Swarm Intelligence

From Natural to Artificial Systems

Swarming – The Definition

- aggregation of similar animals, generally cruising in the same direction

- Termites swarm to build colonies
- Birds swarm to find food
- Bees swarm to reproduce

Why do animals swarm?

- To forage better
- To migrate
- As a defense against predators

- Social Insects have survived for millions of years.

Swarming is Powerful

- Swarms can achieve things that an individual cannot

Swarming – Example

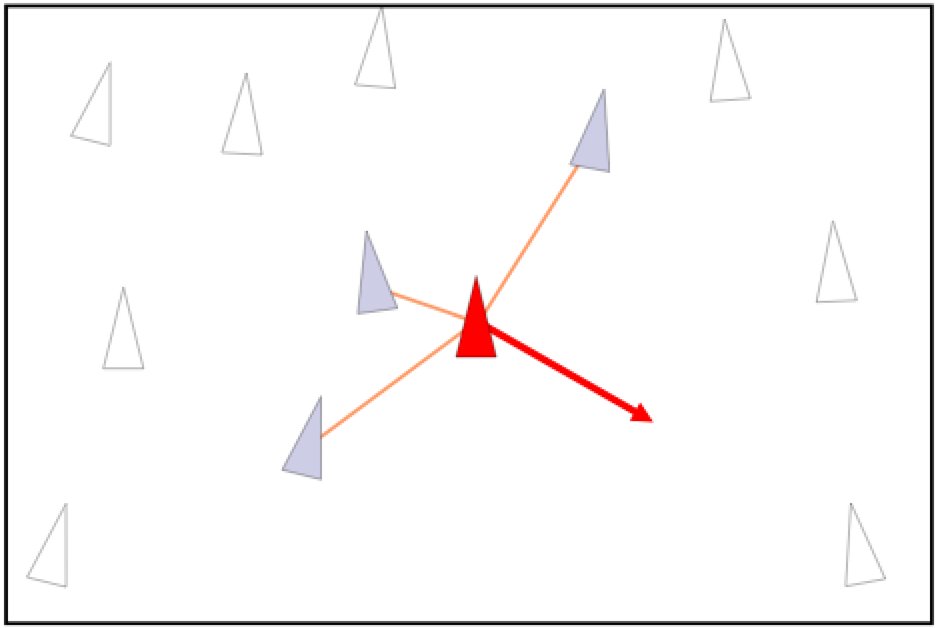
- Bird Flocking

- "Boids" model was proposed by Reynolds
  - Boids = Bird-oids (bird like)

