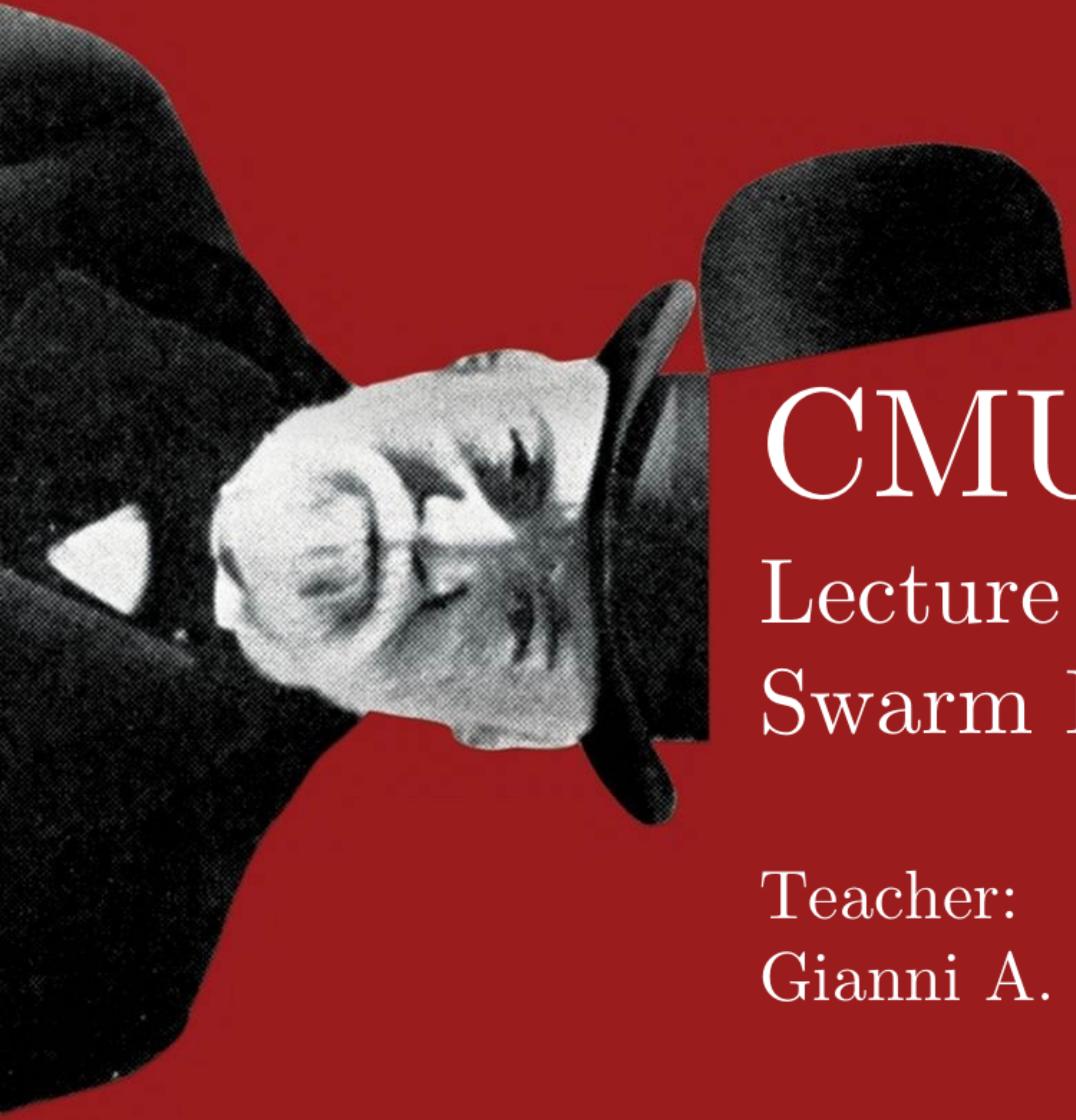


CMU 15-781

Lecture 26: Swarm Intelligence II

Teacher:
Gianni A. Di Caro

PARTICLE SWARM OPTIMIZATION (PSO)

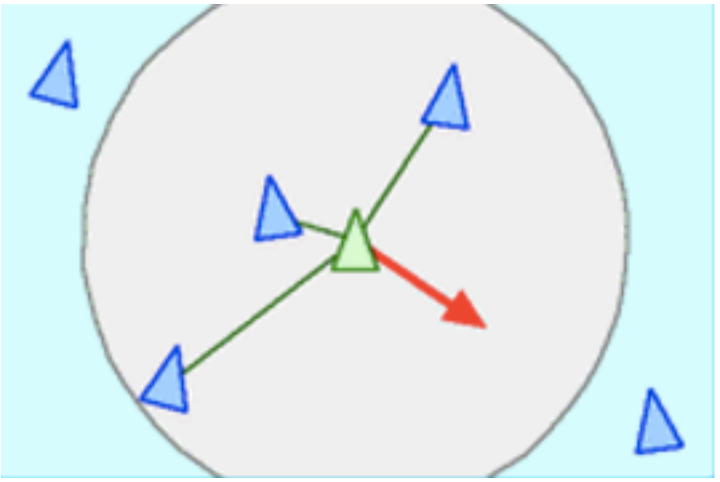
J. Kennedy and R. Eberhart, Particle Swarm Optimization. Proceedings of the Fourth IEEE Int. Conference on Neural Networks, 1995.

- **A population based optimization technique** inspired by **social behavior of bird flocking/roosting** or fish schooling
- A PSO swarm member/agent (a particle) iteratively modifies *a complete solution*

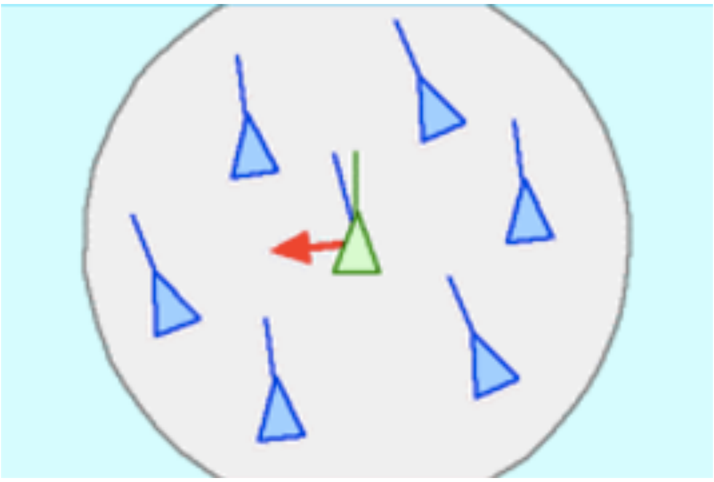
Individual swarm members establish a *social network* and can profit from the discoveries and previous experience of the other members of the swarm

BACKGROUND: REYNOLDS' BOIDS

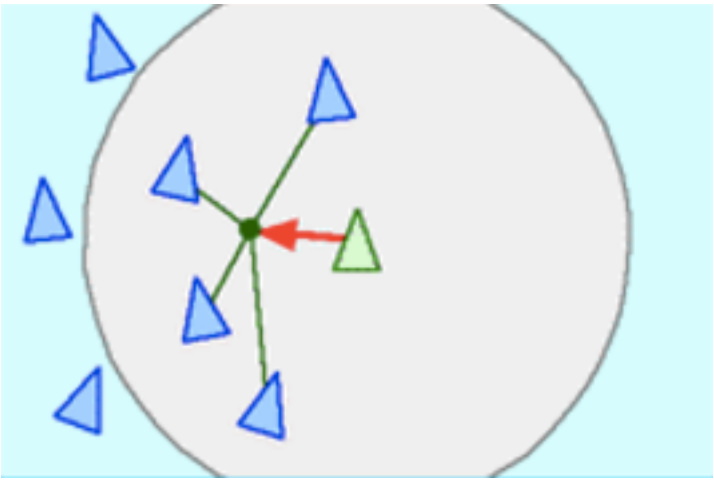
Reynolds created a model of **coordinated animal motion** in which the agents (boids) obeyed **three simple local rules:**



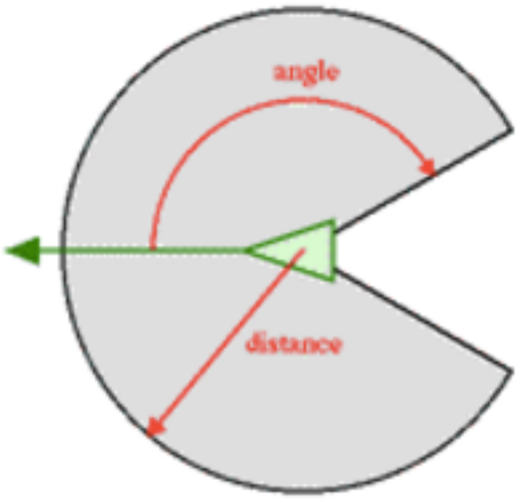
Separation: steer to avoid crowding local flockmates



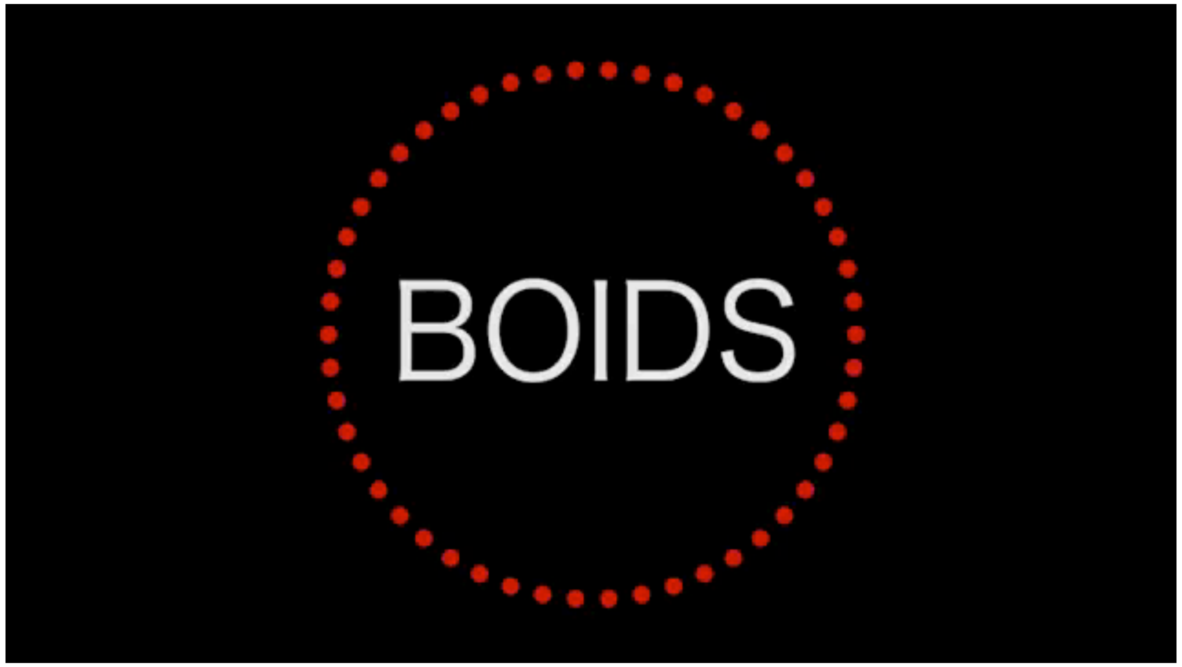
Alignment: steer towards the average heading of local flockmates



Cohesion: steer to move toward the average position of local flockmates



Reynolds, C.W.: Flocks, herds and schools: a distributed behavioral model. Computer Graphics, 21(4), p.25-34, 1987

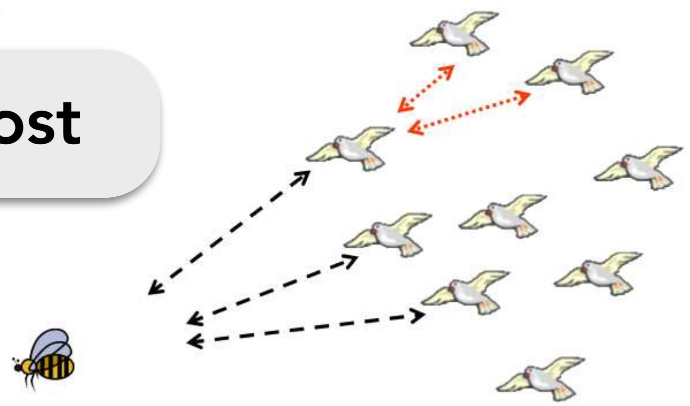


https://www.youtube.com/watch?v=QbUPfMXXQIY

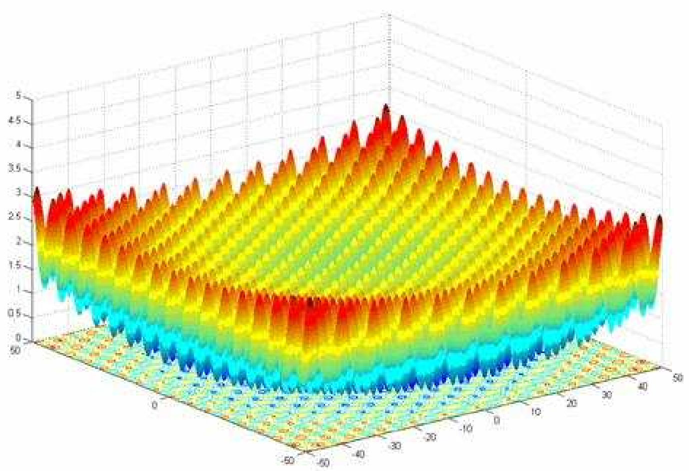
BACKGROUND: ROOST

Kennedy and Eberhart included a **roost** (attraction point) in a simplified Boids-like simulation, such that each agent:

- is attracted to the location of the roost,
- remembers where it was closer to the roost,
- shares information with its neighbors about its closest location to the roost

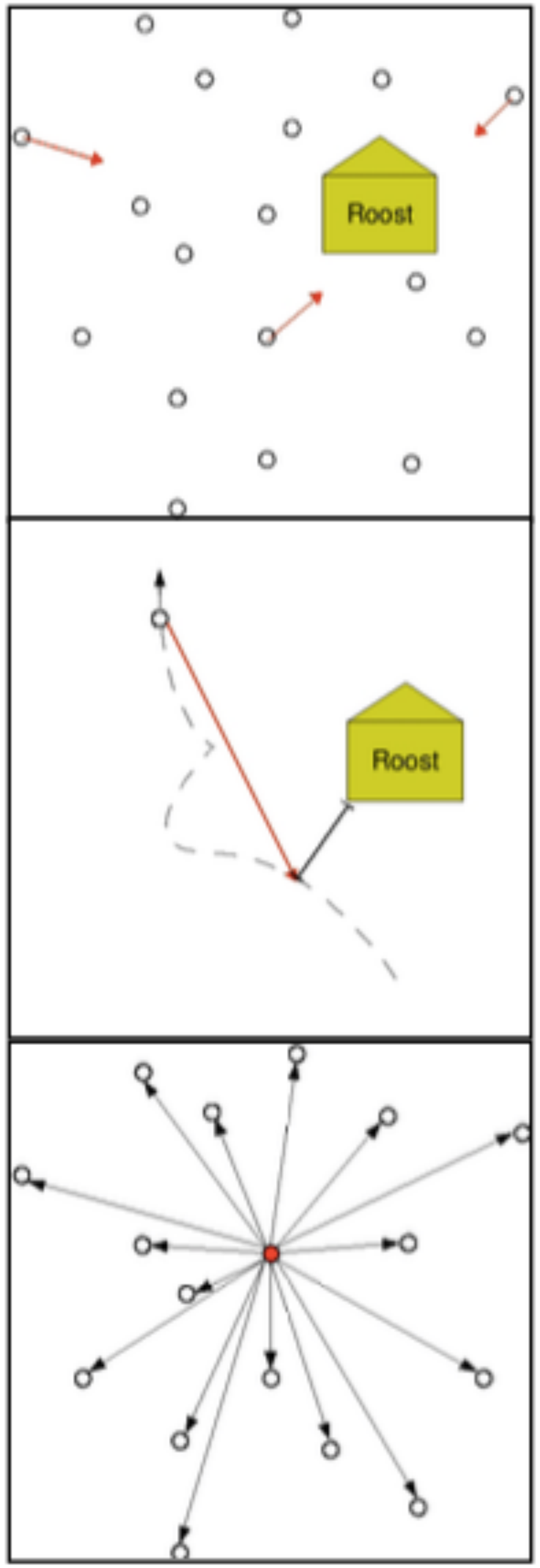


Eventually, all agents land on the roost



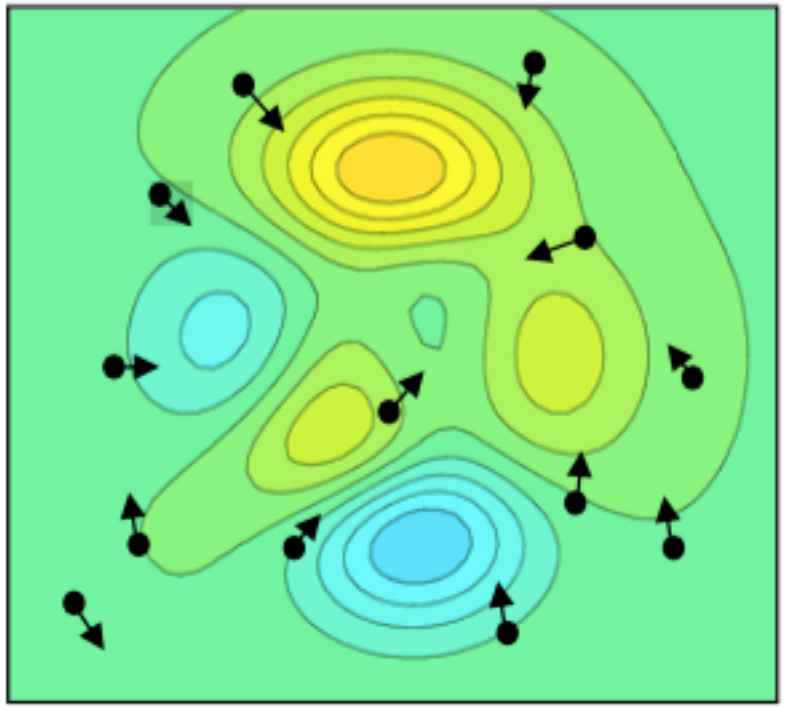
What if:

- roost = (unknown) **extremum of a function**
- distance to the roost = **quality of current agent position**



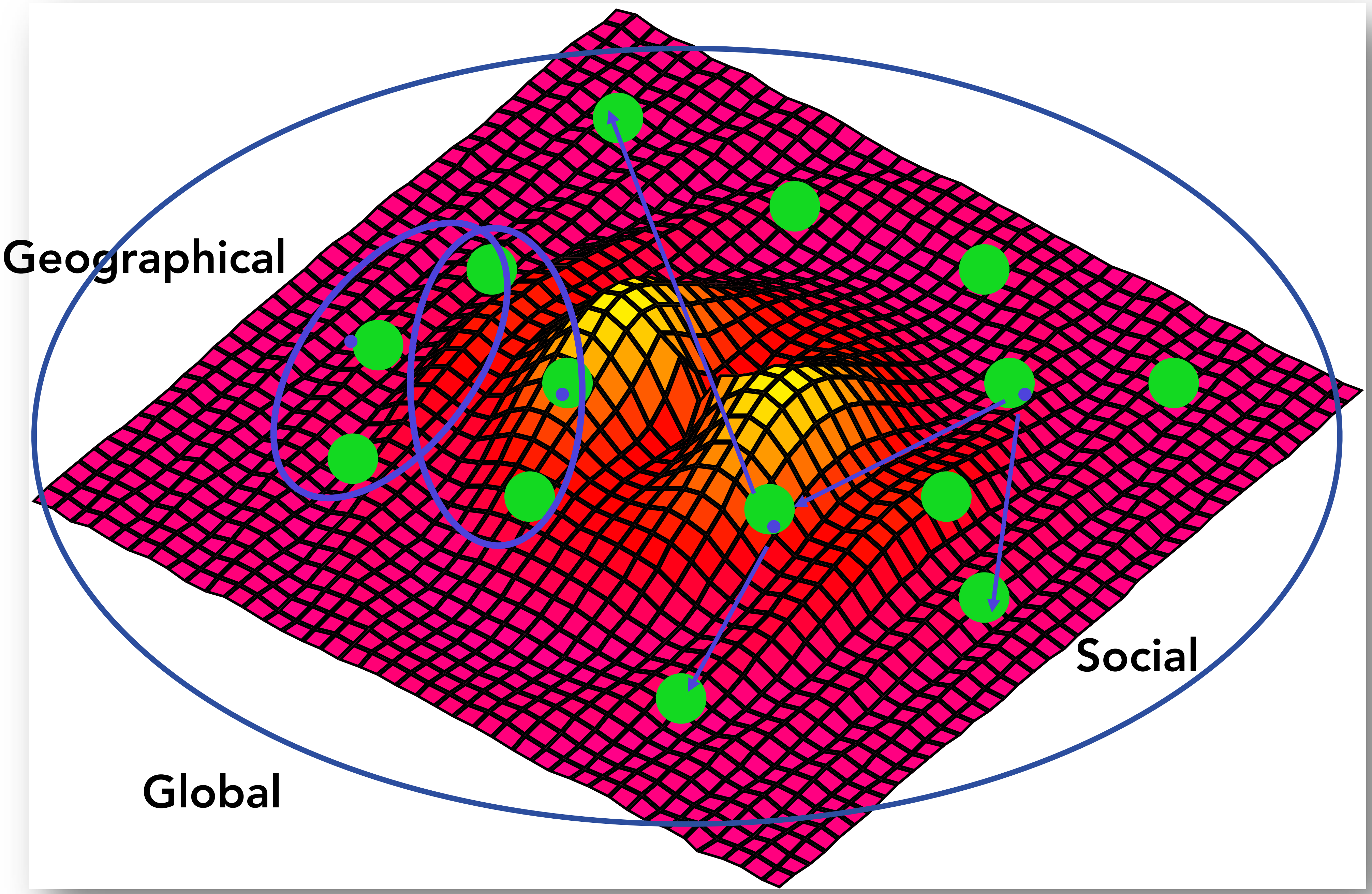
PARTICLE SWARM OPTIMIZATION (PSO)

- PSO consists of a swarm of bird-like **particles**
- Each particle resides at a **position in the search space**
- The *fitness* of each particle represents the **quality of its position**
- The particles *move* over the search space with a certain **velocity**
- Each particle has an **internal state** + network of **social connections**
- The *velocity* (both *direction* and *speed*) of each particle is influenced by its **own best position found so far, *pbest*,** the **best solution that was found so far by its social neighbors, *lbest*,** and/or **the global best so far *gbest***
- "Eventually" the swarm will *converge* to optimal positions



$$\{\vec{x}, \vec{v}, \vec{x}_{pbest}, \mathcal{N}(p)\}$$

NEIGHBORHOODS



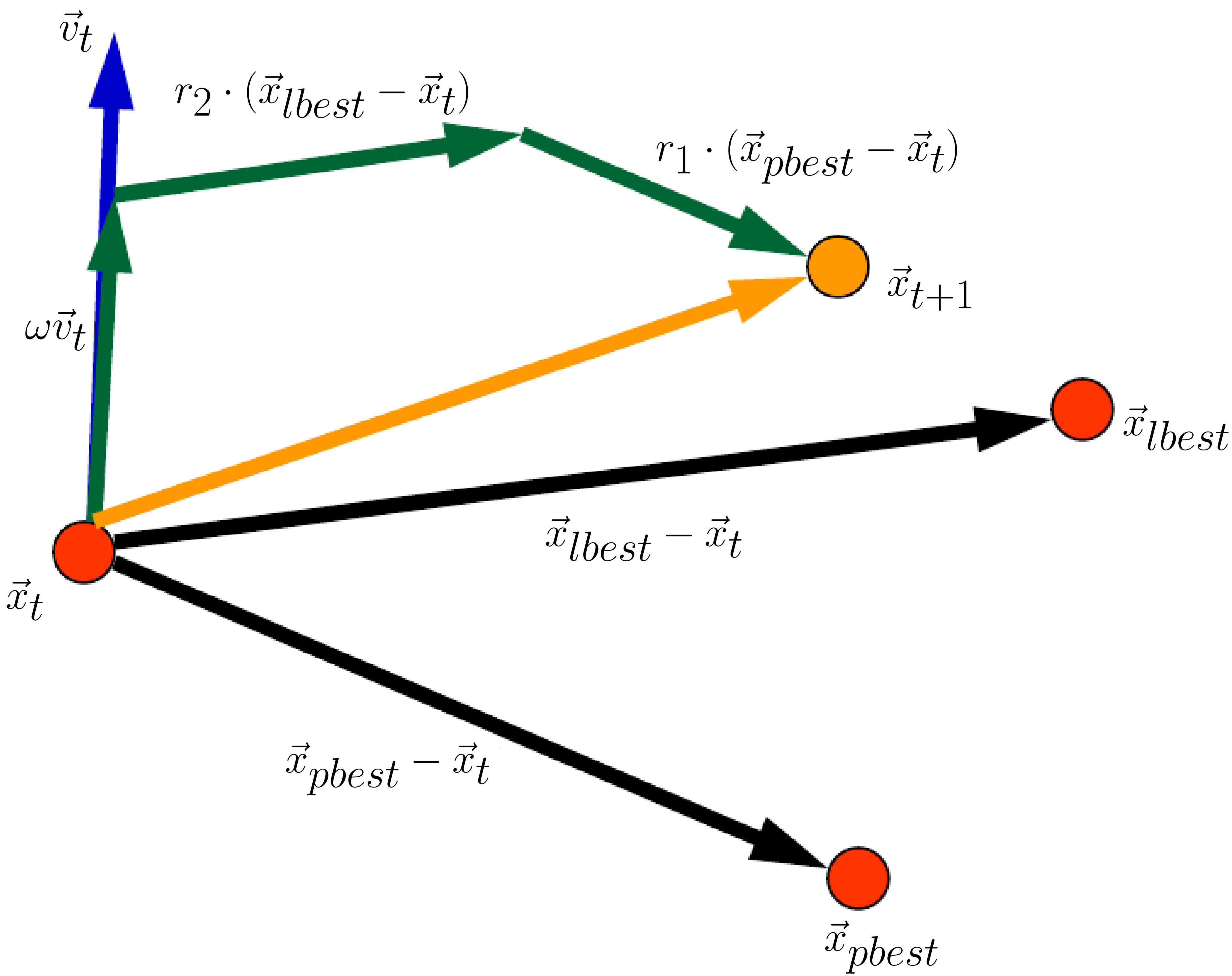
PARTICLE SWARM OPTIMIZATION (PSO)

