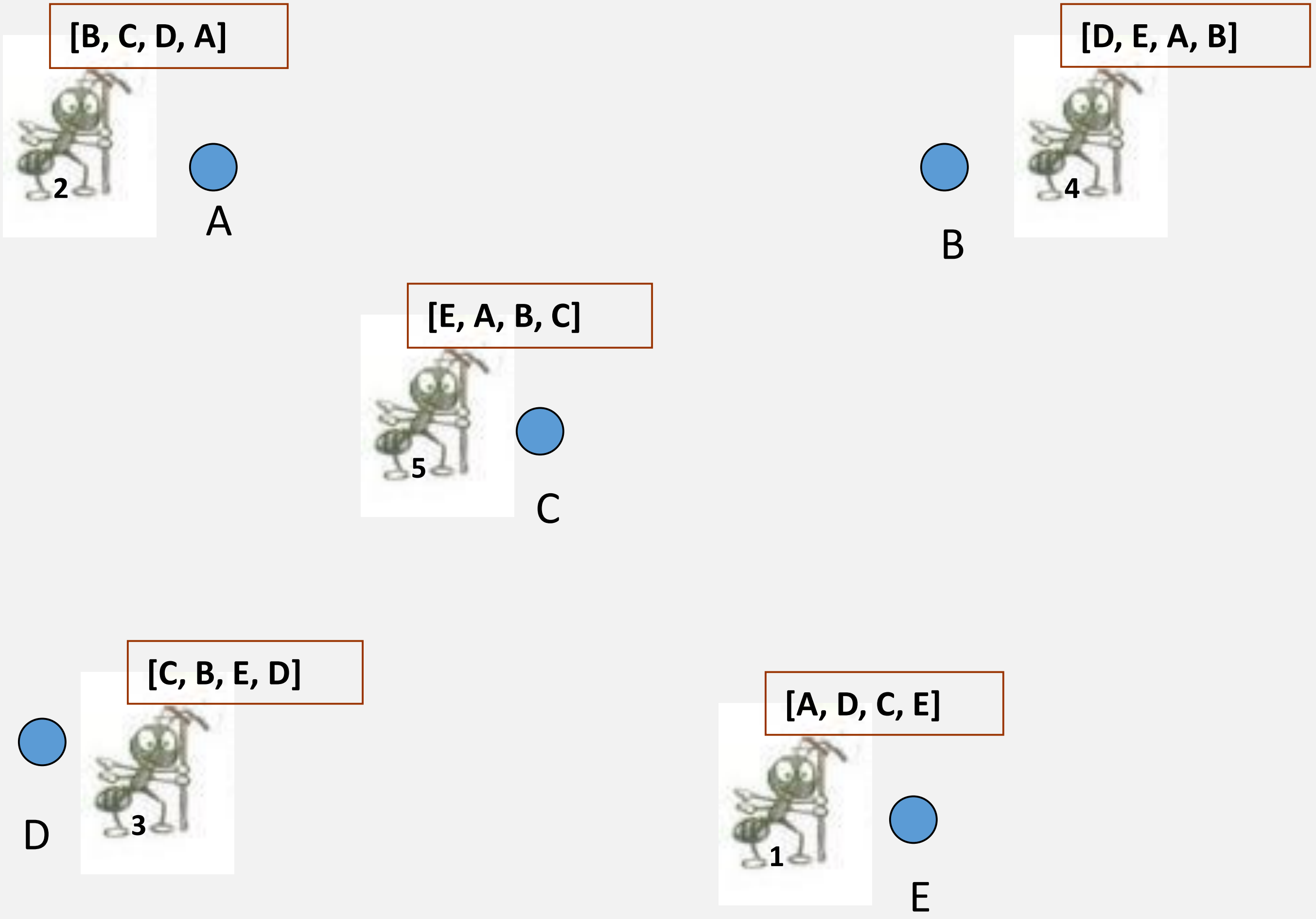
Iteration 2



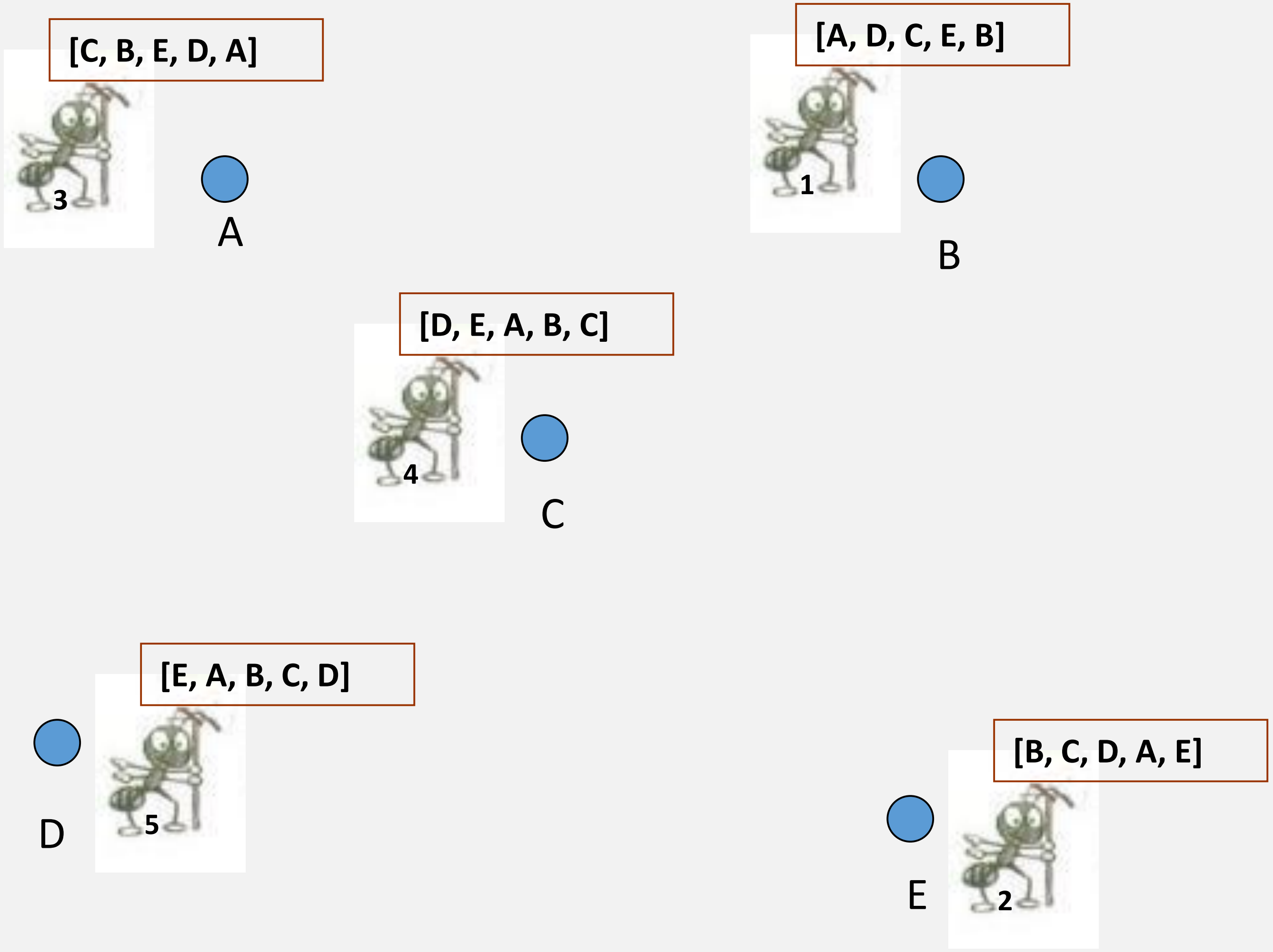
Iteration 3



Iteration 4



Iteration 5



Evaluates the route and updates the pheromone trail

[A, D, C, E, B]

$L_1 = 300$

[B, C, D, A, E]

$L_2 = 450$

[C, B, E, D, A]

$L_3 = 260$

[D, E, A, B, C]

$L_4 = 280$

[E, A, B, C, D]

$L_5 = 420$

$$\Delta\tau_{i,j}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{si } (i,\, j) \in \text{tour} \\ 0 & \text{on the contrary} \end{cases}$$

$$\Delta\tau_{A,B}^{total} = \Delta\tau_{A,B}^{1} + \Delta\tau_{A,B}^{2} + \Delta\tau_{A,B}^{3} + \Delta\tau_{A,B}^{4} + \Delta\tau_{A,B}^{5}$$

Closing of a cycle

- Stop criteria
  - Stagnation
  - Maximum iterations
- Preparation for the next run

End of the first run

Saving the best tour (Sequence and length)

All ants die

New born ants

Conclusions about Ant Systems

- Stochastic search methods.
- Build a solution probabilistically.
- Iteratively add components to partial solutions:
  - Heuristic information
  - Pheromone trail
- A type of reinforcement learning.
- Modify the representation of the problem in each iteration.
- Ants work concurrently and independently.
- Collective interaction via indirect communication leads to good solutions.

Some inherent advantages

- Positive feedback helps in the rapid discovery of good solutions.
- Distributed computing prevents premature convergence.
- Greedy heuristics helps to find an acceptable solution in the early stages of the search process.
- Collective interaction of a population of agents.

Disadvantages of ant systems

- Slower convergence than other heuristics.
- Perform poorly for TSP with more than 75 cities.
- Do not have a centralized processor to guide the AS towards good solutions.

