Swarm Intelligence,
Nature's way to system engineering

Gianni Di Caro

gianni@idsia.ch

IDSIA, USI/SUPSI, Lugano (CH)

Road map

- Basic ideas behind the notion of Swarm Intelligence
- Nature's examples
- Characteristics of Swarm Intelligence design
- Instances of Swarm Intelligence design: COIN and PSO
- Stigmergy and self-organization in insect societies
- The Ant Colony Optimization (ACO) metaheuristic
- Other metaheuristics for combinatorial optimization problems
- Application of ACO to routing in wired networks
- Application of ACO to routing in mobile ad hoc networks (F. Ducatelle)

$\rightarrow$ These overheads are redundant in number and words, but are intended to be a sort of booklet to be used later as a reference, if you will get interested in the topics which will be discussed

Swarm Intelligence: what's this?

- A computational and behavioral metaphor for problem solving that originally took its inspiration from the Nature's examples of collective behaviors (from the end of '90s)

  - *Social insects (ants, termites, bees, wasps): nest building, foraging, assembly, sorting,. . .*
  - *Vertebrates: swarming, flocking, herding, schooling*

- *Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insects and other animal societies [Bonabeau, Dorigo and Theraulaz, 1999]*

- . . . however, we don't really need to "stick" on examples from Nature, whose constraints and targets might differ profoundly from those of our environments of interest . . .

Nature’s examples of SI

Fish schooling (©CORO, CalTech)

Nature’s examples of SI (2)

Birds flocking in V-formation (©CORO, Caltech)

Nature’s examples of SI (3)

Termites’ nest (©Masson)

Nature’s examples of SI (4)

Honeybees’ comb (©S. Camazine)

Nature's examples of SI (5)

Swarm of killer bees

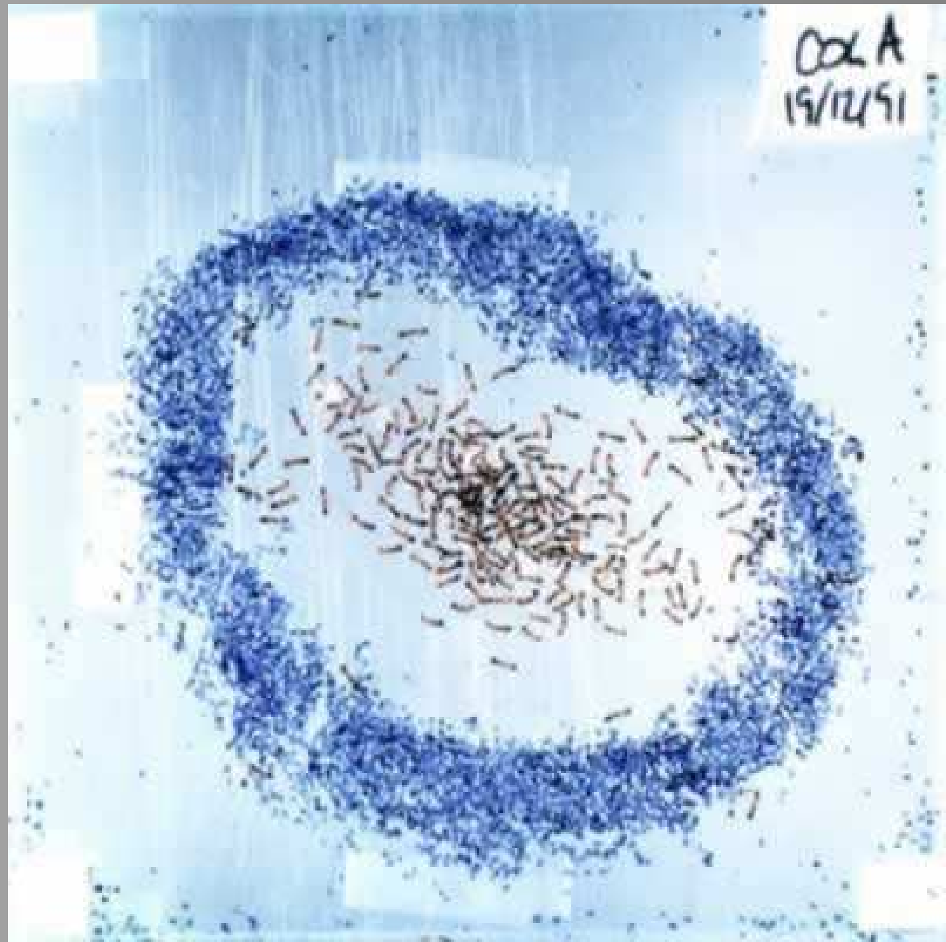
Nature’s examples of SI (6)

Killer bees (©S. Camazine)

Bees’ nest (©S. Camazine)

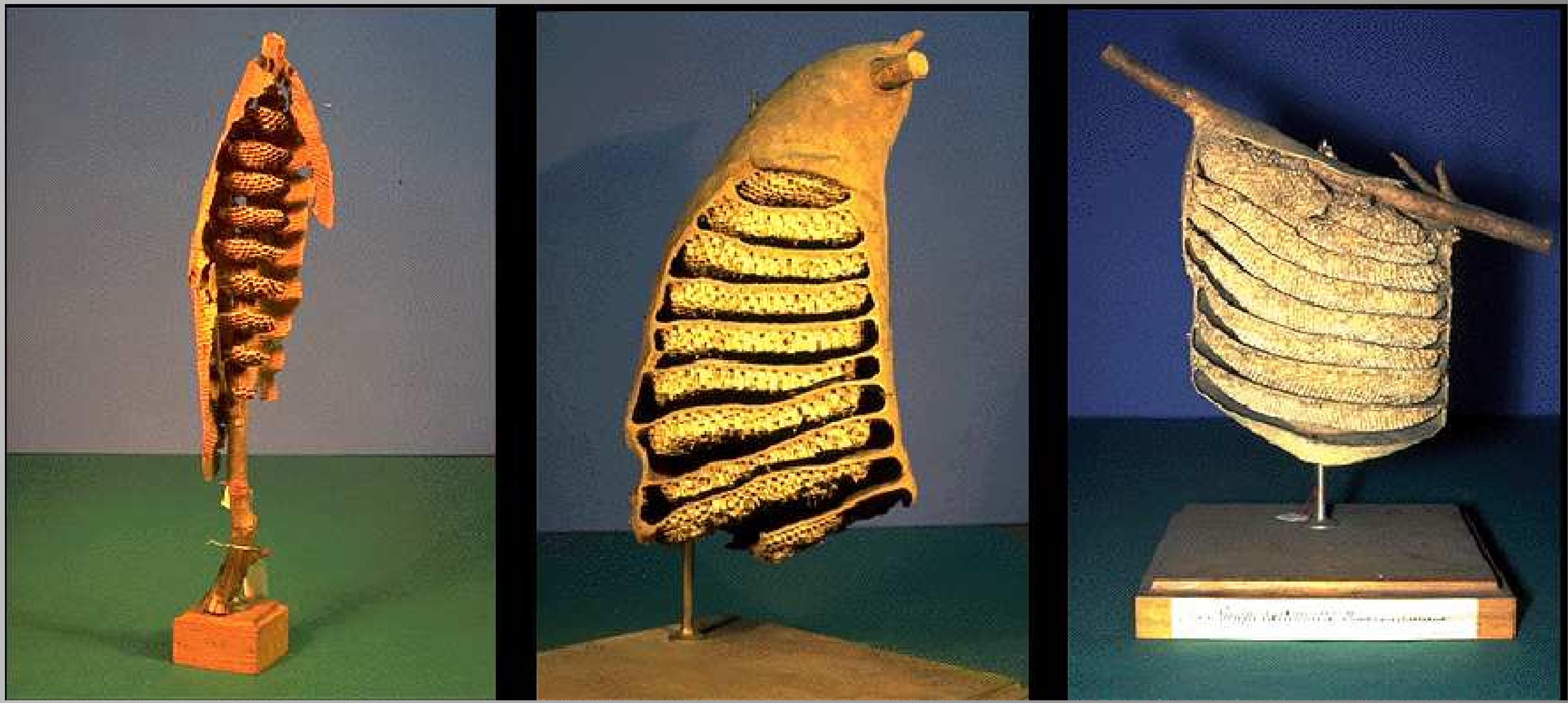
Nature’s examples of SI (7)

Ant chain (©S. Camazine)



Ant wall (©S. Camazine)

Nature’s examples of SI (8)



Wasps’ nest (©G. Theraulaz)

Nature's examples of SI (9)

- Ants: leaf-cutting, breeding, chaining
- Ants: Food catering
- Bees: scout dance

What all these behaviors have in common?

- Distributed society of autonomous individuals/agents
- Control is fully distributed among the agents
- Communications among the individuals are localized
- Stochastic agent decisions (partial/noisy view)
- System-level behaviors appear to transcend the behavioral repertoire of the single (minimalist) agent
- Interaction rules seem to be simple
- The overall response of the system features:
  - Robustness
  - Adaptivity
  - Scalability

I had a dream . . .

. . . I can generate *complexity out of simplicity*: I can put all the previous ingredients in a pot, boil them down and get good, robust, adaptive, scalable algorithms for my problems!

. . . it reminds me of alchemists . . .

. . . but it's just about design choices

There's no magic!

- Task complexity is a *conserved* variable
- Given a *Problem + Constraints + Costs + Optimization Criteria*: How do I solve it?
- The final *design choice* is usually a rather obscure match between designer's expertise, problem's characteristics, constraints and targets

Swarm Intelligence design means . . . ?

- Allocating computing resources to a (large?) number of minimalist units (swarm?)
- No centralized control (not at all?)
- Units interact in a simple and localized way
- Units do not need a representation of the global task
- Stochastic components are important
- . . . and let generate useful global behaviors by *self-organization*

Modular design shifting complexity from modules to protocols

The dark side of SI design

- *Predictability is a problem in distributed bottom-up approaches*
- *Efficiency is another issue* (BTW, are ants efficient?)
- *What's the overall cost?* (self-organization is dissipative)
- *Sometimes is a* lazy *shortcut to problem solution*
- *Loads of parameters to assign* (e.g., how many agents?)
- *Nature had millions of years to "design" effective systems by ontogenetic and phylogenetic evolution driven by selection, genetic recombination and random mutations*, but we have less time. . .

A (tentative) more general definition of SI

- Given a set of $N \gg 1$ communicating and *distributed autonomous agents* (e.g., cells, ants, communication devices, computer processes) each engaged in one or more tasks, and with *no or little centralized control*,
- if from the *local interactions* among the agents results a process of *self-organization* that gives rise to *interesting/useful behaviors at the system level*, we can say that we are observing a phenomenon of *Swarm Intelligence*

- Does *Collective Intelligence* sound better?

- Do we need restriction on aspects like: Nature-inspiration, short-range locality, agent simplicity, awareness of global task, homogeneity,...? $\longrightarrow$ *Parameters*

Key elements of SI

- The swarm lives distributed in some abstract or real *space*
- *Local communication* allows to get nonlinear global behaviors
- Structures resulting from individuals' local interactions develops by a process of *Self-organization*

Self-organization: definitions

- *Self-organization consists of set of dynamical mechanisms whereby structure appears at the global level as the result of interactions among lower-level components. The rules specifying the interactions among the system's constituent units are executed on the basis of purely local information, without reference to the global pattern, which is an emergent property of the system rather than a property imposed upon the system by an external ordering influence [Bonabeau et al., 1997]*

- *More general: any dynamic system from which order emerges entirely as a result of the properties of individual elements in the system, and not from external pressures (e.g., Beńard cellular convection, Belousov-Zhabotinski reactions)*

- *In more abstract terms: self-organization is related to an increase of the statistical complexity of the causal states of the process [Shalizi, 2001]: when a number of units have reached organized coordination, it is necessary to retain more information about the inputs in order to make a statistically correct prediction*

Characteristics of self-organization (in biology)

Basic ingredients:

- Multiple interactions
- Amplifi cation of fluctuations and Randomness
- Positive feedback (e.g., recruitment and reinforcement)
- Negative feedback (e.g., limited number of available foragers)

Signatures:

- Creation of spatio-temporal structures (e.g., foraging trails, nest architectures, social organization)
- Multistability (e.g., ants exploit only one of two food sources)
- Existence of bifurcations when some parameters change (e.g., termites move from a non-coordinated to a coordinated phase only if their density is higher than a threshold value)

IDSIA

Main forms of communication

- *Point-to-point*: antennation, trophallaxis (food or liquid exchange), mandibular contact, direct visual contact, chemical contact, . . . unicast radio contact!
- *Broadcast-like*: the signal propagates to some limited extent throughout the environment and/or is made available for a rather short time (e.g., use of lateral line in fishes to detect water waves, generic visual detection, actual radio broadcast
- *Indirect*: two individuals interact indirectly when one of them modifies the environment and the other responds asynchronously to the modified environment at a later time. This is called *stigmergy* [Grassé, 1959] (e.g., pheromone laying/following, post-it, web)

Ant algorithms, Particle swarms and . . .