- Only three simple rules

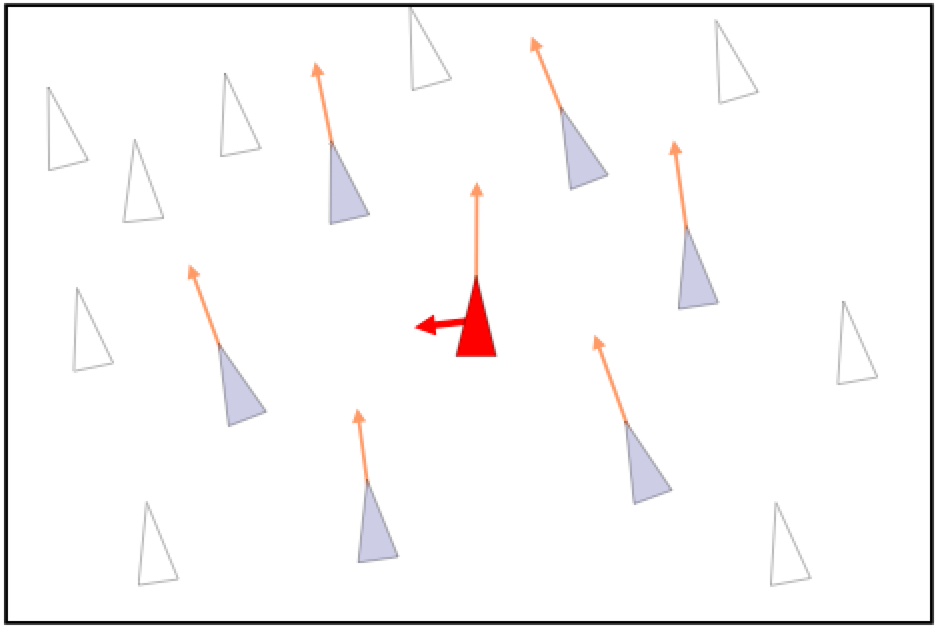
Collision Avoidance

- Rule 1: Avoid Collision with neighboring birds



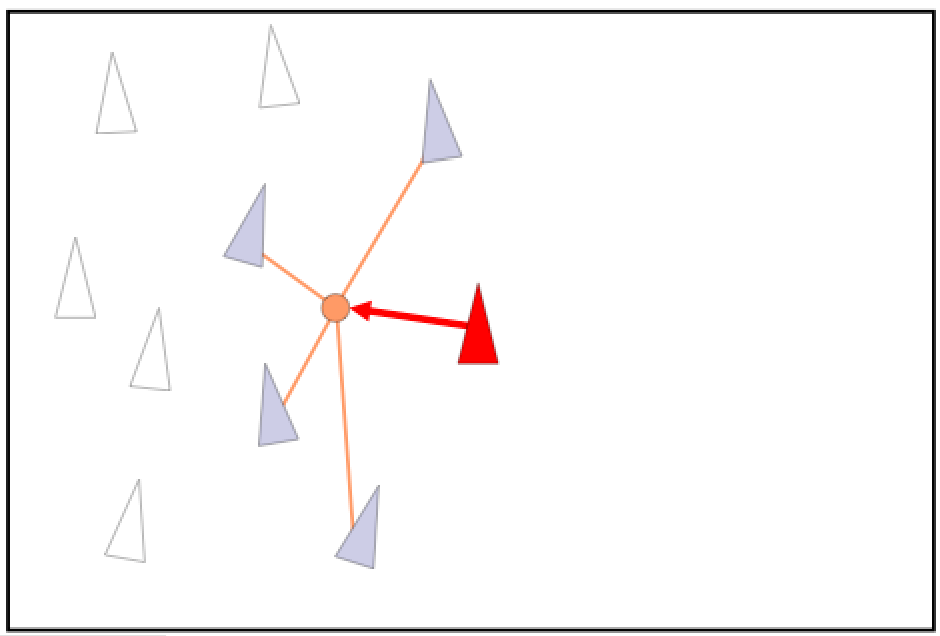
Velocity Matching

- Rule 2: Match the velocity of neighboring birds



Flock Centering

- Rule 3: Stay near neighboring birds



Swarming - Characteristics

- Simple rules for each individual
- No central control
  - Decentralized and hence robust
- Emergent
  - Performs complex functions

Learn from insects

- Computer Systems are getting complicated
- Hard to have a master control

- Swarm intelligence systems are:
    - Robust
    - Relatively simple

Swarm Intelligence - Definition

- “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies” [Bonabeau, Dorigo, Theraulaz: Swarm Intelligence]
- Solves optimization problems

Applications

- Movie effects
  - Lord of the Rings
- Network Routing
  - ACO Routing
- Swarm Robotics
  - Swarm bots

