Particle Swarm Optimization

- Population-based stochastic optimization technique
- Purpose: optimization of continuous nonlinear functions
- Background: bird flocking and fish schooling, artificial life, social systems
- First work: [Eberhart and Kennedy, 1995]
- Popularity: A book [Kennedy and Eberhart, 2001], a recent special issue on IEEE Transaction on Evolutionary Computation [Vol. 8, June 2004], topic of interest in several conferences and workshops
- It's actually a sort of generalization of ***Cellular Automata*** [Wolfram, 1980–], so let's give a quick look to them first

Cellular Automata

- A set of simple automata, that is, finite state machines with few states $S = \{s_1, s_2, \ldots, s_k\}$
- A topology of interconnection among the automata, such that each automaton a_i has n_i *neighbors*, $\mathcal{N}(a_i) = \{a_i^1, a_i^2, \ldots, a_i^{n_i}\}$
- A state-transition function F that depends on the current state $s^i(t)$ of the automaton and on the state of its neighbors $\mathcal{N}(a_i)$
- At discrete time-steps (and either synchronously or asynchronously) each automaton gets the state from its neighbors and possibly change state accordingly

Examples: numeric solution of differential equations, voting in social networks , fluid dynamics, cell behavior

... Loads of theoretical studies more than useful

PSO: background

- Early work on ***simulation of bird flocking*** aimed at understanding the underlying rules that allow smooth flocking [Reynolds, 1984] and roosting behavior [Heppner and Grenader, 1990]
- The rule were supposed simple and based on ***social behavior***: sharing of information and reciprocal respect of the occupancy of physical space
- ***Social sharing of information*** among conspeciates seems to offer an evolutionary advantage
- Target: study of ***human social behavior*** on the basis of bird/fish swarm behavior. The notion of ***change in human social behavior/psychology*** is seen as the analogous of ***change in spatial position in birds***.

PSO: background (2)

- Initial simulation: a population of $N \gg 1$ agents is initialized on a toroidal 2D pixel grid with random position and velocity, $(\bar{x}_i, \bar{v}_i),\ i = 1, \ldots, N$
- At each iteration loop, each agent determines its new speed vector according to that of its nearest neighbor
- A random component is used in order to avoid fully unanimous, unchanging flocking
- All this was not so exciting, but the ***roosting behavior*** of Heppner was intriguing: it looked like a dynamic force such that eventually the birds were attracted to land on a specific location. The roost could be the equivalent of the ***optimum in a search space***!

PSO: background (3)

- In real life, birds don’t know for instance were food is, but if one puts out a bird feeder he/she will see that within hours a great number of birds will find it, even though they had no previous knowledge about it. This looks like the flock dynamics enables members of the flock to capitalize on one another’s knowledge
- The agents can be therefore assimilated to ***solution hunters that socially share knowledge*** while they fly over a solution space. Each agent that has found anything good leads its neighbors toward it. So that eventually they can land on the ***best solution in the field***

PSO: background (3)

- In real life, birds don't know for instance were food is, but if one puts out a bird feeder he/she will see that within hours a great number of birds will find it, even though they had no previous knowledge about it. This looks like the flock dynamics enables members of the flock to capitalize on one another's knowledge

- The agents can be therefore assimilated to ***solution hunters that socially share knowledge*** while they fly over a solution space. Each agent that has found anything good leads its neighbors toward it. So that eventually they can land on the ***best solution in the field***

PSO: the meta-algorithm

Each agents is a particle-like data structure that contains: the coordinates of the current location in the optimization landscape, the best solution point visited so far, the subset of other agents that are seen as neighbors.

```
procedure Particle_Swarm_Optimization()
  foreach particle ∈ ParticleSet do
    init_at_random_positions_and_velocity();
    select_at_random_the_neighbor_set();
  end foreach
  while (¬ stopping_criterion)
    foreach particle ∈ ParticleSet do
      calculate_current_fitness_and_update_memory();
      get_neighbor_with_best_fitness();
      calculate_individual_deviation_between_current_and_best_so_far_fitness();
      calculate_social_deviation_between_current_and_best_neighbor_fitness();
      calculate_velocity_vector_variation_as_weighted_sum_between_deviations();
      update_velocity_vector();
    end foreach
  end while
return best_solution_found();
```