procedure Particle_Swarm_Optimization_for_Minimization($f(x)$)
  foreach *particle* $p \in$ *ParticleSet* **do**
    $(\vec{x}, \vec{v}) \leftarrow$ init_positions_and_velocity();
    $\mathcal{N}(p) \leftarrow$ selection_of_the_ neighbor_set();
    $\vec{x}_{pbest} \leftarrow \vec{x}$; $\vec{x}_{gbest} = \infty$; / * *init personal and global best positions* * /
  end foreach
  while ($\neg$ *stopping_criterion*)
    foreach *particle* $p \in$ *ParticleSet* **do**
      $\vec{x}_{pbest} \leftarrow \arg\max\,(f(\vec{x}), f(\vec{x}_{pbest}))$;
      $\vec{x}_{lbest} \leftarrow$ get_best_so_far_position_from_neighbors($\mathcal{N}(p)$);
      $\vec{\Delta}_{individual} \leftarrow \vec{x}_{pbest} - \vec{x}$;
      $\vec{\Delta}_{social} \leftarrow \vec{x}_{lbest} - \vec{x}$;
      $(\vec{r}_1, \vec{r}_2) \leftarrow$ random_uniform();
      $\vec{v} \leftarrow \omega\vec{v} + w_1\vec{r}_1 \circ \vec{\Delta}_{individual} + w_2\vec{r}_2 \circ \vec{\Delta}_{social}$;   element-wise multiplication operator
      $\vec{x} \leftarrow \vec{x} + \vec{v}$;
      if $f(\vec{x}) < f(\vec{x}_{pbest})$
        $\vec{x}_{pbest} \leftarrow \vec{x}$;
      if $f(\vec{x}) < f(\vec{x}_{gbest})$
        $\vec{x}_{gbest} \leftarrow \vec{x}$;
    end foreach
  end while
return $f(\vec{x}_{gbest})$;

$$\vec{r}_1 = U(0, \phi_1) \quad \vec{r}_2 = U(0, \phi_2)$$

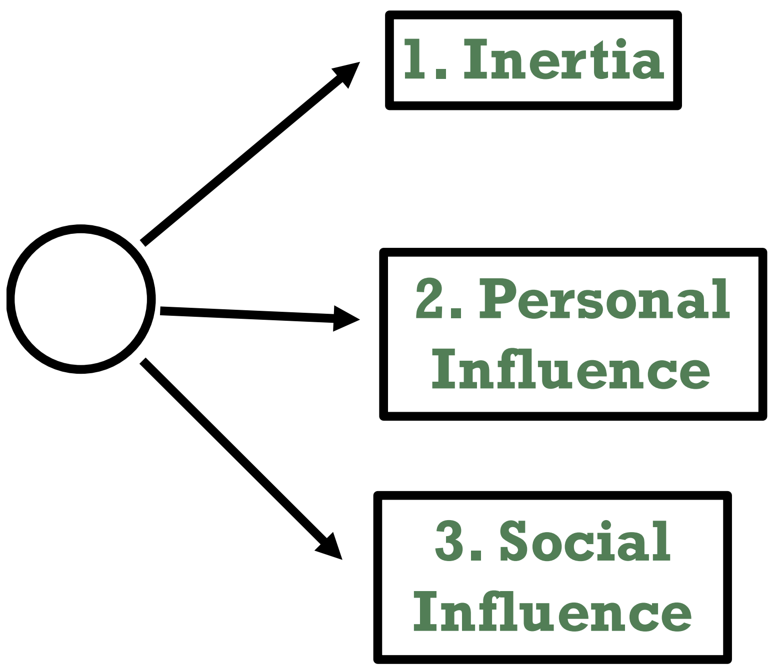
ϕ are **acceleration coefficients** determining scale of forces in the direction of individual and social biases

VECTOR COMBINATION OF MULTIPLE BIASES



VECTOR COMBINATION OF MULTIPLE BIASES

$$\mathbf{v}_i^{t+1} = \underbrace{\mathbf{v}_i^t}_{inertia} + \underbrace{c_1 \mathbf{U}_1^t (\mathbf{pb}_i^t - \mathbf{p}_i^t)}_{personal\ influence} + \underbrace{c_2 \mathbf{U}_2^t (\mathbf{gb}^t - \mathbf{p}_i^t)}_{social\ influence}$$



- Makes the particle move in the same direction and with the same velocity

- Improves the individual
- Makes the particle return to a previous position, better than the current
- Conservative

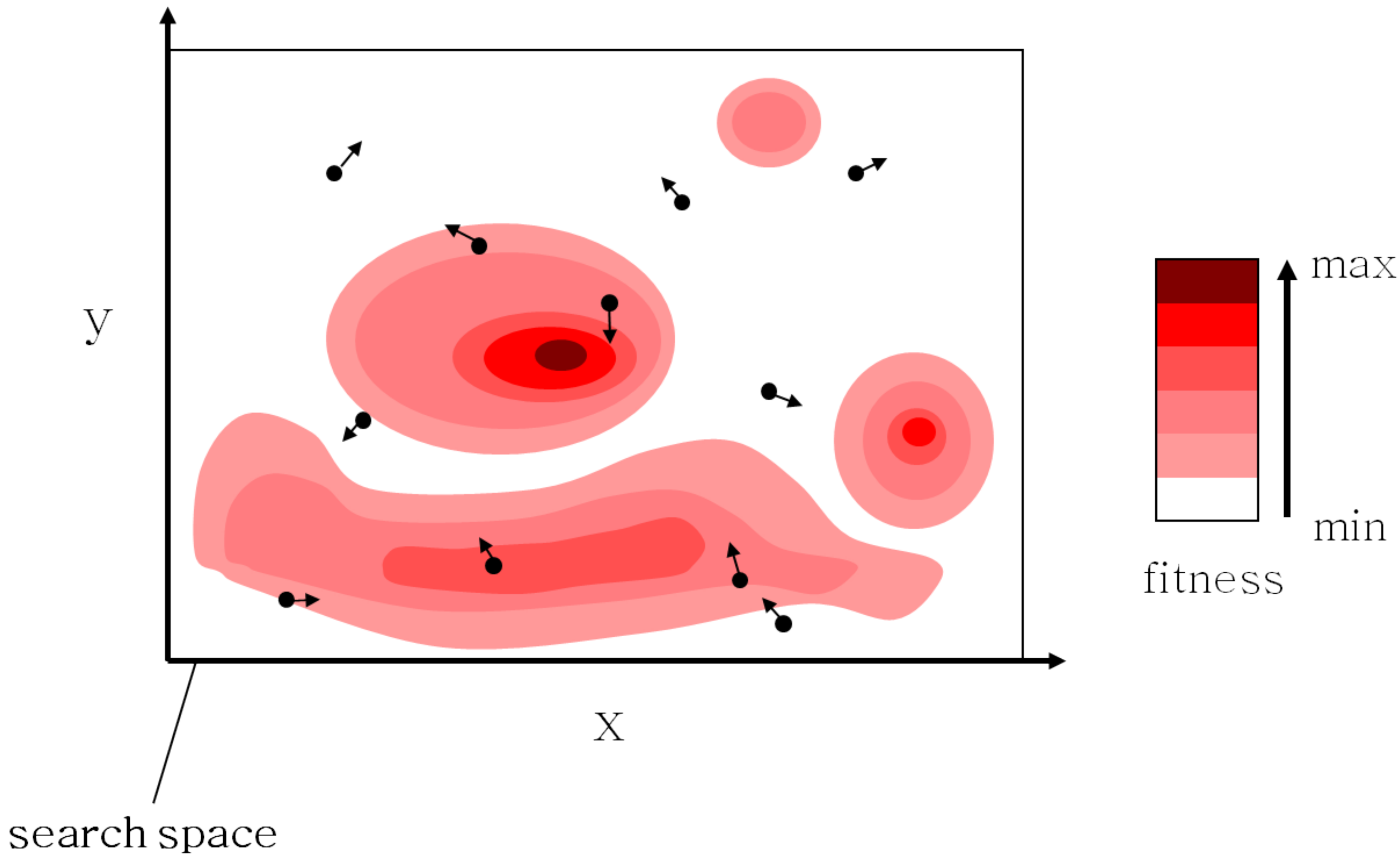
- Makes the particle follow the best neighbors direction

Search for new solutions

$$\mathbf{v}_i^{t+1} = \underbrace{\mathbf{v}_i^t}_{\textbf{Diversification}} + \underbrace{c_1 \mathbf{U}_1^t (\mathbf{pb}_i^t - \mathbf{p}_i^t) + c_2 \mathbf{U}_2^t (\mathbf{gb}^t - \mathbf{p}_i^t)}_{\textbf{Intensification}}$$

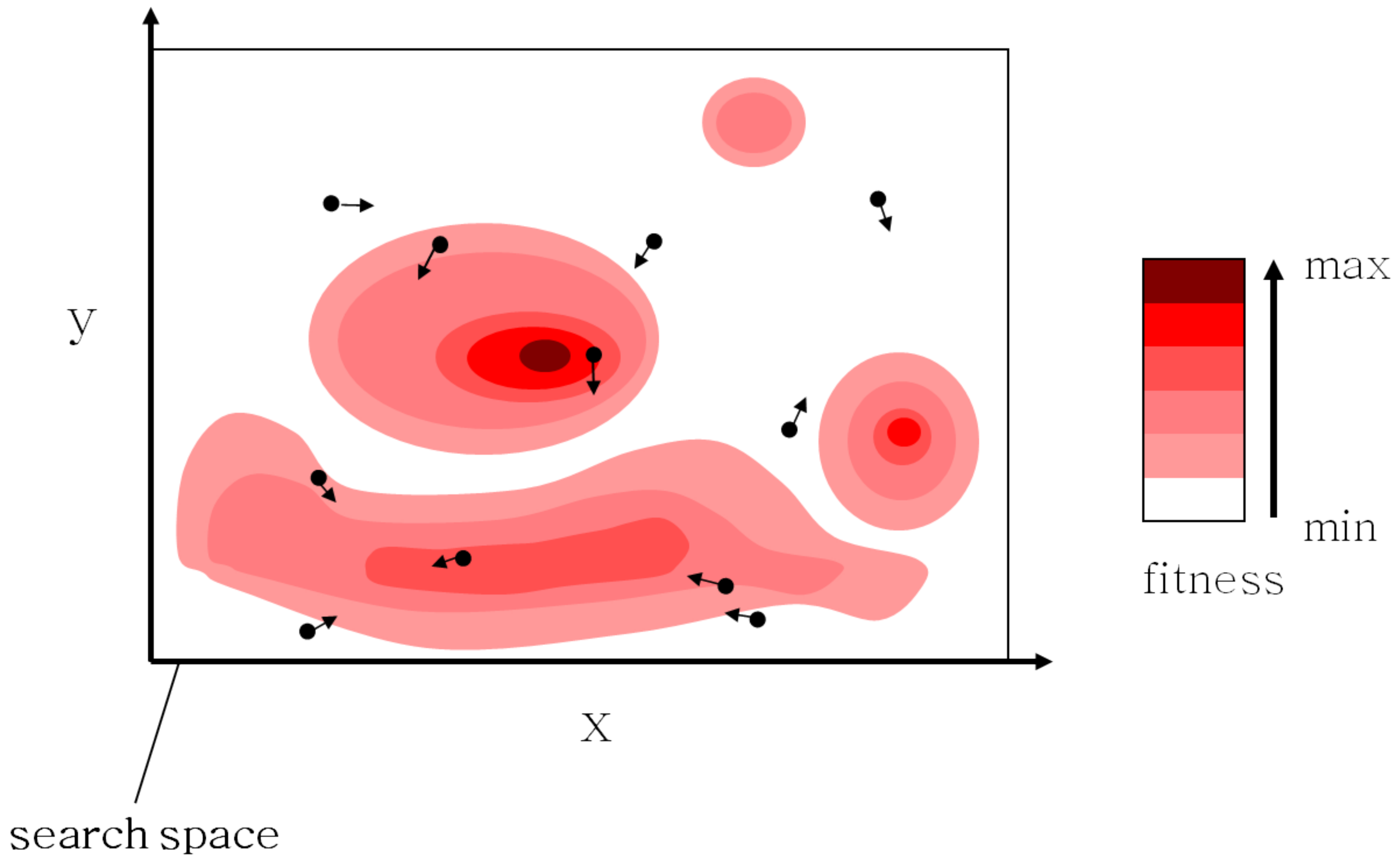
Exploits what good so far

PSO AT WORK (MAX OPTIMIZATION PROBLEM)

