INFO411

Lecture 11: Swarm Intelligence

Lizhi Peng

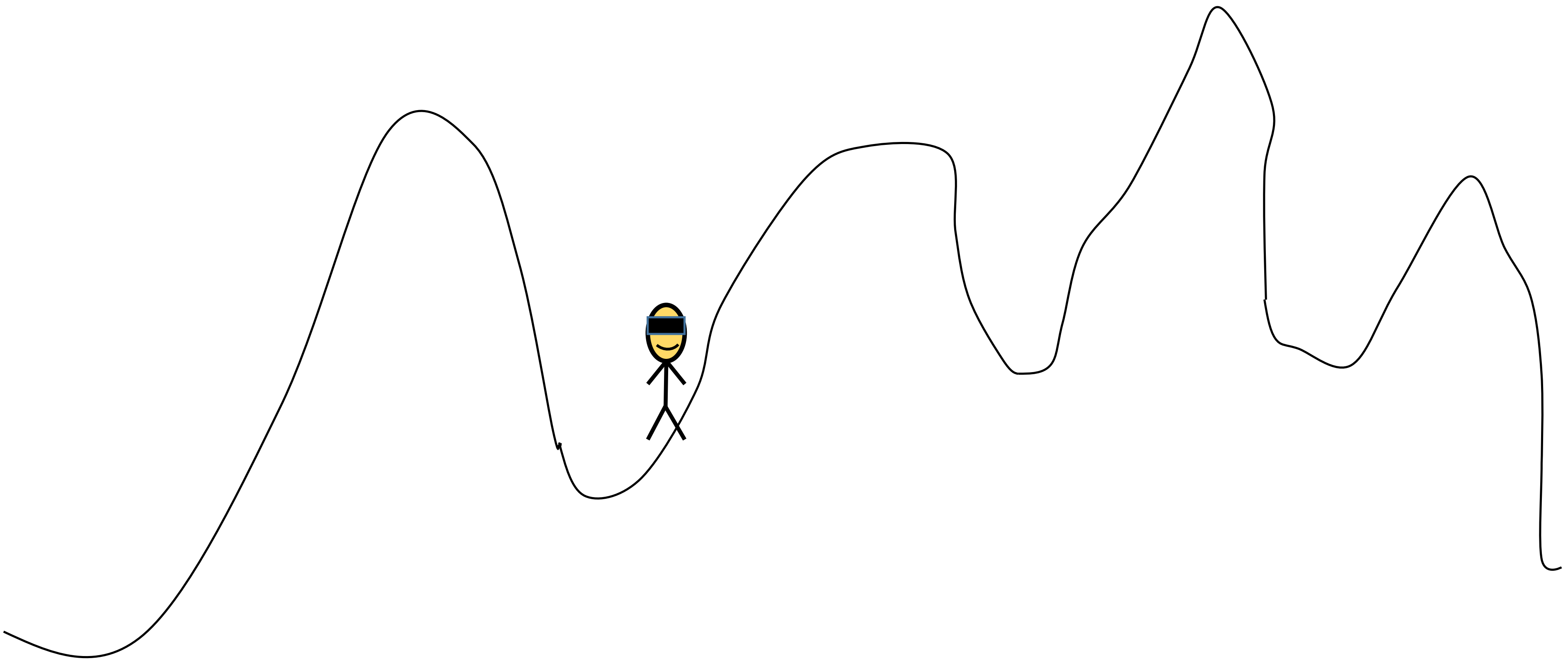
Email: penglizhi.jn@gmail.com

26 Sep, 2017

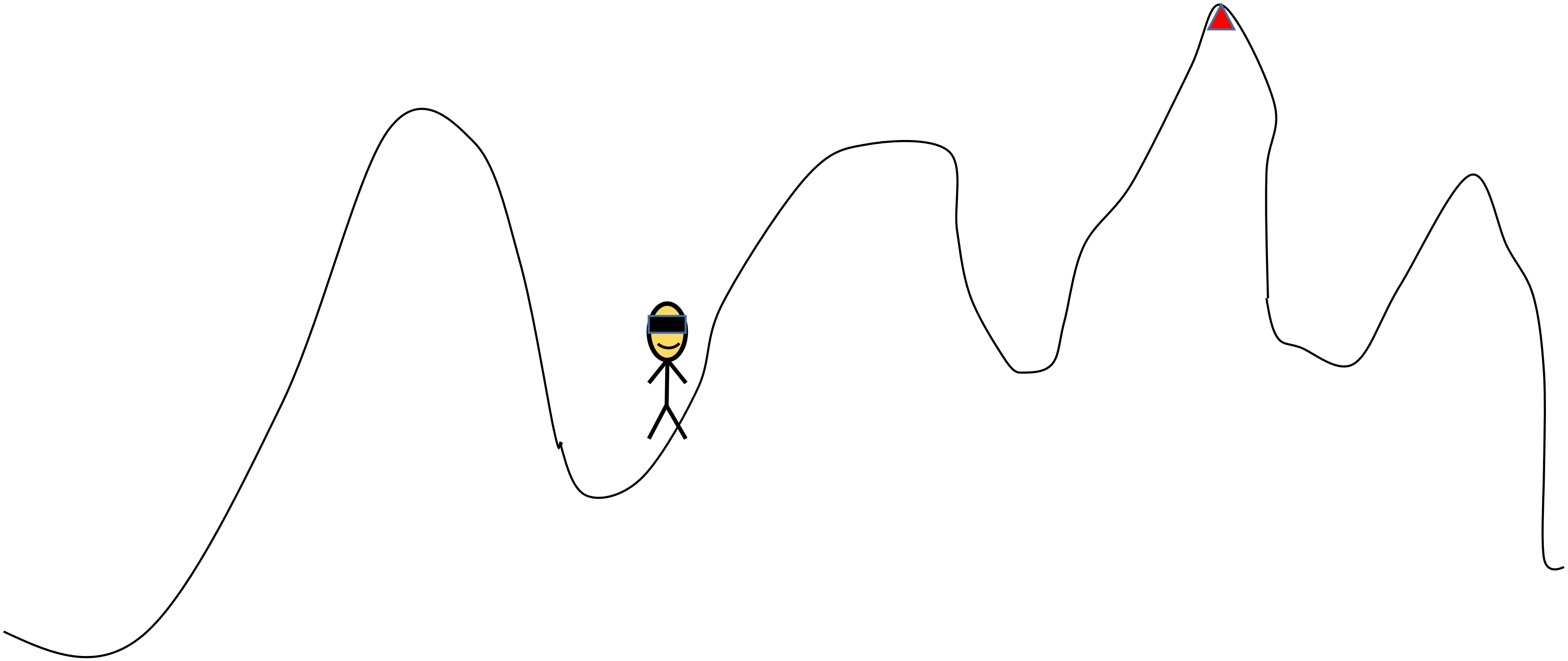
Outline

- **Optimization problem**
- Stochastic algorithms
- Particle Swarm Optimization
- Other swarm intelligence algorithms

Optimization problem

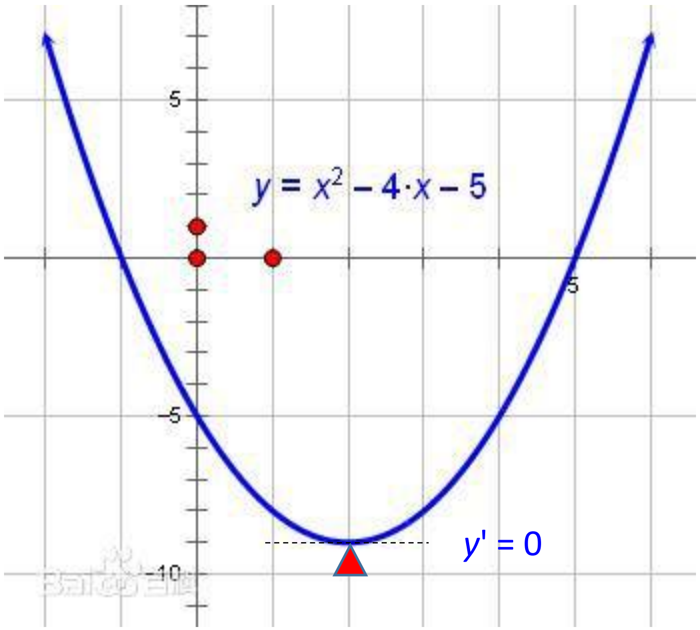


Optimization problem



Optimization problem

- Optimization problem
  - Search (find) the minimum (maximum) value of a function
  - Many real problems
- Simple differentiable functions
  - Use the derivatives



Optimization problem

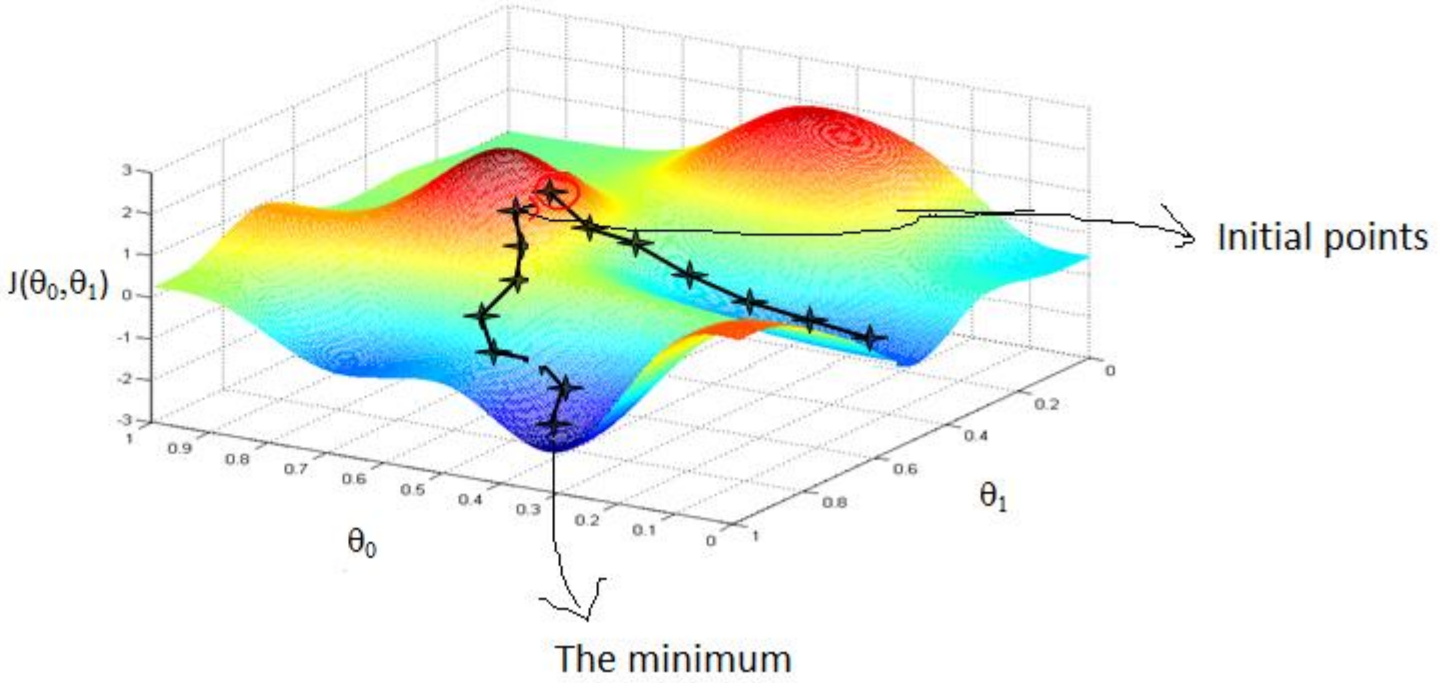
- High-dimensional problems: gradient descent
  - Gradient
    - for a 2-D function *f(x, y)*, its gradient vector on *(x0, y0)* is

$$grad(f) = (\partial f / \partial x, \partial f / \partial y)$$

    - The gradient direction is the “steepest” direction of the function
  - Go down inverse to the gradient direction to find the minimum/maximum value, until the gradient becomes a zero vector