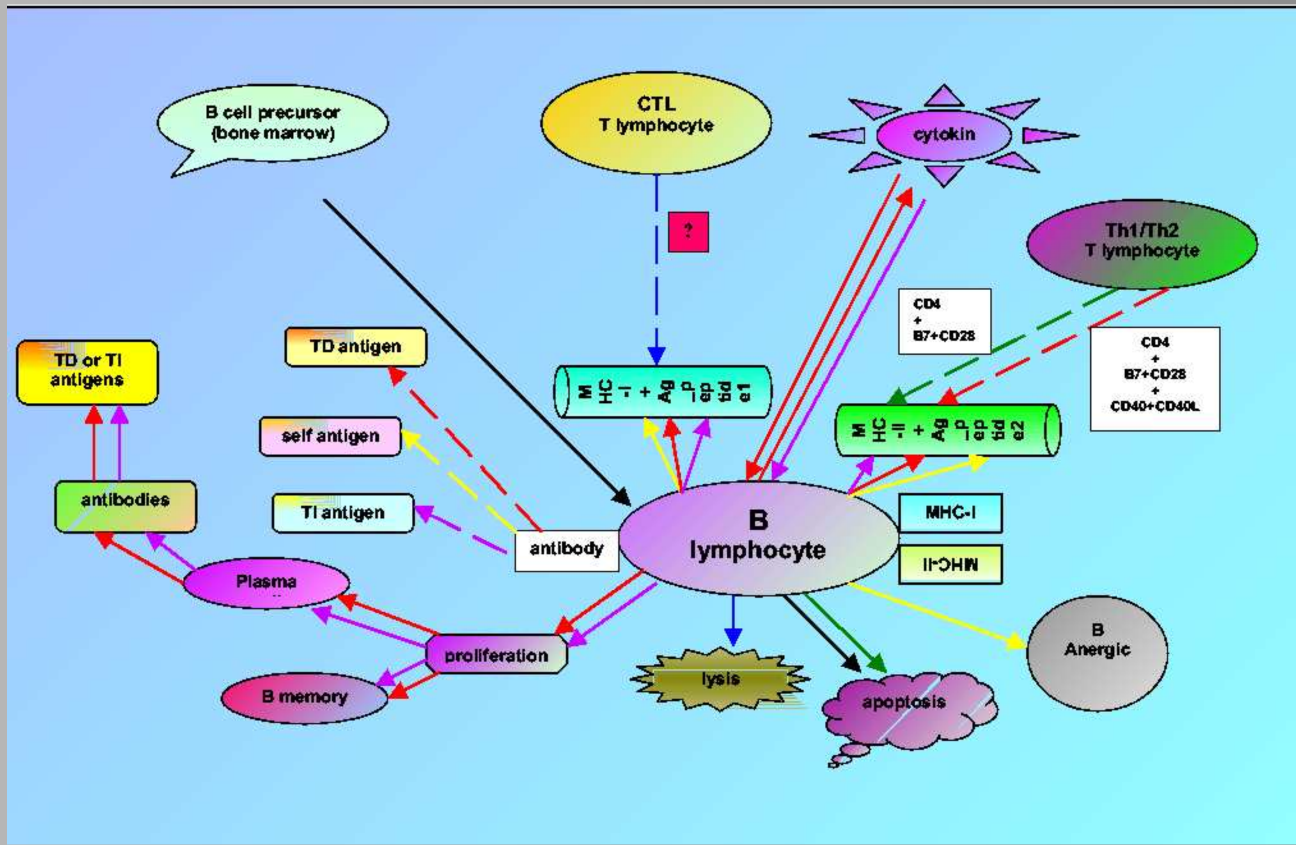
- *Stigmergy* has led to *Ant Algorithms* and in particular to *Ant Colony Optimization (ACO)* [Dorigo & Di Caro, 1999]

- *Broadcast-like* communication is related to schooling and flocking behaviors, that have inspired *Particle Swarm Optimization* [Kennedy & Eberhart, 2001]. Neighbor broadcast is also at the basis of *Cellular Automata* [Wolfram, 1984], one of the early examples of swarm computation

- The use of all the three forms of communication encompasses more general systems showing collective organized behaviors (*COIN, immune systems, cultural algorithms, neural systems, human organizations, mobile ad hoc networks*,. . .)

Examples of SI from biology

- *Immune system*: high diversity, mobility, distributed, dynamic, pipelined strategies, several communication strategies, multi-objective, learning, memory . . .



- *Brains, slime molds, gene regulatory networks* . . .
- *Our body*: a swarm of swarms [Hoffmeyer, 1995]

. . . and from “us”

- *Routing in communications networks*: a system of distributed and adaptive controllers search online for good communication paths between computers
- *Artificial Neural Networks*: artificial neurons connected through artificial synapses that learn to approximate functions, solve classification tasks, control robot motion,. . .
- *Artificial Immune Systems*: patterns (antigens) identification and memorization, coupled with proliferation of pattern matching agents (antibodies), are used for intrusion detection and removal, and content search [Ganguly & Deutsch, 2005]
- *Crowd control*: rush hours in Tokyo’s Shinjuku station, movies like ANTZ or Titanic
- *Economy*: a new view beyond classical economics which denies the direct interaction between individuals [Ormerod, 1998]; *empowerment* in companies

Collective robotics

- *Collective robotics* is attracting a lot of interest: groups of robots play soccer (RoboCup), unload cargo, patrol vast areas, cluster objects, self-assemble (Swarm-bots), and will maybe “soon” participate to war :-(. . .
  - Robot assembly to cross a gap
  - Robot assembly to transport a prey
- Look at RoboCup (www.robocup.org) and Swarm-bots (www.swarm-bots.org)!

Back to the algorithmic frameworks. . .

- *Ant Colony Optimization* (ACO) and *Particle Swarm Optimization* (PSO) are the most popular frameworks based on the original notion of SI (CA?)

- At the core of the design of ACO and PSO there is the specifi c way the agents communicate in the *spatial environment*. These two *optimization frameworks* focus on two different ways of distributing, accessing and using *information in the environment*

- In ACO and PSO agents are rather simple, since they *do not learn at individual level*

- The agents in the *Collective Intelligence* (COIN) framework are reinforcements learners, therefore they can be arbitrarily complex. COIN's design focuses on *generic multi-agent reinforcement learning*. Focus on the role of distributing and managing *utility functions/values* among the agents

- On a complexity scale, from the simplest: (CA, NN), PSO, ACO, COIN

Other related frameworks/keywords

Just names and buzzwords here:

Distributed Artificial Intelligence, Computational Economics, Multi-player Cooperative Game Theory, Evolutionary Computation (Population-based), Artificial Life, Statistical Physics, Markov Fields, Network Theory, Neural Networks, Traffic Theory . . .

(see [Wolpert and Tumer, 2000] for references and discussions)

COIN

The *Collective Intelligence* framework [Wolpert and Tumer, 2000] consists of:

- A large multi-agent system,
- where there is little to no centralized, personalized communication and/or control,
- there is a provided world utility function that rates the possible histories of the full system,
- each agent “runs” a reinforcement learning algorithm (microlearning).

COIN

- Central issues in COIN: *how to map the world utility function into private utility functions for each of the agents?* How the private utility functions can be designed so that each agent can realistically hope to optimize its function, and at the same time the collective behavior of the agents can optimize the world utility?

- The assignment of the rewards to the agents is a critical aspect in *multi-agent reinforcement learning systems*

COIN

- COIN focuses on an *inverse problem*: how to confi gure local dynamical laws and the management of the system-level utility in order to induce the desired global behavior.
  This is the ultimate dream of every engineer dealing with complex systems

- Fixing the agent characteristics and studying the response to different world utilities and distribution of them is also extremely useful: *Economics*!

- COIN is a general mathematical framework. Not straightforward to understand. It points out where the problems are and provides formal tool to reason. However, does not provides straight or automatic design answers. It has been applied to routing and game problems. Not really popular, but worth to give a look at

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, 1984]

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, such that each automaton a_i has n_i neighbors $\mathcal{N}(a_i) = \{a_i^1, a_i^2, \ldots, s_i^{n_i}\}$
- 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

PSO: background

- Early work on *simulation of bird flocking* aimed at understanding the underlying rules of flocking [Reynolds, 1984] and roosting behavior [Heppner & 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 $N >> 1$ 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*

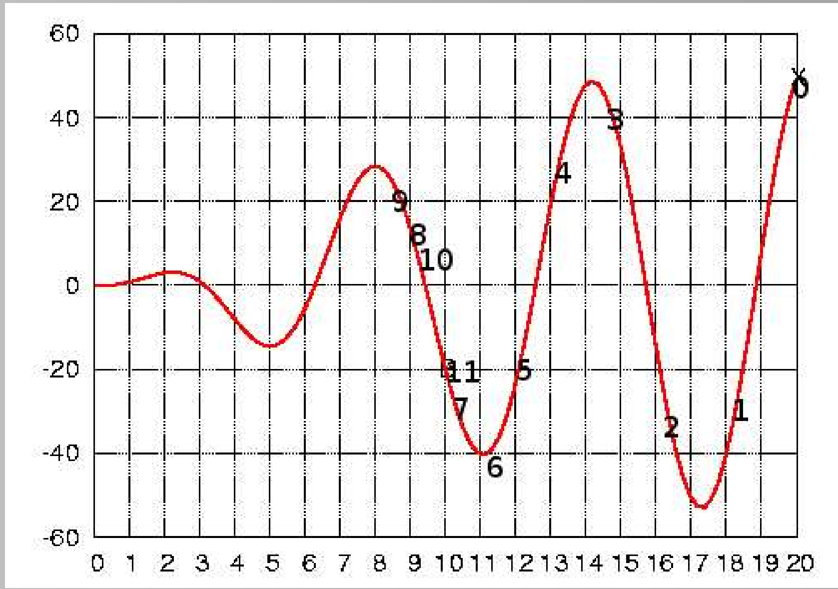
PSO: the meta-algorithm

- *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*

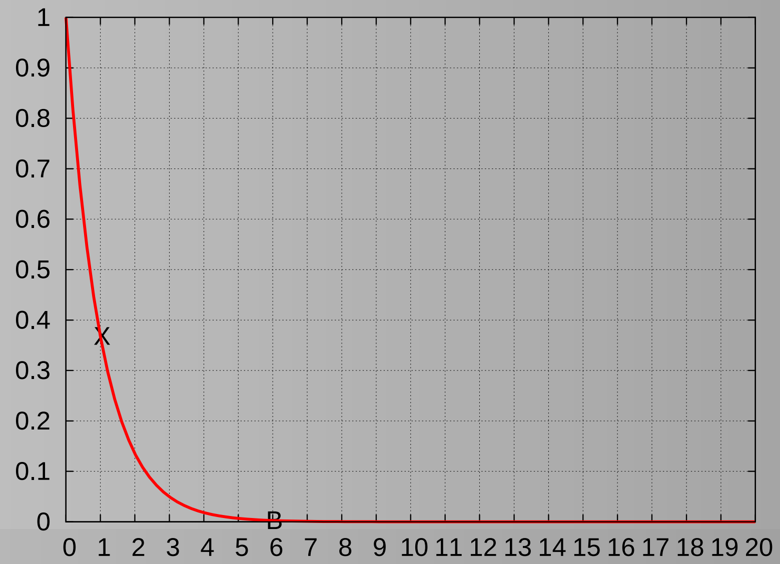
procedure Particle_Swarm_Optimization$()$
  foreach $particle \in ParticleSet$ **do**
    init at random positions and velocity;
    select at random the neighbor set;
  end foreach
  while $(\neg\, stopping\_criterion)$
    foreach $particle \in ParticleSet$ **do**
      calculate current fi tness and update memory;
      get neighbor with best fi tness;
      calculate individual deviation between current and best so far fi tness;
      calculate social deviation between current and best neighbor fi tness;
      calculate velocity vector variation as weighted sum between deviations;
      update velocity vector;
    end foreach
  end while
return $best\_solution\_generated$;

PSO: 1D example, one particle

One particle behavior: sequence of x and x_{best} values



20.00	10.00
18.21	18.21
16.43	16.43
14.64	16.43
13.24	16.43
12.03	16.43
11.06	11.06
10.09	11.06
9.71	11.06
8.85	11.06
9.14	11.06
10.13	11.06



1.0	6.0
1.8	6.0
4.4	6.0
7.2	7.2
10.0	10.0
12.8	12.8
15.6	15.6
18.4	18.4
21.2	21.2
23.9	23.9
26.7	26.7

Some final considerations on PSO

- Equivalent to a real-valued 2D CA where the state of a particle is $(\bar{x}, \bar{v}) \cup (\bar{x}_{best}, f(\bar{x}_{best}, \bar{x}_{best}(\mathcal{N}))$
- 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

Stigmergy and Ant-inspired algorithms

- Stigmergy is at the core of most of all the amazing collective behaviors exhibited by the ant/termite colonies
- Grassé (1959) introduced this term to explain *nest building in termite societies*
- Goss, Aron, Deneubourg, and Pasteels (1989) showed how stigmergy allows *ant colonies to find shortest paths* between their nest and sources of food
- These mechanisms have been reverse engineered to give raise to a multitude of ant colony inspired algorithms based on stigmergic communication and control
- The *Ant Colony Optimization metaheuristic (ACO)* [Dorigo & Di Caro, 1999] is the most popular, general, and effective SI framework based on these principles

Few facts about Social Insects

- Social insects :
  - *Ants*
  - *Termites*
  - Some *bees*
  - Some *wasps*
- 10^{18} living insects (rough estimate)
- 2% of insect are social and most of them are eusocial
- 50% of all social insects are ants
- Total weight ants $\approx$ Total weight humans
  (one ant $1 \div 5$ mg)
- Ants are successfully around since 100 million years, *Home sapiens sapiens* only since 50,000 years

Ant colonies

- Ant colony size: from as few as 30 to millions of workers
- *Work division*:

Reproduction	$\longrightarrow$	Queen
Defense	$\longrightarrow$	Specialized workers
Defense	$\longrightarrow$	Soldiers
Food collection	$\longrightarrow$	Specialized workers
Brood care	$\longrightarrow$	Specialized workers
Nest brooming	$\longrightarrow$	Specialized workers
Nest building	$\longrightarrow$	Specialized workers

Some interesting collective ant behaviors

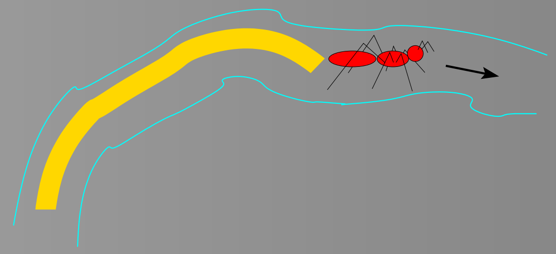
- Nest building and maintaining
- Division of labor and adaptive task allocation
- Discovery of shortest paths between nest and food
- Clustering and sorting (e.g., dead bodies, eggs)
- Structure formation (e.g., deal with obstacles)
- Recruitment for foraging (tandem, group, mass)
- Cooperative transport (e.g., food)

. . . and solitary ones: Ant navigation

- Depends on the sensorial capabilities of ant species as well as on the characteristics of the environment and function within the colony. Can make use of:
  - Visual landmarks (use of memory and learning, encounters with colony mates)
  - Chemical landmarks (pheromone)
  - Compass-based (e.g., *Cataglyphis* desert ant uses light polarization)
  - Dead-reckoning, path integration (calculation of the home vector)
  - Correlated random walk