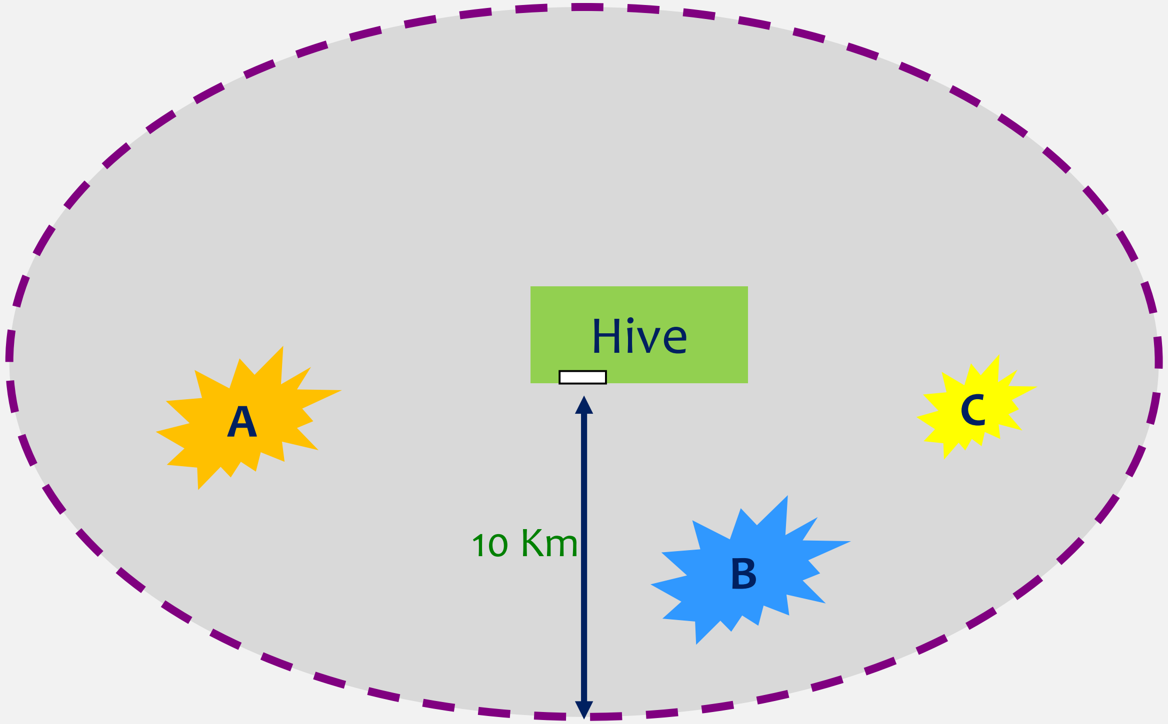
Bees algorithm (BA)

Bees in nature

- Evaluation of food sources:
  - Closeness to beehive
  - Quality
  - Ease of collection
- Employed bees:
  - Associated with a particular food source
  - Transport honey and share their information
- Unemployed bees:
  - Seek sources of food to exploit
  - Scouts
  - Onlookers

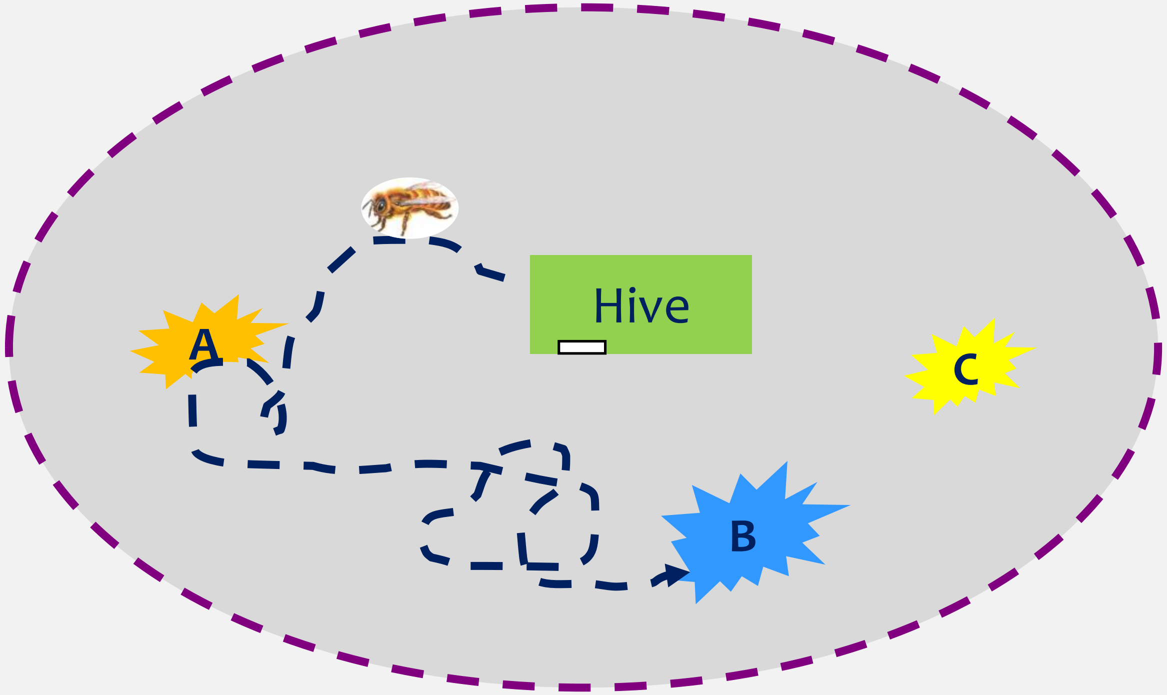
Bees in nature

A colony of honeybees may extend over long distances in multiple directions.



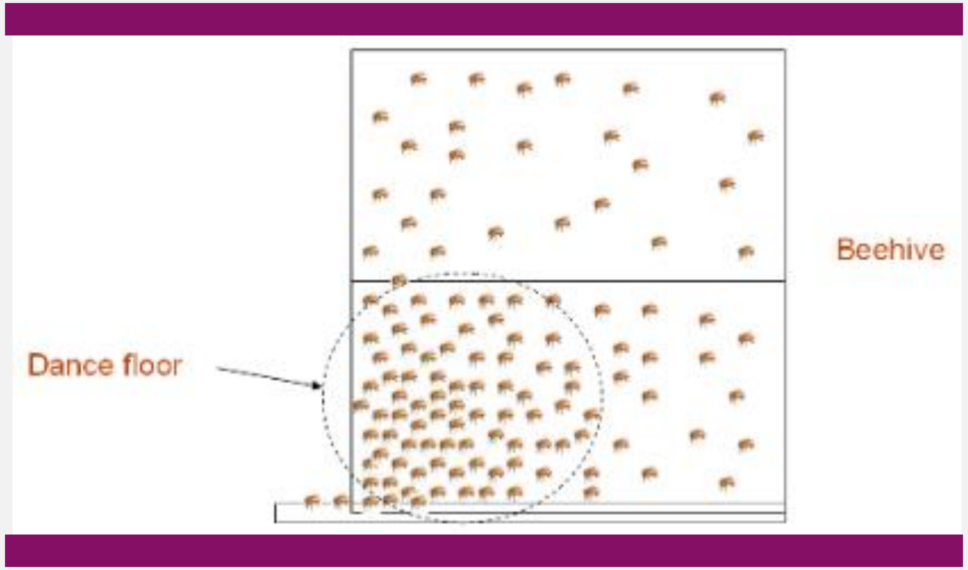
Bees in nature

Scout bees seek randomly from a patch (site) to another.



Bees in nature

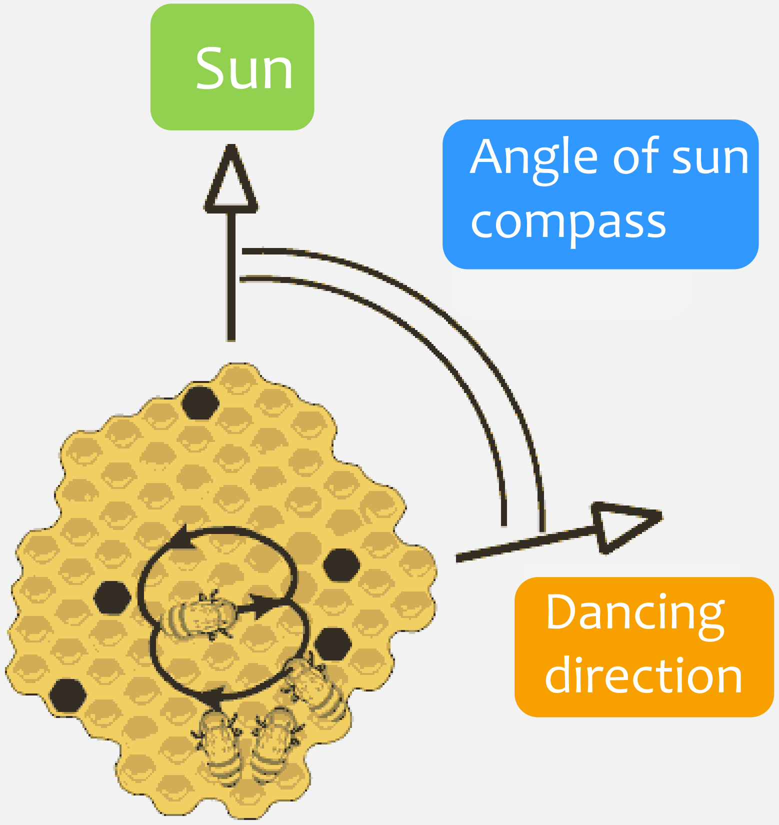
- The **exchange of information between bees** is the most important occurrence in the formation of collective knowledge.
- The communication between bees that is related to the quality of food sources occurs in the **dance zone.**
- The dance is known as the "**Wiggle dance**".
- Bees evaluate different patches according to:
  - The quality of food
  - The amount of energy use



Bees in nature

Bees communicate through wiggle dance the following information:

- The direction of the flower patches (angle between the sun and the patch).
- The distance from the hive (duration of the dance).
- The assessment of the quality (fitness) (frequency of the dance).