Optimization problem

- The main problem: be trapped into local minima (stagnation points)

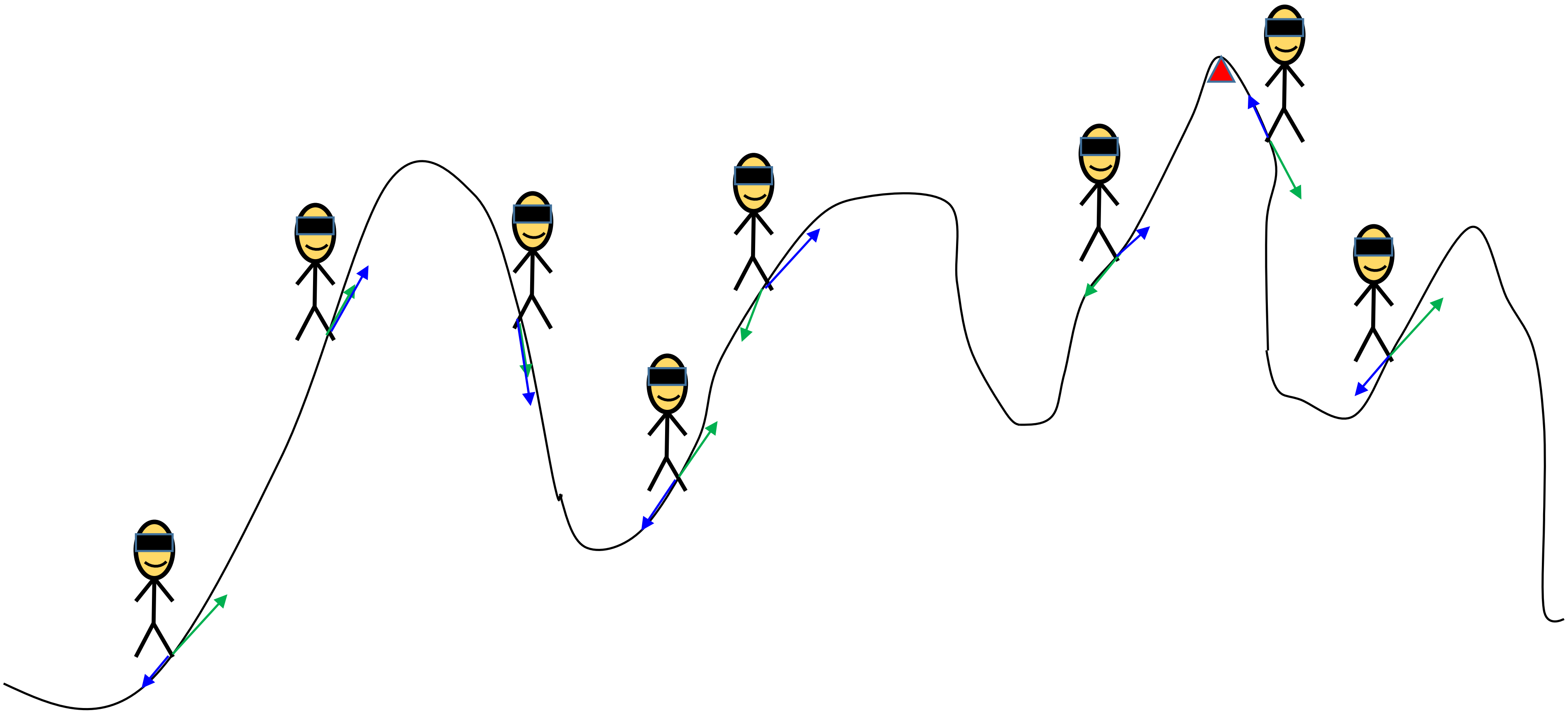


Picture from website http://www.cnblogs.com/pinard/p/5970503.html

Outline

- Optimization problem
- **Stochastic algorithms**
- Particle Swarm Optimization
- Other swarm intelligence algorithms

Stochastic algorithm



Stochastic algorithm

- Stochastic algorithm
  - A group of candidate solutions (**individuals**), form so called **population**
  - The population is generated and updated using random **operators (learning rules)**
  - The population is updated **iteratively**, until the **termination condition** is satisfied

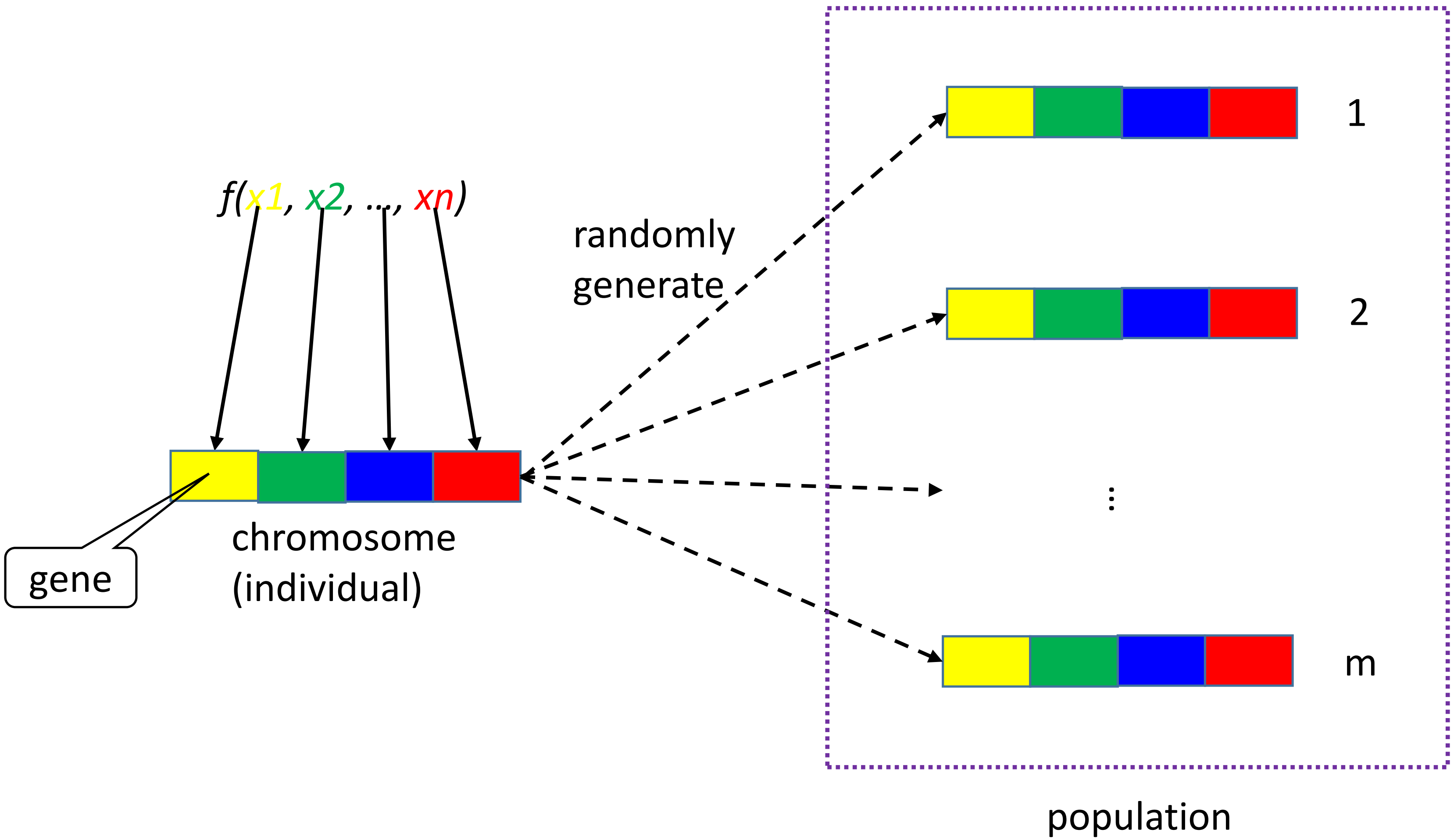
Stochastic algorithm

- Features
  - Global search abilities
  - Nature inspired: simulating bio-evolutionary procedure, birds foraging, …
- Categories
  - Genetic algorithm
  - Swarm intelligence algorithm

Genetic algorithm (GA)

- Basic Ideas
  - Simulating bio-evolutionary procedures
    - Chromosome: candidate solution
    - Selection operator
    - Crossover (combination) operator
    - Mutation operator
  - Operations are randomly executed according to some probabilities