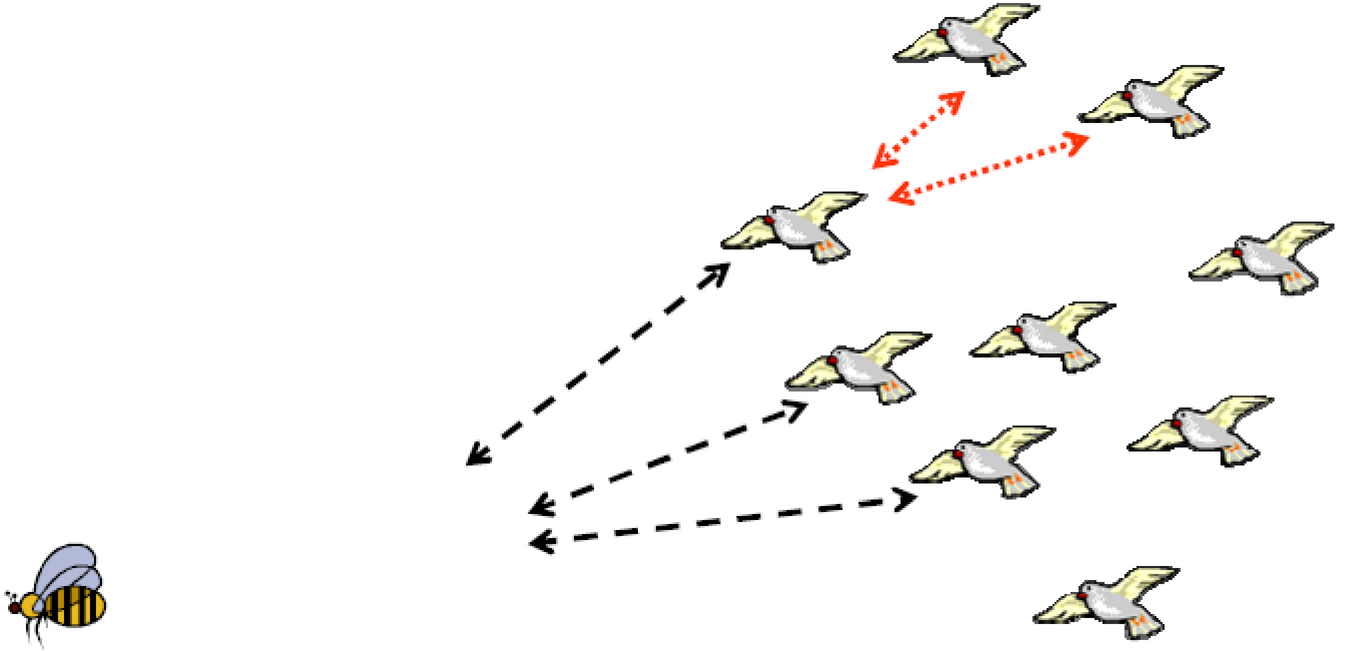
Roadmap

- Particle Swarm Optimization
  - Applications
  - Algorithm
- Ant Colony Optimization
  - Biological Inspiration
  - Generic ACO and variations
  - Application in Routing
- Limitations of SI
- Conclusion

Particle Swarm Optimization

Particle Swarm Optimization

- Particle swarm optimization imitates human or insects social behavior.
- Individuals interact with one another while learning from their own experience, and gradually move towards the goal.
- It is easily implemented and has proven both very effective and quick when applied to a diverse set of optimization problems.



- Bird flocking is one of the best example of PSO in nature.
- One motive of the development of PSO was to model human social behavior.

Applications of PSO

- Neural networks like Human tumor analysis, Computer numerically controlled milling optimization;
- Ingredient mix optimization;
- Pressure vessel (design a container of compressed air, with many constraints).

Basically all the above applications fall in a category of finding the global maxima of a continuous, discrete, or mixed search space, with multiple local maxima.

Algorithm of PSO

- Each particle (or agent) evaluates the function to maximize at each point it visits in spaces.
- Each agent remembers the best value of the function found so far by it (pbest) and its co-ordinates.
- Secondly, each agent know the globally best position that one member of the flock had found, and its value (gbest).

Algorithm – Phase 1 (1D)

- Using the co-ordinates of pbest and gbest, each agent calculates its new velocity as:

$$v_i = v_i + c_1 \times rand() \times (pbestx_i - presentx_i) + c_2 \times rand() \times (gbestx - presentx_i)$$

where $0 < rand() < 1$

$$presentx_i = presentx_i + (v_i \times \Delta t)$$

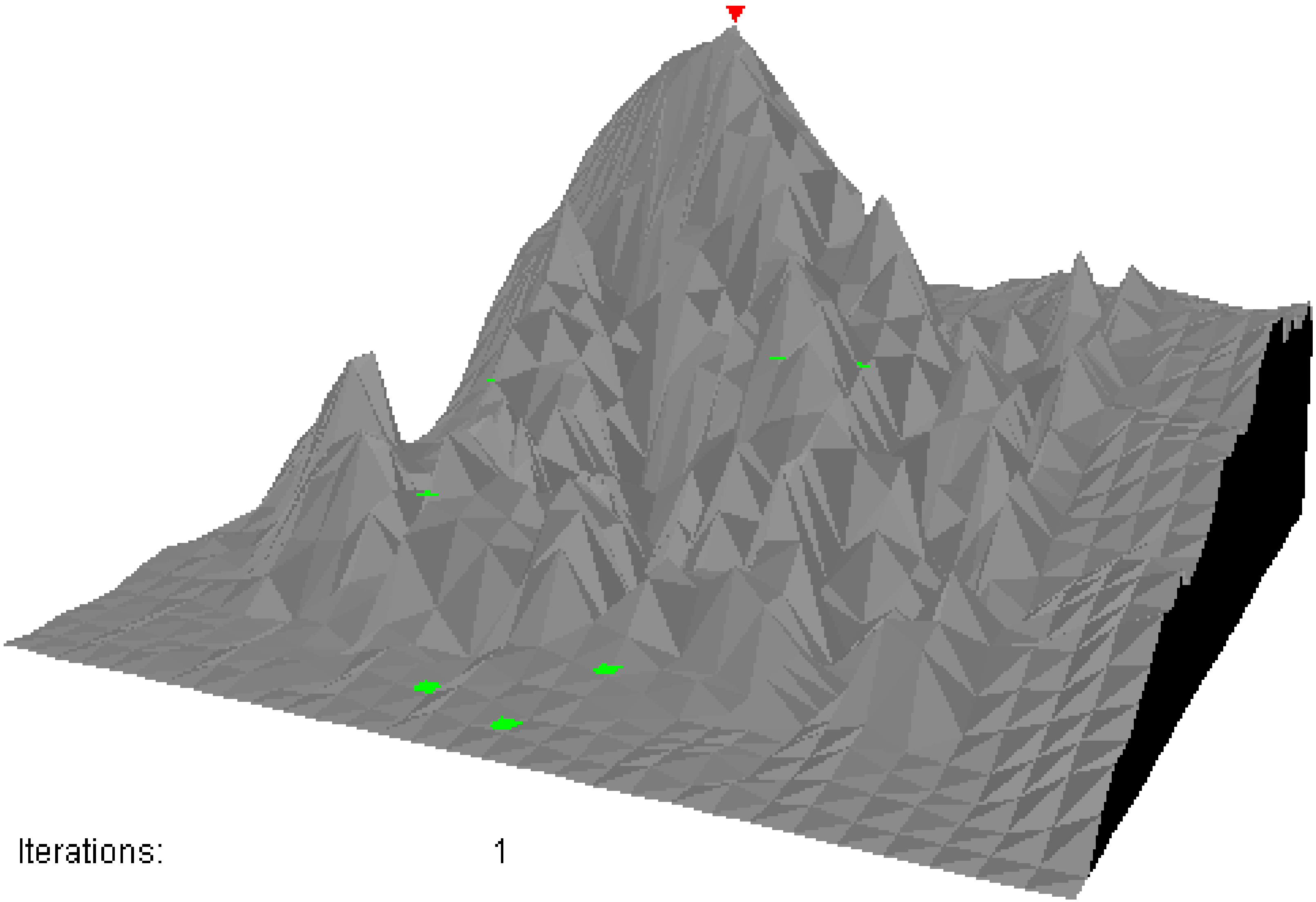
Algorithm – Phase 2 (n-dimensions)