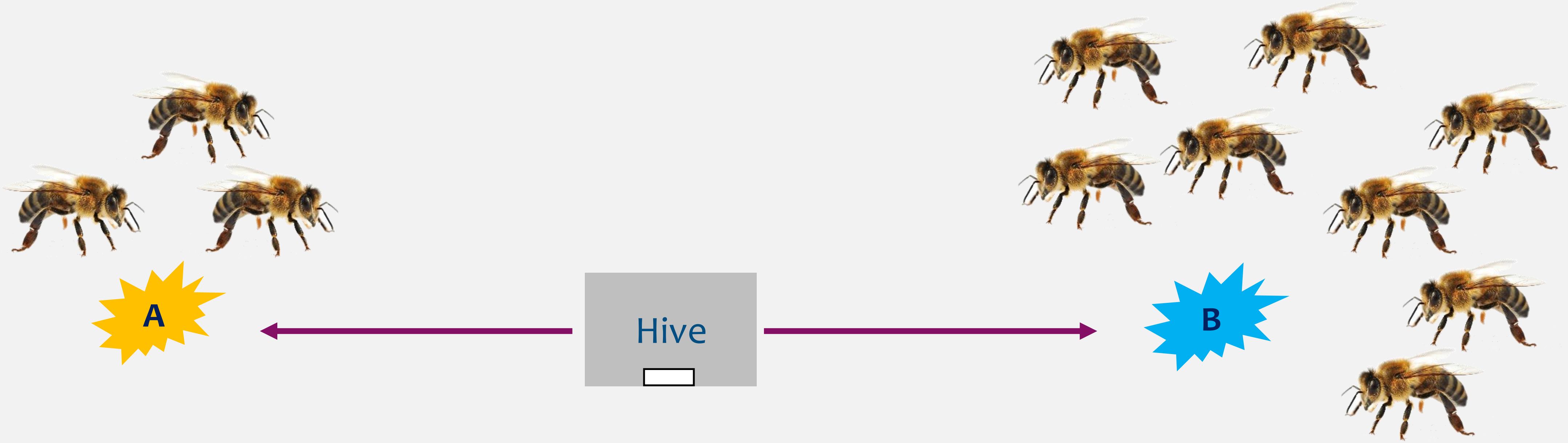


Bees in nature

- The information helps the colony to send bees accurately.
- Onlooker bees follow the dancing bee to the patch to get food quickly and efficiently.
- The same patch will be announced again at the wiggle dance when returning to the hive if this still remains a good enough food source (depending on the level of food) and more bees will be recruited for that source.
- More bees visit patches of flowers with abundant amounts of nectar or pollen.

Bees in nature

Then according to their ability, some patches can be visited by more bees or may be abandoned.



Basic bees algorithm (BA)

The **bees algorithm** is an optimization algorithm inspired in the honey bees' natural feeding behavior to find an optimal solution and works as follows:

Initialize a population of **n** scout bees with random solutions.

1. Assess the fitnesses of the population.
2. Repeat until a stop criterion is reached:
    1. Select **m** sites for neighborhood search.
    2. Recruit bees for selected sites (more bees for the best **e** sites) and assess their fitnesses.
    3. Select the bee with the best fitness for each patch.
    4. Assign the remaining (**n-m**) bees to randomly search and evaluate their fitnesses.

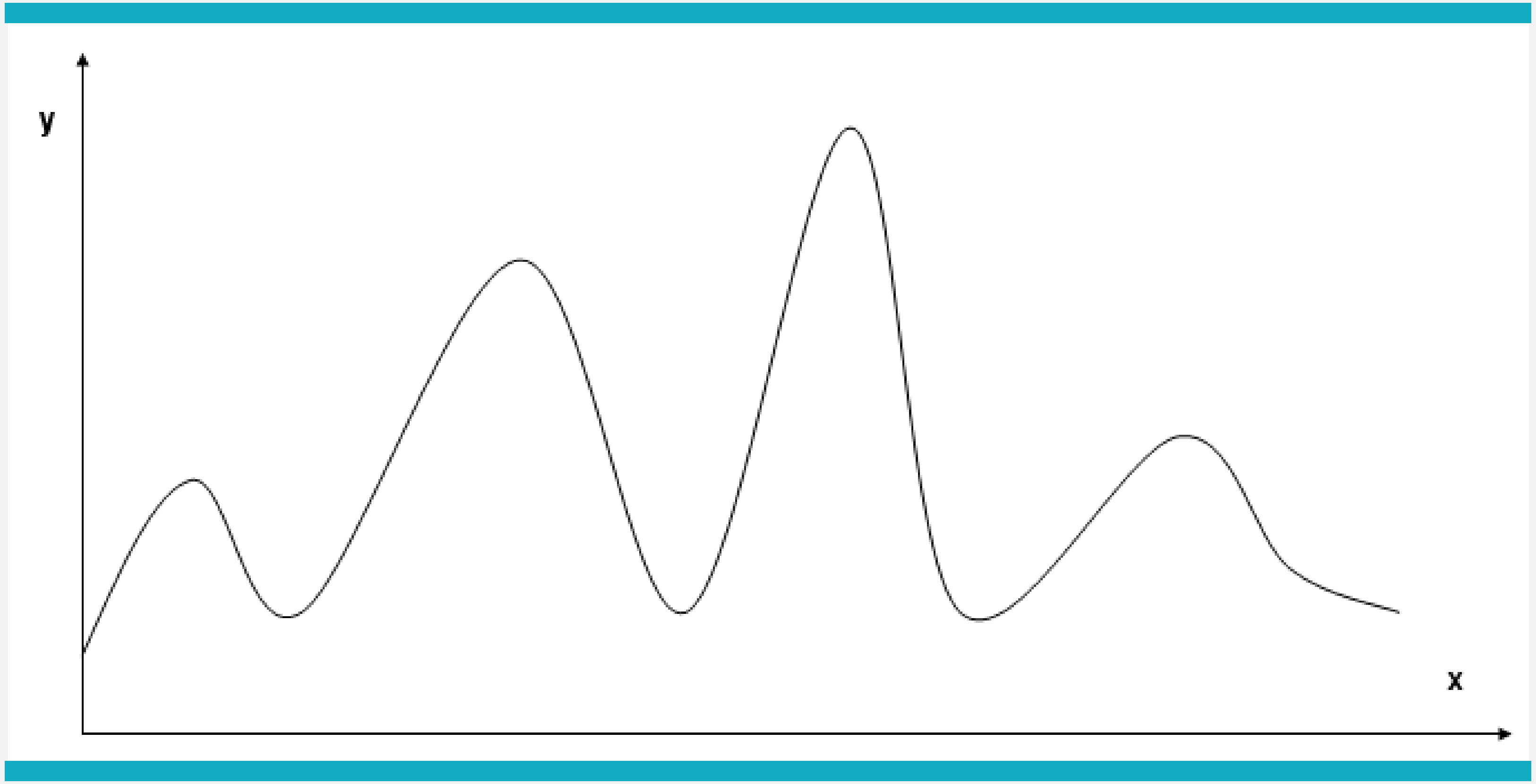
Parameters for the BA algorithm

The BA algorithm requires to set the following parameters:

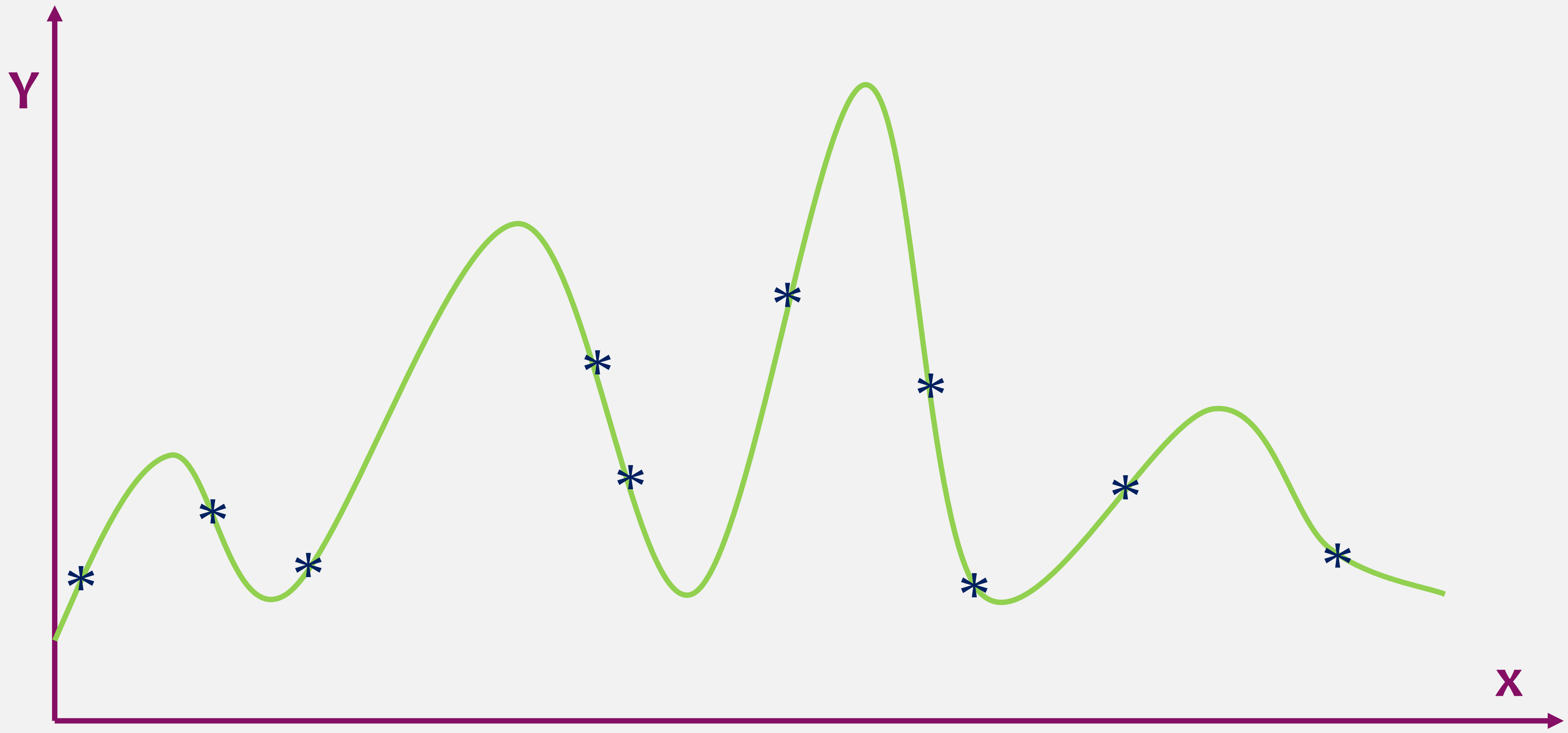
- Population size (total of scout bees) (**n**)
- Number of selected sites (**m**) of the **n** visited
- Number of the best sites (**e**) of the **m** selected
- Number of bees recruited for **e** best sites (**nos** or **n2**)
- Number of bees recruited for other (**m-e**) selected sites (**nsp** or **n1**)
- Initial size of patches (**ngh**) including site, neighborhood, and unemployment criteria
- Maximum number of algorithm iterations (**imax**)