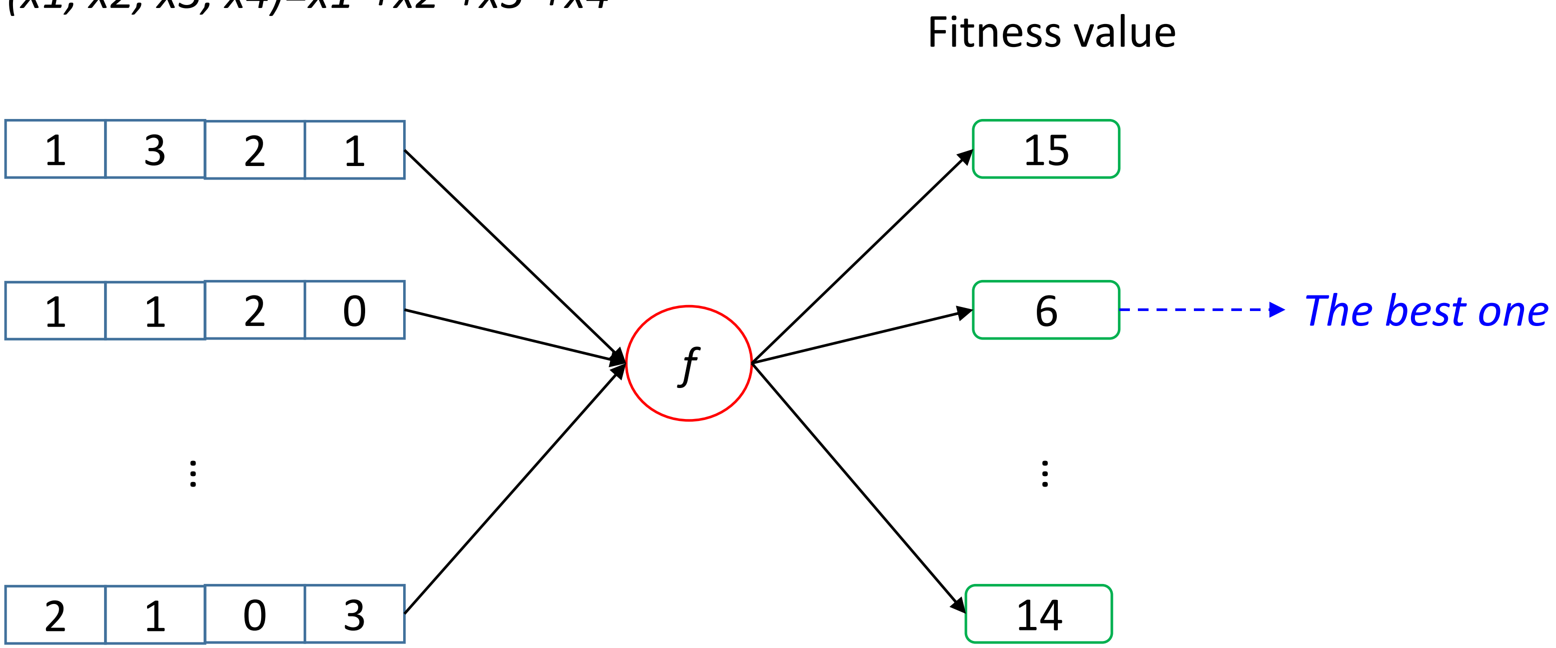
GA--Coding



f(x1, x2, …, xn)
randomly
generate
1
2
⋮
m
gene
chromosome
(individual)
population