PSO: the algorithm

procedure Particle_Swarm_Optimization($Particles$, $Neighbors$, Fitness_Func())
  foreach $particle\ p[i], i = 1, \ldots, Particles$ **do**
    $p[i].\bar{q}_{next} \leftarrow p[i].\bar{q} \leftarrow$ get_random_position();
    $p[i].\bar{v} \leftarrow$ get_random_velocity();
    $p[i].\bar{n} \leftarrow$ get_random_neighbors($Neighbors$); $p[i].best.val \leftarrow -\infty$;
  end foreach
  $s.best \leftarrow -\infty$;
  while $(\neg\ stopping\_criterion)$
    foreach $particle\ p[i], i = 1, \ldots, Particles$ **do**
      $p[i].\bar{q}_{next} \leftarrow p[i].\bar{q}$;
      $pFitness \leftarrow$ Fitness_Func($p[i]$);
      if $(pFitness > p.best.val)$
        $p[i].best.val \leftarrow pFitness$; $p[i].best.\bar{q} \leftarrow p[i].\bar{q}$;
      end if
      if $(pFitness > s.best.val)$
        $s.best.val \leftarrow pFitness$; $s.best.\bar{q} \leftarrow p[i].\bar{q}$;
      end if
      $n \leftarrow$ get_neighbor_with_best_fi tness($p[i].\bar{n}$);
      foreach $d = 1, \ldots, ProbDimensions$ **do**
        $indFactor \leftarrow indWeight$ * random($iMin, iMax$);
        $socFactor \leftarrow socWeight$ * random($socMin, socMax$);
        $\Delta p \leftarrow p.best[d] - p.q[d]$;
        $\Delta n \leftarrow n.best[d] - p.q[d]$;
        $\Delta \leftarrow (indFactor * \Delta p) + (socFactor * \Delta n)$;
        $p.v[d] \leftarrow p.v[d] + \Delta$; $p.v[d] \leftarrow$ keep_velocity_in_a_min_max_range();
        $p.q[d]_{next} \leftarrow p.q[d] + p.v[d]$;
      end foreach
    end foreach
  end while
return $s.best$;

Some final considerations on PSO

- Equivalent to a ***real-valued 2D CA*** where the state of a particle is $(\bar{q}, \bar{v}) \cup (\bar{q}_{best}, F(\bar{q}_{best}))$
- The neighborhood relationship is ***not transitive***, however other choices can be selected
- Social networks can be asymmetric (A is connected to B but B might not care about A)
- An update of the state (position on the optimization landscape) is calculated as a tradeoff between ***individual*** and ***social knowledge***
- Tested on benchmarks for continuous functions (e.g., [van den Berg and Engelbrecht, 2004]) and NN training (10–50 particles). Performance comparable to genetic algorithms, but simpler to design and analyze