A simple example

- The following figure shows a mathematical function to which we want to find its global maximum:

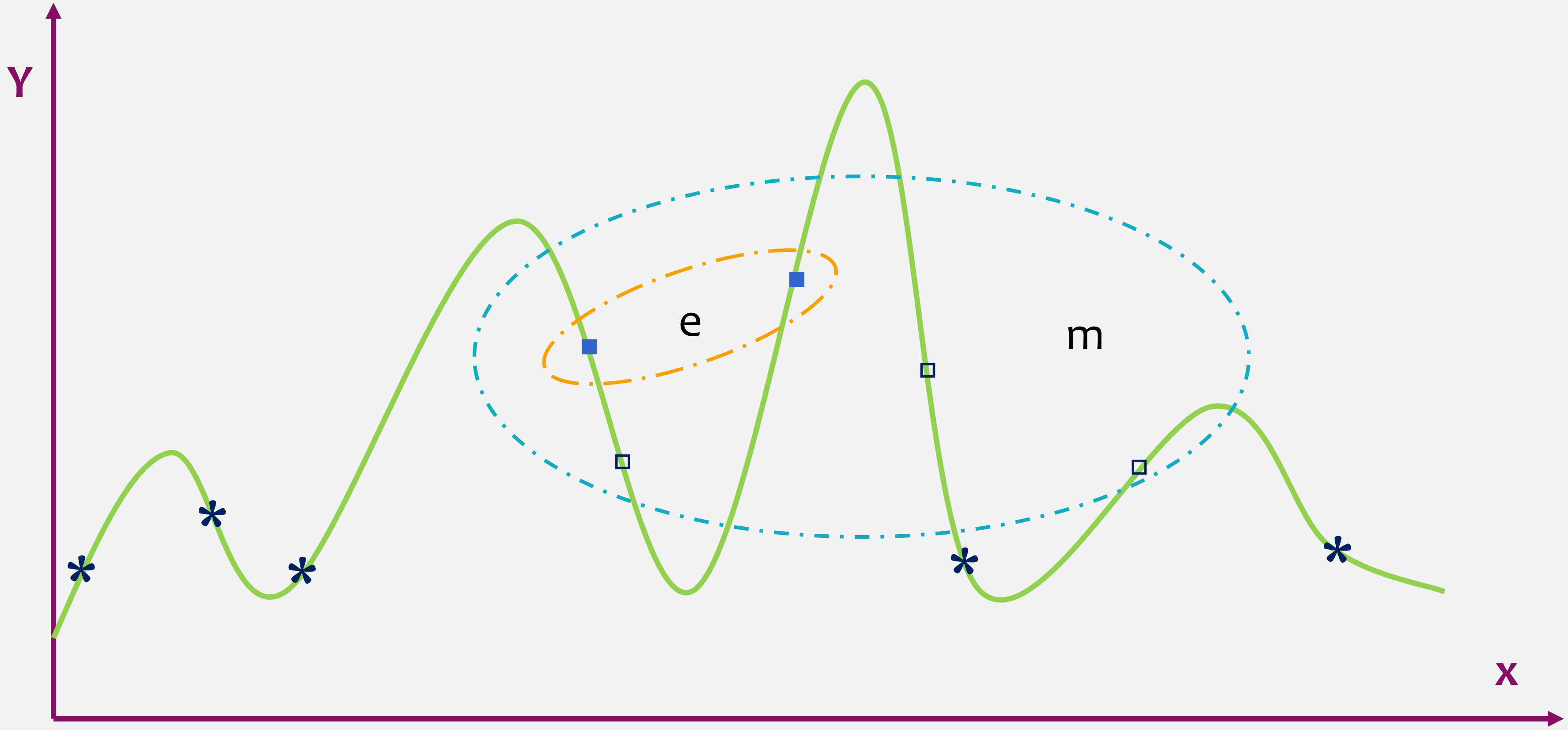


A simple example



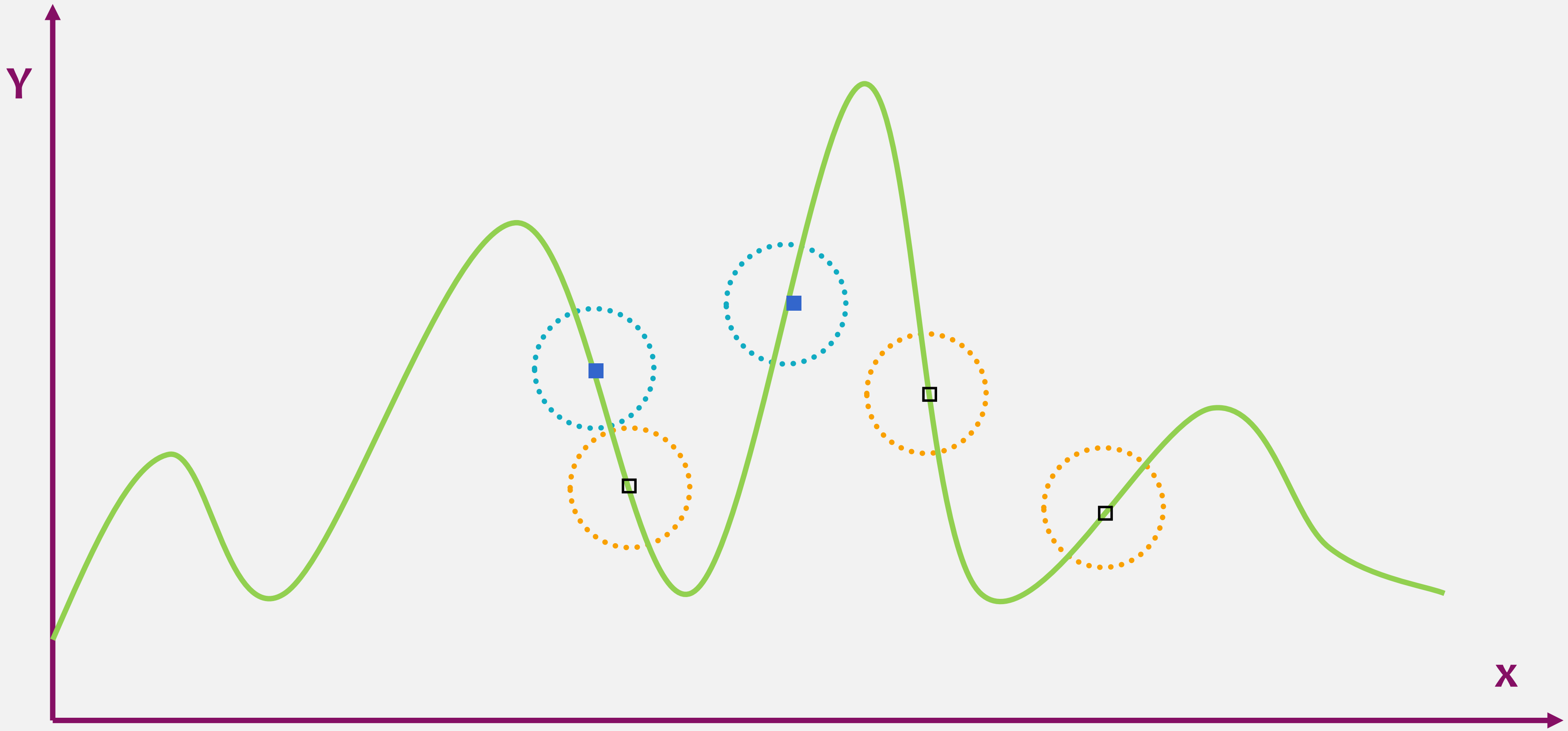
1. Initializes a population of **n = 10** employed bees with a random search and assesses their fitnesses.