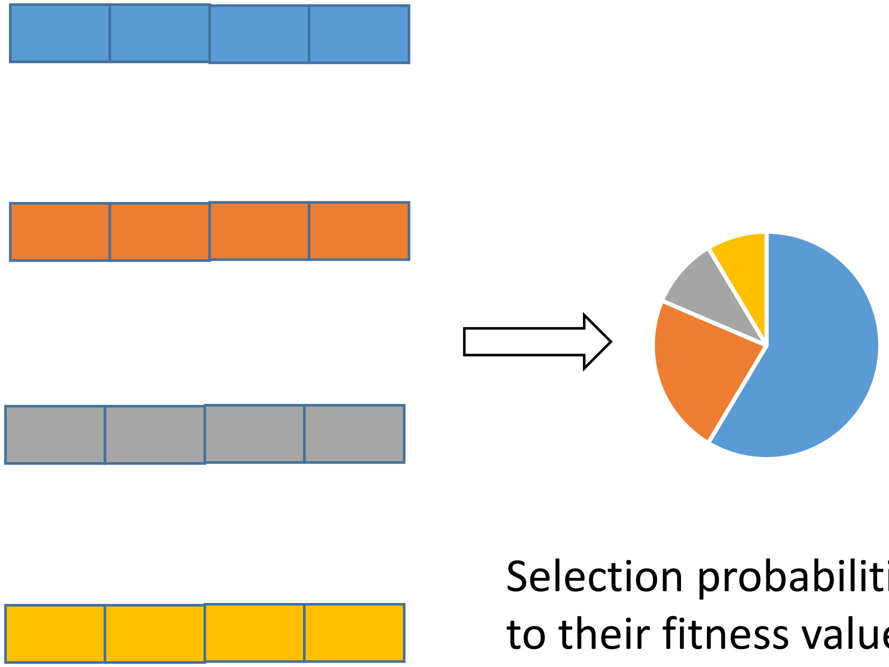
GA—Fitness evaluation

$f(x1, x2, x3, x4) = x1^2 + x2^2 + x3^2 + x4^2$



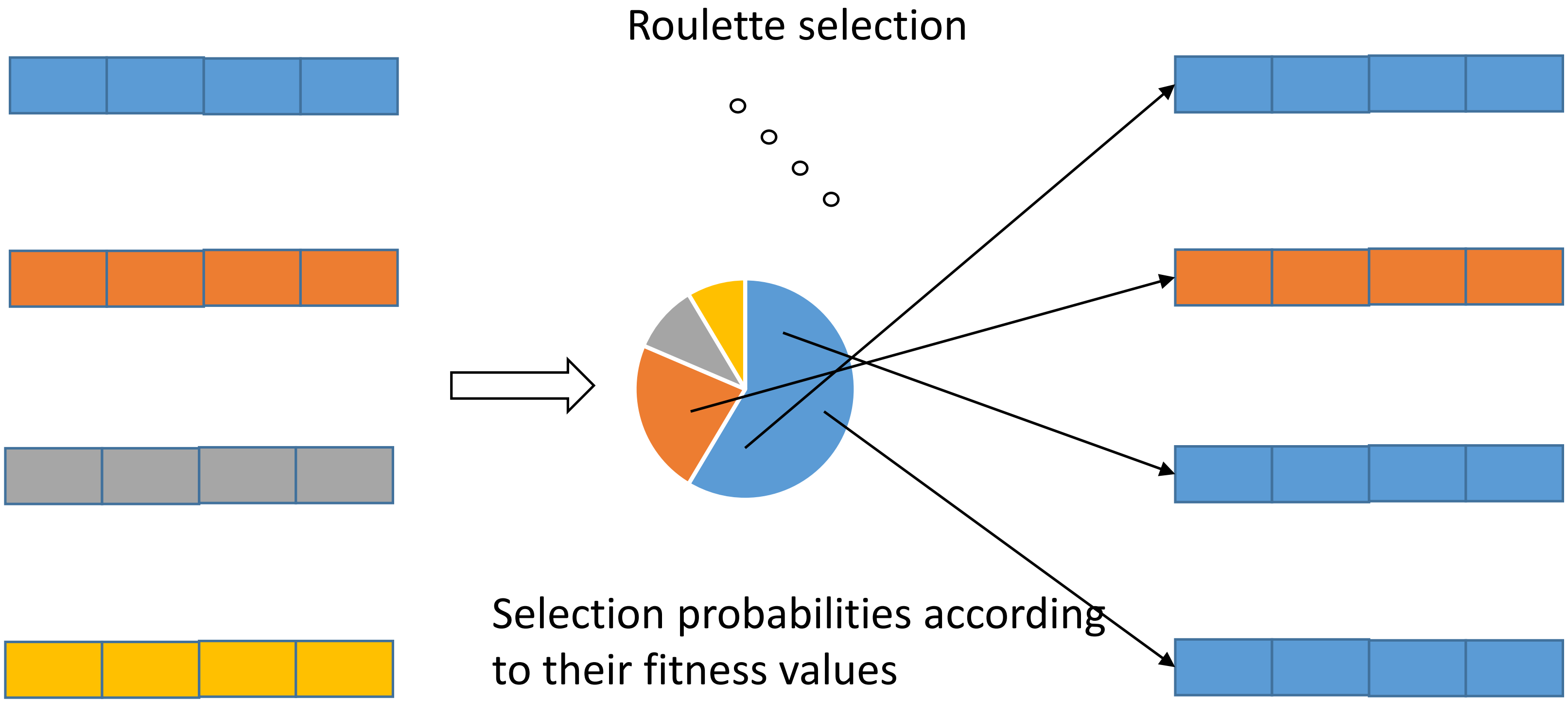
GA--Selection

Roulette selection

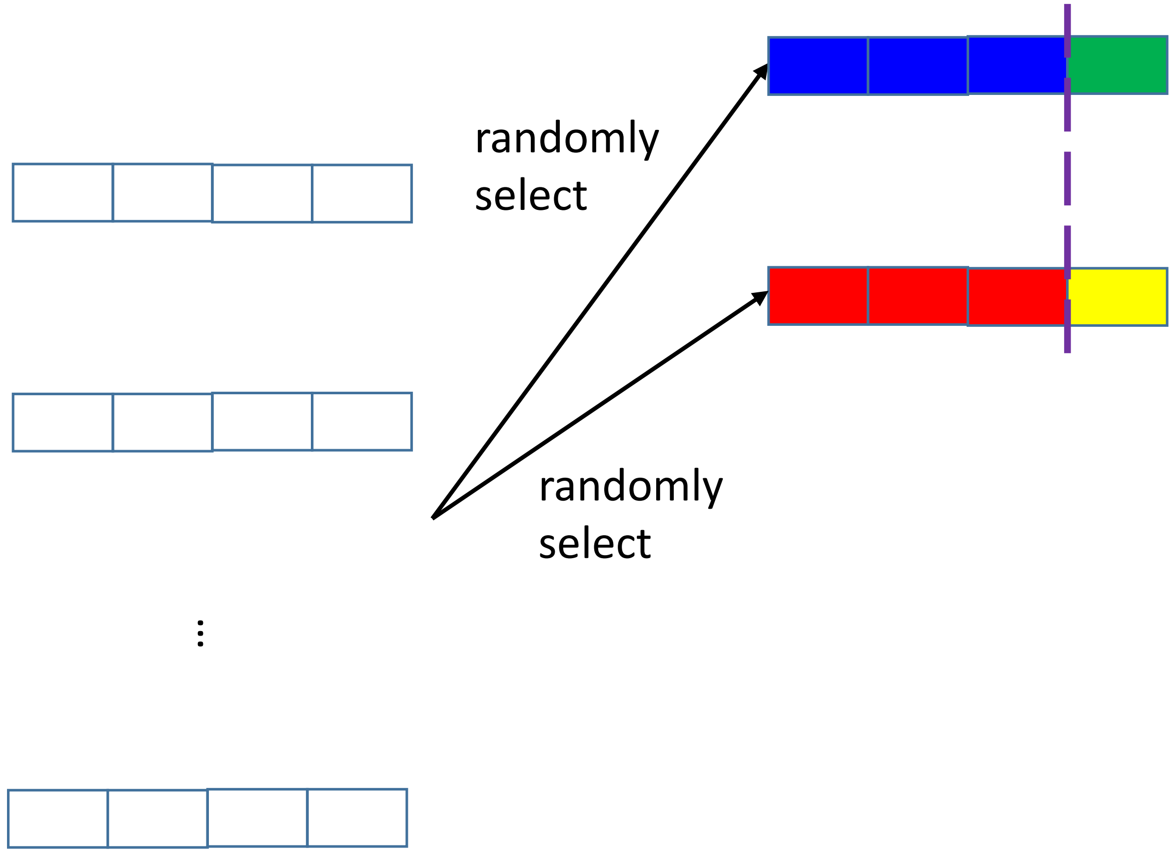


Selection probabilities according to their fitness values

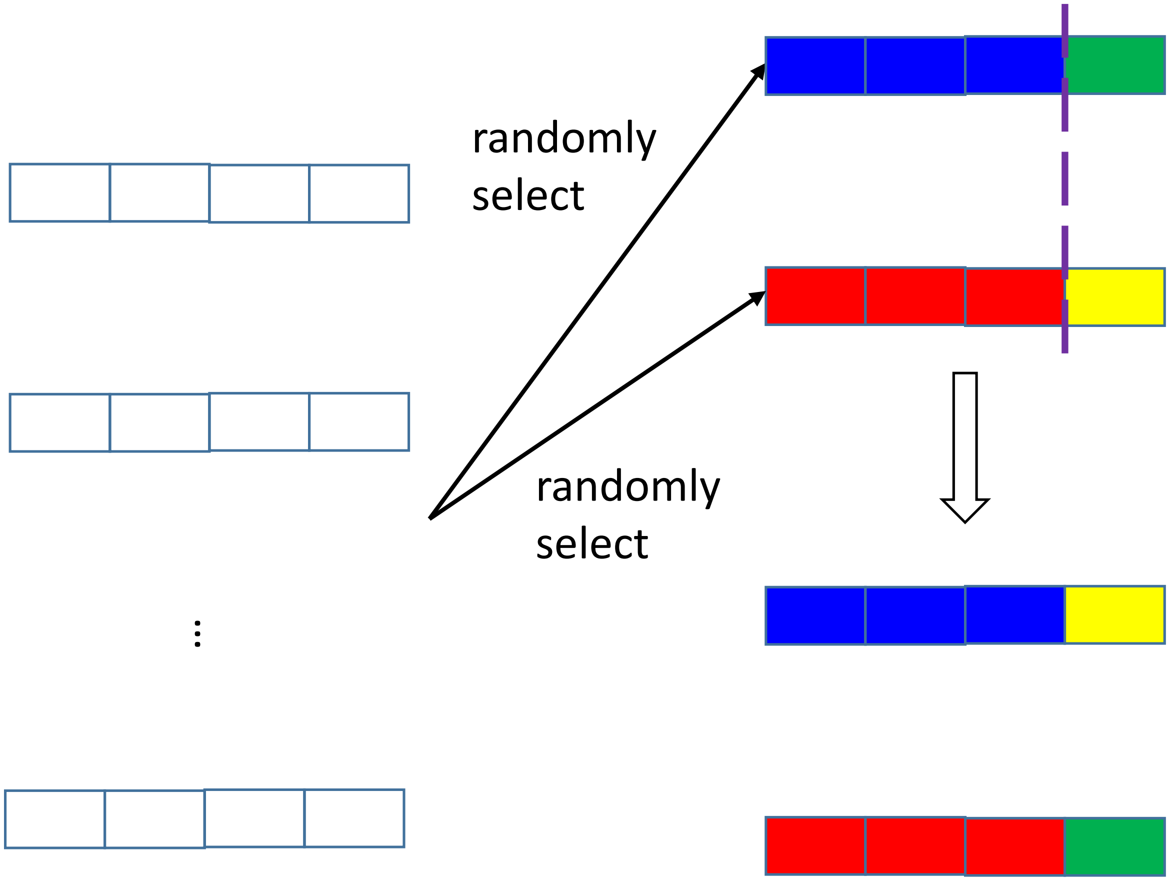
GA--Selection



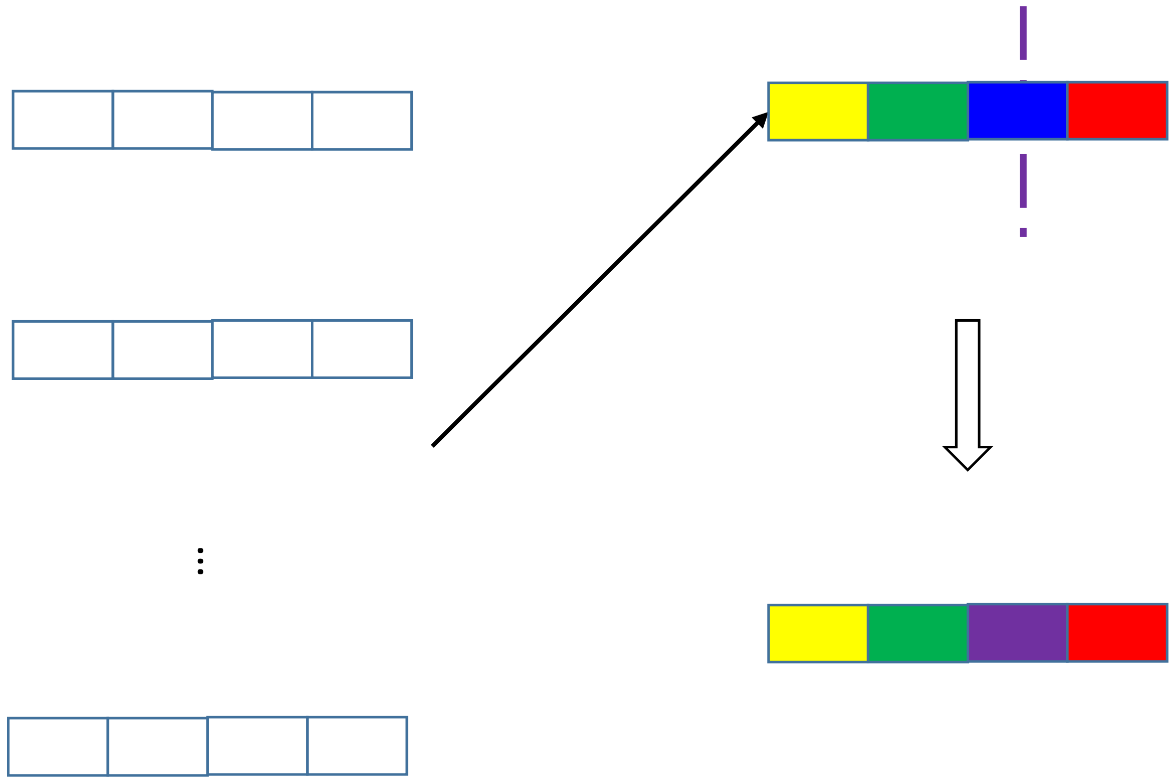
GA--Crossover



GA--Crossover



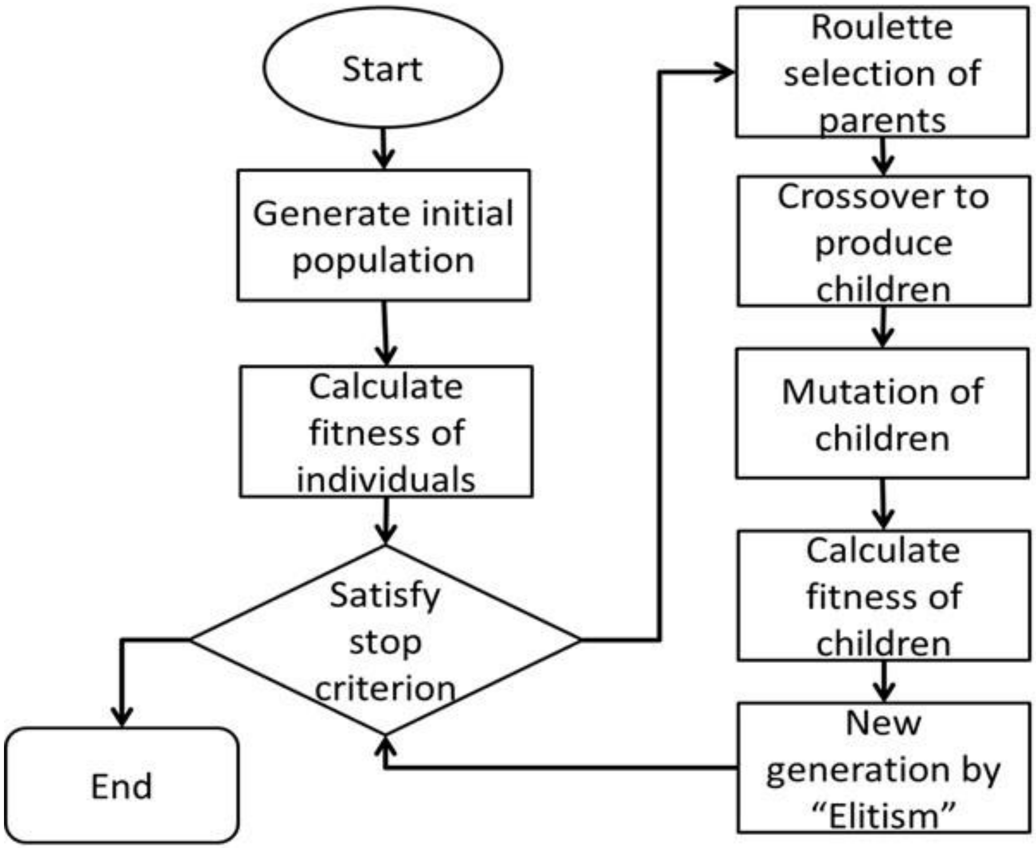
GA--Mutation



GA--Probabilities

- Crossover probability p_c
  - Controls the crossover rate
- Mutation probability p_m
  - Controls the mutation rate
  - Usually is a small number, e.g. 0.05

GA—Algorithm framework



Picture from Open-I https://openi.nlm.nih.gov/

Other GAs

- Gene Programming (GP)
- Gene expression programming (GEP)
- Immune algorithm (IA)
- Differential evolution (DE)
- Population-based incremental learning (PBIL)
- …