ACO' background: Shortest paths, stigmergy

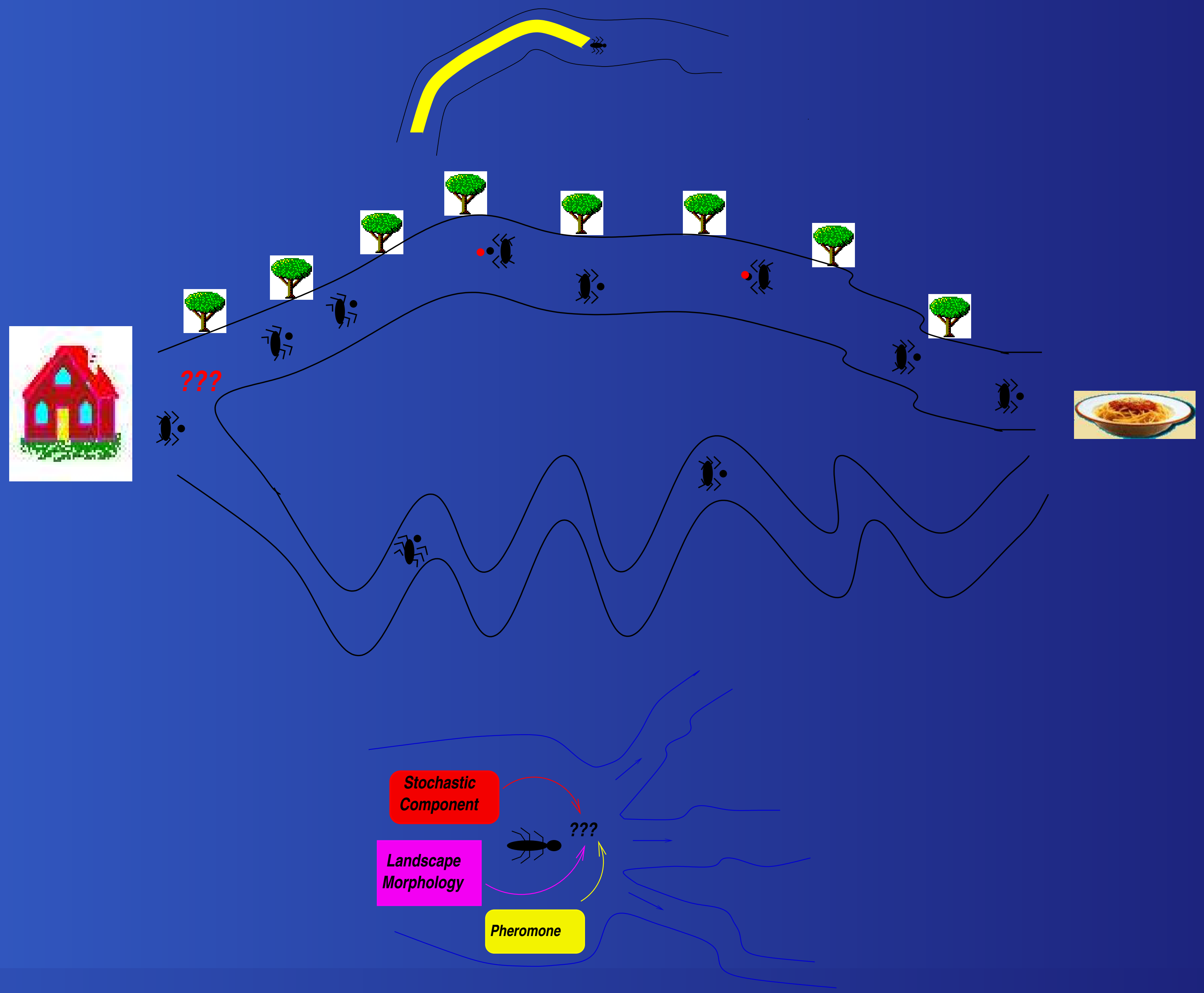
Outline

- Biological experiments that pointed out the shortest path behavior of ant colonies
- Ingredients that make the shortest path behavior happening
- Usefulness of solving shortest path problems
- Stigmergy: indirect communication giving rise to self-organization
- Definition and examples of stigmergic variables
- Design of a stigmergy-based multi-agent system

Pheromone and Shortest Paths

- Several ant species have ***trail-laying/trail-following behavior*** when ***foraging*** [Hölldobler and Wilson, 1990]. While moving, individual ants deposit on the ground a volatile chemical substance called ***pheromone***, forming in this way ***pheromone trails***. Ants can smell pheromone and, when choosing their way, they tend to choose, in probability, the paths marked by stronger pheromone concentrations. In this way pheromone trails constitute a sort of attractive ***potential field***
- Pheromone trails allows the ants to find their way back to food sources (or to the nest), and can be used by other ants to find the location of the food sources discovered by their nestmates.

Ant behavior illustrated



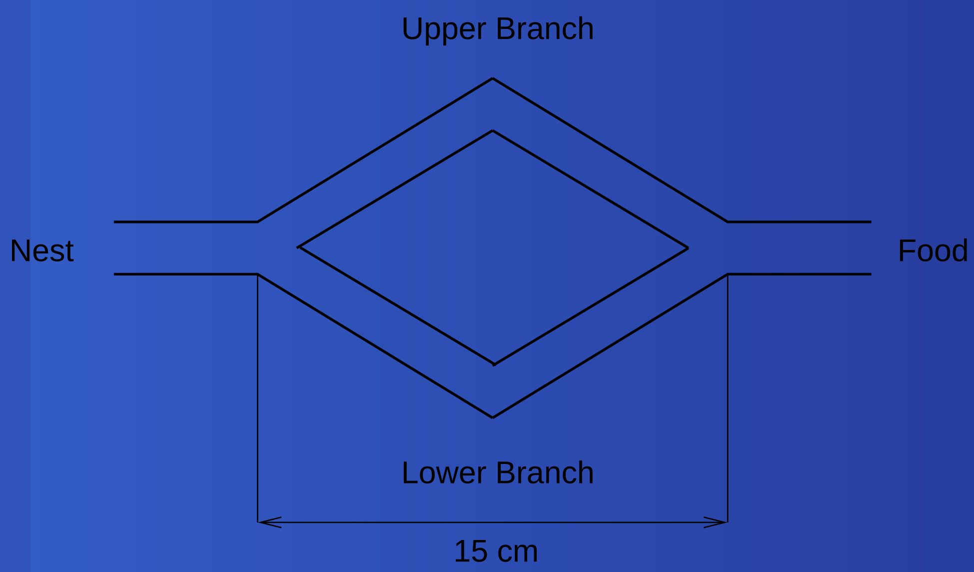
More on pheromone and shortest paths. . .