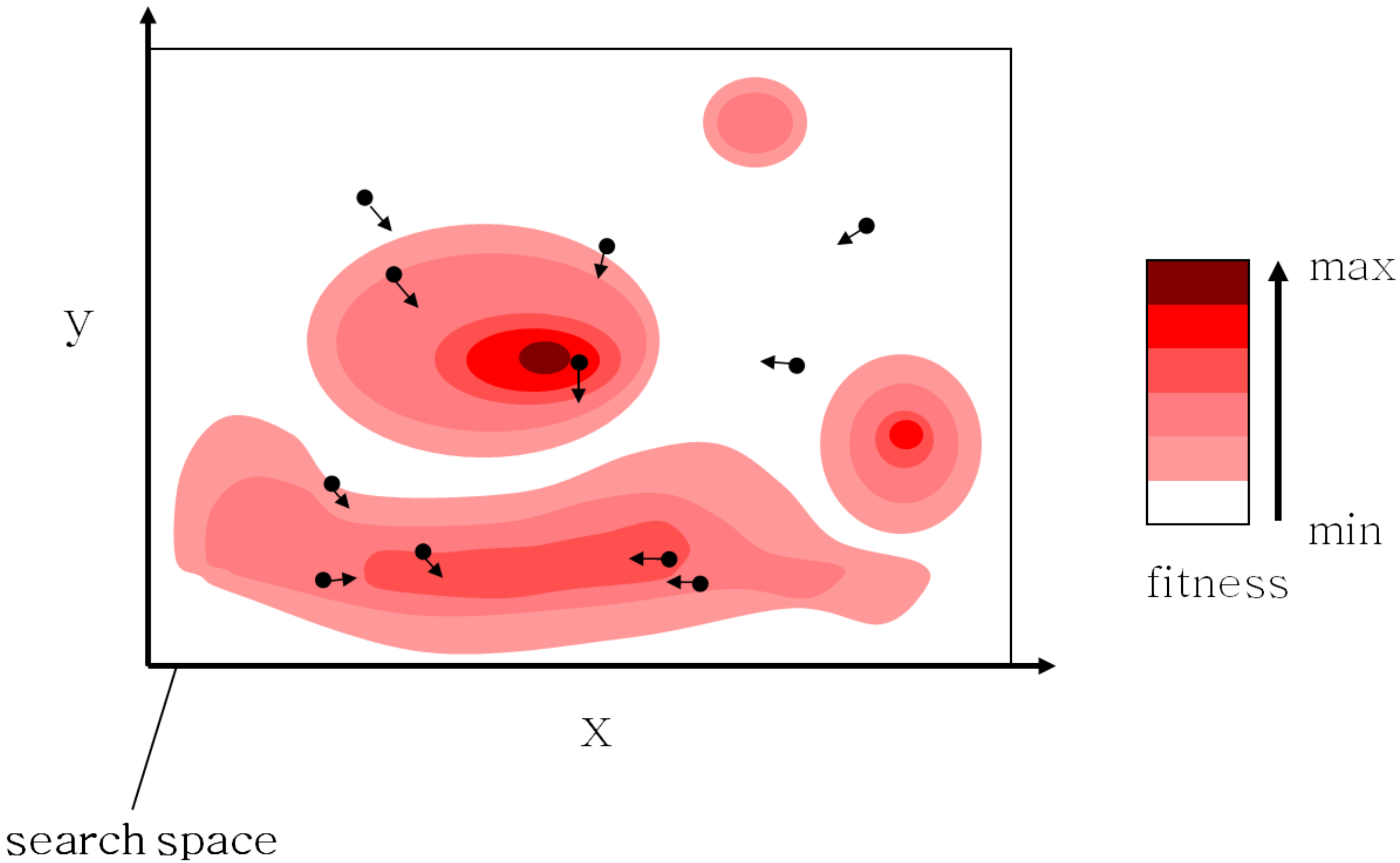


Example slides from Pinto et al.