A simple example



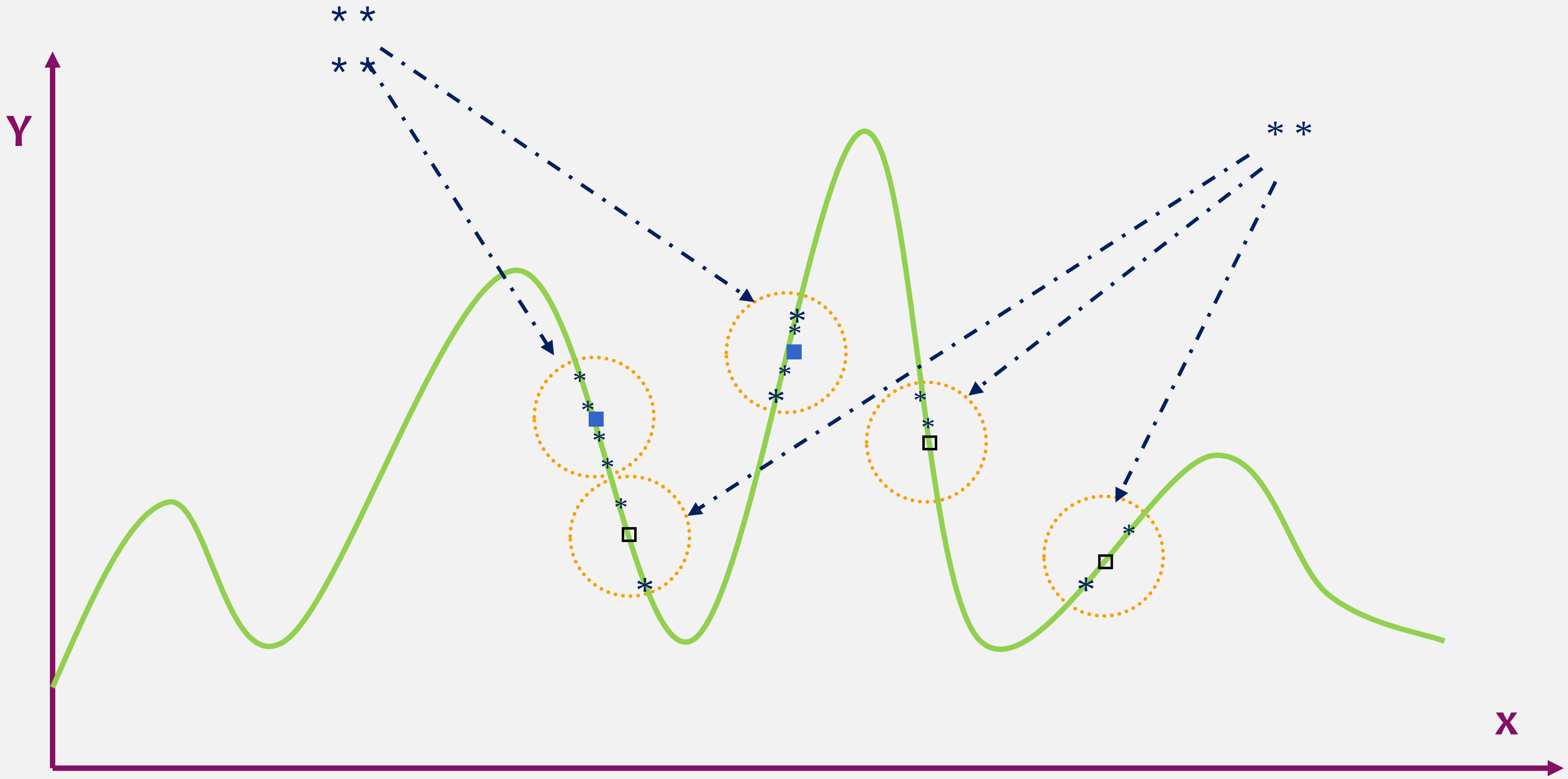
2. Select the **m = 5** best sites for a neighborhood search: **e = 2** elite bees "▪" and other **m-e = 3** selected bees "▫".

A simple example



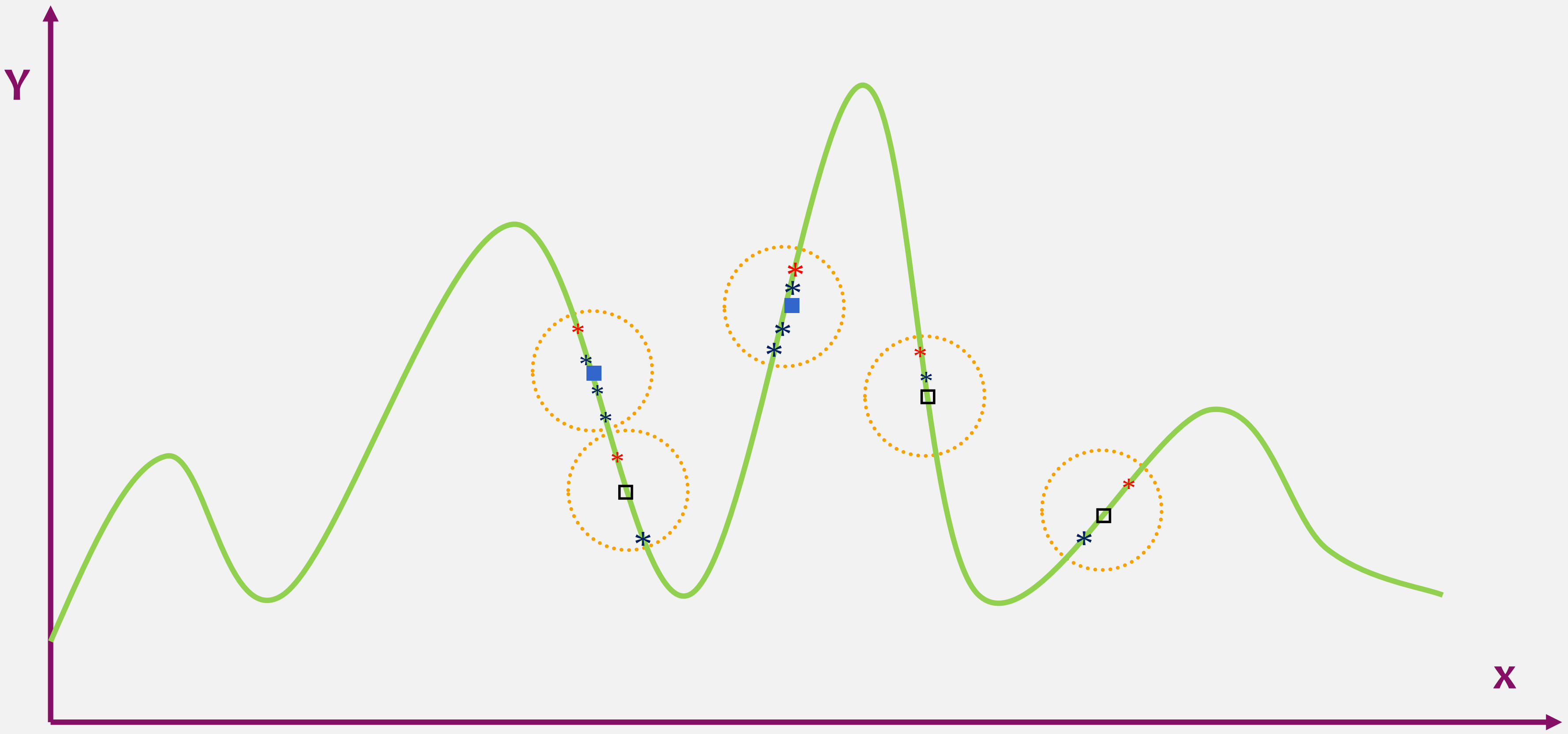
3. Determine the neighborhood size (patch size = **ngh**).

A simple example



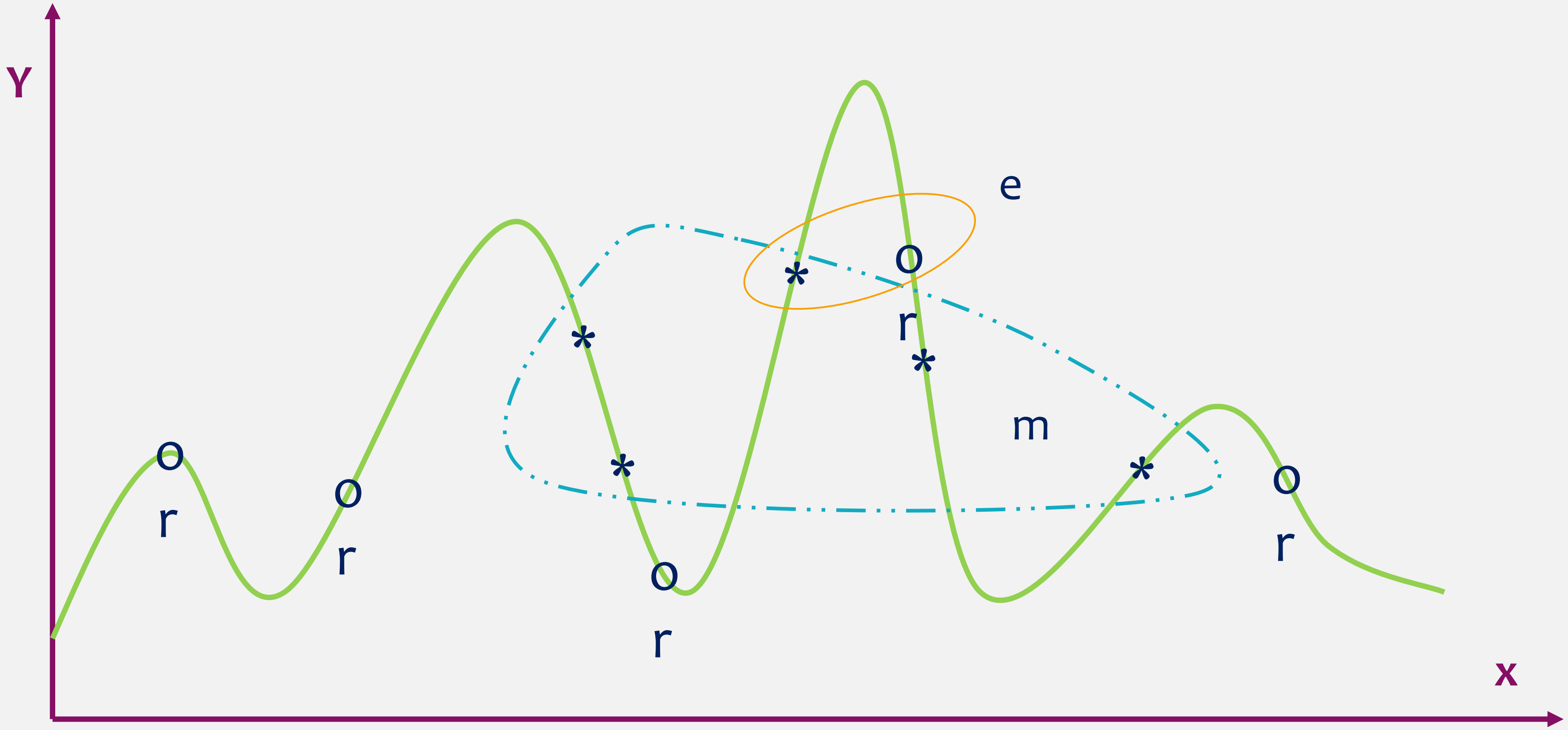
4. Recruits onlooker bees for the selected sites (more bees for **e** elite sites (**n2 = 4**) and fewer bees for the other **m-e** sites (**n1 = 2**).