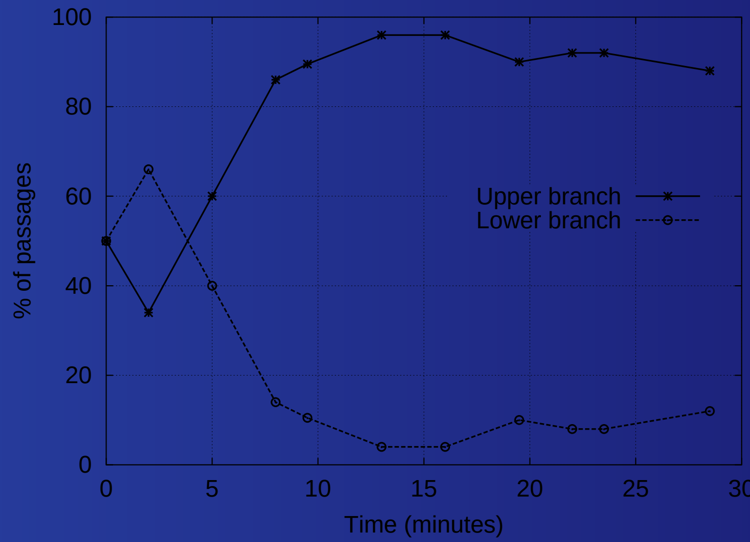
- Pheromone trails act as a sort of ***distributed dynamic collective memory*** of the colony, a repository of all the most recent ***foraging experiences*** of the ants belonging to the same colony. By continually updating and sensing this chemical repository the ants can indirectly communicate and influence each other through the environment

- This basic form of ***indirect communication***, coupled with a form of positive feedback, can be enough to allow the colony *as a whole* to discover, when only few alternative paths are possible, the ***shortest path connecting a source of food to the colony's nest*** [Experiments by Aron, Beckers, Deneubourg, Goss, Pasteels et al., 1977–]

The binary bridge experiment

- After an initial transitory phase lasting few minutes during which some oscillations can appear, ants tend to converge on the same path [Deneubourg, Aron, Goss and Pasteels, 1990]





Probabilistic model for binary bridge

- The amount of pheromone on a branch is proportional to the number of ants which have been using the branch in the past (pheromone is assumed persistent). The probability of choosing a branch at a certain time depends on the total amount of pheromone on the branch, which, in turn, is proportional to the number of ants which have used the branch until that moment:

$$
\begin{aligned}
P_U(m) &= \frac{(U_m + k)^h}{(U_m + k)^h + (L_m + k)^h}, \\
P_L(m) &= 1 - P_U(m).
\end{aligned}
$$

- The dynamics regulating the ant choices are:

$$
\begin{aligned}
U_{m+1} &= U_m + 1 \qquad \text{if } \psi \leq P_U,\ \psi = \mathcal{U}(0,1) \\
U_{m+1} &= U_m \qquad \text{otherwise,}
\end{aligned}
$$

- Monte Carlo simulations were run to test the model versus real data: results of simulations were in agreement with the experiments with real ants for $k \approx 20$ and $h \approx 2$

Binary bridge with unequal branches (1)

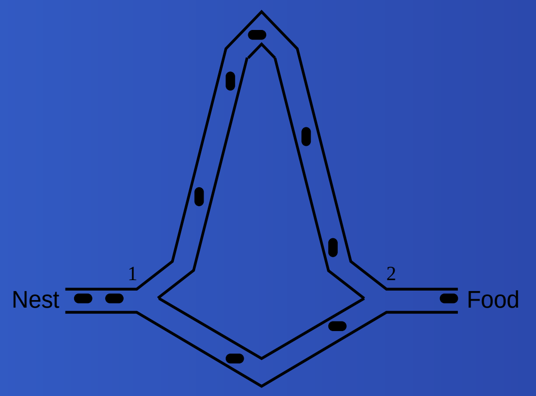
- If the branches of the bridges are of *different length*, then the pheromone field can lead the majority of the ants in the colony to select the ***shortest between the two available paths*** [Goss, Aron, Deneubourg and Pasteels, 1989]