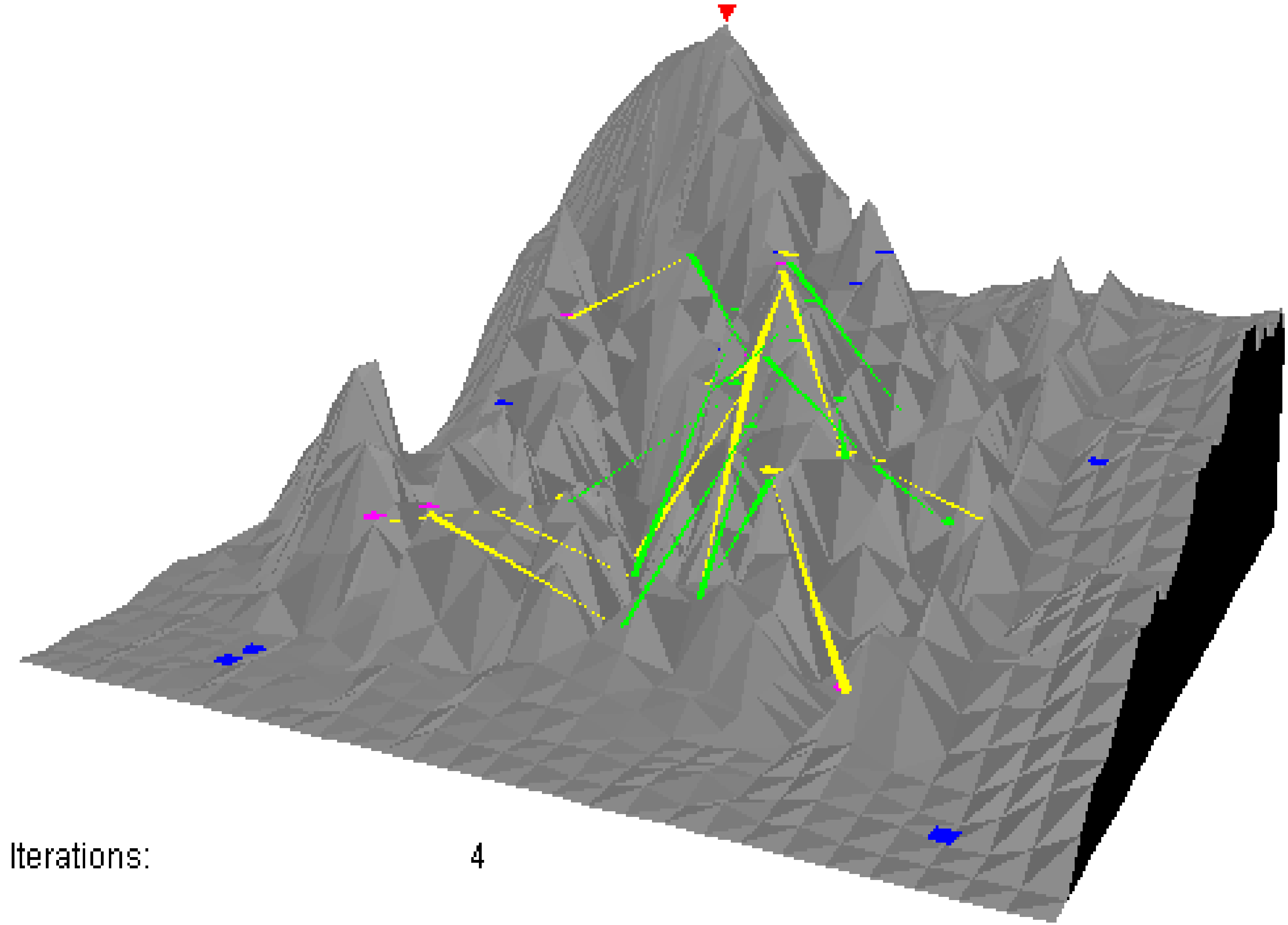
- In n-dimensional space :