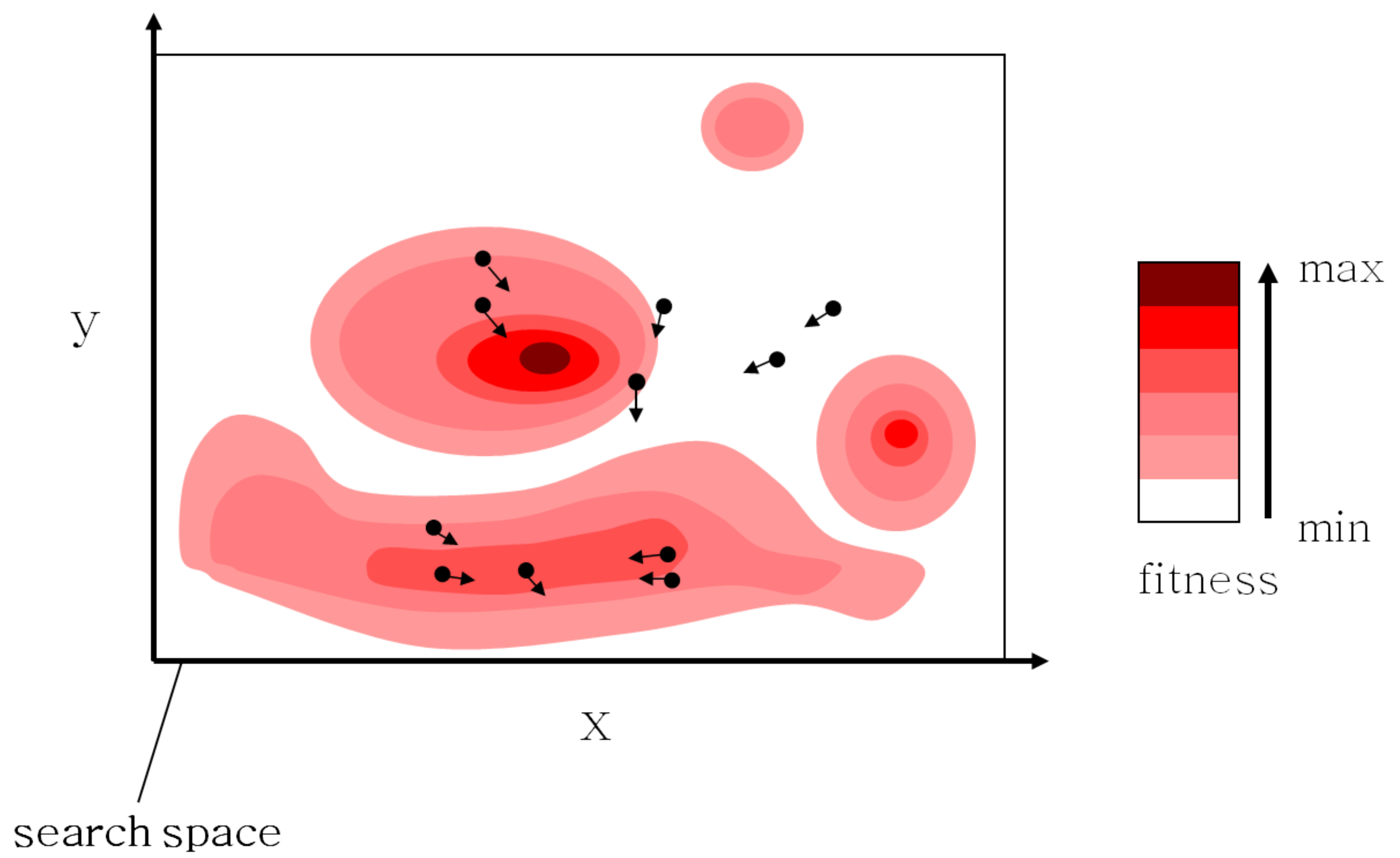
PSO AT WORK



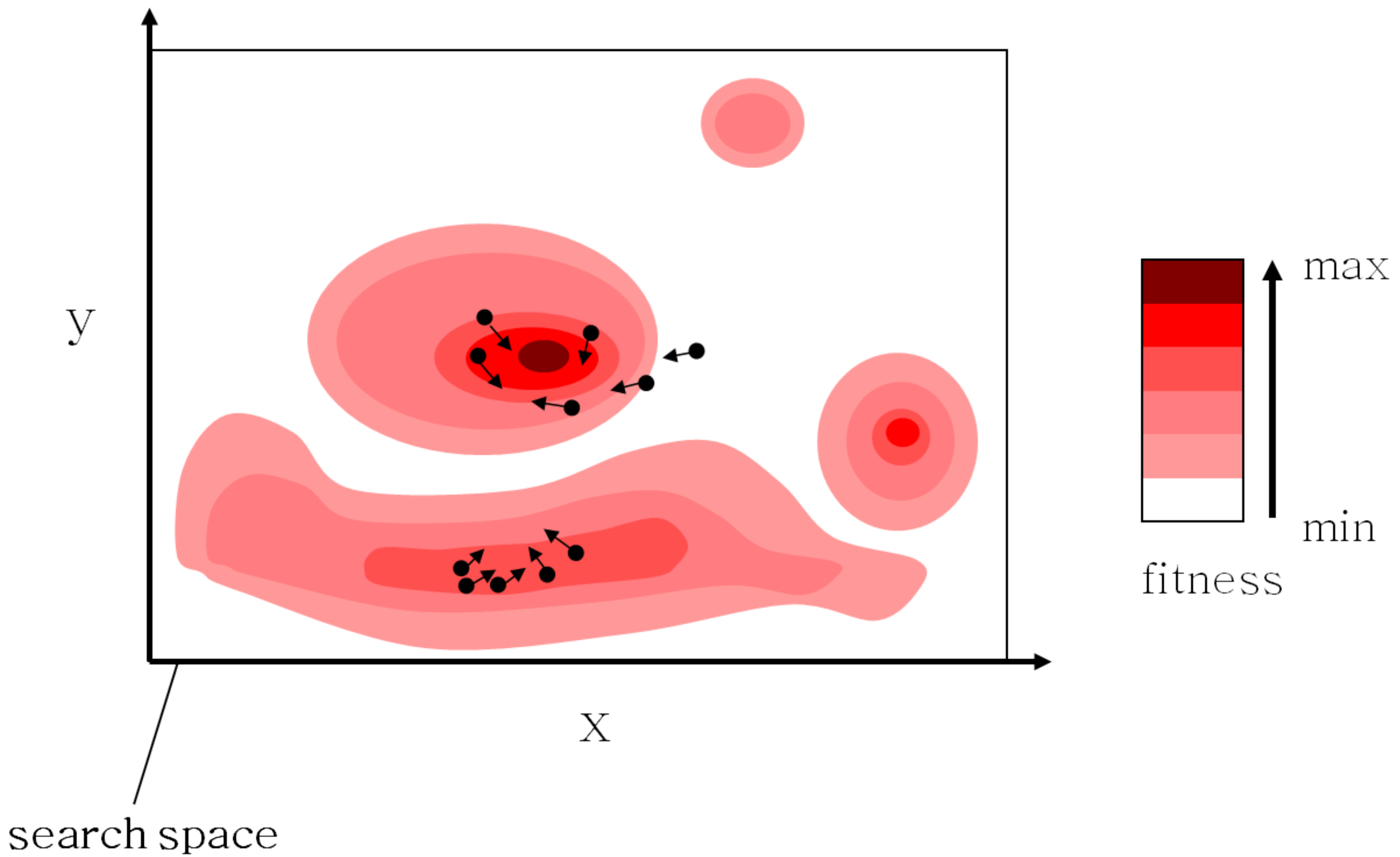
PSO AT WORK



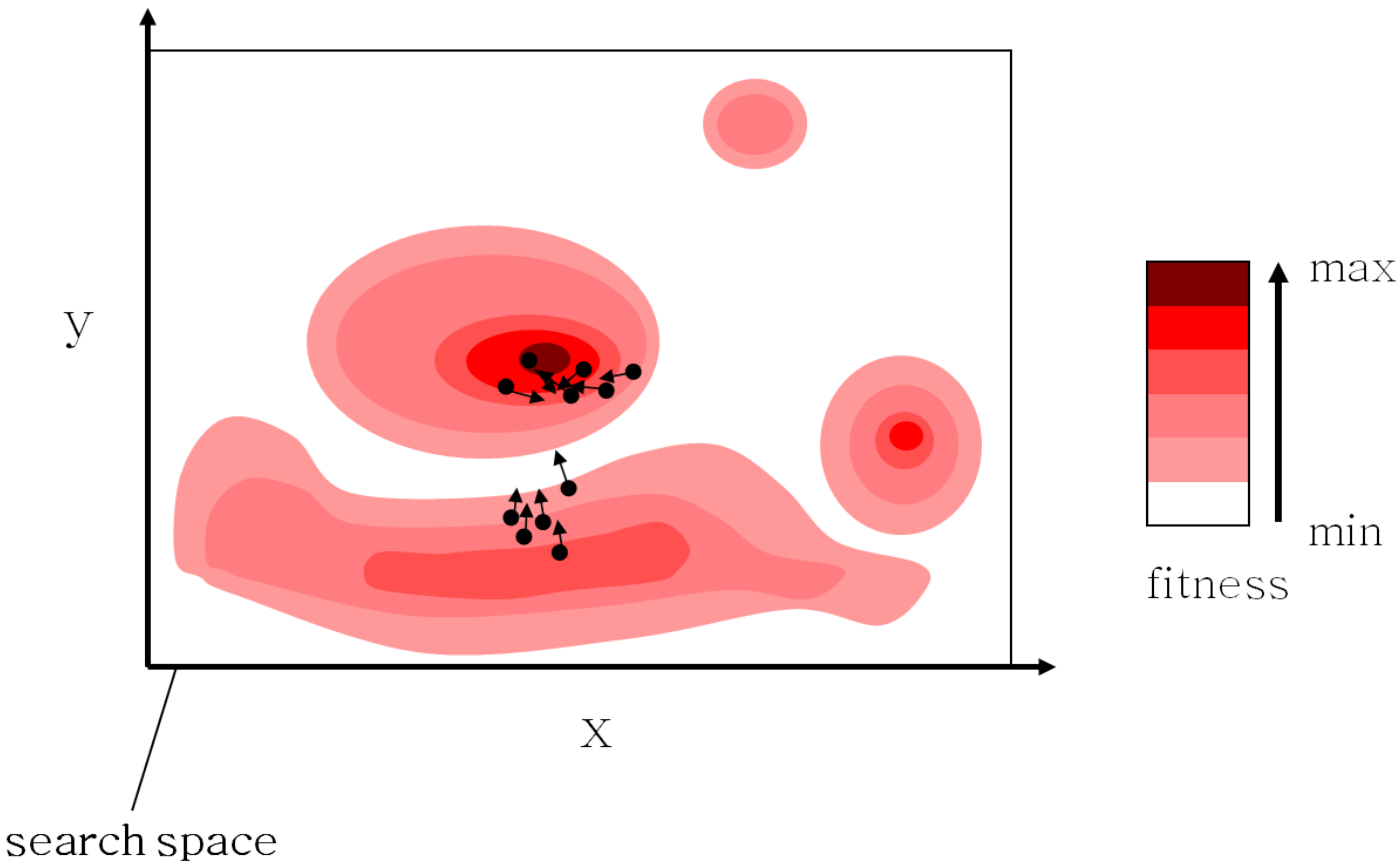
PSO AT WORK



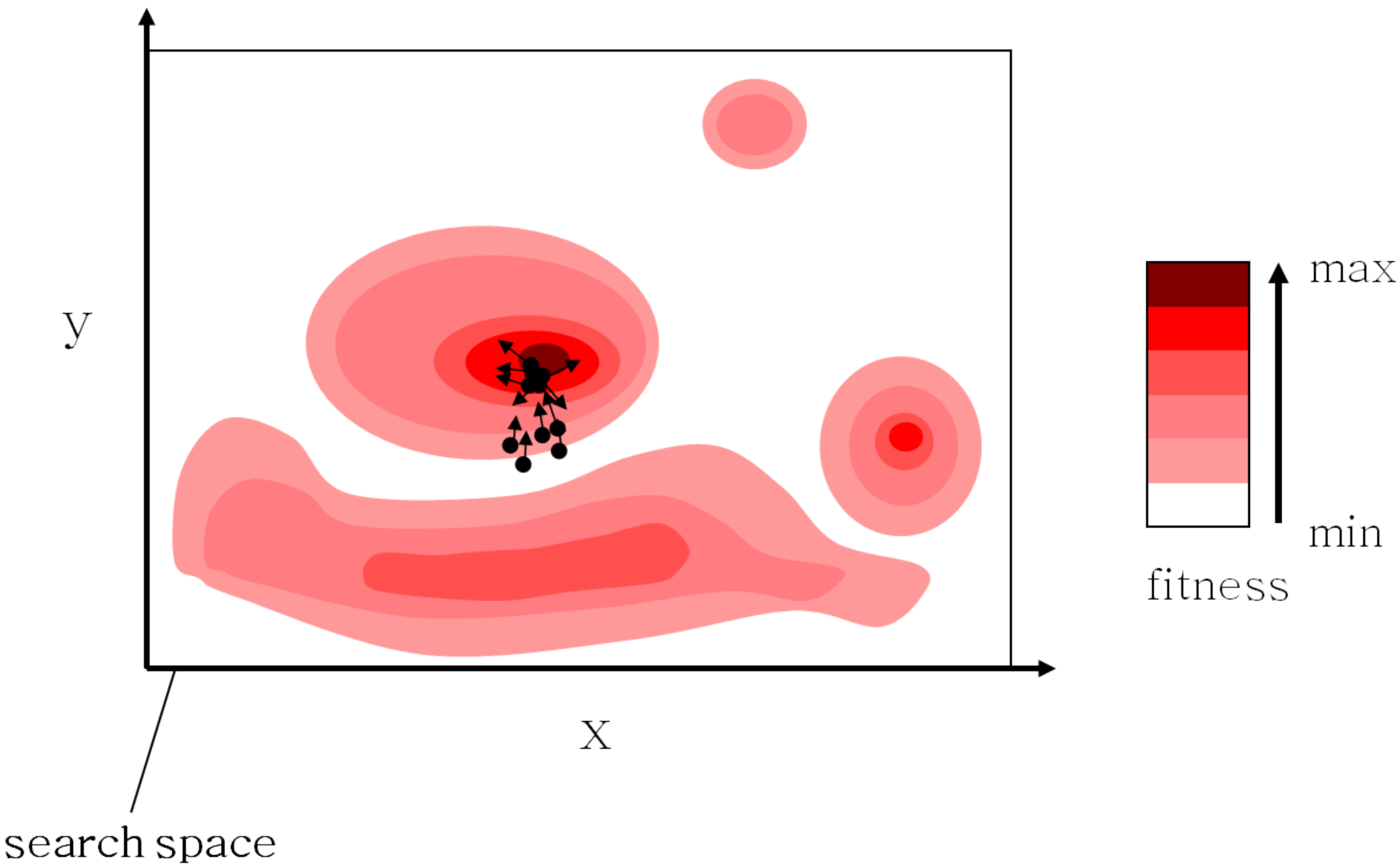
PSO AT WORK



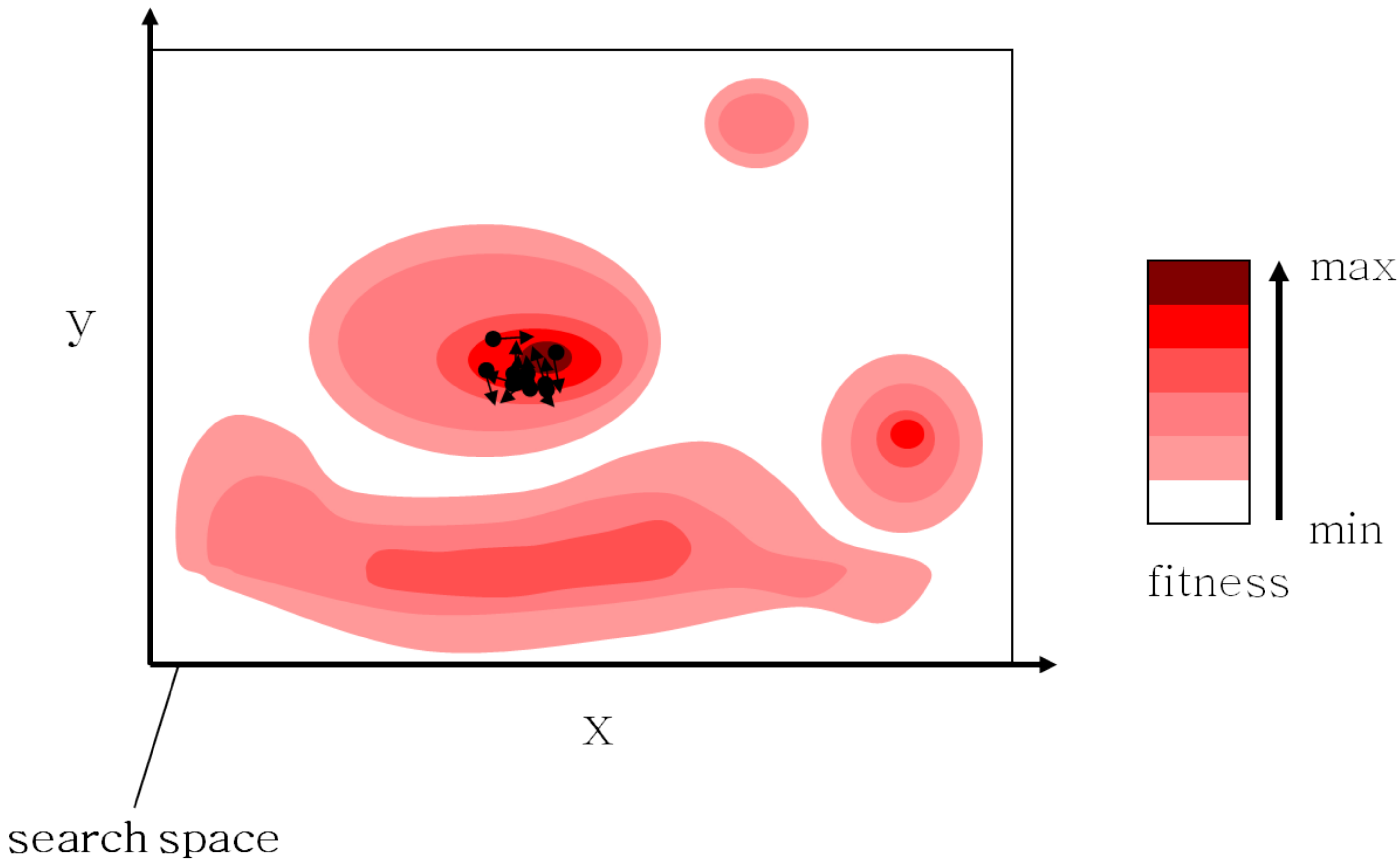
PSO AT WORK



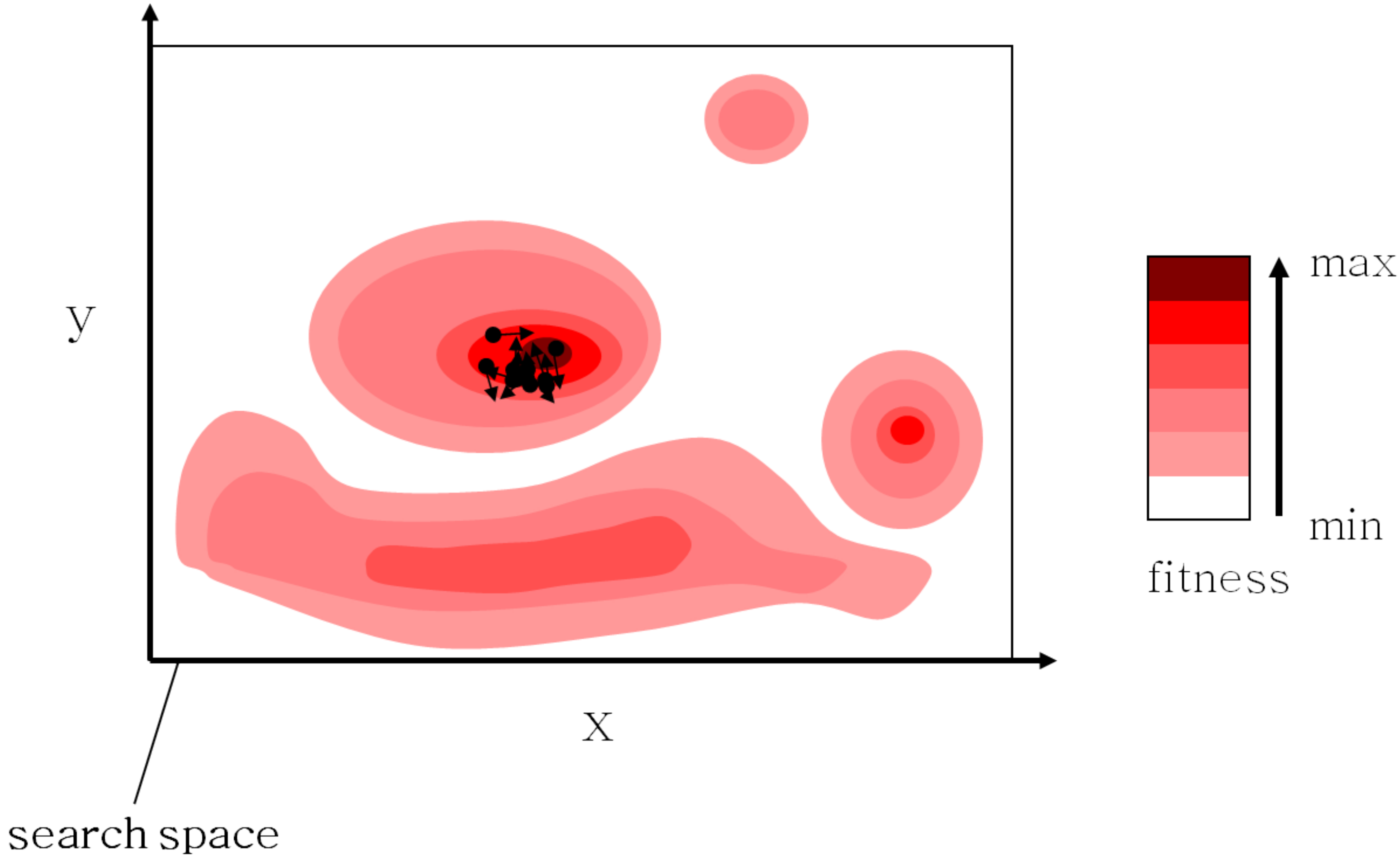
PSO AT WORK



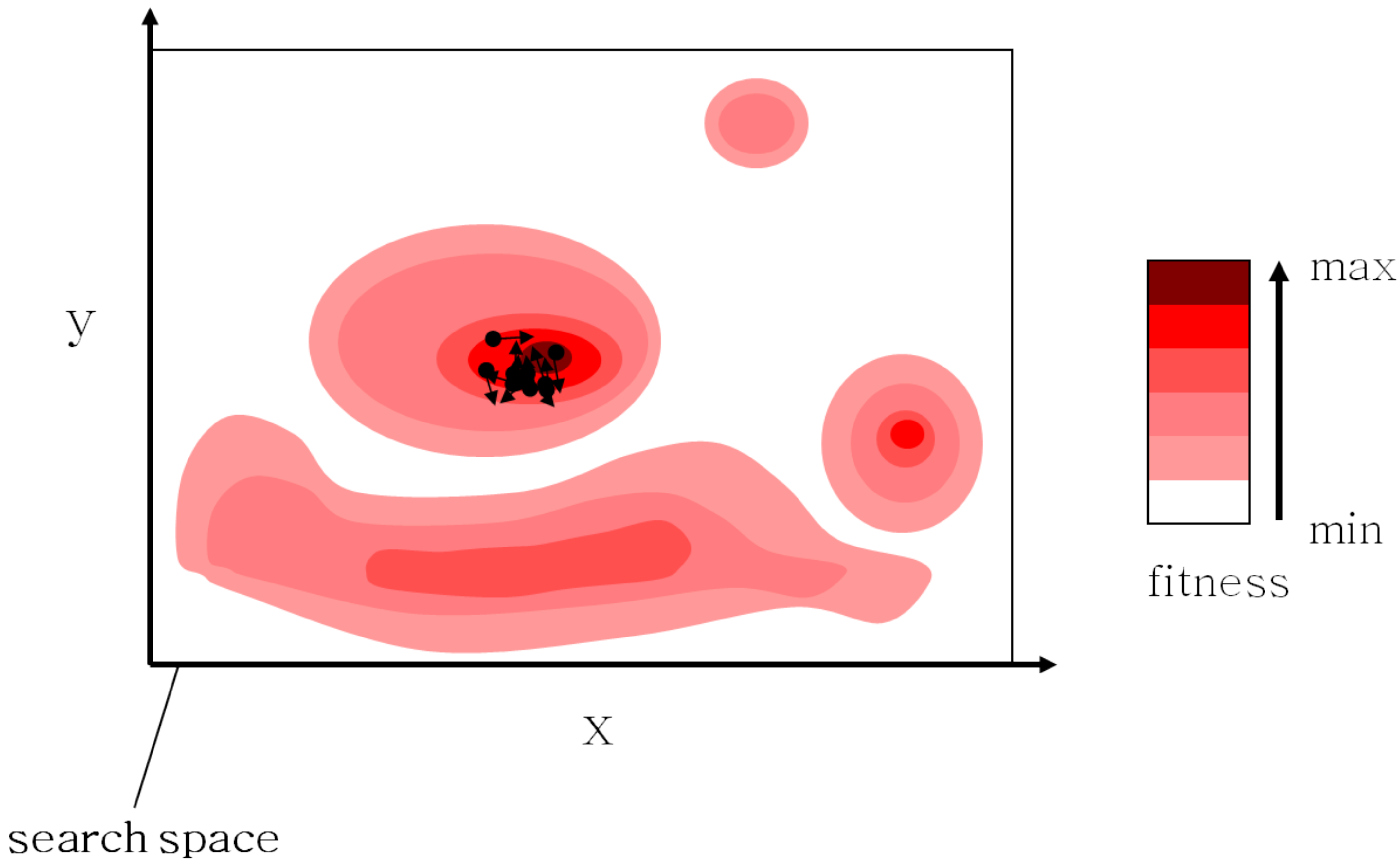
PSO AT WORK



PSO AT WORK



PSO AT WORK



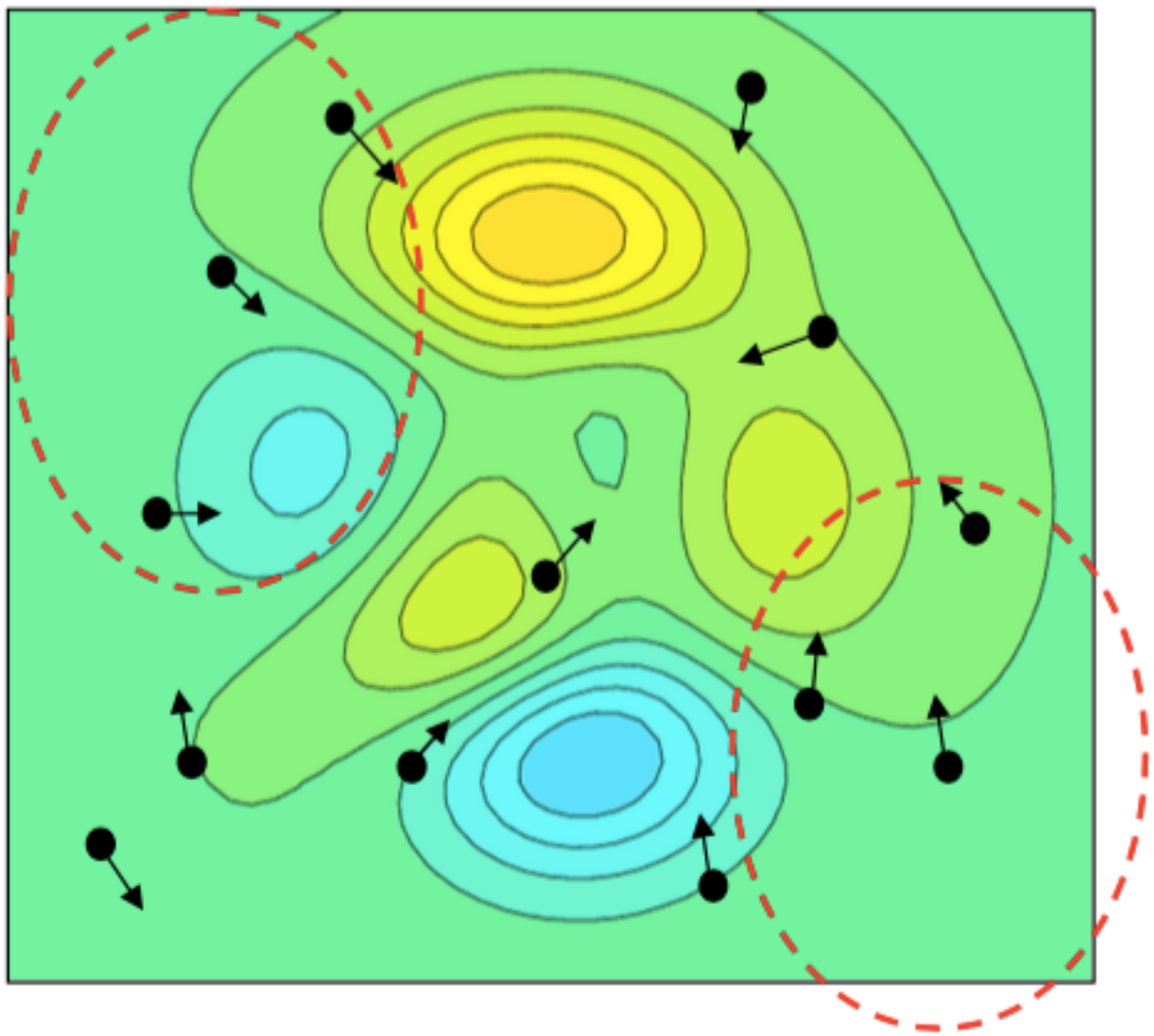
PSO VS. ACO

- Birds flocking/roosting vs ant pheromone laying/following
- Iterative solution modification vs. Repeated solution construction
- Social network of point-to-point information exchange vs. Stigmergy, environment-mediated communications
- Both are based on the use of a population of solutions