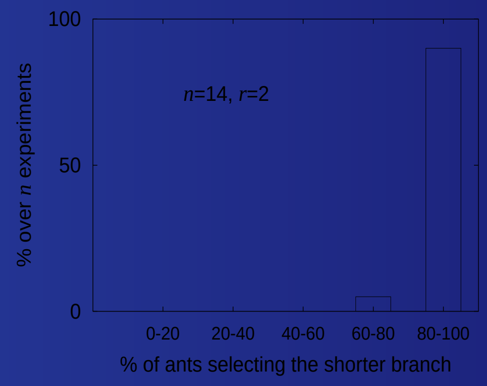
- The first ants able to arrive at the food source are those that traveled following the shortest branch. Accordingly, the pheromone that these same ants have laid on the shortest branch while moving *forward* towards the food source makes this branch marked by more pheromone than the longest one. The higher levels of pheromone present on the shortest branch stimulate these same ants to *probabilistically* choose again the shortest branch when moving *backward* to their nest. Recursive behavior: the very fact of choosing a path increases its probability of being chosen again in the near future (***autocatalytic effect***)

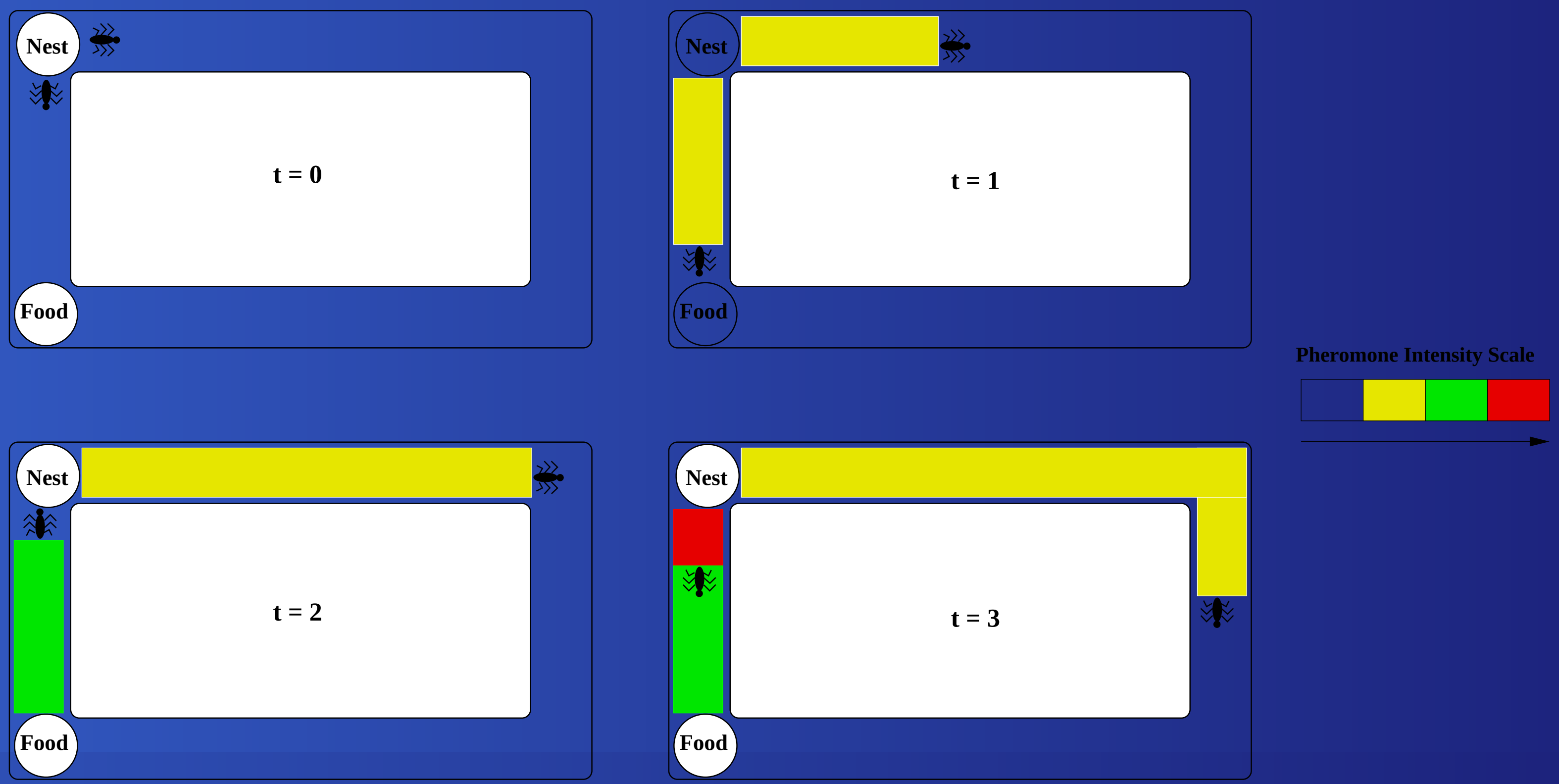
Binary bridge with unequal branches (2)