A simple example



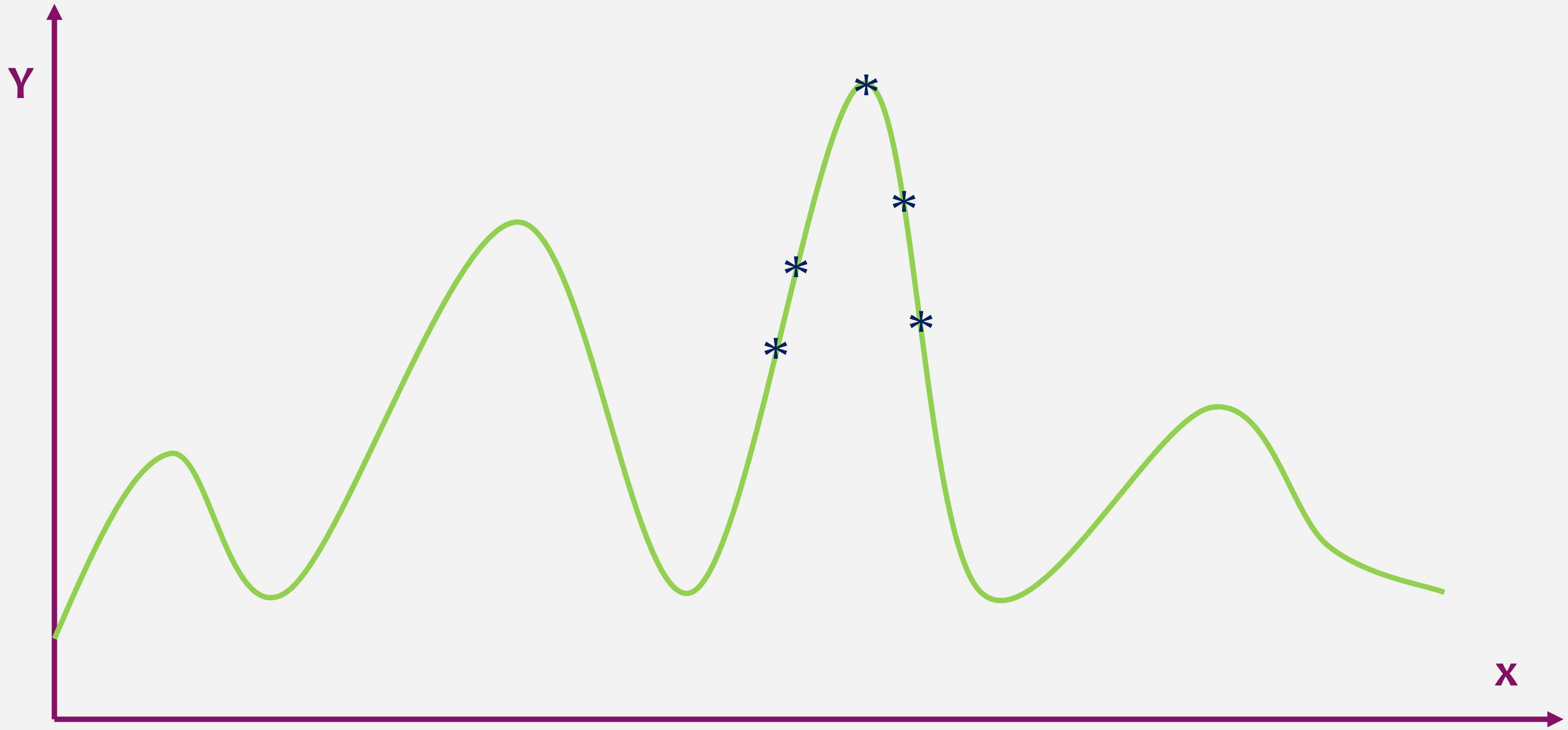
5. Select the bee with the greatest fitness for each site.

A simple example



6. Assign the **n-m = 5** remaining bees as scouts ("o“) for random search and repeat the process until the stop criterion is reached.

A simple example



At the end: It is expected to find the global maximum.

BA applications

- Optimization functions.
- Solving the traveling salesman problem.
- Classifier training as neural networks.
- Electronic design.
- Mechanical design.
- Optimization of digital filters.
- Design for fuzzy control.
- Data grouping (solving the local optimum for the K-Means algorithm).
- Robot control.

Advantages and disadvantages of BA

- Advantages of BA
  - Very efficient to find optimal solutions.
  - Overcomes the problem of local optima.

- Disadvantages of BA
  - Uses a number of adjustable parameters.
  - The parameter values may be established by conducting a small number of experiments.

Particle Swarm Optimization (PSO)

The basic idea

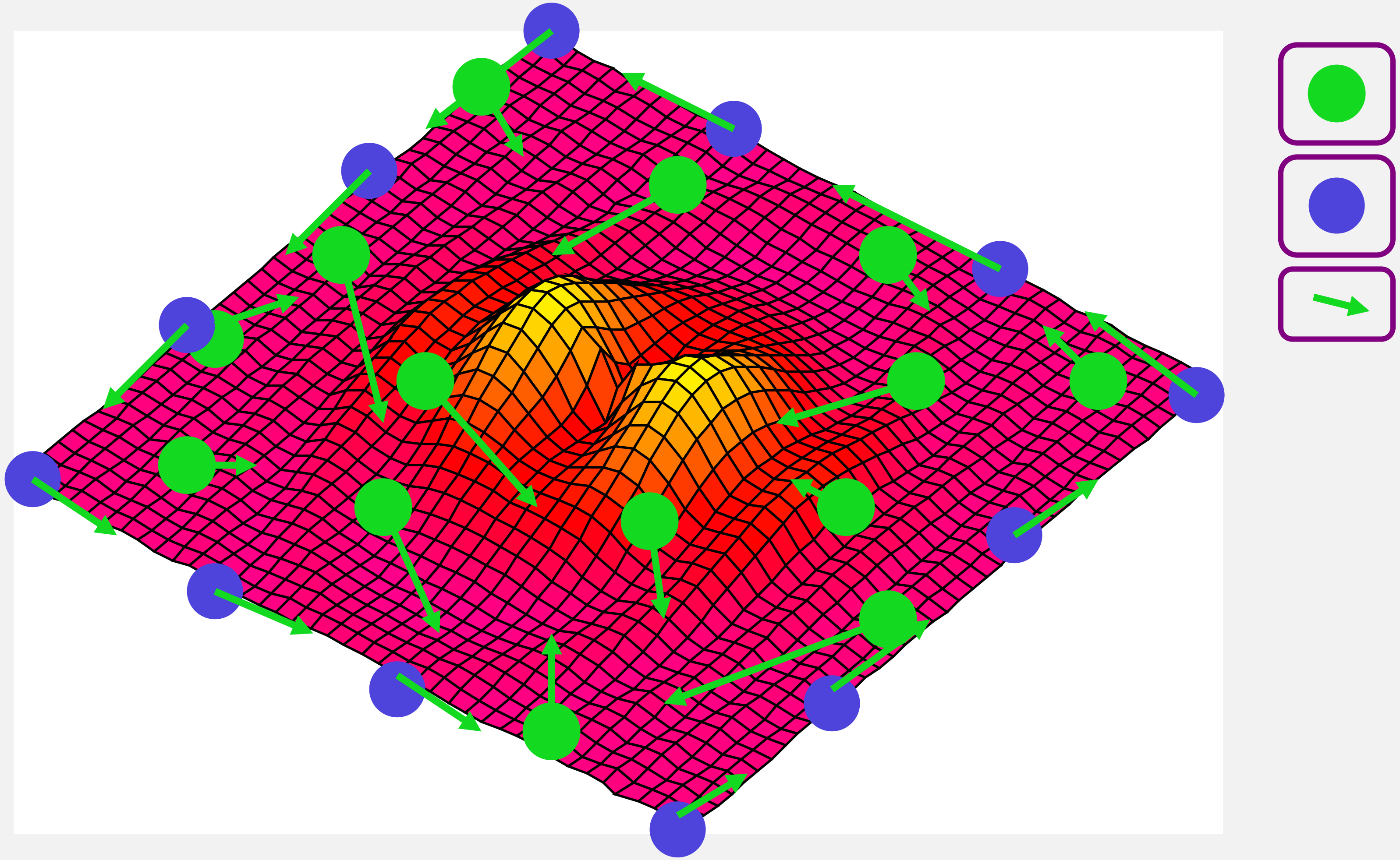
- Each particle is searching for the optimum.
- Each particle is moving and therefore has a velocity.
- Each particle remembers the position it had when it got its best result so far (personal best).
- But this is not good enough by itself; the particles need help deciding where to look.

The basic idea II

- **The particles in the swarm cooperate.** They exchange information about what they have discovered in the places they have visited.

- **Cooperation is very simple.** In the basic PSO it is like this:
  - One particle has a neighborhood associated with it.
  - A particle knows the fitnesses of its neighbors, and uses the position of the one with the greatest fitness.
  - This position is simply used to adjust the velocity of the particle.