- Both sample the solution space and are global optimizers
- Both are quite straightforward to implement in parallel / distributed architectures
- Both are relatively simple to implement and perform at the SoA

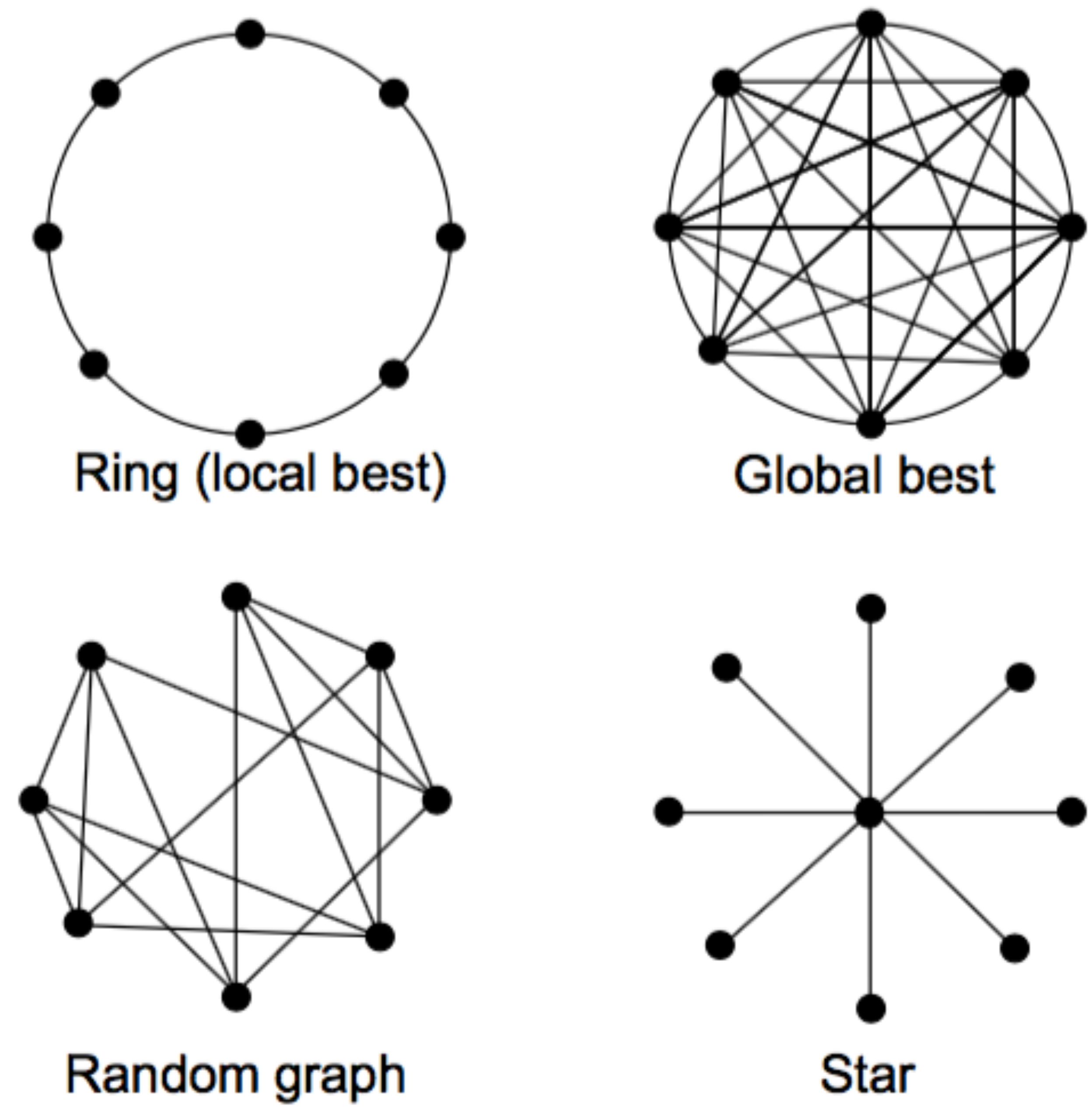
GOOD AND BAD POINTS OF BASIC PSO

- **Advantages**
  - Quite insensitive to scaling of design variables
  - Simple implementation
  - Easily parallelized for concurrent processing
  - Derivative free
  - Very few algorithm parameters
  - Very efficient global search algorithm
- **Disadvantages**
  - Tendency to a fast and premature convergence in mid optimum points
  - Slow convergence in refined search stage (weak local search ability)

GOOD NEIGHBORHOOD TOPOLOGY?