- Moving backward, additional pheromone is released on the shortest path. In this way, pheromone is laid on the shortest branch at a *higher rate* than on the longest branch. This reinforcement of the pheromone intensity on the shorter paths is the result of a form of ***implicit path evaluation***: the shorter paths are completed earlier than the longer ones, and therefore they receive ***pheromone reinforcement more quickly***
- Therefore, for a same number of ants choosing either the shortest or the longest branch at the beginning, since the pheromone on the shortest branch is accumulated at a higher rate than on the longest one, the choice of the shortest branch becomes more and more attractive for the subsequent ants at both the decision points

Binary bridge with unequal branches (3)







Ingredients of the shortest path behavior

- Population (colony) of foraging ants (autonomous, asynchronous, concurrent, distributed)
- Forward-backward path following
- Pheromone laying and sensing (shared distributed memory, distributed control)
- Pheromone-biased stochastic decisions (local fi eld)
- Autocatalysis
- Implicit path evaluation
- Iteration over time

The colony realizes *concurrent computation*. *Multiple paths* are repeatedly tried out back and forth and some information related to each followed path is released on the environment. Stochastic decisions are based on the *local* pheromone content.
Implicit path evaluation, coupled with autocatalysis, results in distributed and collective *path optimization*. Given enough time, and depending on the number of ants, relative length of the paths, pheromone evaporation, etc., this can result in the convergence onto the shortest path

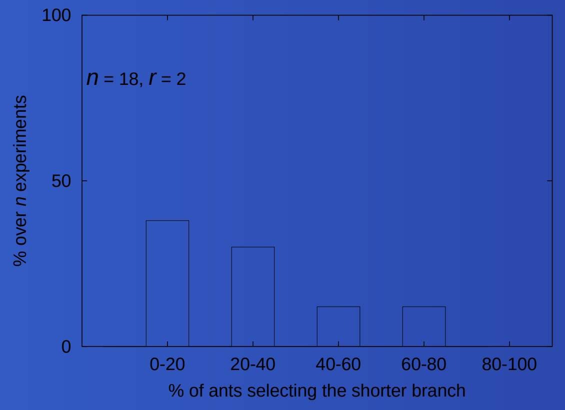
Ant repertoire vs. colony repertoire

- A single ant is capable of ***building a solution*** (cf. with PSO: every particle IS a solution point, or with the Immune System where none of the agents provides a solution)
- However, it is only the simultaneous presence and synergistic action of a swarm of ants that makes possible the shortest path finding behavior
- This ***self-organized behavior*** is a property of the colony and of the concurrent presence of all the discussed ingredients. It is not a property of the single ant

Solving shortest path problems is useful

- This class of problems is a very important one and encompasses a vast number of other problems. Graphs whose nodes represent possible alternatives/states and whose edges represent distances/losses/rewards/costs associated to node transitions are graphical models for a huge number of practical and theoretical decision and optimization problems
- In general, any ***combinatorial optimization*** or ***network flow*** problem can modeled in the terms of shortest paths
- Several general and very effective techniques to solve general shortest path problems are available: *label setting* techniques (e.g., ***Dijkstra algorithm*** [Dijkstra, 1959]), *label correcting* techniques (e.g. Bellman-Ford / ***dynamic-programming*** algorithms [Bellman, 1957, 1958; Ford and Fulkerson, 1962; Bertsekas, 1995], and *rollout algorithms* [Bertsekas, Tsitsiklis and Wu, 1997]. The literature on the subject is extensive
- The novelty of ants is that they solve it in a ***distributed*** and ***dynamic*** way using ***environment-mediated*** communication

Pheromone evaporation



The characteristics of the dynamic processes regulating ***pheromone evaporation*** play a central role determining the conditions for shortest path behavior and to favor *exploration* and *adaptation*

- Both pheromone evaporation and the characteristics of the ant ***stochastic decision policy*** are strictly related to the choice of the *tradeoff* between ***exploitation*** (e.g., of a path that seems to be particularly good) and ***exploration*** (e.g., of new alternative paths). This is a major issue for any system engaged in a search process repeated over time (e.g., ACO, Simulated Annealing)

- The literature on the mathematical and practical aspects of the *exploration/exploitation dilemma* (and on the related *bias/variance dilemma*) is vast [Kearns and Singh, 1998; Thrun, 1992; Robert and Casella, 1999; Kirkpatrick, Gelatt and Vecchi, 1982]