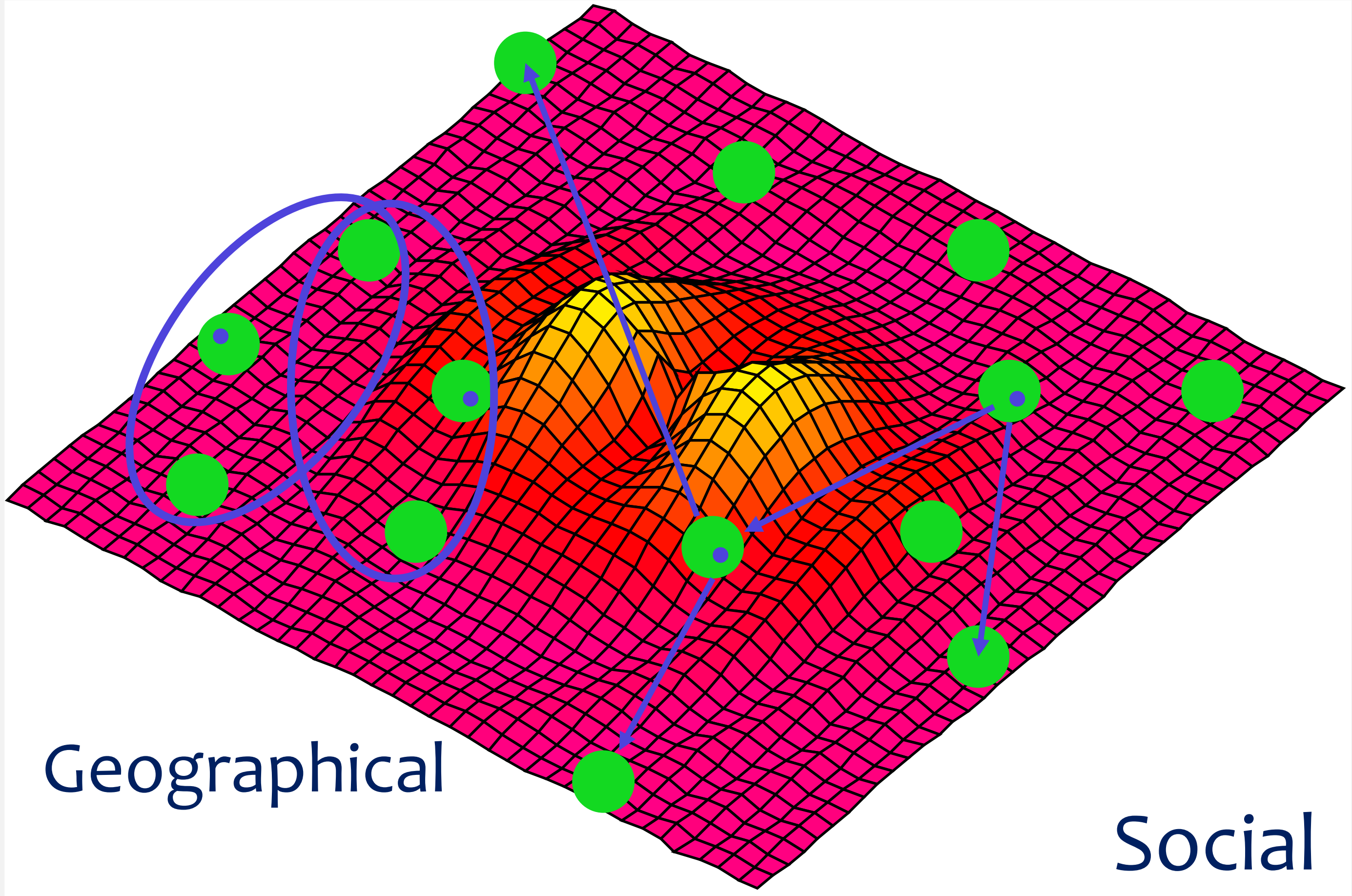
Initialization: Positions and velocities



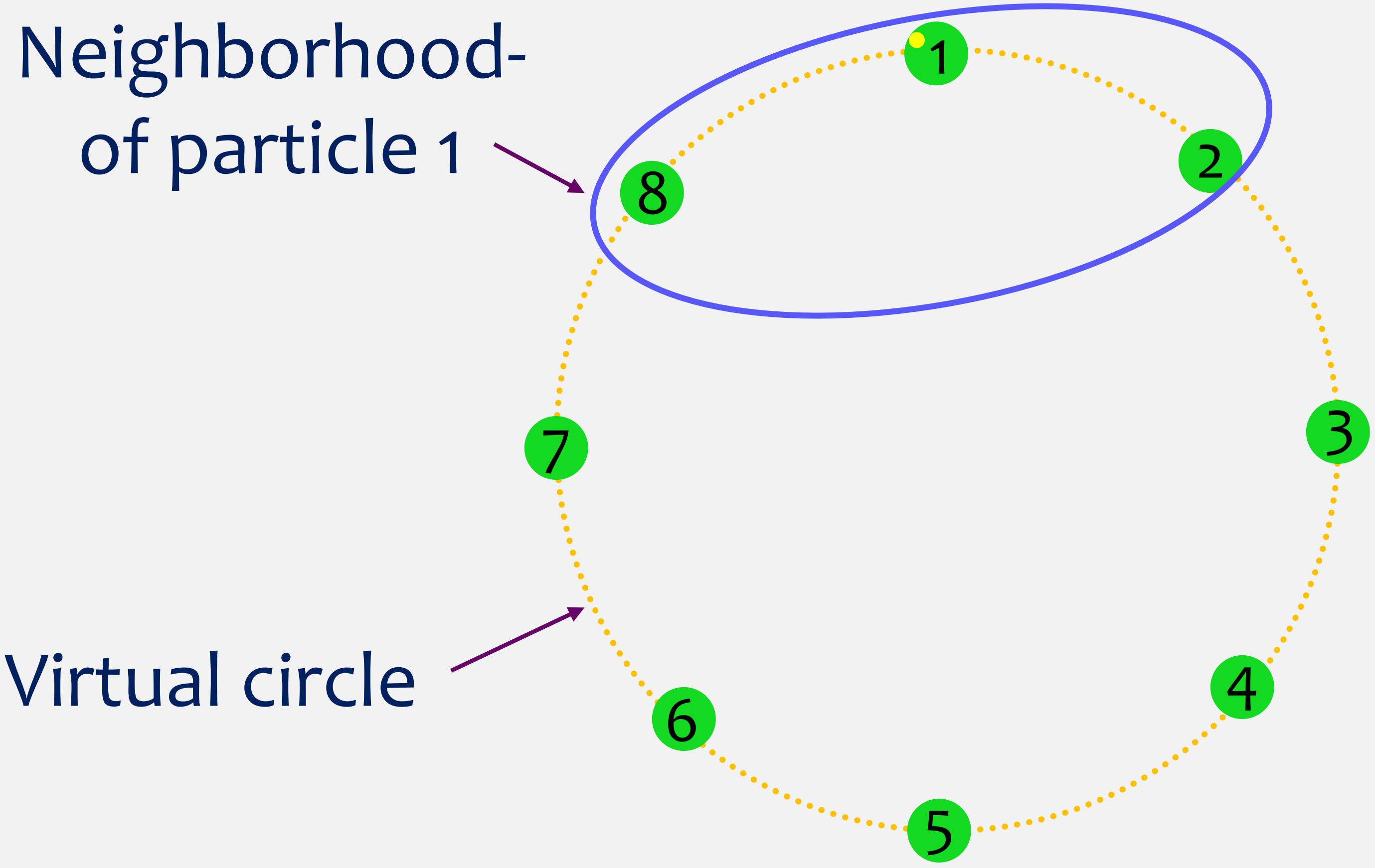
How a particle moves?

- At each time step, a particle must move to a new position. It does this by adjusting its velocity.
  - The adjustment is essentially this:
    - The current velocity PLUS.
    - A portion randomly weighted in the direction of its personal best PLUS.
    - A portion randomly weighted in the direction of its best neighbor.
- Having calculated the new velocity, its position is simply its previous position plus its new velocity.