Geographical neighborhoods



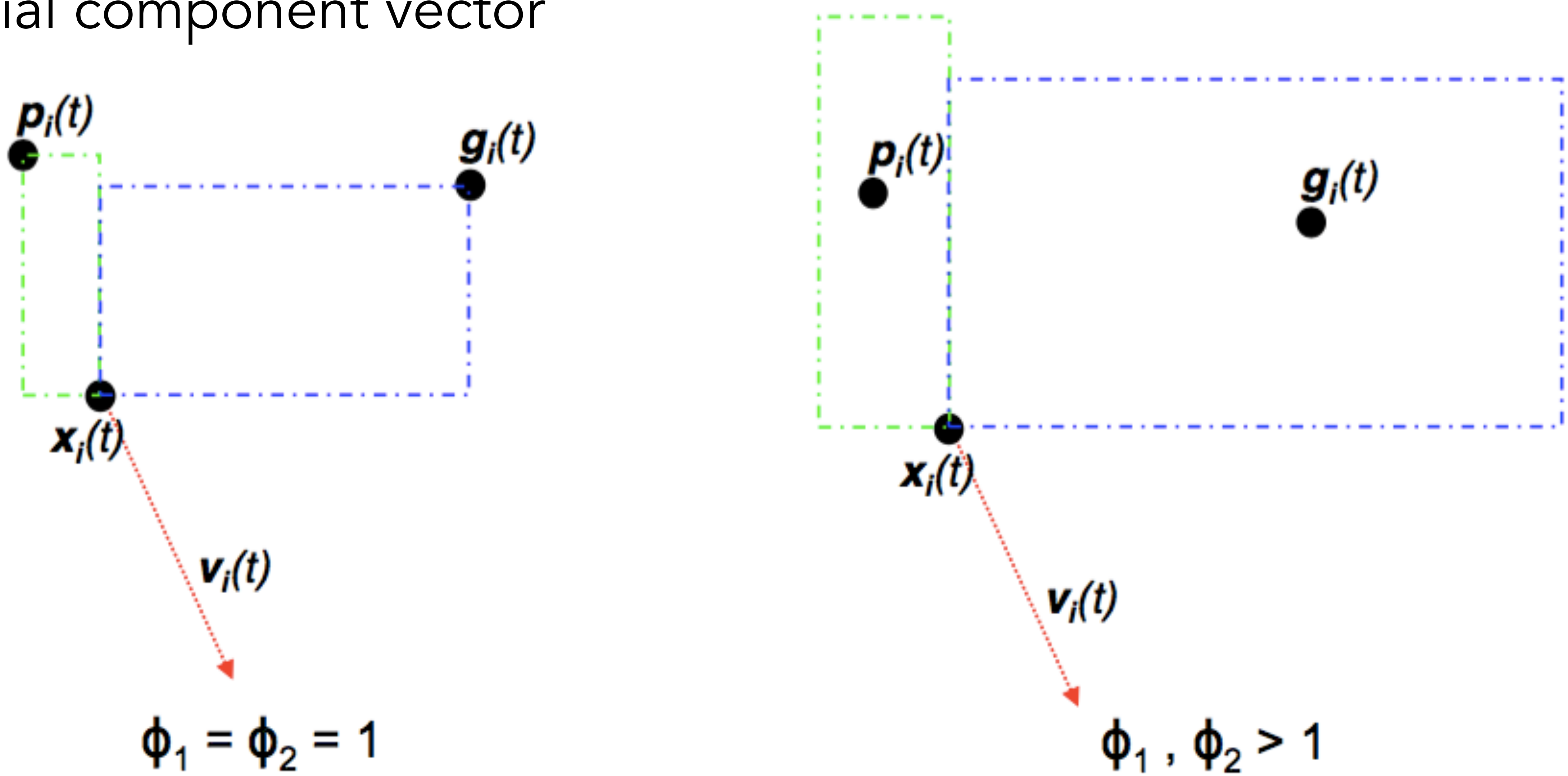
Communication network topologies

GOOD NEIGHBORHOOD TOPOLOGY?

- Also considered were:
  - Clustering topologies (islands)
  - Dynamic topologies
  - …
- No clear way of saying which topology is the best
- **Exploration / exploitation dilemma**
- Some neighborhood topologies are better for local search others for global search
  - *lbest* neighborhood topologies seems better for global search,
  - *gbest* topologies seem better for local search

ACCELERATION COEFFICIENTS

- The boxes show the distribution of the random vectors of the attracting forces of the *local best* and *global best*
- The acceleration coefficients determine the **scale distribution** of the random individual (cognitive) component vector and the social component vector

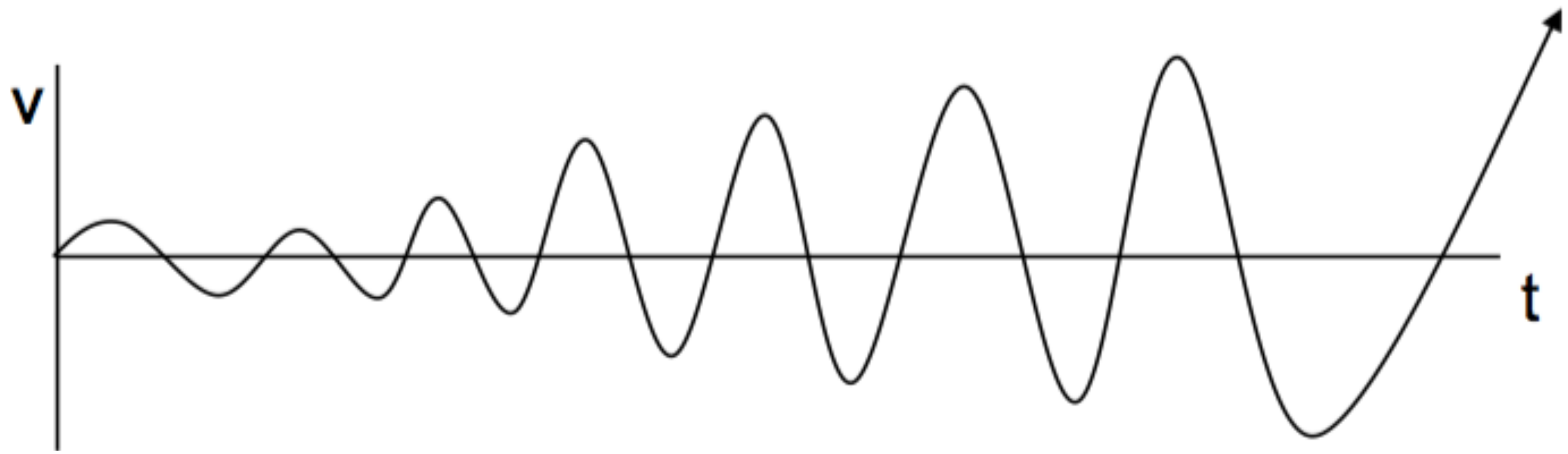


ACCELERATION COEFFICIENTS

- $\phi_1 > 0$, $\phi_2 = 0$ particles are independent hill-climbers
- $\phi_1 = 0$, $\phi_2 > 0$ swarm is one stochastic hill-climber
- $\phi_1 = \phi_2 > 0$ particles are attracted to the average of p_i and g_i
- $\phi_2 > \phi_1$ more beneficial for unimodal problems
- $\phi_1 > \phi_2$ more beneficial for multimodal problems
- low ϕ_1, ϕ_2 smooth particle trajectories
- high ϕ_1, ϕ_2 more acceleration, abrupt movements
- Adaptive acceleration coefficients have also been proposed, for example to have ϕ_1, ϕ_2 decreased over time (e.g., Simulated Annealing)

ORIGINAL PSO: ISSUES

- The acceleration coefficients should be set sufficiently high
- High acceleration coefficients result in **less stable systems** in which the velocity has a tendency to explode!



- To fix this, the velocity **v** is usually kept within the range $[-\mathbf{v}_{max}, \mathbf{v}_{max}]$
- However, limiting the velocity does not necessarily prevent particles from leaving the search space, nor does it help to guarantee convergence :(

INERTIA COEFFICIENT

- The **inertia weight** ω was introduced to control the velocity explosion

$$\vec{v} \leftarrow \omega \vec{v} + \vec{r}_1 \circ \vec{\Delta}_{individual} + \vec{r}_2 \circ \vec{\Delta}_{social}$$

- If ω, ϕ_1, ϕ_2 are set "correctly", this update rule allows for convergence without the use of $\mathbf{v}_{max}$
- The inertia weight can be used to control the **balance between exploration and exploitation:**
  - $\omega \geq 1$: velocities increase over time, swarm diverges
  - $0 < \omega < 1$: particles decelerate, convergence depends on ϕ_1, ϕ_2

CONSTRICTION COEFFICIENT

- Take away some 'guesswork' for setting ω, ϕ_1, ϕ_2
- An "elegant" method for preventing explosion, ensuring convergence and eliminating the parameter $\mathbf{v}_{max}$
- The constriction coefficient was introduced as:

$$\vec{v}_i \leftarrow \chi \cdot \left(\vec{v}_i + \vec{U}(0,\phi_1) \otimes (\vec{p}_i - \vec{x}_i) + \vec{U}(0,\phi_2) \otimes (\vec{g}_i - \vec{x}_i)\right)$$

With $\phi = \phi_1 + \phi_2 > 4$ and $\chi = \dfrac{2}{\varphi + \sqrt{\varphi^2 - 4\varphi}}$

Clerc, M. Kennedy, J., 'The particle swarm - explosion, stability, and convergence in a multidimensional complex space', *Evolutionary Computation, IEEE Transactions on* , vol. 6, no. 1, 58-73 (2002).