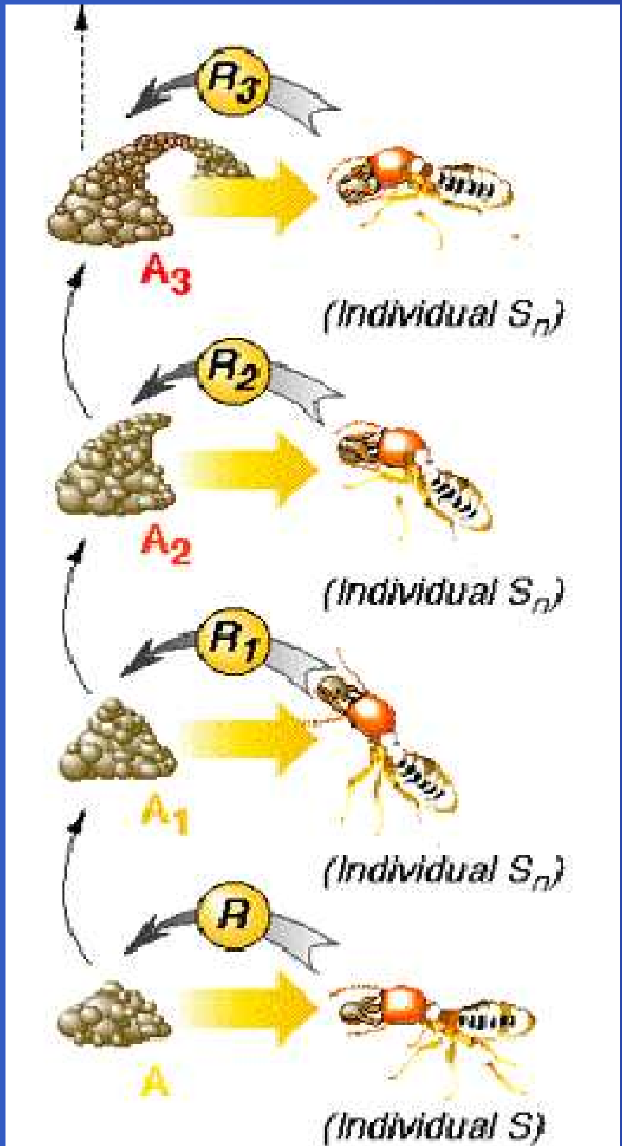
Stigmergy

- Stigmergy expresses the general idea of using indirect communication mediated by physical modifi cations of the environment to activate and coordinate self-organizing behaviors (in a colony of insects):

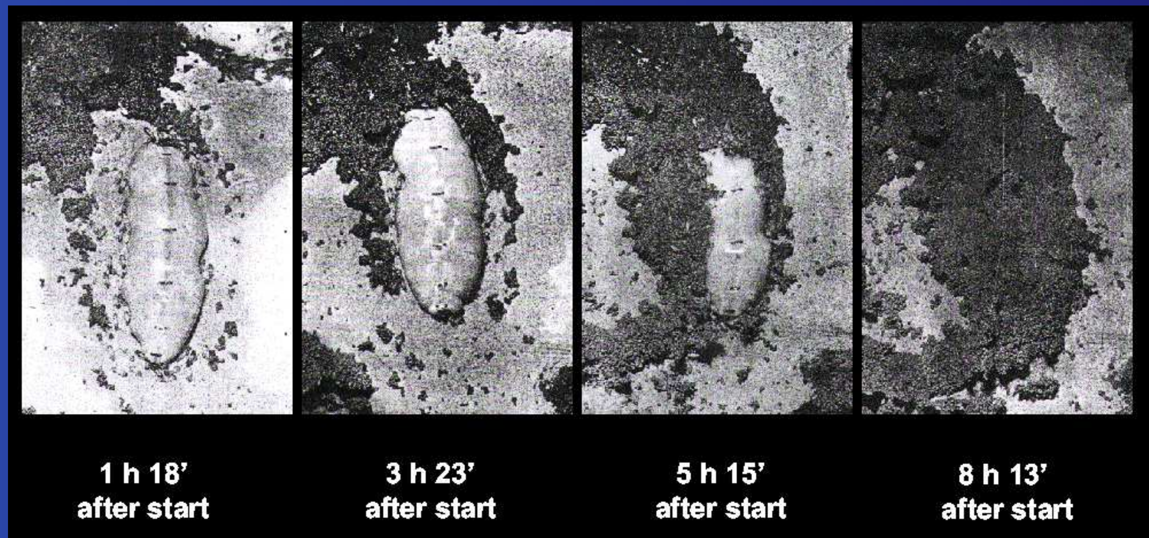
  The coordination of tasks and the regulation of constructions does not depend directly on the workers, but on the constructions themselves. The worker does not direct his work, but is guided by it. It is to this special form of stimulation that we give the name **stigmergy** (stigma, sting; ergon, work, product of labour = stimulating product of labour) [Grassé, 1959]