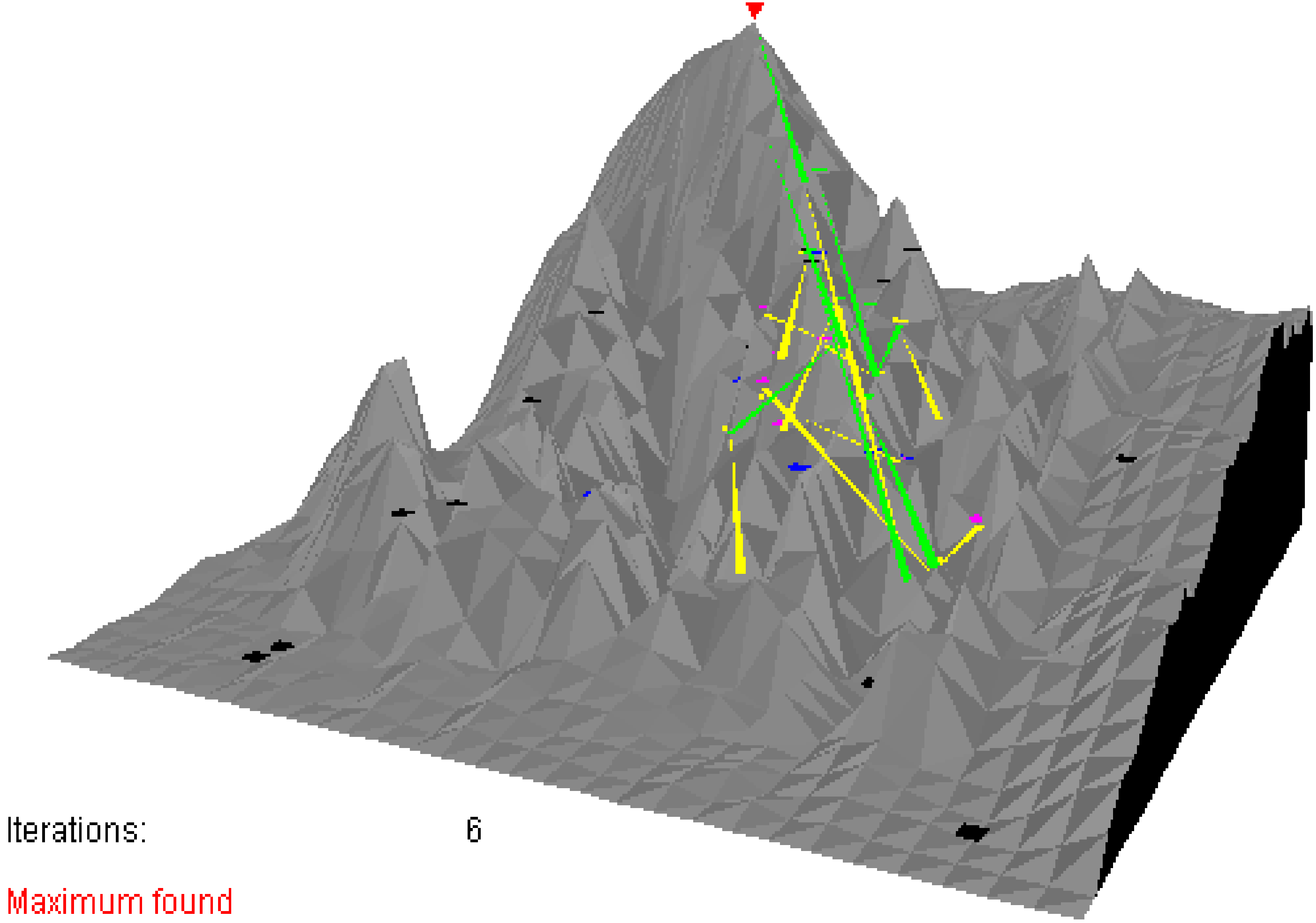
$$\vec{v}_i = \vec{v}_i + \overbrace{\boxed{\text{rand}() \times \vec{c}_1 \otimes (\overrightarrow{pbest}_i - \overrightarrow{present}_i)}}^{\text{cognitive component}}$$
$$+ \underbrace{\boxed{\text{rand}() \times \vec{c}_2 \otimes (\overrightarrow{gbest} - \overrightarrow{present}_i)}}_{\text{social component}}$$