Pheromone laying-attraction is the key

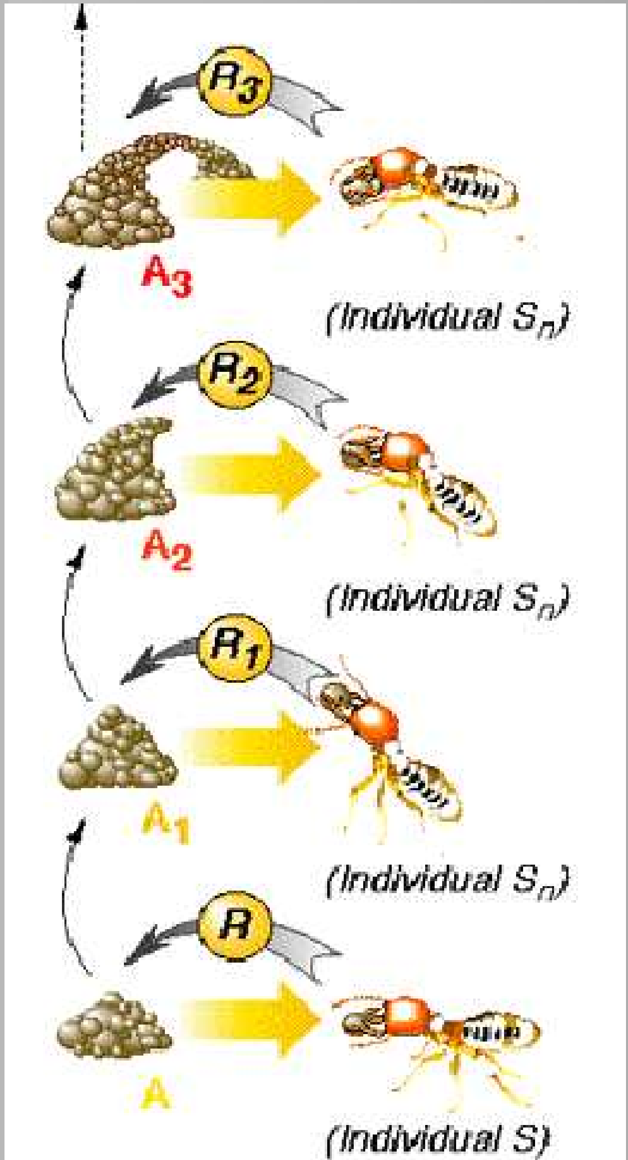
- While walking, the ants lay on the ground a volatile chemical substance, called *pheromone*



- Pheromone distribution *modifi es the environment*(the way it is perceived) creating a sort of *attractive potential fi eld*for the ants

- *This is useful for retracing the way back, for mass recruitment, for labor division and coordination, to fi nd shortest paths. . .*

Termite nest building



- Grassé observed that insects are capable to respond to so called *signifi cant stimuli*which activate a genetically encoded reaction. In turn, this reaction as new signifi cant stimuli, generating a *recursive feedback* that can lead to a phase of a global coordination
- Stigmergy = incite to work

Stigmergy and stigmergic variables

- *Stigmergy* means 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

- The local environment's variables whose value determine in turn the characteristics of the agents' response, are called *stigmergic variables*

- Stigmergic communication and the presence of stigmergic variables is expected (depending on parameter setting) to give raise to a *self-organized global behaviors*

- *Blackboard/post-it, style of asynchronous communication*

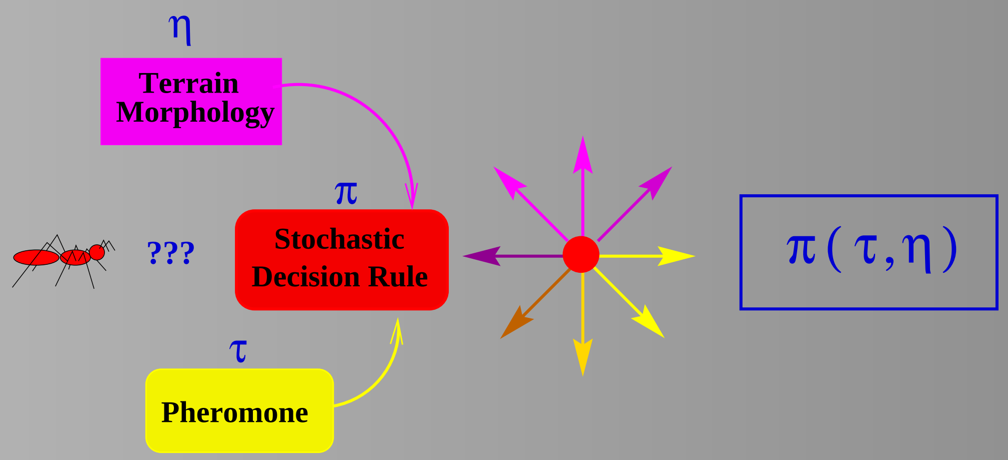
Examples of stigmergic variables

- *Leading to diverging behavior at the group level:*
  - The height of a pile of dirty dishes floating in the sink
  - 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]

- *Leading to converging behavior at the group level:*
  - Intensity of pheromone trails in ant foraging: shortest paths!

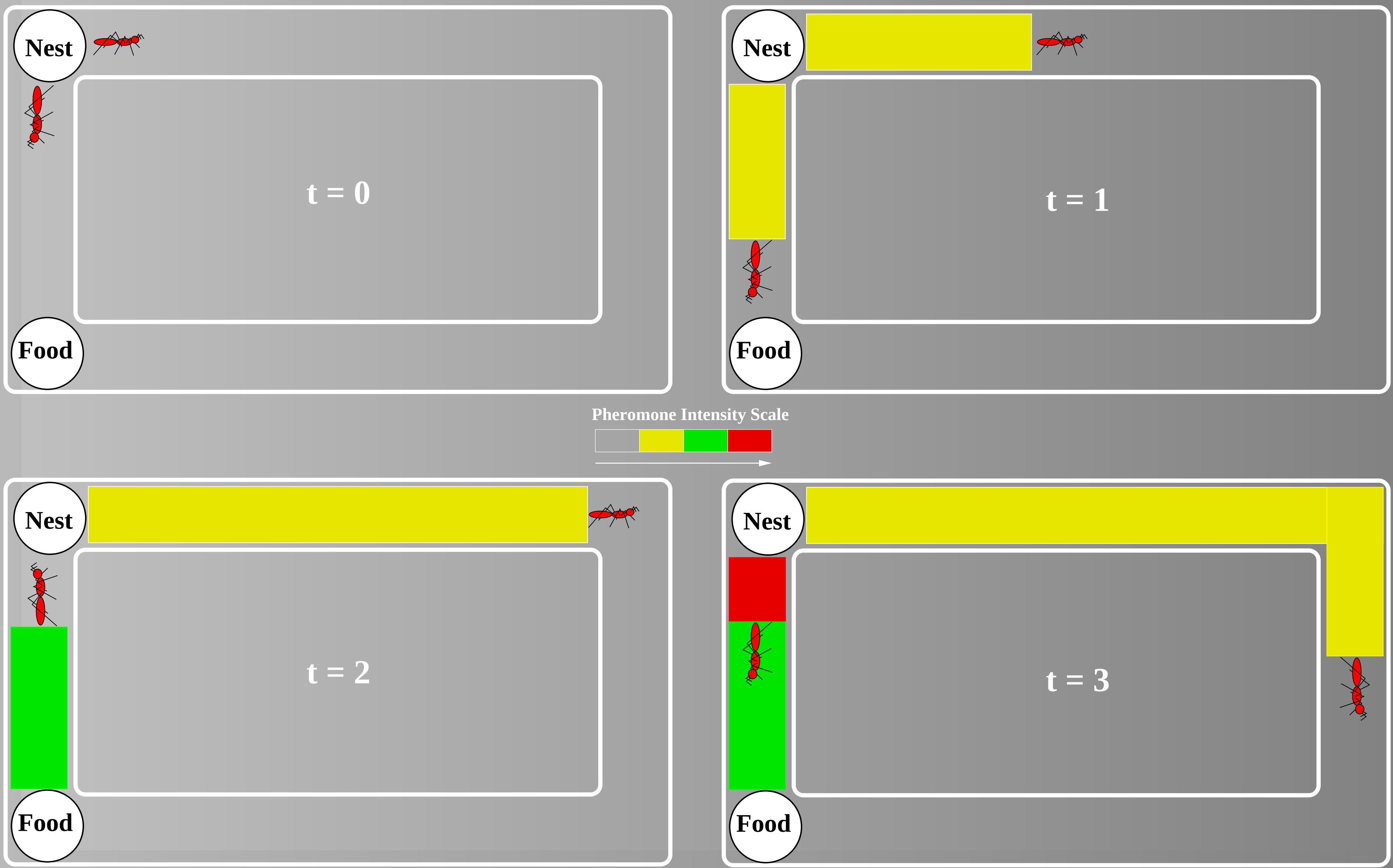
Shortest path behavior in ant colonies

- While walking, at each step a *routing decision* is issued. Directions locally marked by *higher pheromone intensity* are preferred according to some *probabilistic rule*:

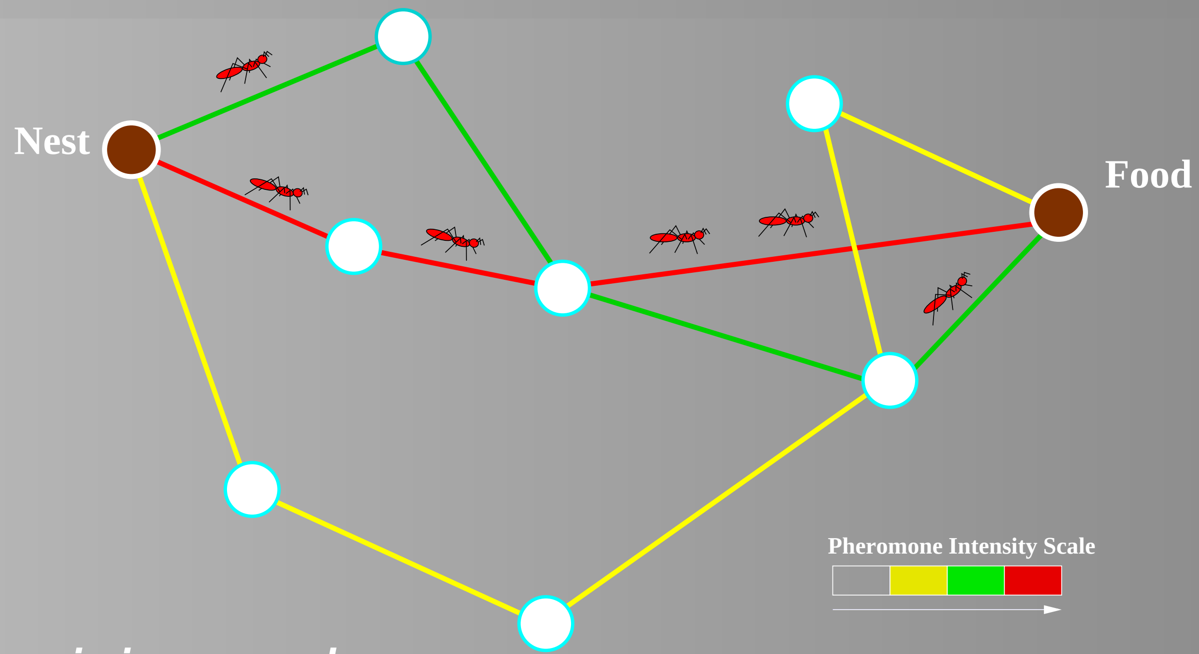


- This basic pheromone laying-following behavior is the main ingredient to allow the colony converge on the *shortest path* between the nest and a source of food

Ant colonies: Pheromone and shortest paths



Ant colonies in a more complex discrete world



- Multiple *decision nodes*
- A path is *constructed* through a *sequence of decisions*
- Decisions must be taken on the basis of local information only
- A traveling *cost* is associated to node *transitions*
- *Pheromone intensity locally encodes decision goodness as collectively estimated by the repeated path sampling*
- Are ant colonies able to "solve" such complex problems?

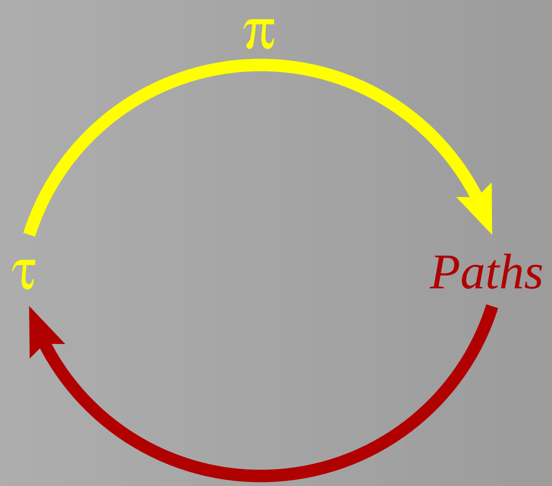
Ant colonies: Ingredients for shortest paths

- A number of concurrent autonomous (simple?) agents (ants)
- Forward-backward path following/sampling
- Multiple paths are tried out and implicitly evaluated
- Local laying and sensing of pheromone
- Stochastic step-by-step decisions biased by pheromone
- Positive feedback effect (local reinforcement of good decisions)
- Persistence / evaporation of the pheromone field
- Iteration over time of the path sampling actions
- Convergence onto the shortest path?

What pheromone represents in abstract terms?