Swarm intelligence

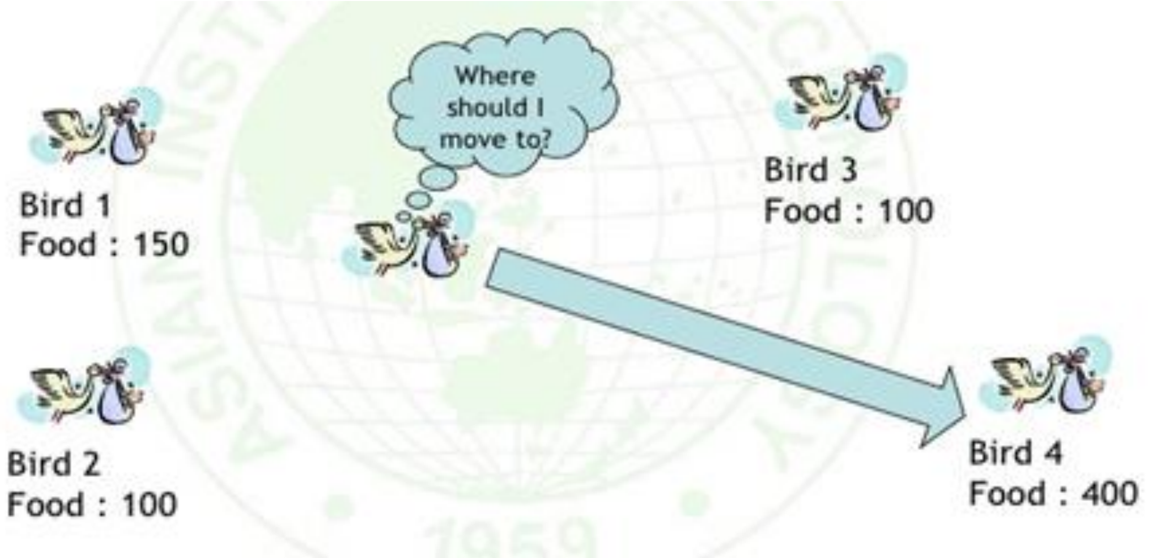
- Another large stochastic algorithm family
- Basic idea: simulating the intelligent behaviours of animals, e.g. birds, ants, and bees
- Share some features with GAs
  - Population based
  - Iterative search
- Differences to GAs
  - No crossover and mutation
  - Individuals move in the solution space, just like animals

Outline

- Optimization problem
- Stochastic algorithms
- **Particle Swarm Optimization**
- Other swarm intelligence algorithms

Particle swarm optimization (PSO)

- Simulating the behaviours of bird/fish swarm foraging to search the best solution of the target problem
- Introduced by Kennedy and Eberhart in 1995 (Kennedy et al., 1995)

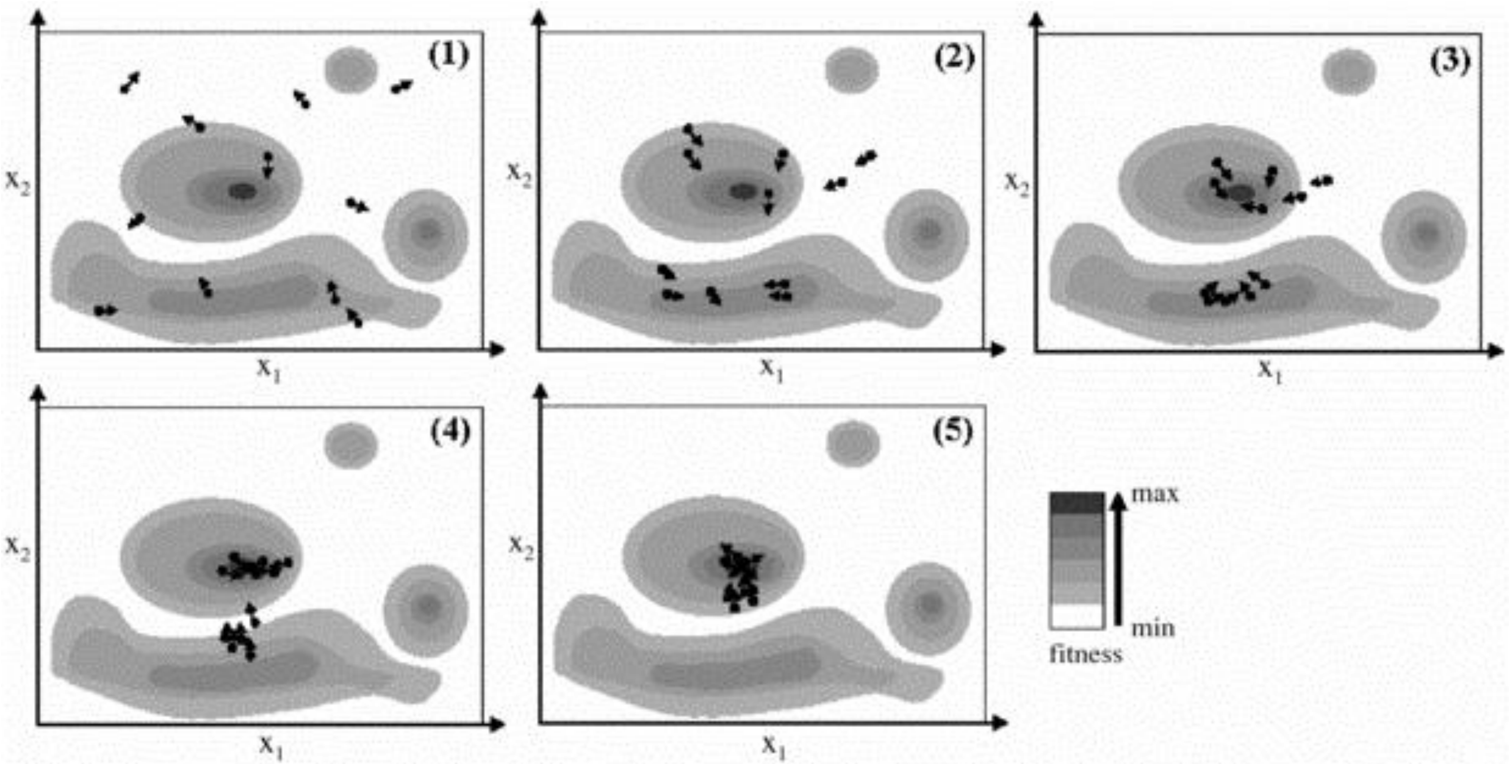


Picture from Open-I https://openi.nlm.nih.gov/

Particle swarm optimization (PSO)

- Particle
  - Position (*x*): a candidate solution
  - Velocity (*v*): let the particle move iteratively
- Population (Swarm)
  - A group of particles
  - Initialized randomly

Particle swarm optimization (PSO)



Picture from article: Li Y, Yao D, Yao J, et al. A particle swarm optimization algorithm for beam angle selection in intensity-modulated radiotherapy planning[J]. Physics in medicine and biology, 2005, 50(15): 3491.

Particle swarm optimization (PSO)

- Learning rule

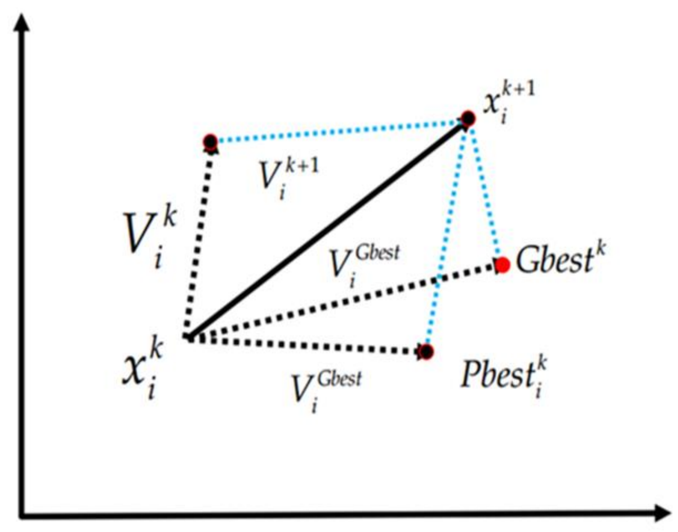
$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{p}_i - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{g} - \mathbf{x}_i(t))$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1)$$

- p_i is the personal best position (p-best) the *i*-th particle found so far
- g is the global best position (g-best) all particles found so far
- w is the inertia weight
- c_1, c_2 are the acceleration coefficients
- r_1, r_2 are two random numbers between 0 and 1.

Particle swarm optimization (PSO)

- The velocity includes three components
  - p-best learning
  - g-best learning
  - the last velocity



Particle swarm optimization (PSO)

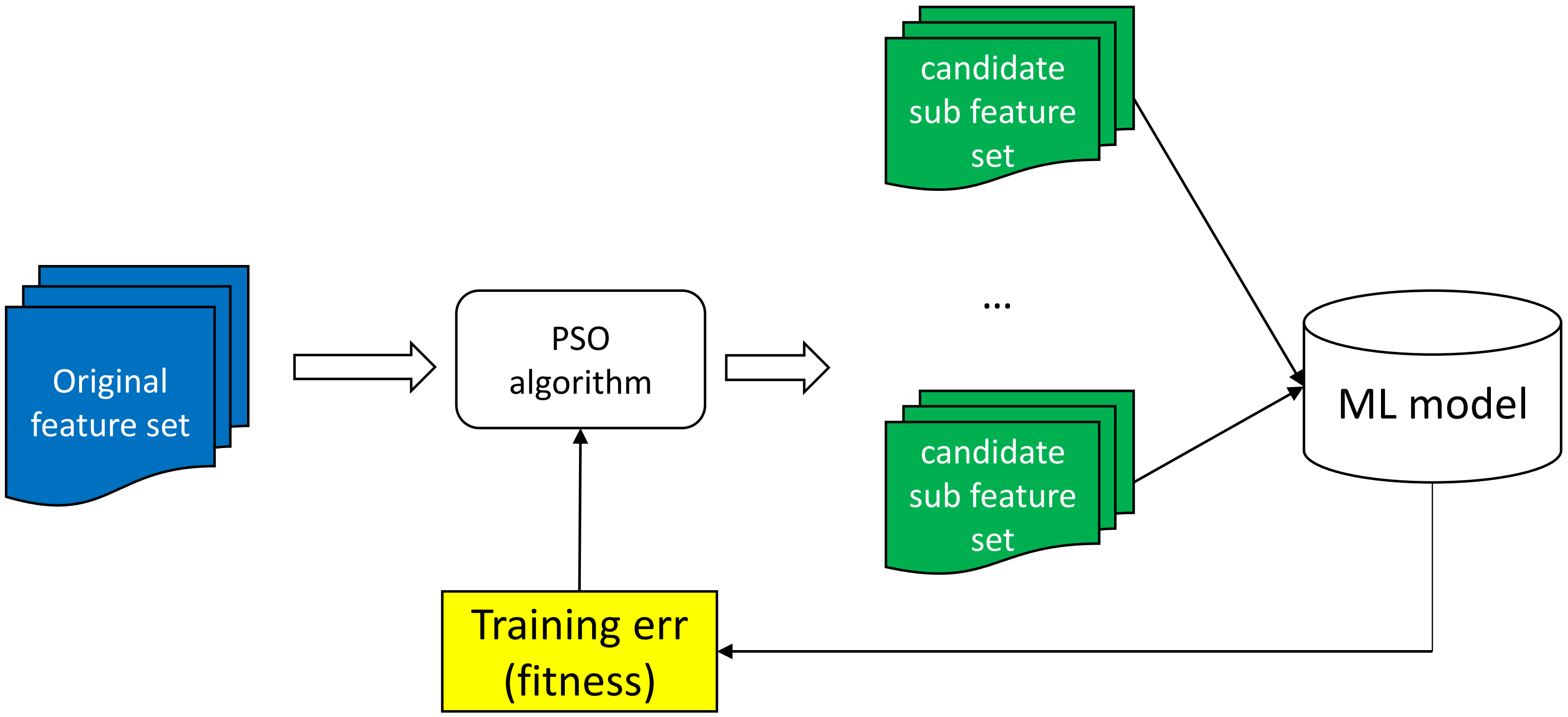
- PSO applications
  - Shortest path
  - Power grid schedule
  - Resource schedule
  - Industrial design
  - …