FULLY INFORMED PSO (FIPS)

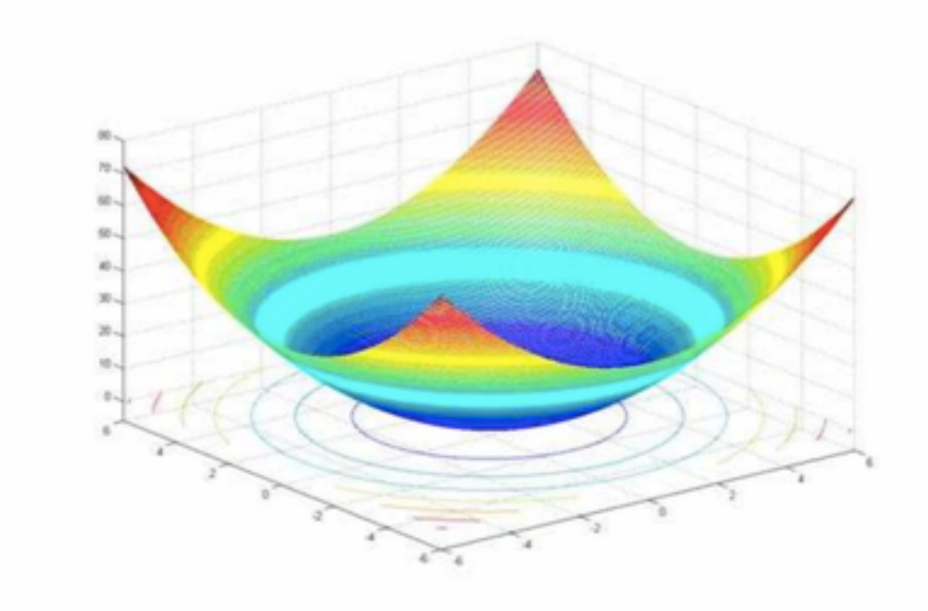
- Each particle is affected by all of its K neighbors
- The velocity update in FIPS is:

$$\begin{cases} \vec{v}_i = \chi \cdot \left(\vec{v}_i + \frac{1}{K_i} \sum_{n=1}^{K_i} \vec{U}(0,\varphi) \otimes \left(\vec{p}_{nbr_n} - \vec{x}_i \right) \right) \\ \vec{x}_i = \vec{x}_i + \vec{v}_i \end{cases}$$

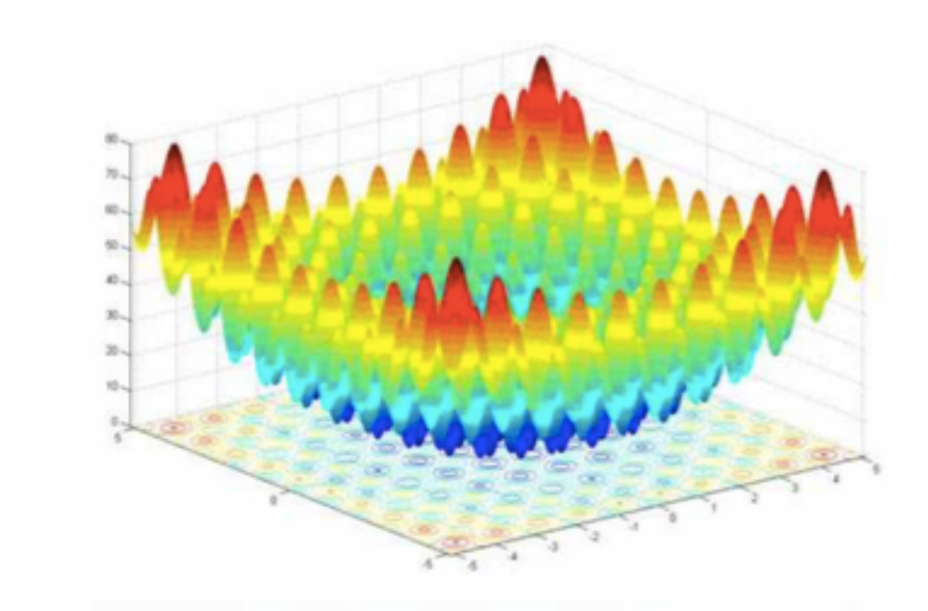
- FIPS outperforms the canonical PSO's on most test-problems
- The performance of FIPS is generally more dependent on the neighborhood topology (global best neighborhood topology is recommended)

R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," IEEE Trans. Evol. Comput., vol. 8, pp.204–210, June 2004.

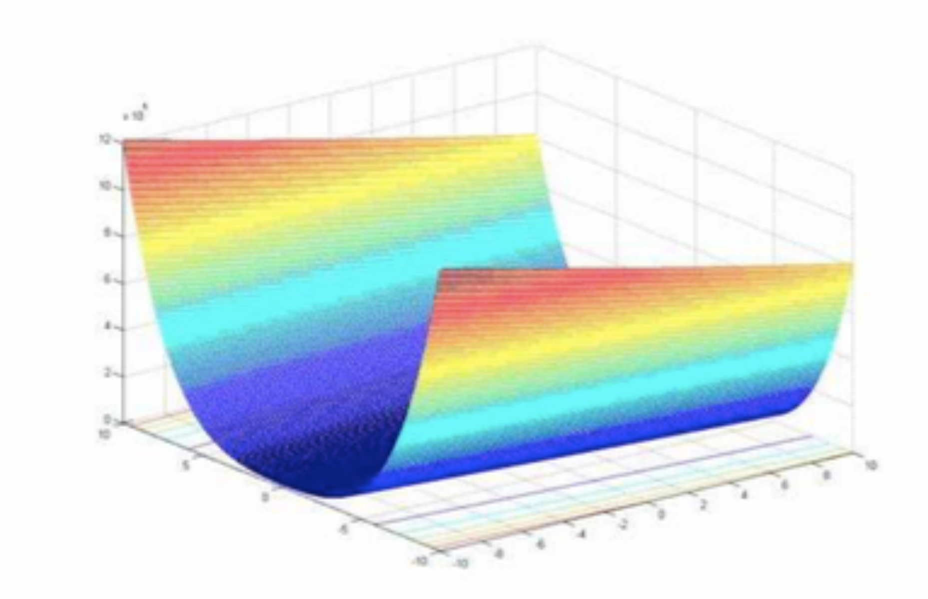
TYPICAL BENCHMARK FUNCTIONS



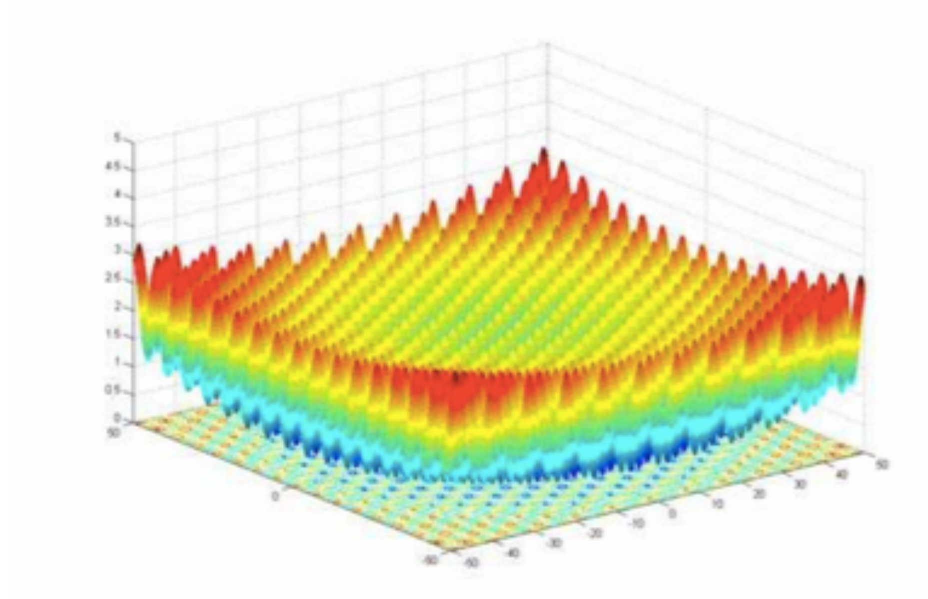
Sphere



Rastrigin



Rosenbrock

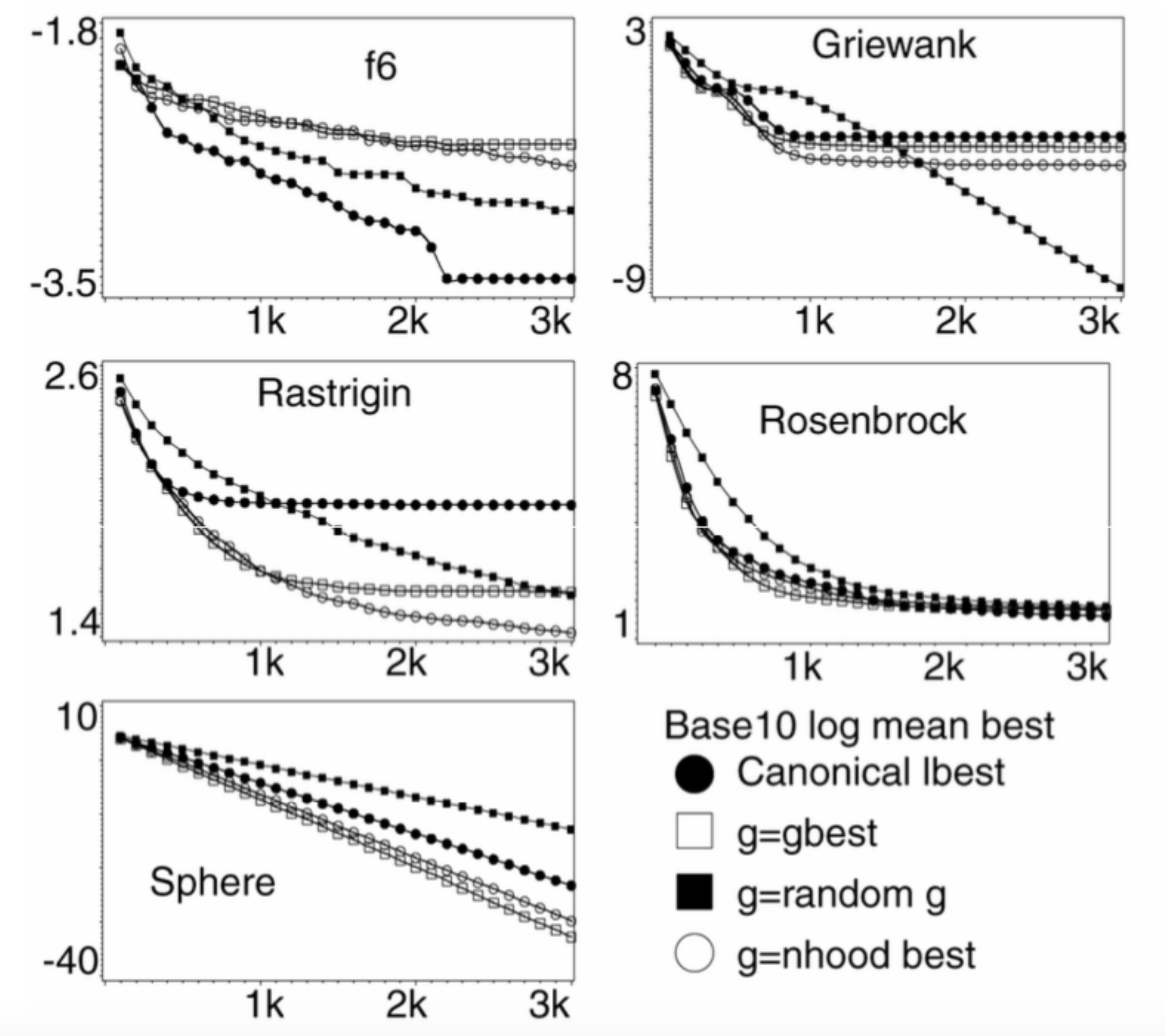


Griewank

Videos showing the time evolution of PSO algorithms:

http://www.youtube.com/watch?v=N2dtKHBhpxw
http://www.youtube.com/watch?v=VR3LSq99ebs
http://www.youtube.com/watch?v=ML2vuzqw1ok
http://www.youtube.com/watch?v=VAASmSSsFaY

PERFORMANCE VARIANCE



BINARY / DISCRETE PSO

- A simple modification to the continuous one
- Velocity remains continuous using the original update rule
- Positions are updated using the velocity as a **probability threshold** to determine whether the j-th component of the i-th particle is a zero or a one

$$x_{ij} = \begin{cases} 1 & \text{if } \tau < s(v_{ij}) \\ 0 & \text{otherwise} \end{cases} \quad \text{with} \quad s(x_{ij}) = \frac{1}{1 + \exp(-x_{ij})}$$

J. Kennedy and R. Eberhart. A discrete binary version of the particle swarm algorithm. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 4104-4108, IEEE Press, 1997

ANALYSIS, GUARANTEES

- Hard because:
  - **Stochastic search** algorithm
  - **Complex group dynamics**
  - **Performance depends on the search landscape**
- Theoretical analysis has been done with simplified PSOs on simplified problems

- Graphical examinations of the trajectories of individual particles and their responses to variations in key parameters
- Empirical performance distributions

SUMMARY PSO

- Inspired by social and roosting behaviors in bird flocking
- Easy to implement, easy to get good results with "wise" parameter tuning (but just a few parameters)
- Computationally light
- Exploitation-Exploration dilemma
- A number of variants
- A few theoretical properties (hard to derive for general cases)
- Mostly applied to continuous function optimization, but also to combinatorial optimization, and robotics / distributed systems

REFERENCES

- Computational Intelligence, A. Engelbrecht, Wiley, 2007
- Ant Colony Optimization, M. Dorigo, T. Stuetzle, MIT Press, 2002
- Swarm Intelligence: From Natural to Artificial Systems, E. Bonabeau, G. Theraulaz, M. Dorigo, Oxford University Press, 1999
- Swarm Intelligence, J. Kennedy, R. Eberhart, Y. Shi, Morgan Kaufmann, 2001
- Self-organization in biological systems, S. Camazine, J.L. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, E. Bonabeau, Princeton University Press, 2001