- *Distributed, dynamic, and collective memory of the colony*
- *Learned goodness of a local move (routing choice)*
- Circular relationship: pheromone trails modify environment $\longrightarrow$ locally bias ants decisions $\longrightarrow$ modify environment

Pheromone distribution biases path construction



Outcomes of path construction are used to modify pheromone distribution

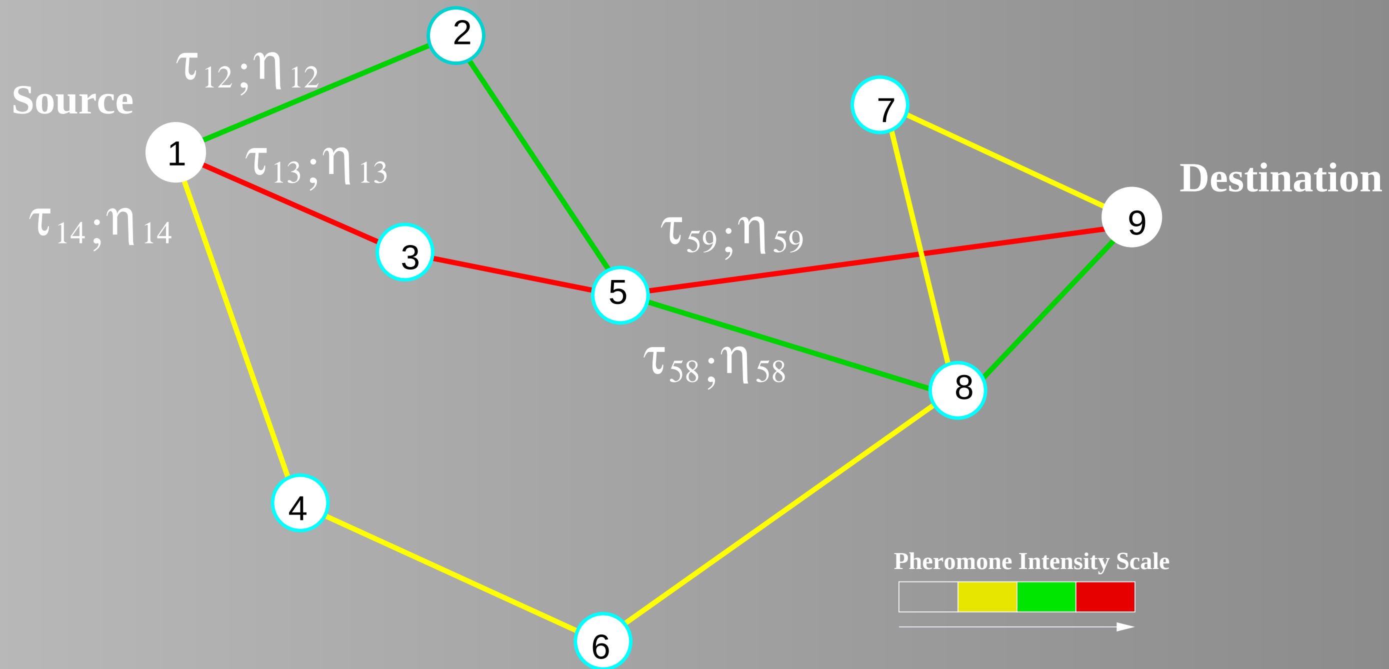
A meta-strategy for shortest path problems

- By reverse engineering ant colonies' shortest path behavior we get an effective metaheuristic, *ACO*, based on repeated path sampling and distributed/collective decision learning through reinforcements, *to solve shortest path problems* . . .
- . . . in a possibly *fully distributed and adaptive way*
- . . . and we know that shortest paths are a very general model for *combinatorial optimization and decision problems!*

ACO: general architecture

procedure ACO_metaheuristic()
  while ($\neg$ *stopping_criterion*)
    schedule_activities
      ant_agents_construct_solutions_using_pheromone();
      pheromone_updating();
      daemon_actions();   /* OPTIONAL */
    end schedule_activities
  end while
return *best_solution_generated*;

ACO: From natural to artificial ant colonies(1)

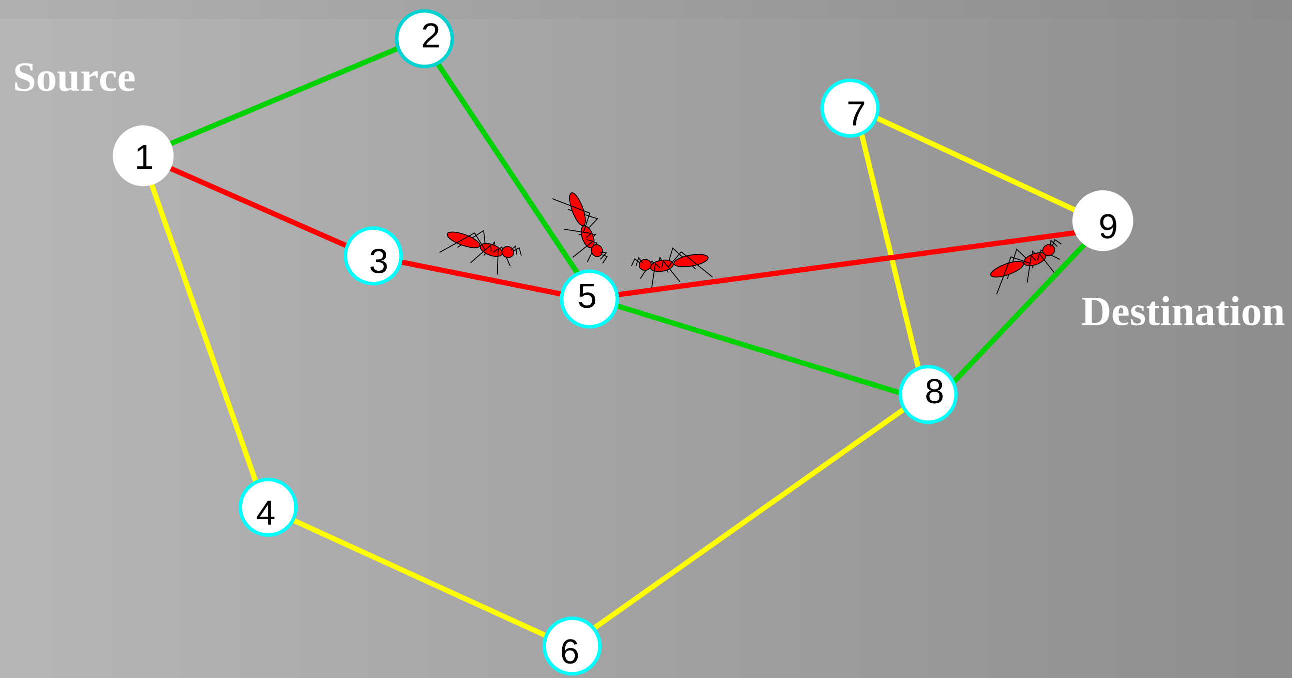


- Each node i holds an *array of pheromone variables:*
  $\vec{\tau}_i = [\tau_{ij}] \in \mathbf{R}, \forall j \in \mathcal{N}(i)$ $\quad\longrightarrow$ Learned through path sampling

- and an equivalent array of *heuristic variables:*
  $\vec{\eta}_i = [\eta_{ij}] \in \mathbf{R}, \forall j \in \mathcal{N}(i)$ $\quad\longrightarrow$ Resulting from other sources $\uparrow$

- $\tau_{ij} = q(j|i)$: estimated *quality/goodness/utility* of moving to next node j conditionally to the fact of being in i

ACO: From natural to artificial ant colonies(2)



- Each ant is an *autonomous agent that constructs a path* $\mathcal{P}_{1\rightarrow 9}$ $\longrightarrow$ proposes a solution to the problem
- There might be one or more ants *concurrently active* at the same time. Ants do not need synchronization
- Next hops are selected through a *stochastic decision policy*

$$\boxed{\pi_{\epsilon}(i;\ \vec{\tau}_i, \vec{\eta}_i)}$$

ACO: Ant-routing table and decision policy

- The values of τ_i and η_i at each node i must be combined and given a relative weight in order to assign a precise goodness value to each locally available next hop $j \in \mathcal{N}(i)$:

$$\boxed{\mathcal{A}_i(j) = f_\tau(\boldsymbol{\tau}_i, j) \circ f_\eta(\boldsymbol{\eta}_i, j)}$$

- $\mathcal{A}_i(j)$ is called the *(Ant-routing table)*: it summarizes all the information locally available to make next hop selection.

  Examples: $\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}, \quad \alpha\tau_{ij} + (1-\alpha)\eta_{ij}$

- $\mathcal{A}_i(j)$ values are used by π_ϵ to take a probabilistic decision:
  - *Example - Random-proportional:* $p_{ij} = \dfrac{\mathcal{A}_i(j)}{\sum_{k\in\mathcal{N}(i)} \mathcal{A}_i(k)}$
  - *Example - ϵ-greedy:*

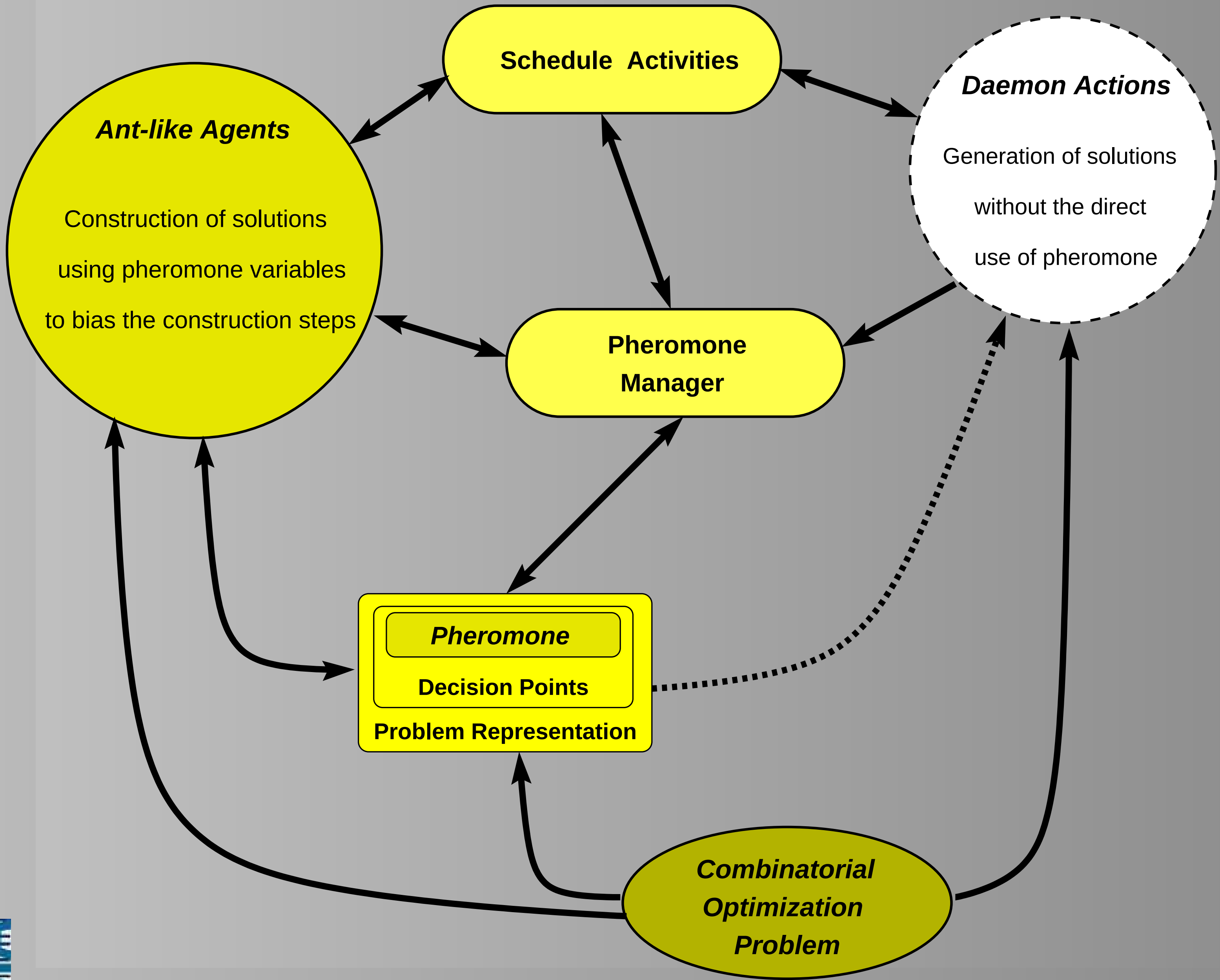
$$\begin{cases} \textit{if } p_b > p_u : & p_{ij} = 1 \textit{ if } j = \arg\max \mathcal{A}_i(j),\ 0 \textit{ otherwise} \\ \textit{else} : & p_{ij} = 1/|\mathcal{N}(i)|,\ \forall j \in \mathcal{N}(i) \end{cases}$$

ACO: Some important issues to clarify. . .

- What the decision nodes / pheromone variables represent (states?)