Particle swarm optimization (PSO)

- PSO applications for machine learning
  - Feature selection
    - Chuang L Y, Chang H W, Tu C J, et al. Improved binary PSO for feature selection using gene expression data[J]. Computational Biology and Chemistry, 2008, 32(1): 29-38.
    - Wang X, Yang J, Teng X, et al. Feature selection based on rough sets and particle swarm optimization[J]. Pattern recognition letters, 2007, 28(4): 459-471.
  - Model parameter optimization
    - Mendes R, Cortez P, Rocha M, et al. Particle swarms for feedforward neural network training[C]//Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. IEEE, 2002, 2: 1895-1899.
    - Peng L, Zhang H, Yang B, et al. A new approach for imbalanced data classification based on data gravitation[J]. Information Sciences, 2014, 288: 347-373.

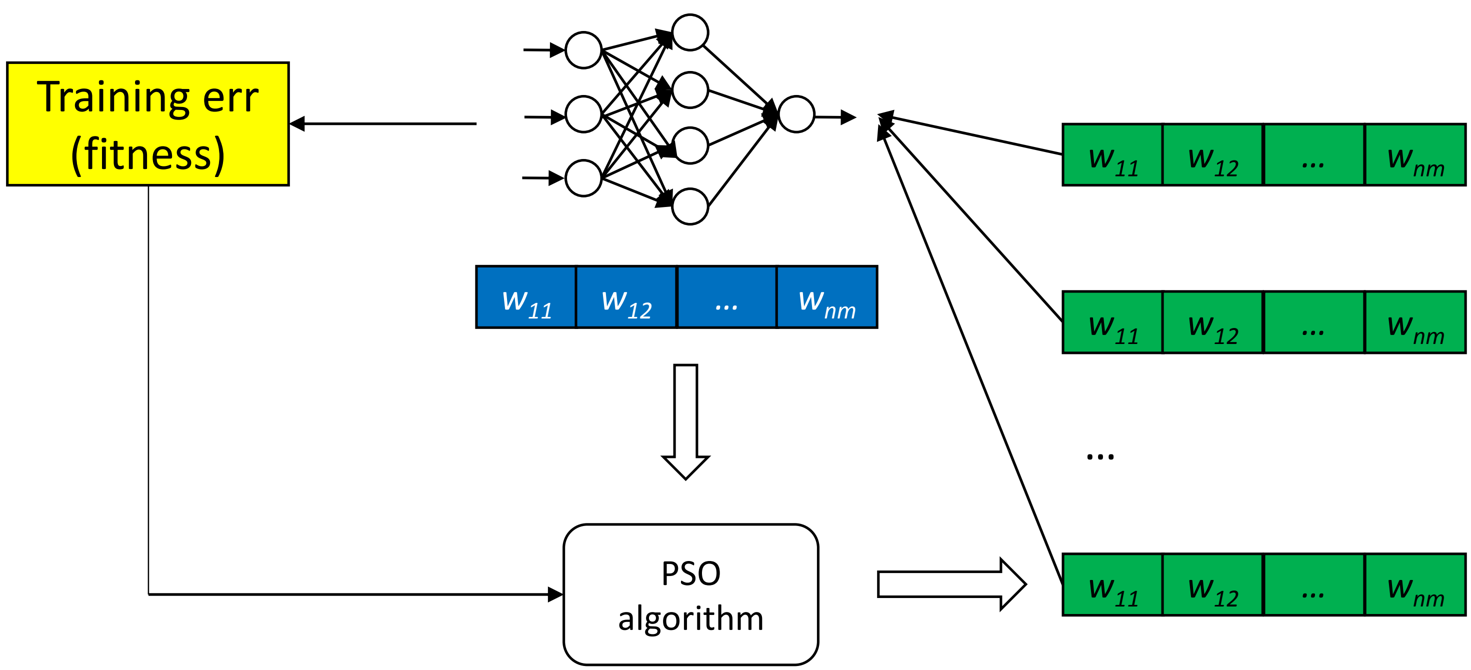
Particle swarm optimization (PSO)

- Feature selection using PSO



Particle swarm optimization (PSO)

- Model parameter optimization
  - weight optimization of ANN



Particle swarm optimization (PSO)

- Premature convergence
  - The most significant problem of basic PSO algorithm
  - All particles are trapped into a local minimum
- Solution
  - Keeping population diversity to enhance the global search (exploration) ability

PSO variants

- Cooperative co-evolution
  - De-compose a high dimensional problem into several subcomponents, which correspond to the same number of sub-populations
  - All sub-populations evolve simultaneously, and the results are then combined
  - Examples
    - CCPSO2 (Li et al., 2012)
    - DSPLSO (Yang et al., 2016)

PSO variants

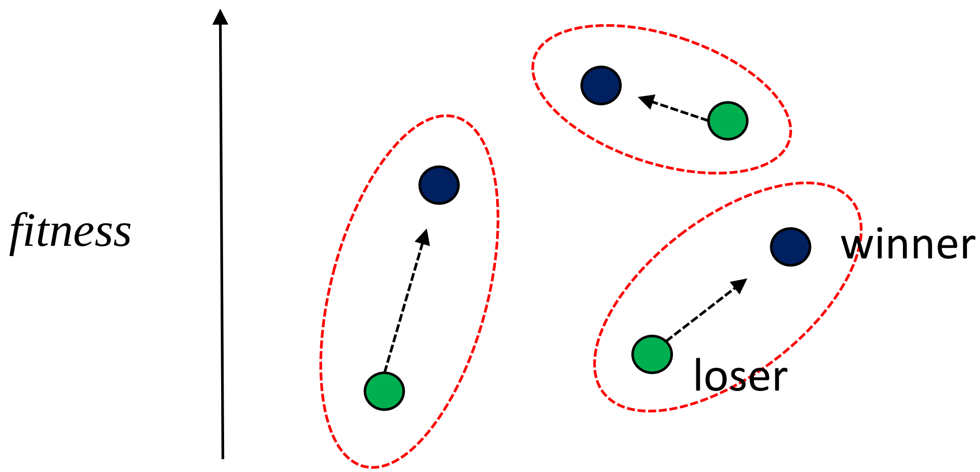
- Hybrid algorithms
  - Use other algorithms' operators, e.g. genetic algorithm, to enhance the search ability of PSO
  - Examples
    - DEPSO (Xu et al., 2010)
    - BBPSO (Kennedy, 2003)

PSO variants

- Random exemplar learning
  - CSO (Cheng et al., 2015): particles compete pairwise, the loser learns from the winner
  - PLPSO: Probability learning particle swarm optimization

Competitive swarm optimization (CSO)

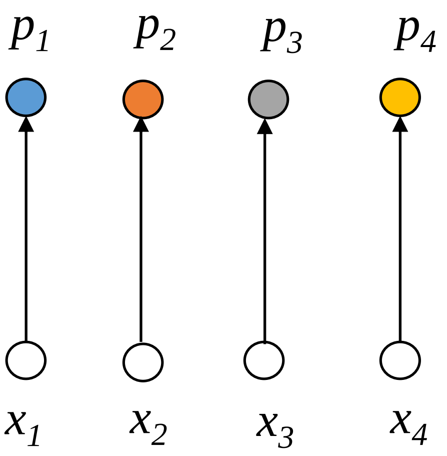
- Particles compete pairwise, the loser learns from the winner

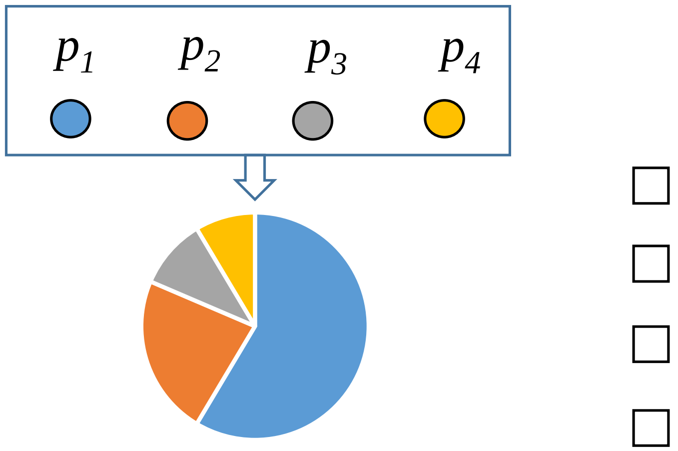


Probability learning particle swarm optimization (PLPSO)