- Grassé observed that insects are capable to respond to so called *significant stimuli* which activate a genetically encoded reaction. In social insects, the effects of these reactions can act as new signifi cant stimuli for both the insect that produced them and for other insects in the colony, generating a recursive feedback that can lead to a phase of a ***global coordination***

Termite nest building



The combination of stigmergy and physical characteristics of the environment gives rise to astonishing results



A more operative definition

- We call ***stigmergy*** any form of indirect communication among a set of possibly concurrent and distributed agents which happens through acts of local modification of the environment and local sensing of the outcomes of these modifications [Dorigo, Di Caro and Gambardella, 1999]

- The local environment's variables whose value determine in turn the characteristics of the agents' response, are called ***stigmergic variables*** [Dorigo, Bonabeau and Theraulaz, 1999]. Stigmergic communication and the presence of stigmergic variables is expected (depending on parameter setting) to give raise to a synergy of effects resulting in ***self-organized global behaviors***

Examples of stigmergic variables

- The height of a pile of dirty dishes floating in the sink
- Pheromone intensity in trail-following and shortest path behavior in ant colonies
- Nest energy level in foraging robot activation [Krieger and Billeter, 1998]
- Level of customer demand in adaptive allocation of pick-up postmen [Bonabeau et al., 1997]

- Two main model of response: ***converging*** and ***diverging*** behavior at the group level

Threshold models of labor division

- Diverging behavior is explained by ***threshold models of labor division and task allocation***:

$$\begin{aligned} s &= \text{stimulus intensity} \\ \theta &= \text{internal response threshold} \\ T_\theta(s) &= \text{response function: probability of performing} \\ &\ \text{the task as a function of s} \end{aligned}$$

$$\begin{aligned} \text{if } s \ll \theta &\rightarrow T_\theta(s) \approx 0 \\ \text{if } s \gg \theta &\rightarrow T_\theta(s) \approx 1 \end{aligned}$$

- The internal response threshold can change over time according to some evolution or learning.
  A typical choice for T is:

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n}$$

Designing a stigmergy-based system

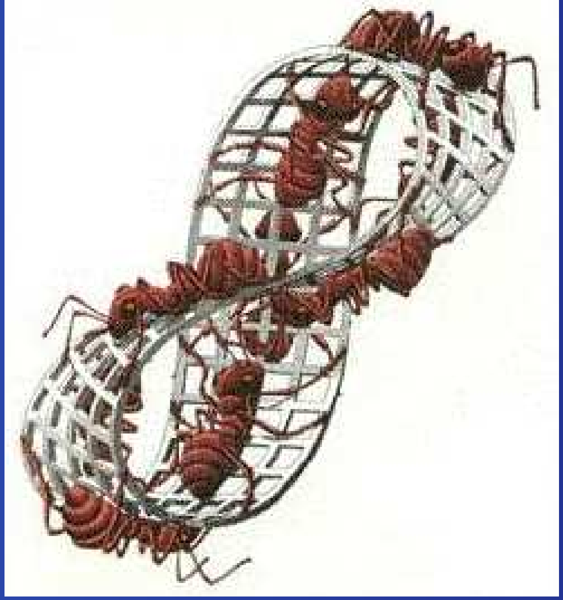
When a stigmergic approach is adopted to ***design a multi-agent system***, the focus is put on the defi nition of:

- effective stigmergic variables, such that robust coordination and synergy can result,
- an effective system of realizing in practice the indirect agent communication ***by exploiting environment structure***

- The central issue is the defi nition of the***protocols*** (interfaces), rather than of the *modules* (agents) of the system [Csete and Doyle, 2002] (simple, cheap?)

- Protocols here are rules that prescribe allowed interfaces between modules, permitting system functions that ***could not be achieved by isolated modules***
- Protocols facilitate the addition of new protocols, and simplify modeling and abstraction
- A good protocol supplies ***global robustness, scalability, evolvability***, and, in the end, allows to fully exploit the potentialities of the modules and of modularity

ACO and other (meta)heuristics for CO



Outline

- Notes on combinatorial optimization and algorithmic complexity
- Construction and modification metaheuristics: two complementary ways of searching a solution space
- Finally, Ant Colony Optimization explained
- Other population-based metaheuristics for optimization problems: Genetic Algorithms, Cultural Algorithms, Cross-entropy
- Single agent metaheuristics: Local Search, Tabu Search, Rollout algorithms