- When and how pheromone variables are updated (evaluation)

ACO's logical diagram can help to understand



Decisions are based on state features

- Pheromone variables represent the decision variables that are the object of learning
- In principle we should try to learn good state transitions: $\uparrow$ $\tau_{ij} = q(x_j|x_i)$, or, equivalently $\tau_{ij} = q(c_j|x_i)$
  - Computationally unfeasible:
    number of decision variables $>$ number of states
    (neuro-dynamic programming?)
  - It's hard to learn. . .
- *The alternative is to trade optimality for efficiency using* state features *instead of the full states:* $\tau_{ij} = q(c_j|\rho(x_i))$
  The available state information can be used for feasibility $\uparrow$

Pheromone updating

- Ants update pheromone *online step-by-step* $\longrightarrow$ Implicit path evaluation based on on traveling time and rate of updates
- Ant's way is inefficient and risky
- The right way is *online delayed* + *pheromone manager filter*:
  - *Complete the path*
  - *Evaluate*
  - *"Retrace" and assign credit / reinforce the goodness value of the decision (pheromone variables) that built the path*
  - *Total path cost can be safely used as reinforcement signal*

    Example TSP: $s = (c_1, c_3, c_5, c_7, c_9), \quad J(s) = L$

    $$\tau_{13} \leftarrow \tau_{13} + 1/L, \quad \tau_{35} \leftarrow \tau_{35} + 1/L, \ldots$$
- Online step-by-step decrease for exploration (e.g., ACS)
- If states: online step-by-step + bootstrapping is ok
- *Offline*: *daemon*, *evaporation*: $\tau_{ij} \leftarrow \rho\tau_{ij}, \; \rho \in [0, 1]$,

Designing an ACO algorithm

- Representation of the problem $\rightarrow$ pheromone model $\vec{\tau}$
- Heuristic variables $\vec{\eta}$
- Ant-routing table $\mathcal{A}$
- Stochastic decision policy π_ϵ
- Solution evaluation $J(s)$
- Policies for pheromone updating
- Scheduling of the ants
- Daemon components
- Pheromone initialization, constants, $\ldots$

Best choices for static/centralized problems

- Last component as state feature for the pheromone model
- Problem's costs or lower bounds as heuristic variables
- Multiplicative or additive ant-routing functions
- ϵ-greedy a random proportional for decisions
- *Elitist strategies for pheromone updating*
- Few ants at-a-time for a large number of iterations
- *Problem-specific local search daemon procedures*
- *Bounded pheromone ranges*

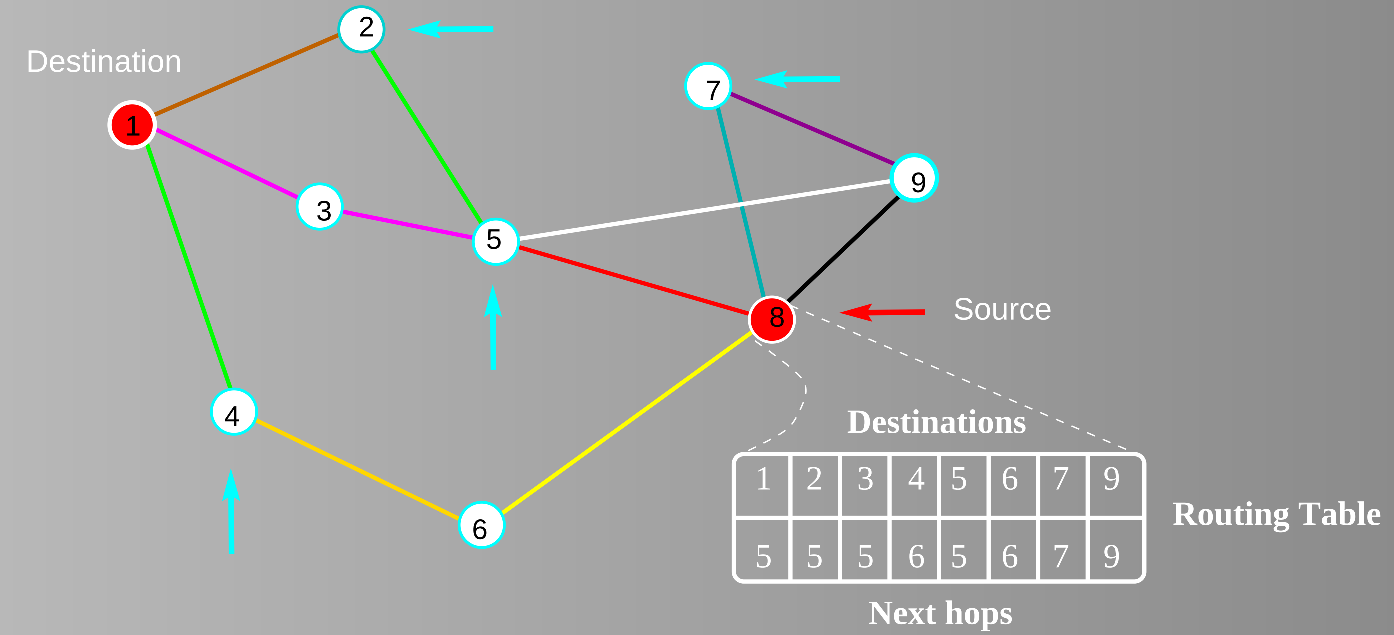
Applications and performance

- *Traveling salesman:* state-of-the-art / good performance
- *Quadratic assignment:* good / state-of-the-art
- *Scheduling:* state-of-the-art / good performance
- *Vehicle routing:* state-of-the-art / good performance
- *Sequential ordering:* state-of-the-art performance
- *Shortest common supersequence:* good results
- *Graph coloring and frequency assignment:* good results
- *Bin packing:* state-of-the-art performance
- *Constraint satisfaction:* good performance
- *Multi-knapsack:* poor performance
- *Timetabling:* good performance
- *Optical network routing:* promising performance
- *Set covering and partitioning:* good performance
- *Parallel implementations and models:* good parallelization effi ciency

Application of ACO to routing problems

- Straightforward mapping
- Very good matching: multi-agent, adaptive, distributed
- Cheap/controllable realistic online Monte Carlo simulation
- No need of daemon components to get top performance
- Real-world problems of great practical interest
- Innovative design components
- Open to future developments in networks
  (Autonomic view and Traffic Engineering)

Routing in telecommunication networks (1)



- *Online and distributed data flow allocation problem*
- Building and maintaining at each node a *routing table* that *maps destinations to next hops*
- Centralized approaches are usually unfeasible
- Many different types of networks

Routing in telecommunication networks (2)

- Routing consists of two basic tasks:
  - Collecting and keeping up-to-date *local state* information (e.g. *link costs, topological connectivity*)
  - Exchanging this information and/or collecting similar, non-local, information to locally get a *global view*, and use this view to route data
- Local state information can be set:
  - Using reliable a priori knowledge (optimal routing)
  - Offline / manually
  - Partly online (e.g., topology), partly offline (e.g., costs)
  - Online following variations → *Adaptive routing*
- Adaptivity → optimization of resources *(Traffic Engineering)*

The dark side of adaptivity

- Paths inconsistencies (loops), instabilities, performance oscillations. . .
- Non-trivial setting of control parameters (time scales. . .)
- Problems with algorithms at the transport layer (e.g., TCP)
- Security issues

Link-state and distance-vector implementations

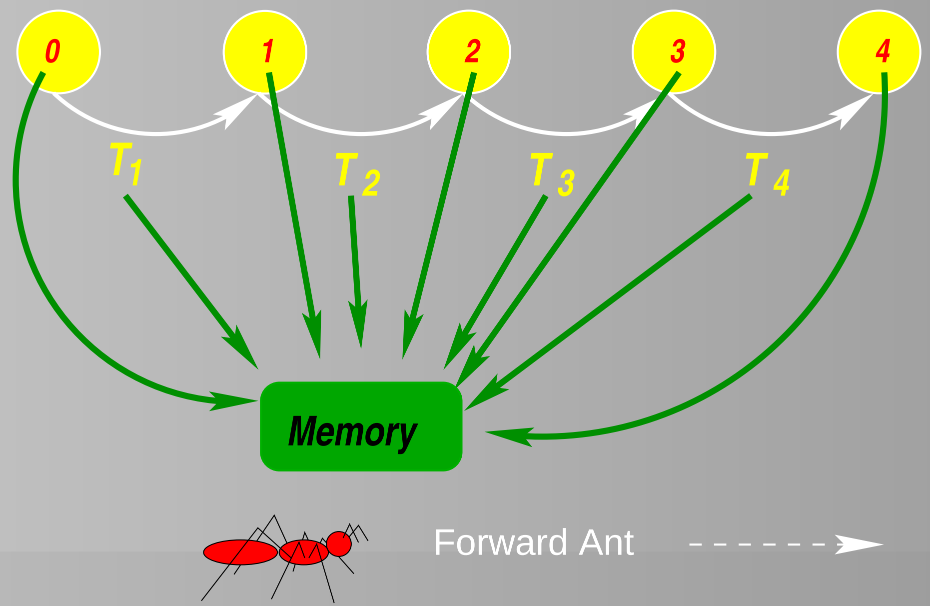
- Extensively used on *Internet* as IGP and BGP (OSPF, RIP)
- OSPF is used in larger Autonomous Systems
- *Topology adaptive* but not really traffic-adaptive
- *Local* estimations (costs, topology) have *global impact* (link-state flooding, distance estimates bootstrapping)
- *Single-path routing*
- *Deterministic*
- *No exploratory actions*
- Only *passive observation* of local conditions and *proactive propagation of information*, no active gathering of information
- Robust / complex protocols, but not in the direction of Traffic Engineering

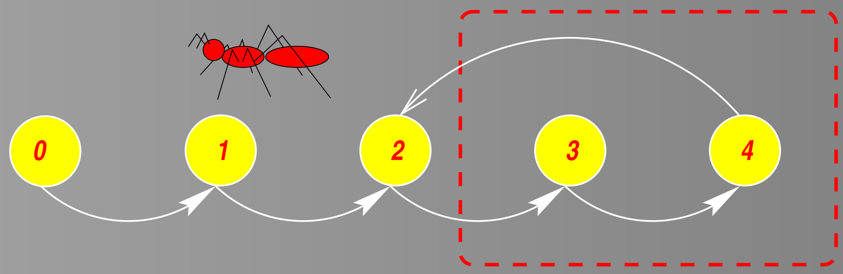
AntNet & AntNet-FA

- First ACO algorithms for *datagram networks* [Di Caro & Dorigo, 1997, 1998] (Schoonderwerd et al. applied ACO to telephone-like networks in 1996)
- General architecture: straightforward application of ACO
- *Careful design of each component*
- *State-of-the-art performance* and seen as reference algorithm
- AntNet-FA is a smart improvement over AntNet

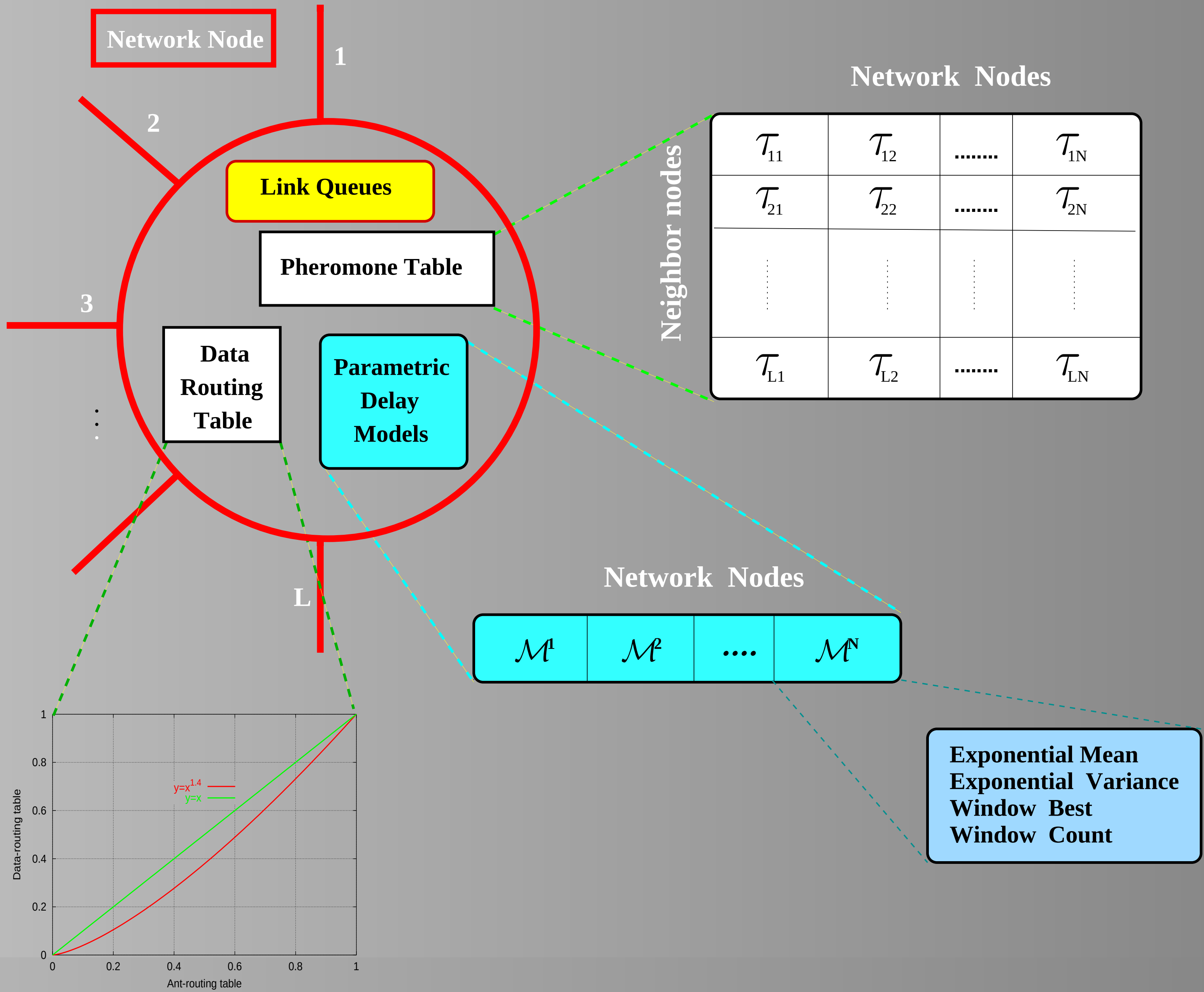
AntNet: algorithm description (1)

- *Proactive generation of Forward Ants*
- An ant faithfully *simulate a data packet*
  - *Discover/sample a good path*
  - *Update routing information*
- Forward ants maintain a private *memory* of each visited node and of the time of the visit (loops are removed)





AntNet: algorithm description (2)

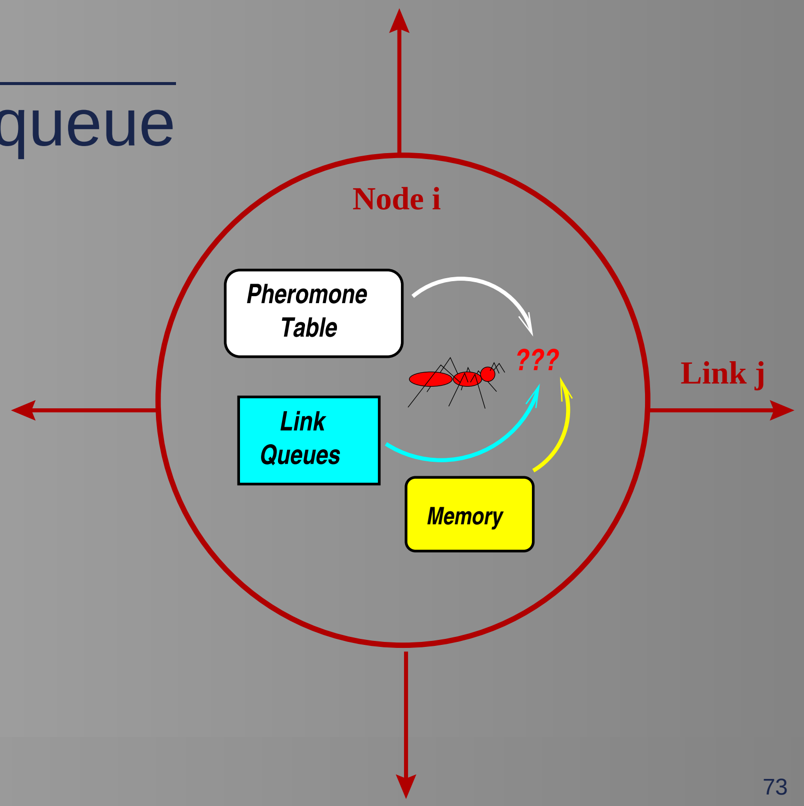


AntNet: algorithm description (3)