Note that the symbol $\otimes$ denotes a point-wise vector multiplication.

```
Randomly generate an initial population
repeat
        for i = 1 to population_size do
                if f(present_i) < f(pbest)
                    then pbest = present_i;
                gbest = best(pbest);
                for d =1 to dimensions do
                        velocity_update();
                        position_update();
                end
        end
until termination criterion is met.
```


Iterations: 1


Iterations: 4

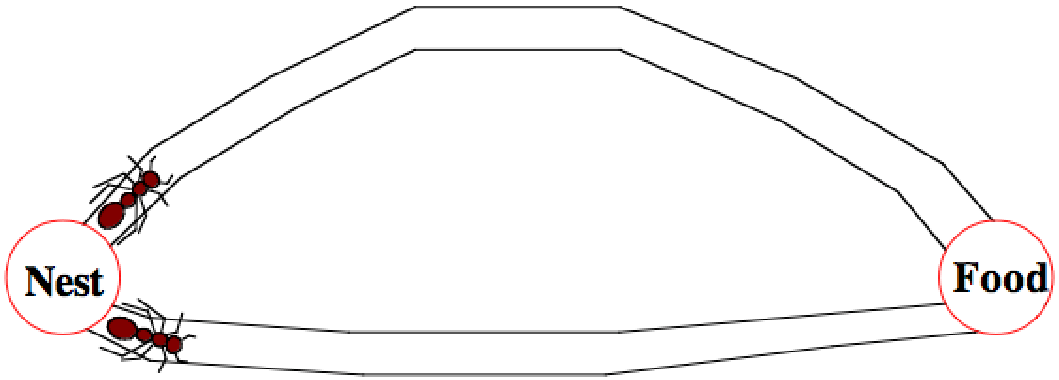

Iterations: 6
Maximum found

Ant Colony Optimization

Ant Colony Optimization - Biological Inspiration

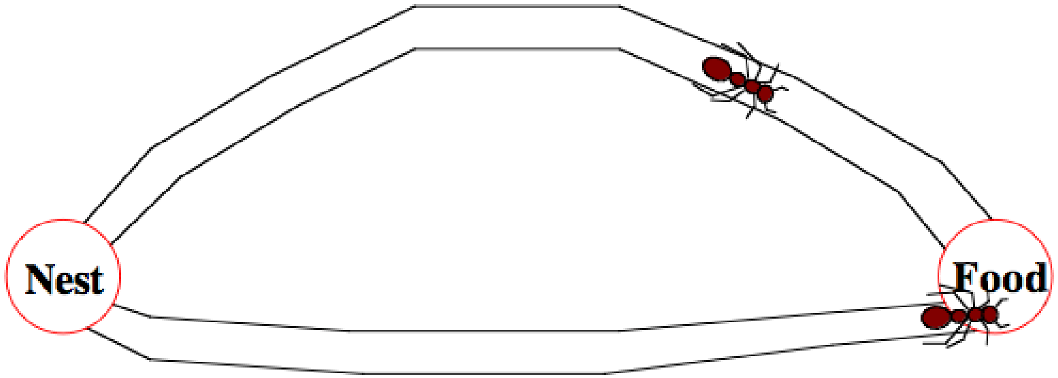
- Inspired by foraging behavior of ants.
- Ants find shortest path to food source from nest.
- Ants deposit pheromone along traveled path which is used by other ants to follow the trail.
- This kind of indirect communication via the local environment is called stigmergy.
- Has adaptability, robustness and redundancy.

Foraging behavior of Ants



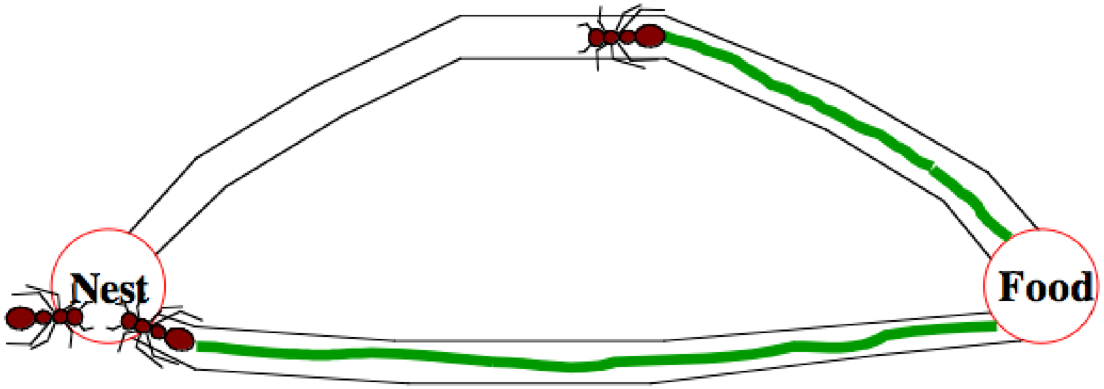
- 2 ants start with equal probability of going on either path.