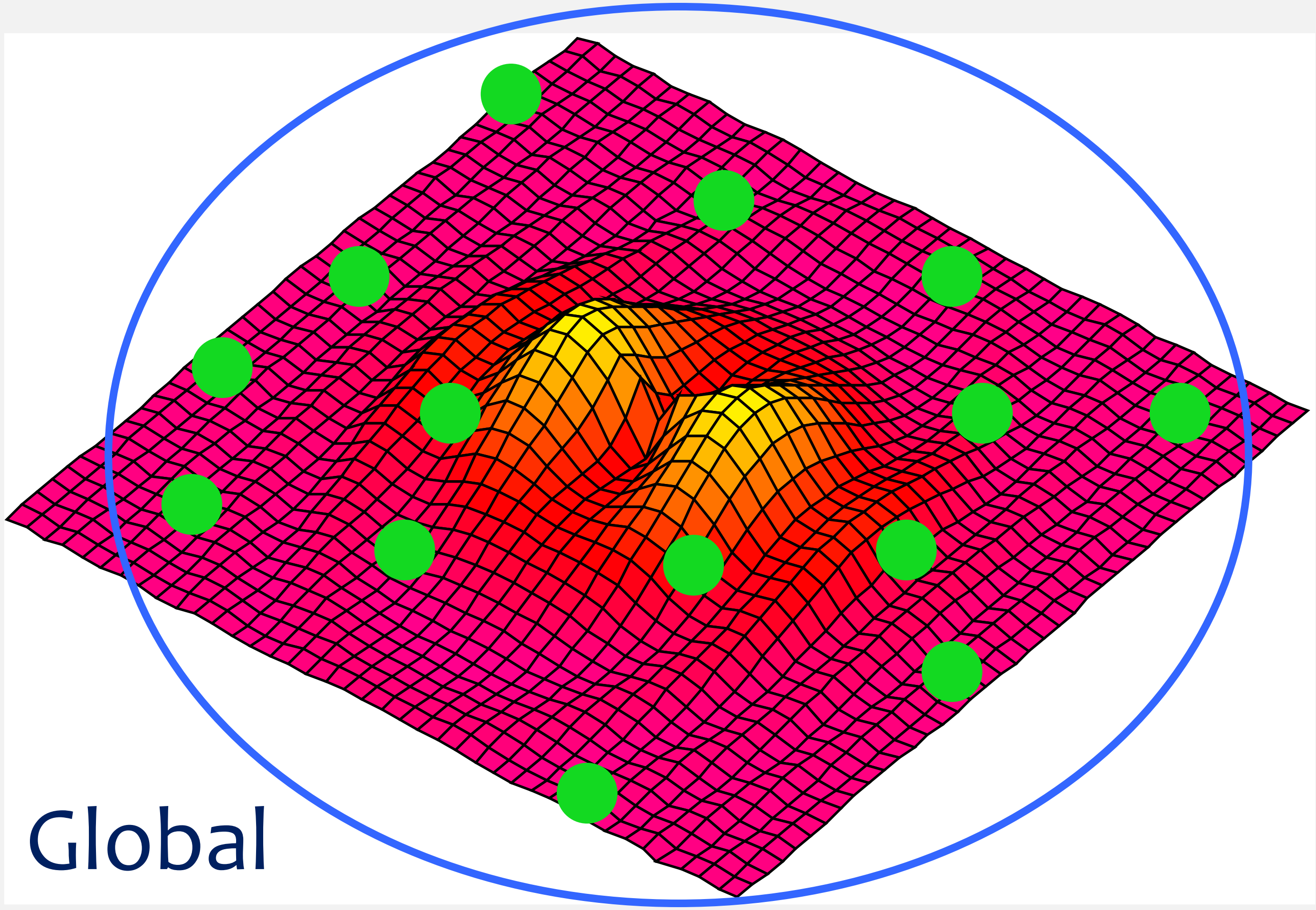
Neighborhoods



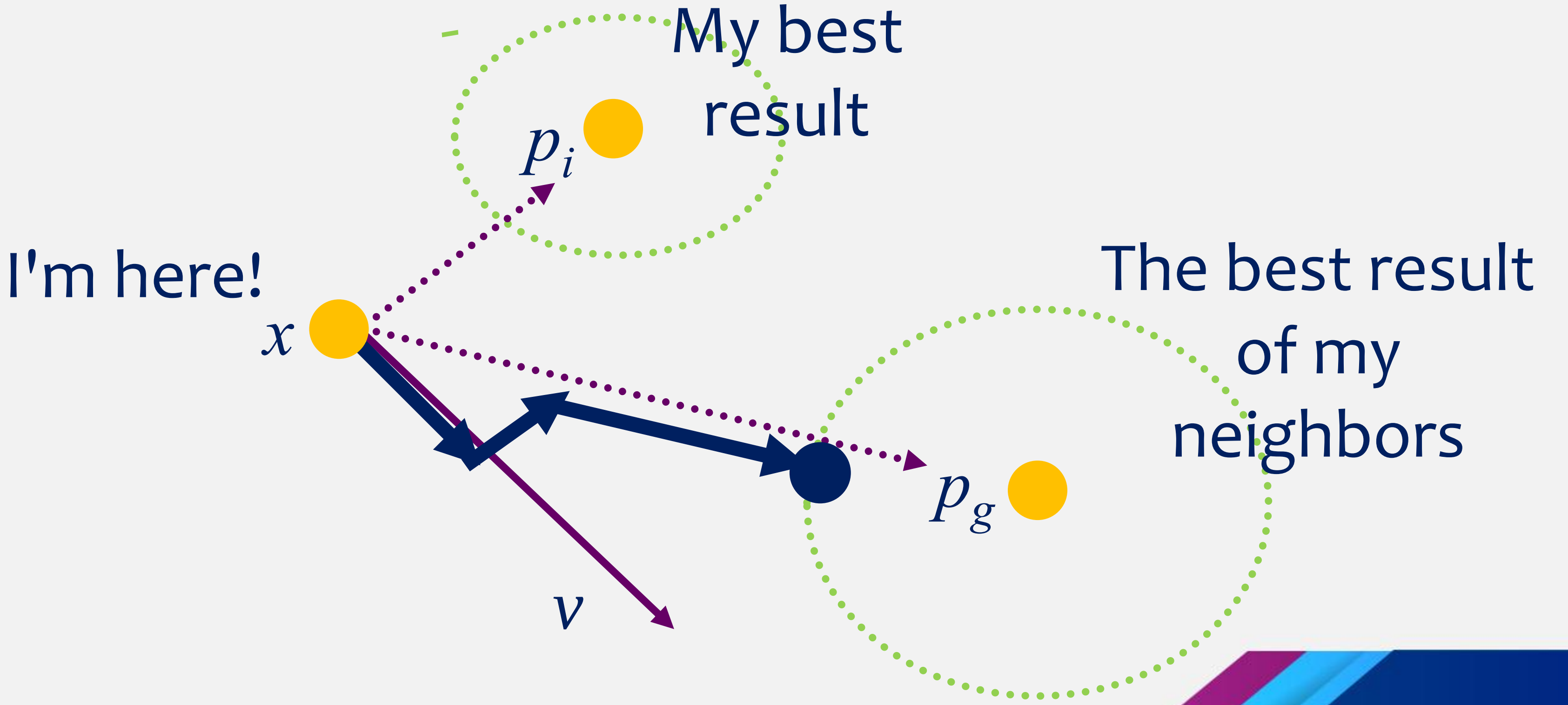
Circular neighborhood



Neighborhoods



- Particles adjust their positions according to a "psychosocial compromise" between what an individual considers comfortable and what society says.



Equations

Equation (a) to adjust the velocity

```
vel[]= C0 * vel[]
     + C1 * rand() * (pbest[] - pos [])
     + C2 * rand() * (gbest[] - pos [])
```

(In the original method, C0 = 1, but now many researchers play with this parameter)

Equation (b) to adjust the position

```
pos[] = pos[] + vel[]
```

Pseudocode for PSO

```
Repeat for each particle
    Initializes the particle
End

Repeat
    Repeat for each particle
        Calculate its fitness value
        If the fitness value is better than its best personal
             It sets its current value as the new pbest
    End

    Choose the particle with the overall best fitness as gbest

    Repeat for each particle
        Calculate the new velocity of the particle
                   with equation (a)
        Calculate the new position of the particle
                   with equation (b)
    End

While maximum iterations or minimum error criterion is not reached
```

Pseudocode for PSO

- The particle velocities in each dimension are subject to a maximum speed **Vmax** which is a parameter specified by the user.

- If the sum of the accelerations will cause that Vmax is exceeded in one dimension, then the velocity in this dimension would be established as Vmax.

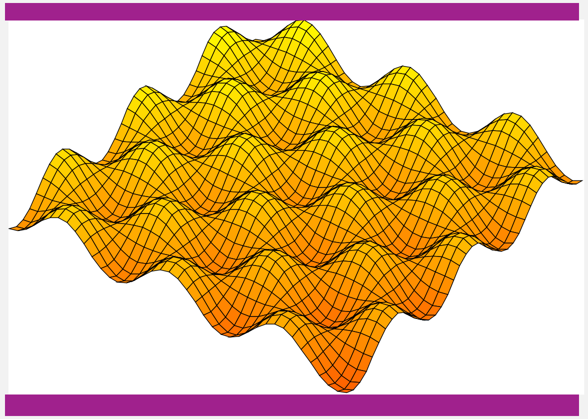
How to choose the parameters?

- **Number of particles**

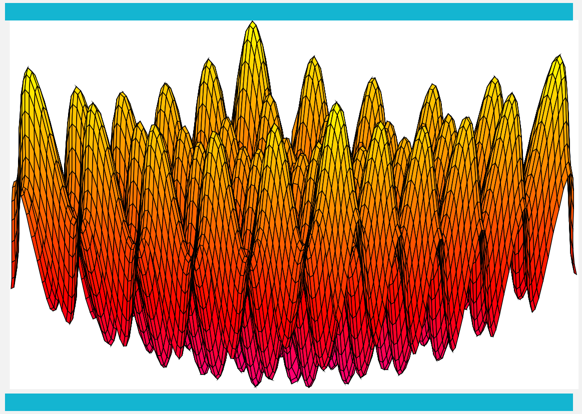
  (10-50) are reported as usually sufficient.
- **C1** (Importance of personal best)
- **C2** (Importance of best neighbor)
- Usually C1 + C2 = 4. Without better reason than as product of empirical results.
- **Vmax** - too low → too slow;
too high → too unstable.

Some functions that are frequently used to test optimization algorithms with real values.

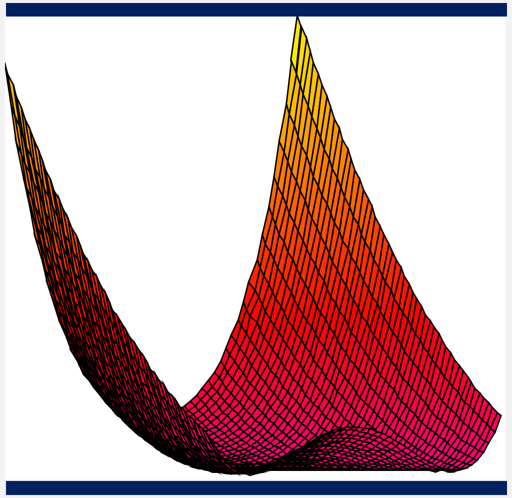
Griewank



Rastrigin



Rosenbrock




30%
20%
10%