- *Next hop nodes* are selected according to a *stochastic decision policy* π parametrized by:
  - *Pheromone table* $\boldsymbol{\tau}_i$
  - Heuristic variables $\boldsymbol{\eta}_i$ = *Status of local link queues*
  - *Memory* of the nodes visited so far

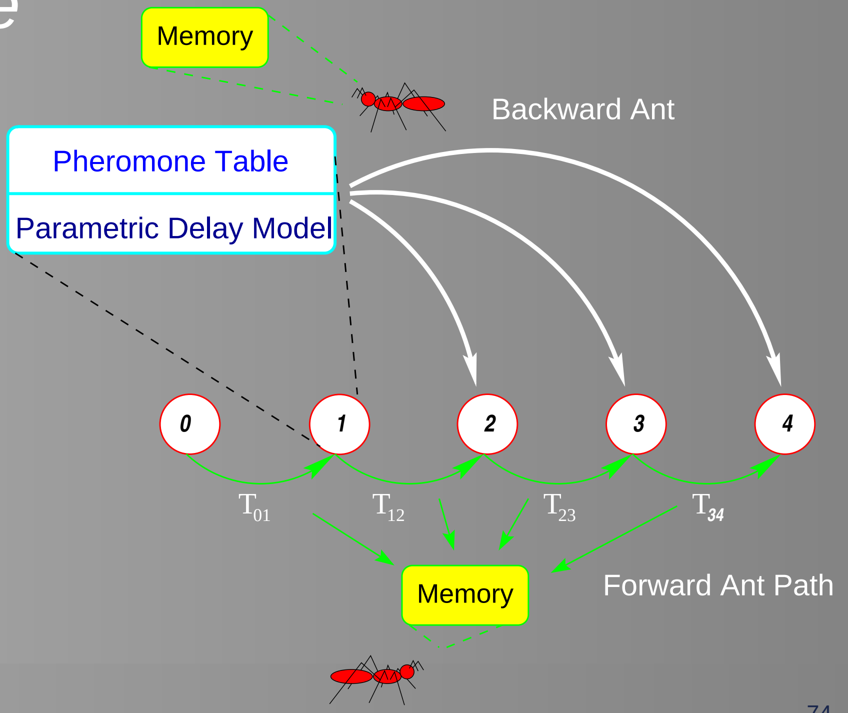
$$\eta_{ij}(t) \propto \frac{1}{\text{expected waiting time at } j\text{'s queue}}$$

$$p^d_{ij}(t) \propto \alpha\tau^d_{ij}(t) + (1-\alpha)\eta_{ij}(t)$$



AntNet: algorithm description (4)

- At destination d the Forward Ant a_k becomes a *Backward Ant* and *retraces the path*
- At each node i the Backward Ant, coming from neighbor j:
  - *Updates the Parametric Delay Model* $\mathcal{M}_i^d$
  - *Evaluates the path*: $r_{ij}^d(k) = J(T_{i \to d}^k, \mathcal{M}_i^d)$
  - *Updates the pheromone table and the routing table* with r_{ij}^d



AntNet-FA: improving AntNet

- In AntNet the path trip time $T_{i \to j}$ is the actual time experienced by the ant
- In AntNet-FA Forward ants make use of high priority queues (they fly!)
- The trip time $T_{i \to d}$ is calculated *during the backward journey, estimating the waiting time at the link* j to calculate the one-hop time:

$$T_{i \to j} = d_j + \frac{q_j}{b_j}$$

- $T_{i \to j}$ is an *up-to-date* estimate of the time to hop from $i \to j$

What is the outcome of the ant actions?

- *Proactive exploration and route adaptation*
- At each node a *bundle* of datagram paths are available
- Each choice has a *goodness value* (pheromone) which is online adapted to the traffic patterns ↓
- *Data are spread stochastically (multi-path routing)*
- The less good paths are *backup paths*
- Automatic *load balancing*
- *Robust* wrt ant failures and to parameter setting
- No global propagation of local estimates
- Active non-local information gathering
- Shortcomings: TCP, short-lived loops, topological adaptivity

Critical design components

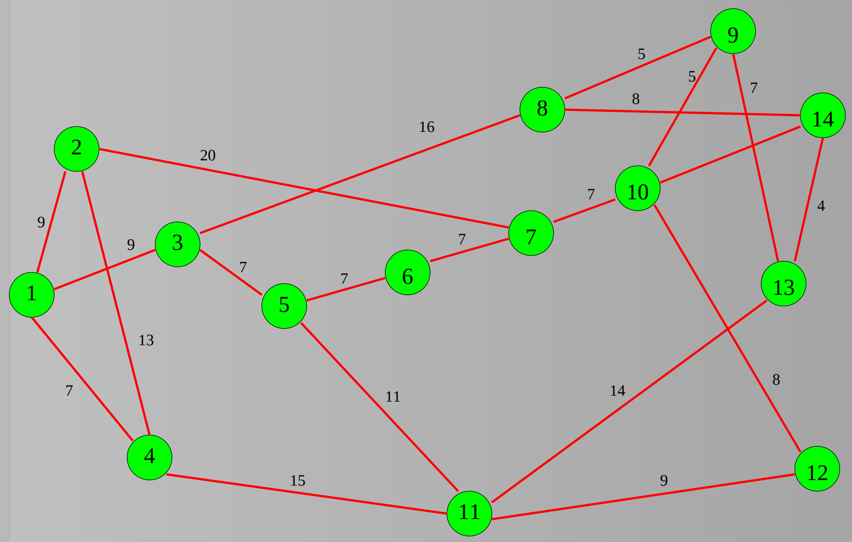
- *Ant generation:* when, where, how?
- *Decision policy:* instantaneous vs. longer-term view
- *Evaluation and rewarding of sampled paths:* metrics (AntNet-FA), non-stationarity, hidden state, learning rate

Experimental setup for AntNet(-FA)

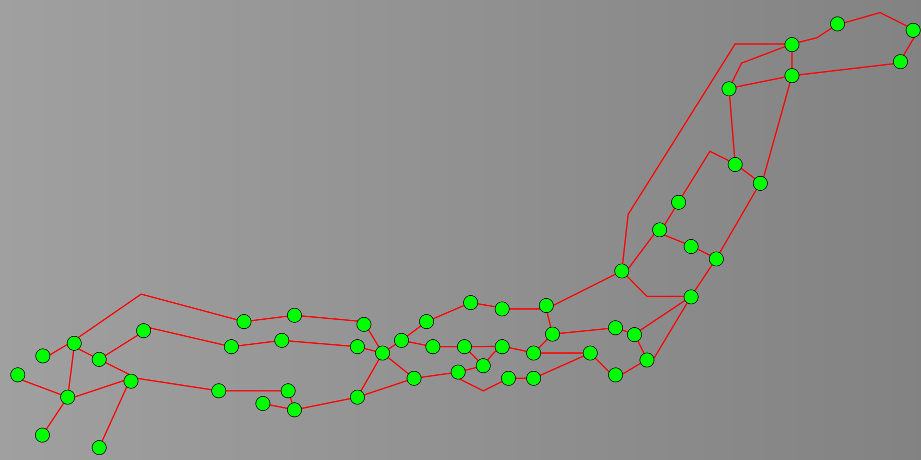
- Extensive simulation studies
- Realistic experimental setup for:
  - Network topology and physical characteristics
  - Protocol for data transmission
  - Spatial and Temporal Traffic Patterns
  - Algorithms to compare the performances

Networks

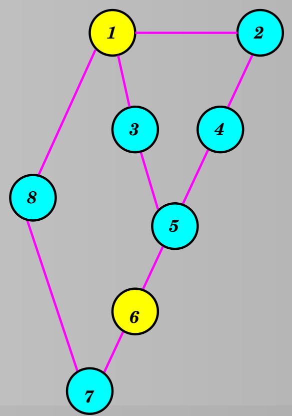
NSFNET-T1 (14, 21, 1.5) - US backbone (1987)



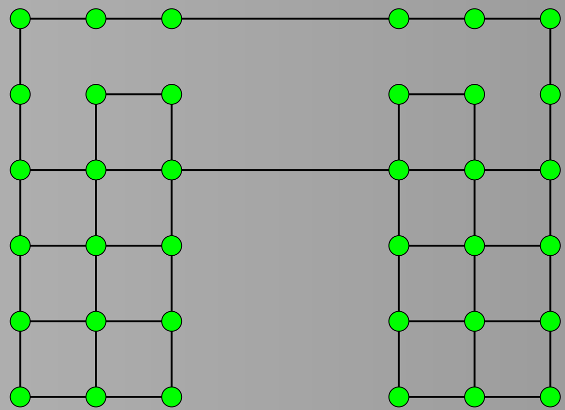
NTTnet (57, 162, 6) - NTT network (?)



Simple Net (8, 9, 10)



6x6 Net (36, 99, 10) - (Boyan & Littman, 1994)



Random Nets (100/150, ≈ 3, 10)

Traffic patterns

Data Transmission Protocol

- Best-effort Datagram traffi c
- IP-like protocol
- Discarding packet for no buffer space
- Failure situations not considered
- No arrival acknowledgment or error notifi cation packets
- Simple Flow control mechanism based on a static production window

Data sessions

- Negative exp distribution for sessions' inter-arrival times, global size, and packet sizes
- Traffi c patterns obtained by the combination of three basic traffi c types:
  - Poisson (Spatially Uniform (UP) and Random (RP))
  - Constant Bit Rate (CBR)
  - Hot Spots (HS)

IDSIA

Algorithms used for comparison

Static - Link-state

- *OSPF*: Minimum cost paths, current IGP Internet algorithm

Adaptive - Link-state

- *SPF*: Link-state prototype, Adaptive link costs, last ARPANET algorithm

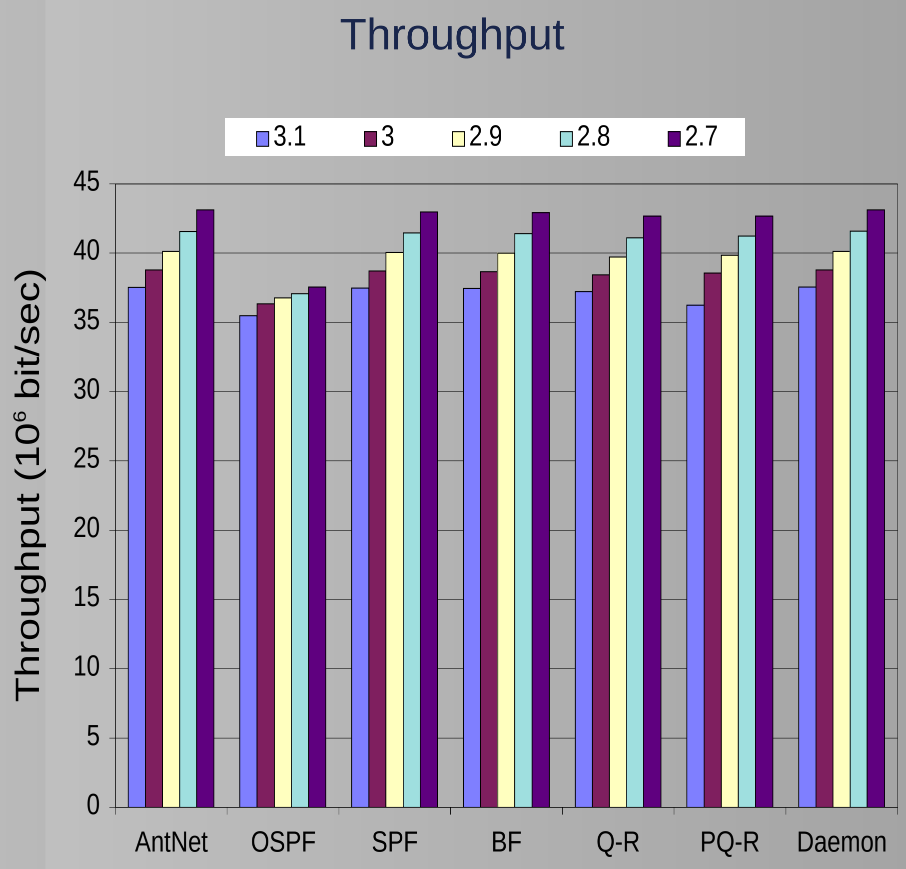
Adaptive - Distance-Vector

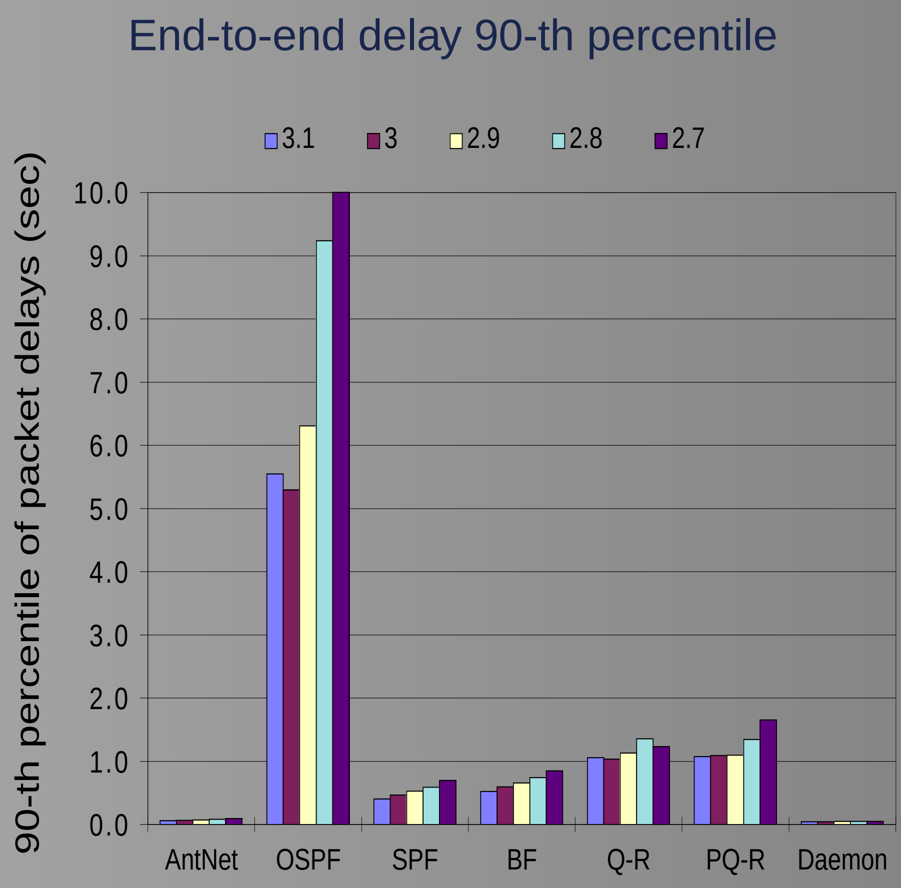
- *BF*: Asynchronous Bellman-Ford prototype, Adaptive link costs, ARPANET
- *Q-Routing*: Asynchronous Bellman-Ford with online updates and Q-Learning-like rule
- *PQ-Routing*: Q-Routing with a system to learn a model of the link queues