Foraging behavior of Ants



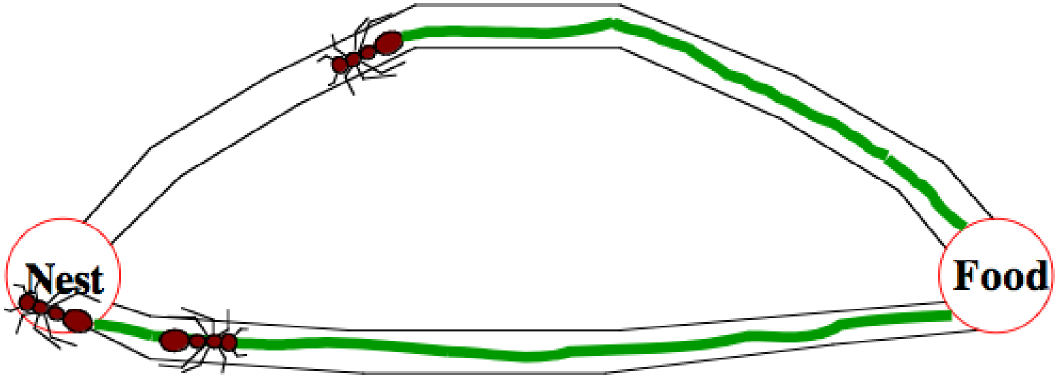
- The ant on shorter path has a shorter to-and-fro time from it’s nest to the food.

Foraging behavior of Ants



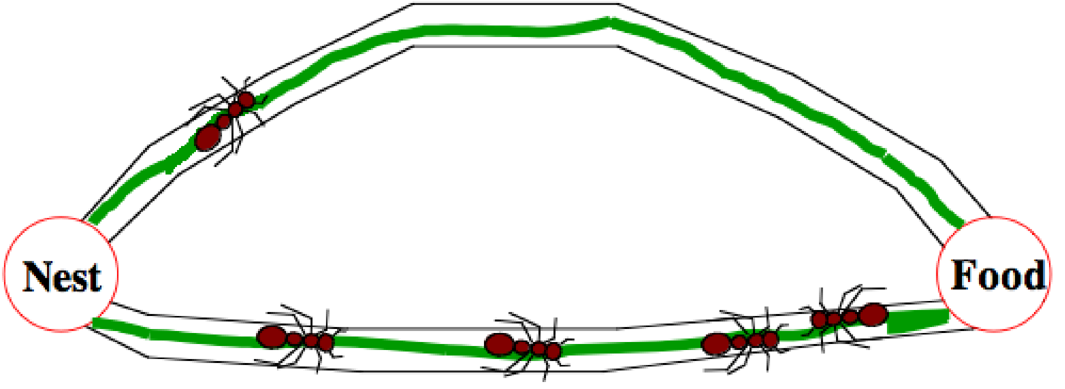
- The density of pheromone on the shorter path is higher because of 2 passes by the ant (as compared to 1 by the other).

Foraging behavior of Ants



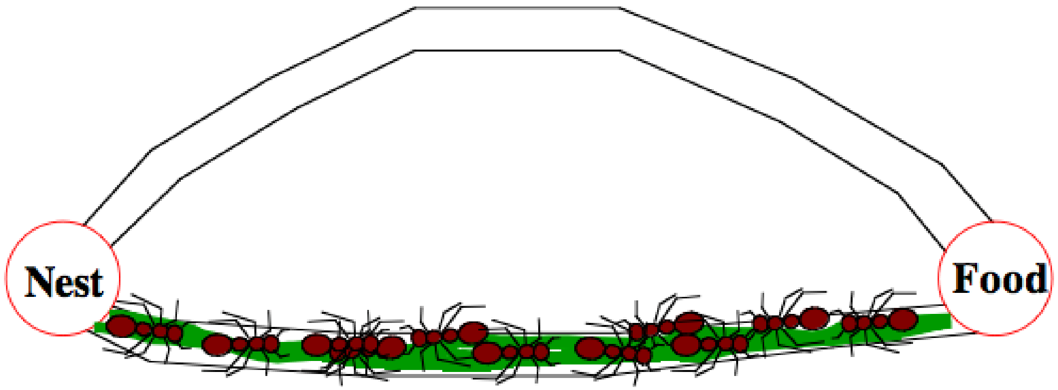
- The next ant takes the shorter route.

Foraging behavior of Ants



- Over many iterations, more ants begin using the path with higher pheromone, thereby further reinforcing it.

Foraging behavior of Ants



- After some time, the shorter path is almost exclusively used.