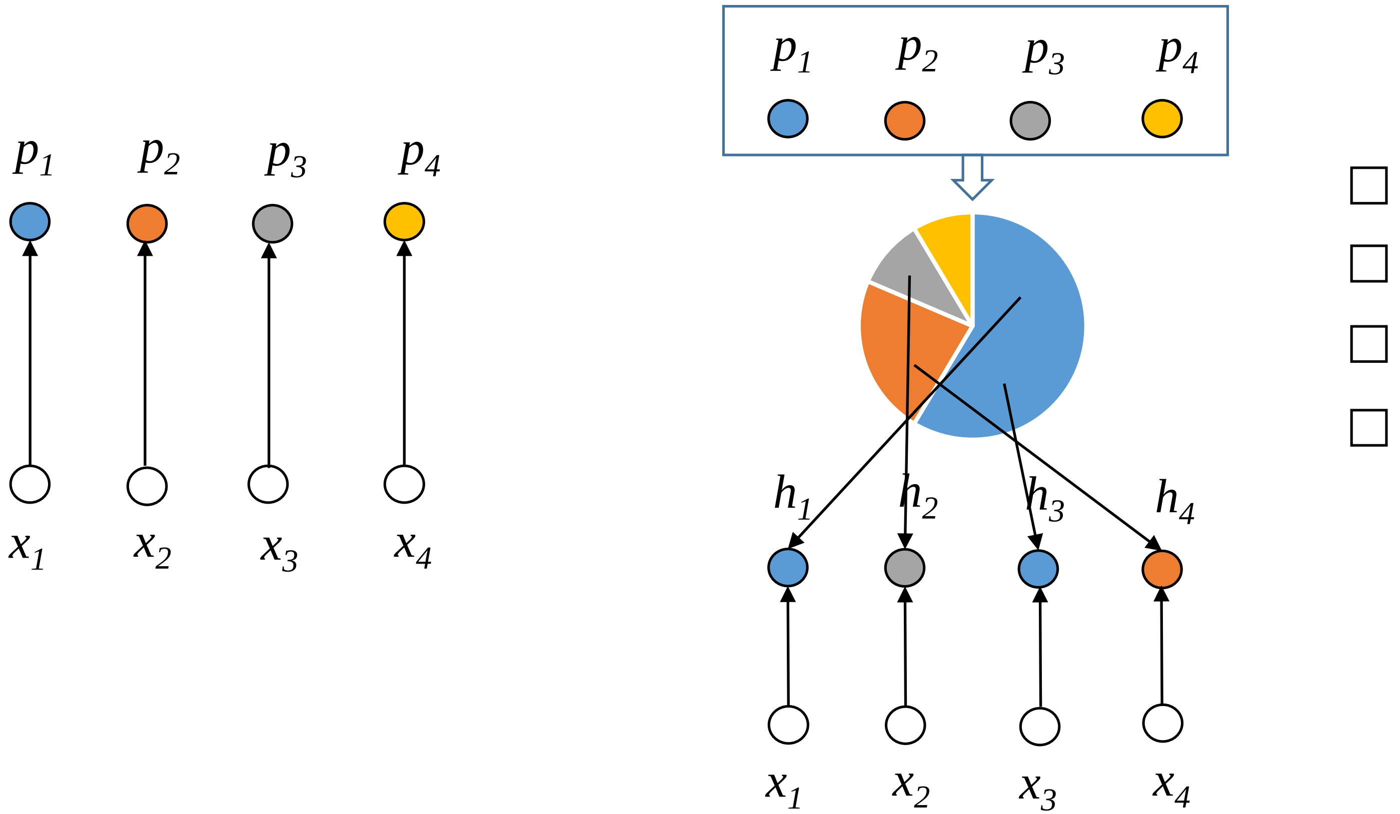
- P-best learning of the standard PSO
  - Is a kind of elitism learning, good for learning efficiency (positive side)
  - Simplex learning target (exemplar) leads to premature convergence (negative side)
- Our Idea
  - A variant of p-best learning
  - Let a particle learns from different p-bests according to their probabilities

PLPSO





PLPSO



PLPSO

- Definition of the probability of p_i (the *i*-th p-best)
  - Historical merit of p_i

$$\mu_i^h = \max_{j \in [1,n]} f(\mathbf{p}_j(t)) - f(\mathbf{p}_i(t))$$

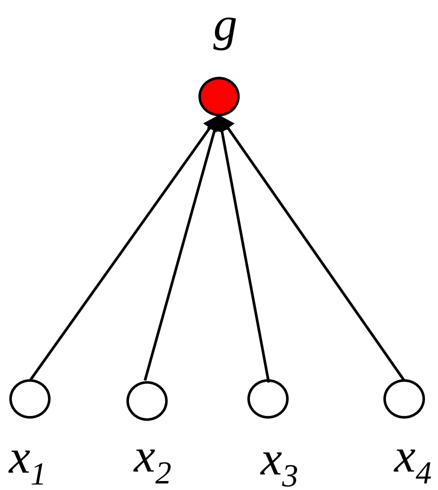
  - Probability of p_i

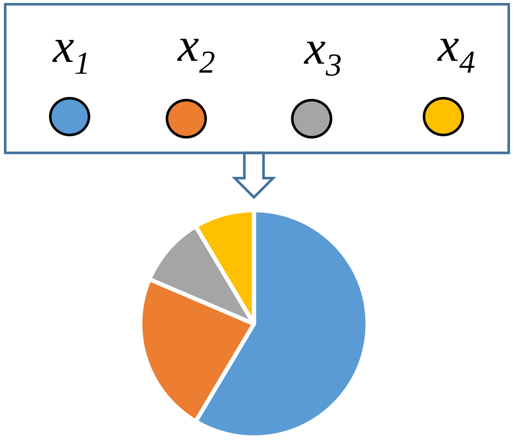
$$\pi_i^h = \frac{\mu_i^h}{\sum_j \mu_j^h}$$

PLPSO

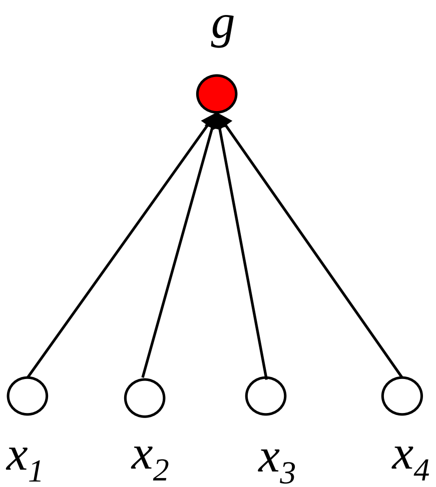
- G-best learning of the standard PSO
  - Is a pure elitism learning
  - Only has one exemplar: the global best position all particles found
- Ideas
  - Discard g-best learning
  - Let a particle learns from another particle according to its probability

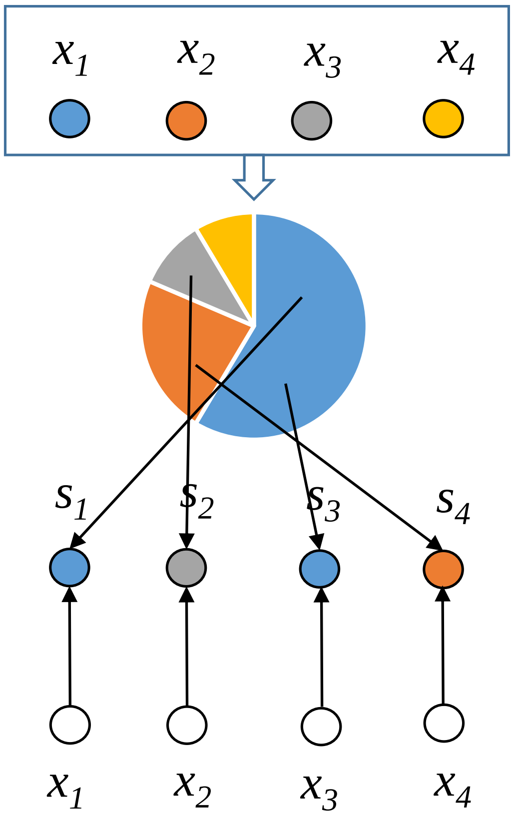
PLPSO





PLPSO





Probability based social learning

- Problem of worse exemplar
  - It is possible for a particle to select a exemplar worse than it, $f(s_i)>f(x_i)$
  - Learning from a bad exemplar is not good for the search procedure
- Solution
  - If $f(s_i)>f(x_i)$, the learner particle **does not move**
  - Such a particle is called an **idle particle**

PLPSO

$$\mathbf{v}_i(t+1) = \begin{cases} r_1\mathbf{v}_i(t) \\ \;+ r_2 c(\mathbf{h}_i(t) - \mathbf{x}_i(t)) \\ \;+ r_3(\mathbf{s}_i(t) - \mathbf{x}_i(t)), & \text{if } f(\mathbf{s}_i(t)) < f(\mathbf{x}_i(t)) \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{x}_i(t+1)$$

- Features:
  - No inertia weight
  - No acceleration coefficient for the social learning item

Outline

- Optimization problem
- Stochastic algorithms
- Particle Swarm Optimization
- **Other swarm intelligence algorithms**

Ant colony optimization (ACO)

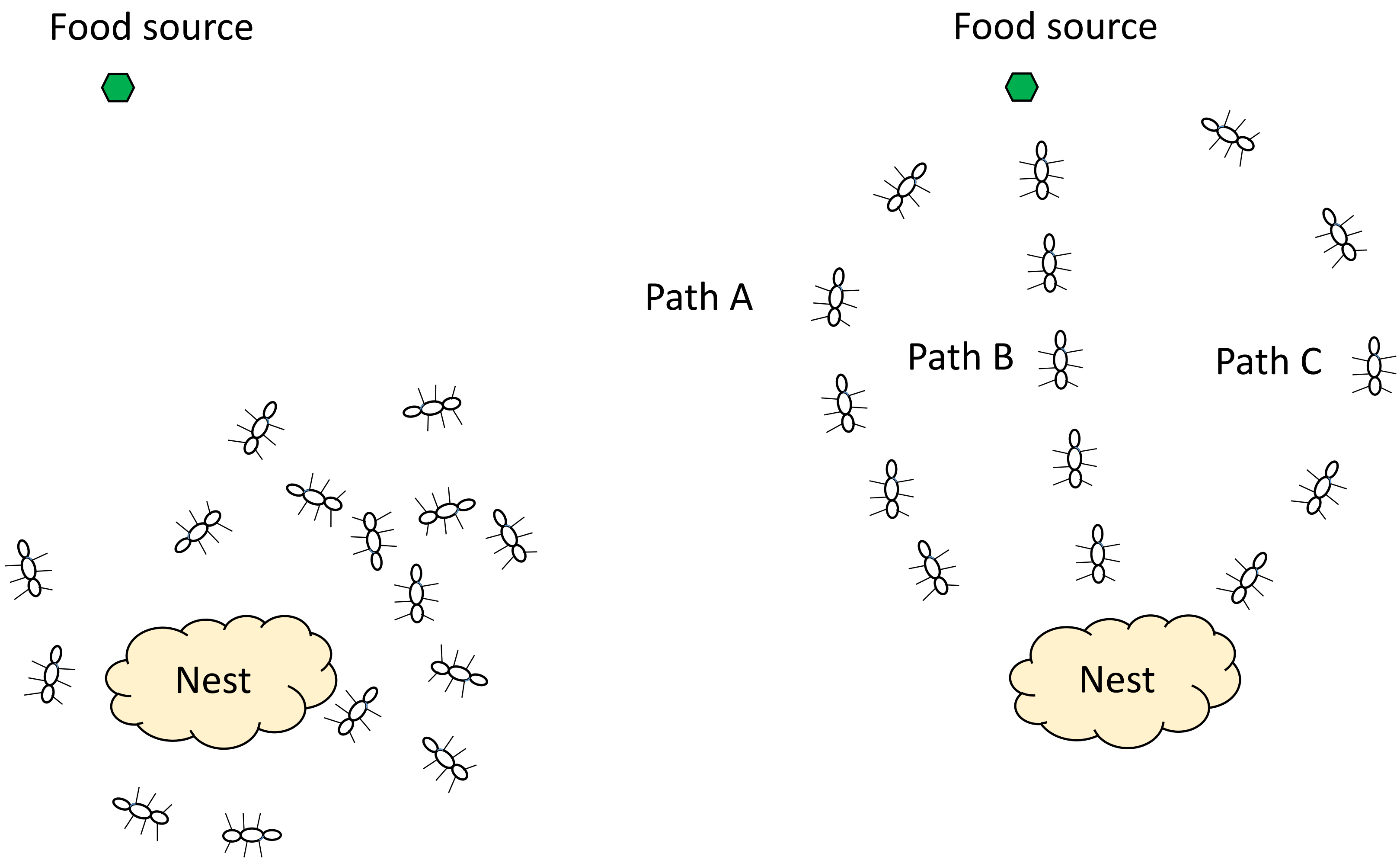
- Simulating the behaviours of ant swarm finding foods
- Introduced by Dorigo in 1992 (Dorigo, 1992)
- More complicated than PSO