Ideal

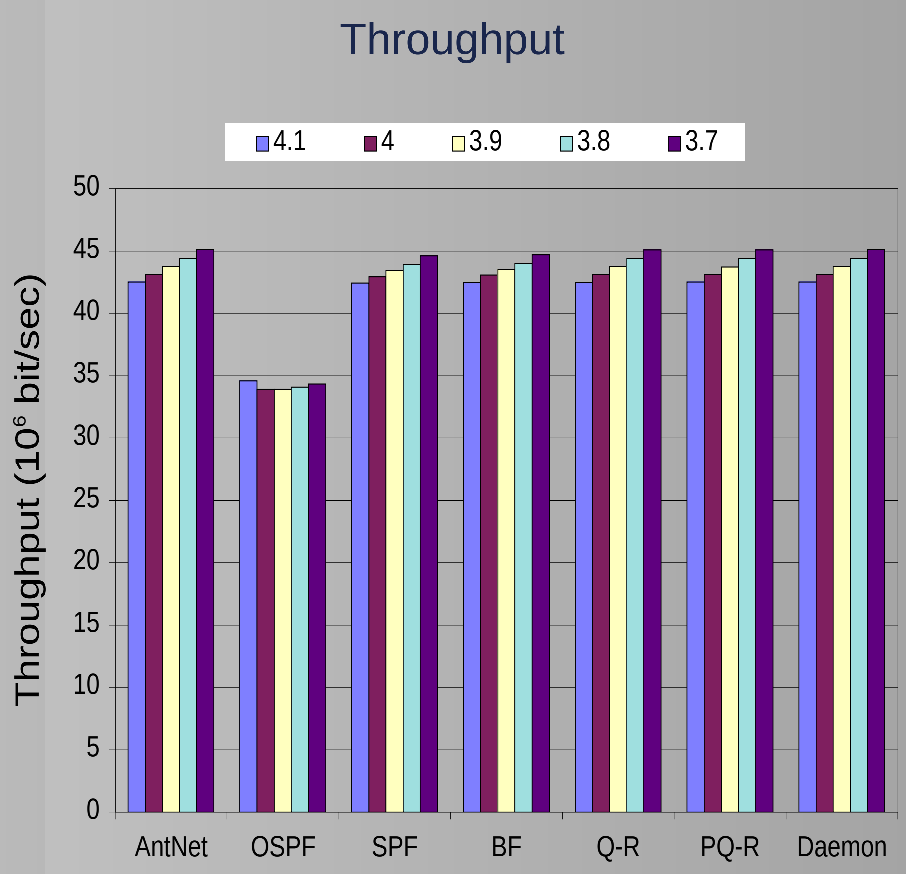
- *Daemon*: Access the state of all the net queues, empirical bound on performance