Generic ACO

- Formalized into a metaheuristic.
- Artificial ants build solutions to an optimization problem and exchange info on their quality vis-à-vis real ants.
- A combinatorial optimization problem reduced to a construction graph.
- Ants build partial solutions in each iteration and deposit pheromone on each vertex.

Ant Colony Metaheuristic

Algorithm 1 The Ant Colony Optimization Metaheuristic

```
Set parameters, initialize pheromone trails
while termination condition not met do
    ConstructAntSolutions
    ApplyLocalSearch (optional)
    UpdatePheromones
end while
```

- ConstructAntSolutions: Partial solution extended by adding an edge based on stochastic and pheromone considerations.
- ApplyLocalSearch: problem-specific, used in state-of-art ACO algorithms.
- UpdatePheromones: increase pheromone of good solutions, decrease that of bad solutions (pheromone evaporation).

Various Algorithms

- First in early 90's.
- Ant System (AS):
  - First ACO algorithm.
  - Pheromone updated by all ants in the iteration.

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ used edge } (i,j) \text{ in its tour,} \\ 0 & \text{otherwise,} \end{cases}$$

  - Ants select next vertex by a stochastic function which depends on both pheromone and problem-specific heuristic $n_{ij} = \frac{1}{d_{ij}}$

Various Algorithms - 2

- MAX-MIN Ant System (MMAS):
  - Improves over AS.
  - Only best ant updates pheromone.
  - Value of pheromone is bound.

$$\tau_{ij} \leftarrow \left[(1-\rho)\cdot\tau_{ij} + \Delta\tau_{ij}^{\text{best}}\right]_{\tau_{min}}^{\tau_{max}}$$

$$\Delta\tau_{ij}^{\text{best}} = \begin{cases} 1/L_{\text{best}} & \text{if } (i,j) \text{ belongs to the best tour,} \\ 0 & \text{otherwise.} \end{cases}$$

  - L_{best} is length of tour of best ant.
  - Bounds on pheromone are problem specific.

Various Algorithms - 3

- Ant Colony System (ACS):
  - Local pheromone update in addition to offline pheromone update.
  - By all ants after each construction step only to last edge traversed.
  - Diversify search by subsequent ants and produce different solutions in an iteration.
  - Local update: $\tau_{ij} = (1-\varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0$
  - Offline update: $\tau_{ij} \leftarrow \begin{cases} (1-\rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij} & \text{if } (i,j) \text{ belongs to best tour} \\ \tau_{ij} & \text{otherwise.} \end{cases}$

Theoretical Details

- Convergence to optimal solutions has been proved.
- Can’t predict how quickly optimal results will be found.
- Suffer from stagnation and selection bias.

ACO in Network Routing

Amit Bharadwaj

Ant like agents for routing

- Intuitive to think of ants for routing problem
- Aim is to get shortest path
- Start as usual
  - Release a number of ants from source, let the age of ant increases with increase in hops
  - decide on pheromone trails i.e biasing the entries in routing table in favor of youngest ant
- Problem – Ants at an node do not know the path to destiation, can't cahnge table entry

Routing continued ...

- Possible Solutions
  - first get to dest. and then retrace
    - Needs memory to store the path
    - And intelligence to revert the path
  - Leave unique entries on nodes
    - a lot of entries at every node
- Observation – At any intermediate node, ant knows the path to source from that node.
  - now leave influence on routing table having entry "route to source via that link"

Routing contd ...

- Now at any node it has information about shortest path to dest., left by ants from dest.
- The ant following shortest path should have maximum influence
- A convenient form of pheromone can be inverse of age + constant
- The table may get frozen, with one entry almost 1, add some noise f i.e probabilty that an ant choses purely random path

Dealing with congestion

- Add a function of degree of congestion of each node to age of an ant
- Delay an ant at congested node, this prevents ants from influencing route table

SI - Limitations

- Theoretical analysis is difficult, due to sequences of probabilistic choices
- Most of the research are experimental
- Though convergence in guaranteed, time to convergence is uncertain

Scope

- Startup !!
  - Bluetronics, Smartintel
- Analytic proof and models of swarm based algorithm remain topics of ongoing research
- List of applications using SI growing fast
  - Controlling unmanned vehicles.
  - Satellite Image Classification
  - Movie effects

Conclusion

- Provide heuristic to solve difficult problems
- Has been applied to wide variety of applications
- Can be used in dynamic applications

References

- Reynolds, C. W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34.
- James Kennedy, Russell Eberhart. Particle Swarm Optimization, IEEE Conf. on Neural networks – 1995
- www.adaptiveview.com/articles/ ipsop1
- M.Dorigo, M.Birattari, T.Stutzle, Ant colony optimization – Artificial Ants as a computational intelligence technique, IEEE Computational Intelligence Magazine 2006
- Ruud Schoonderwoerd, Owen Holland, Janet Bruten - 1996. Ant like agents for load balancing in telecommunication networks, *Adaptive behavior, 5(2)*.