Picture from Wikipedia https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms

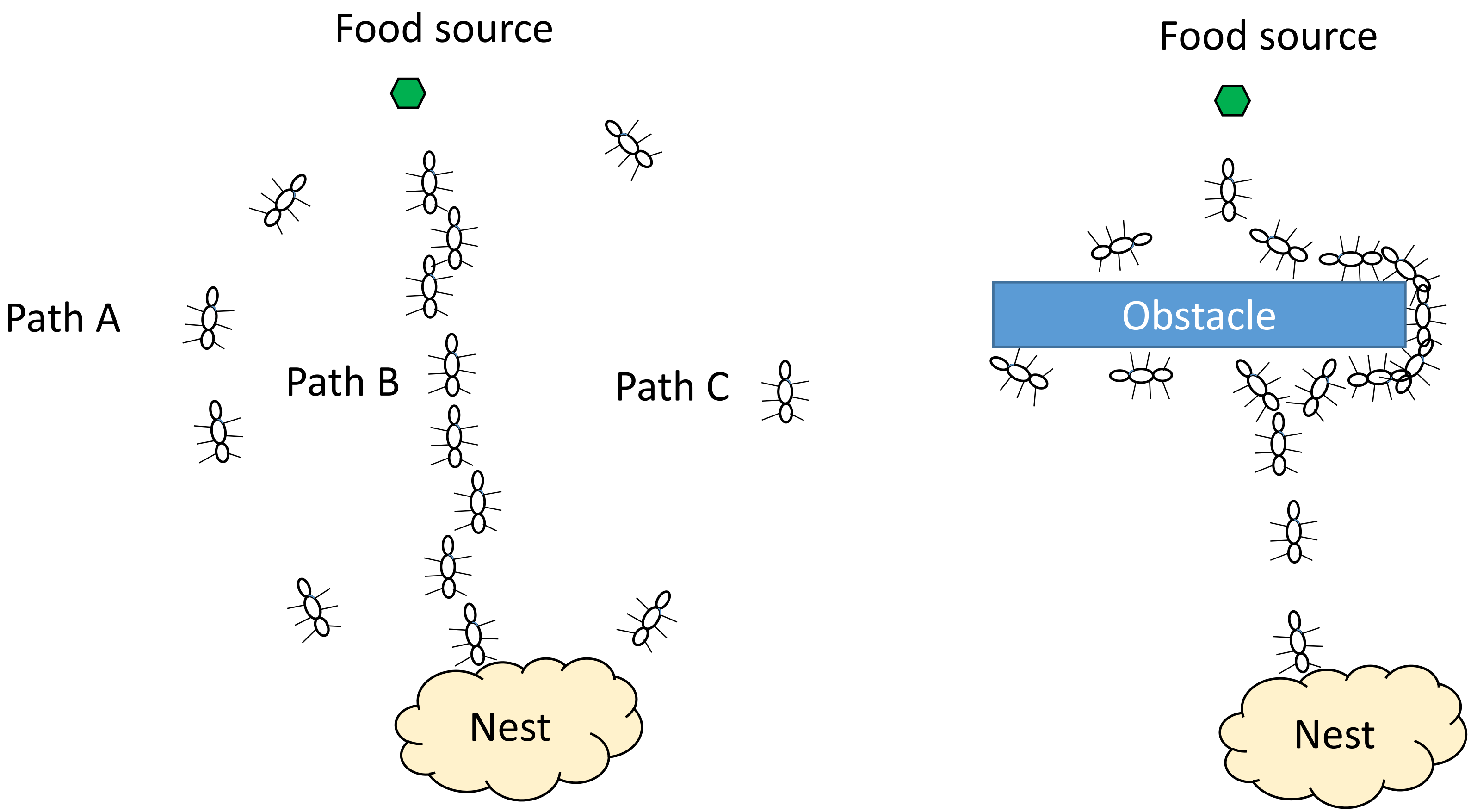
Ant colony optimization (ACO)

- Goal: searching the optimal path between the nest and the food source
- Ideas and concepts
  - Path between food source and nest: candidate solution
  - Ants move randomly according to the probabilities of all possible paths
  - Ants deposit **pheromone** on their trails
  - More pheromones on a path increases the probability it to be followed (attracts more ants)
  - Pheromones evaporate by time

Ant colony optimization (ACO)



Ant colony optimization (ACO)



Artificial bee colony algorithm (ABC)

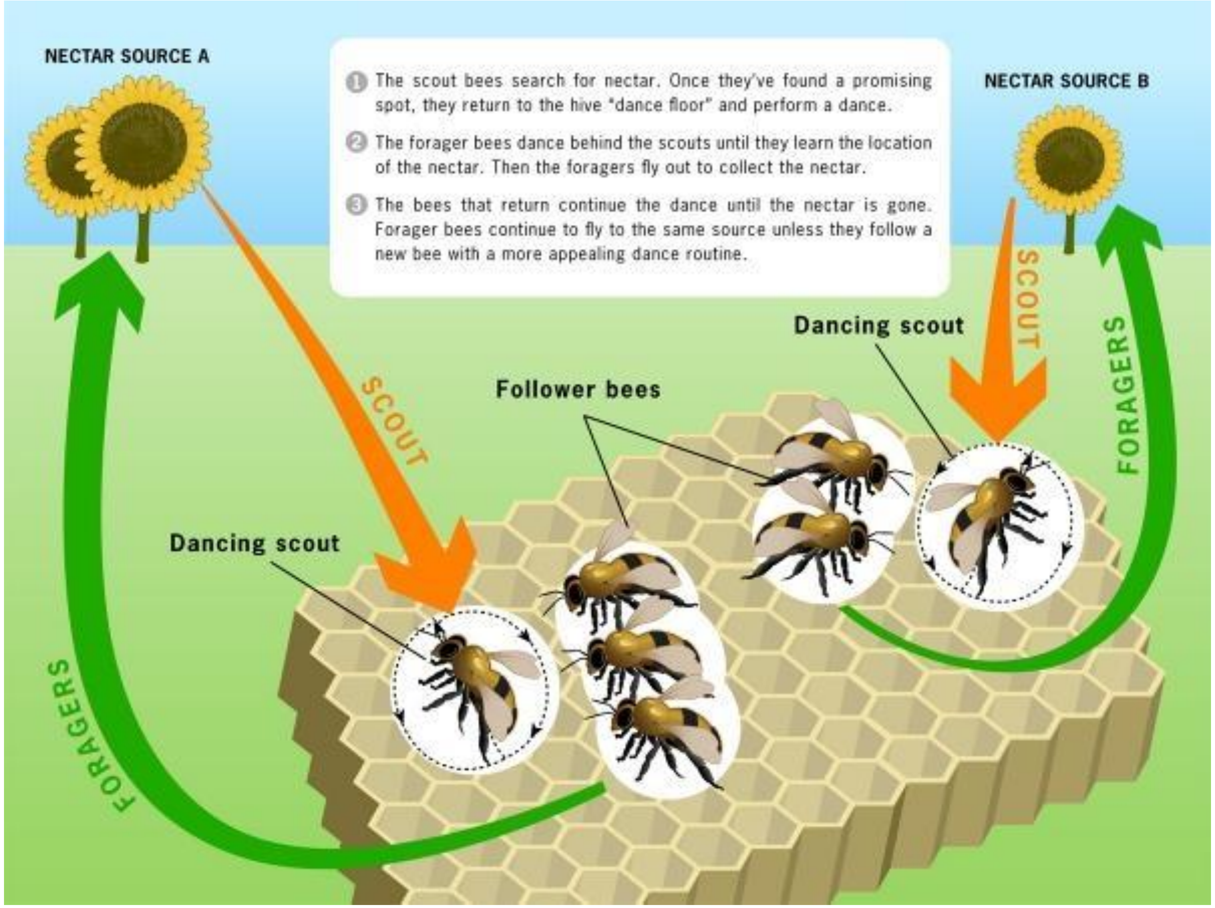
- Simulating the foraging behaviours of honey bees
- Introduced by Karaboga in 2005 (Karaboga, 2005)

Picture from website https://www.shortlist.com/food-drink/these-gifs-show-you-what-would-happen-to-your-food-if-all-the-bees-died/67594

Artificial bee colony algorithm (ABC)

- Ideas and concepts
  - Nectar source: candidate solution
  - Bees
    - Employed (forager) bees: each forager bee stores the information of a nectar source, and dances to share the information
    - Scouter (spy) bees: search for new nectar sources
    - Onlooker (follower) bees: each onlooker bee watches the dances of foragers, chooses one source and goes to the source, then search and evaluate the neighbour area of the source

Artificial bee colony algorithm (ABC)



Picture from website http://thebridge.psgtech.ac.in/?p=717

Thank you!