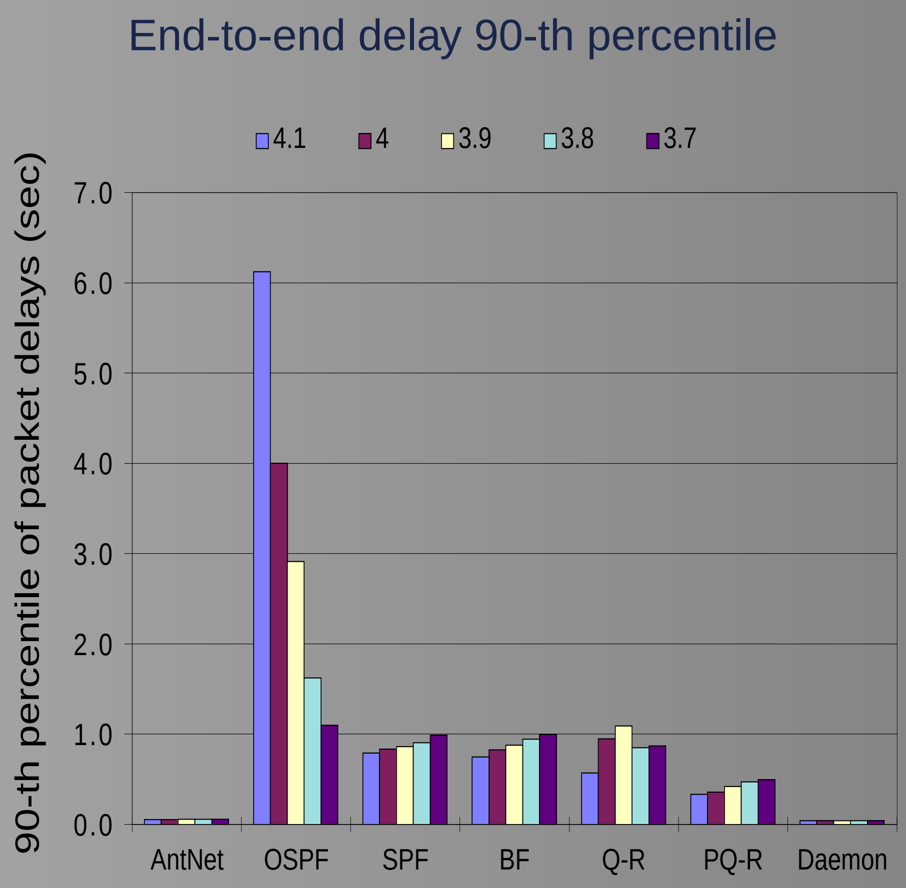
Results - NTTnet UP Load



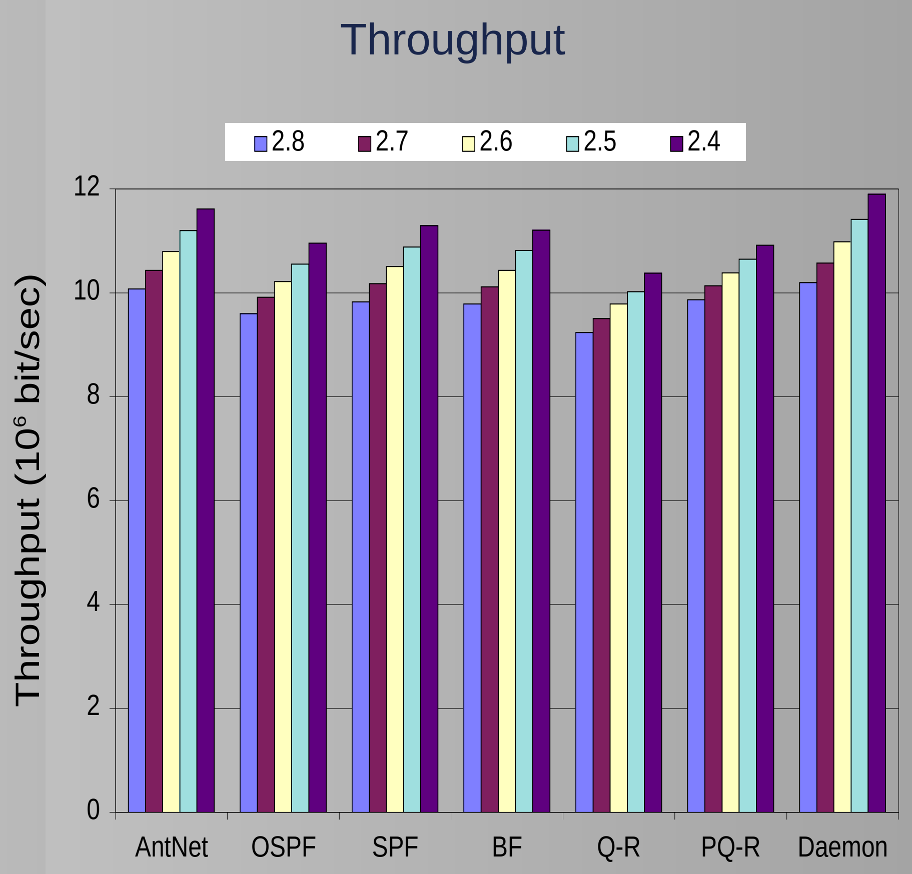


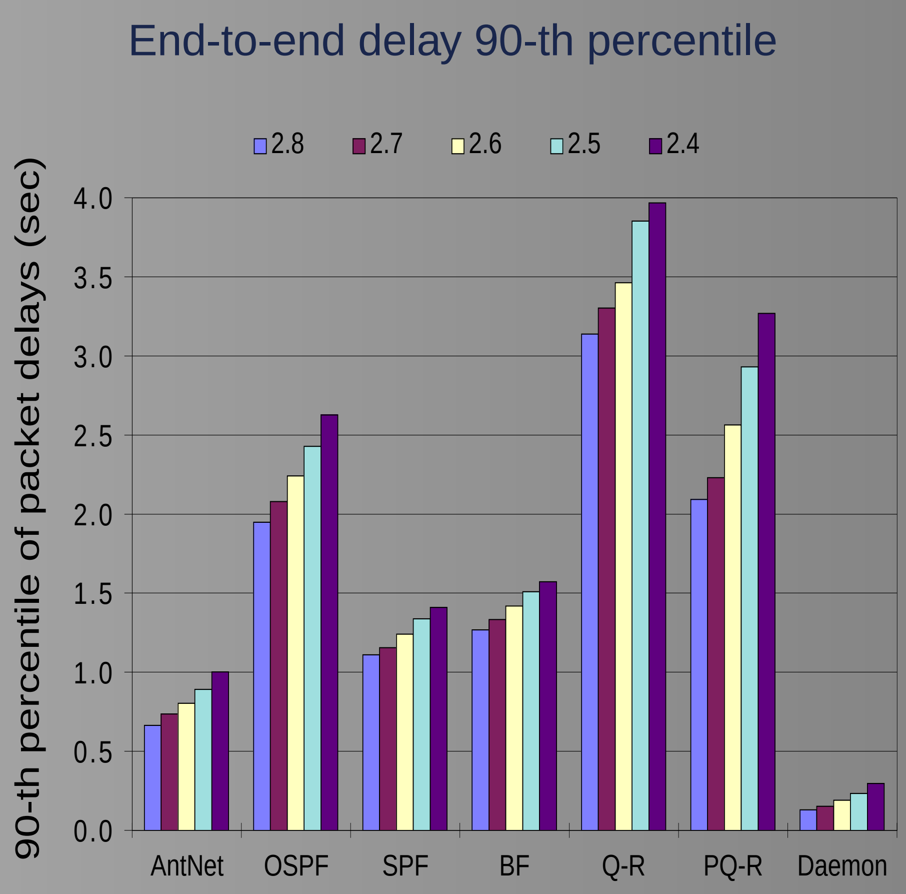
Results - NTTnet UPHS Load





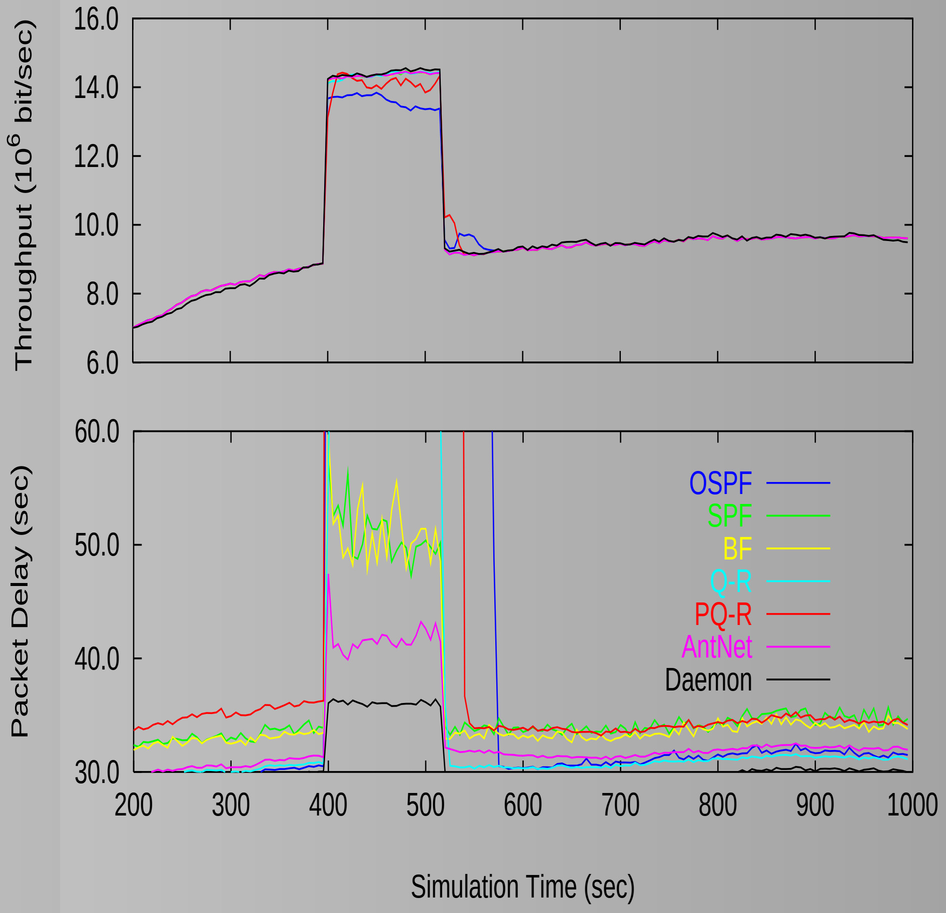
Results - NSFNET RP Load



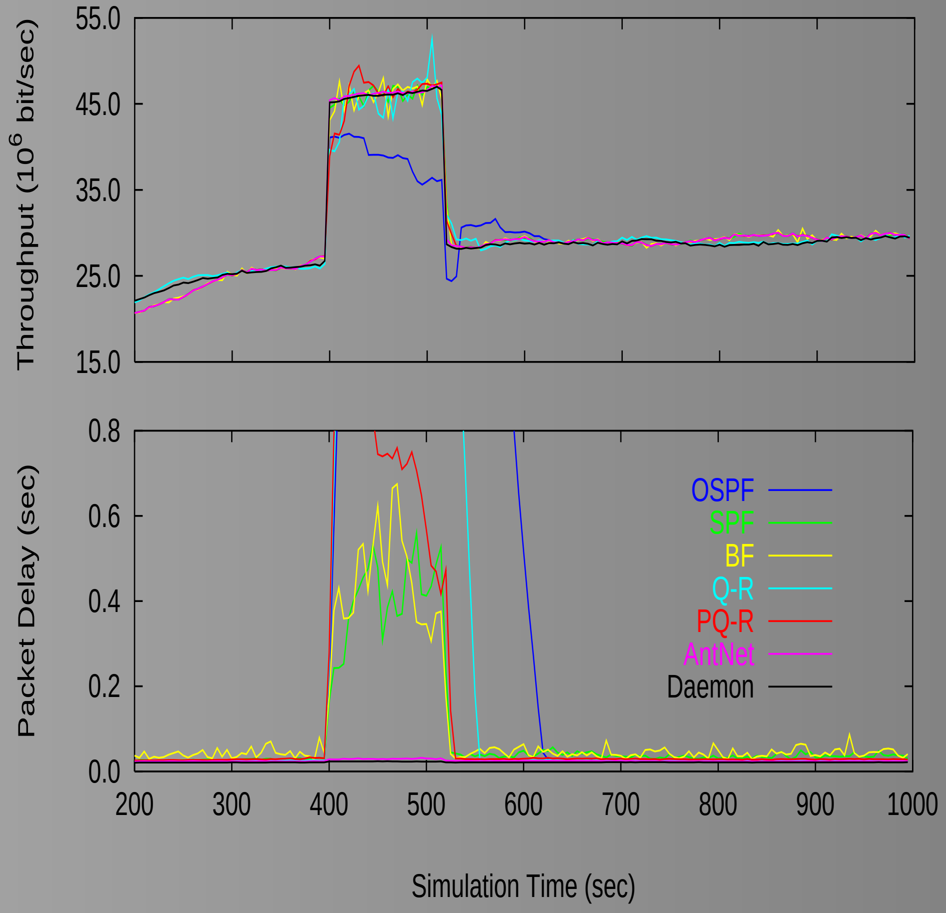


Results - Load Variation

NSFNET: UP Load Variation



NTTnet: UP Load Variation

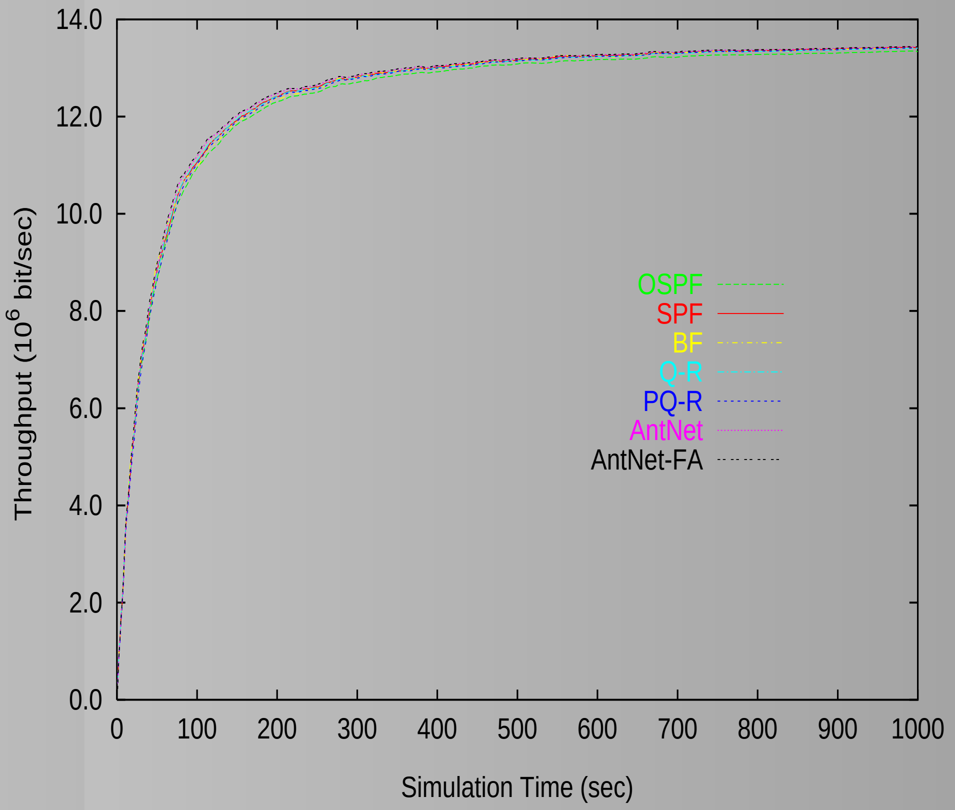


MSIA=3.0, MPIA=0.3, HS=4, MPIA-HS=0.04

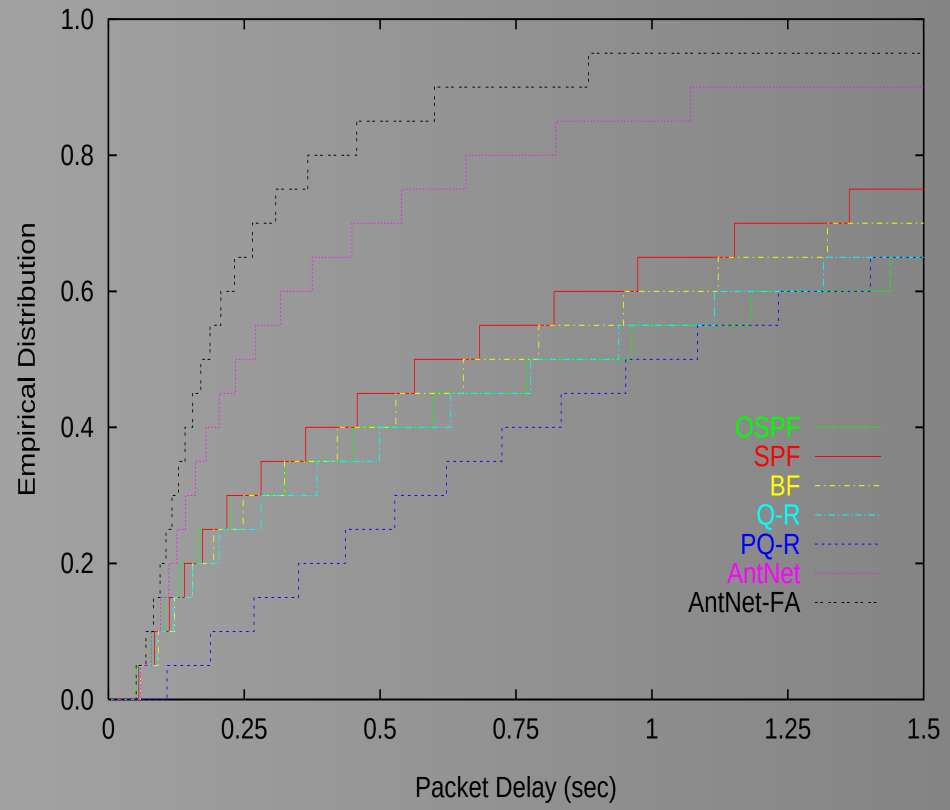
MSIA=4.0, MPIA=0.3, HS=4, MPIA-HS=0.05

100-Nodes RandomNets - UP Load

Throughput



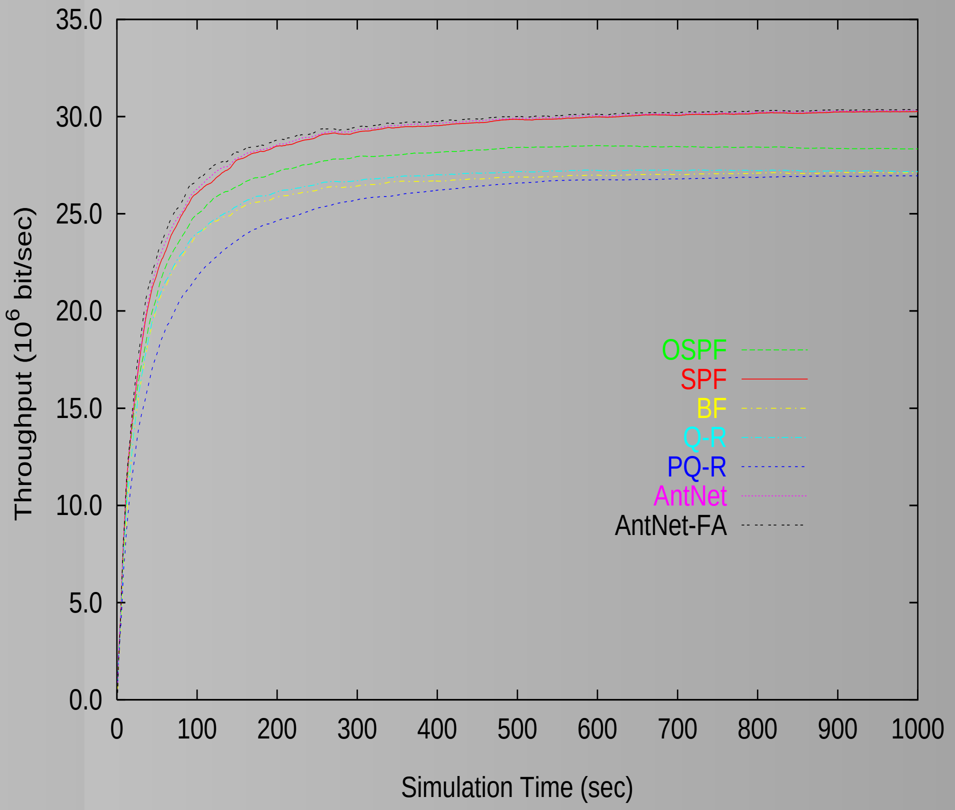
Delay distribution



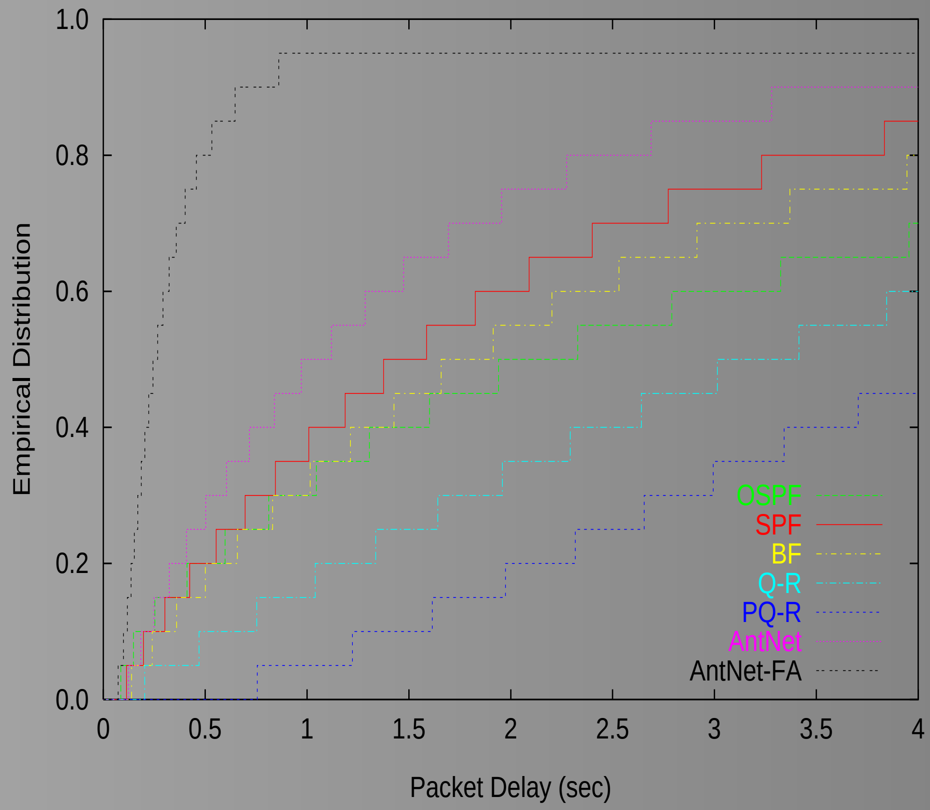
(MSIA = 15.0, MPIA = 0.005)

150-Nodes RandomNets - RP Load

Throughput



Delay distribution



(MSIA = 10.0, MPIA = 0.005)

Routing Overhead

Routing Overhead (10^{-3}) for some of the realized experiments

	AntNet	OSPF	SPF	BF	Q-R	PQ-R	Daemon
NSF - UP	2.39	0.15	0.86	1.17	6.96	9.93	0.00
NSF - RP	2.60	0.16	1.07	1.17	5.26	7.74	0.00
NSF - UPHS	1.63	0.15	1.14	1.17	7.66	8.46	0.00
NTT - UP	2.85	0.14	3.68	1.39	3.72	6.77	0.00
NTT - UPHS	3.81	0.15	4.56	1.39	3.09	4.81	0.00

Routing Overhead = Ratio between the generated routing traffi c and the total available bandwidth

For all the considered algorithms the routing overhead is quite low

The Ant Colony Routing framework

- Flat organization, homogeneous agents, nodes that just hold information, only proactive ant generation, fixed schedule . . . work well in best-effort wired networks but what about more complex environments?

- QoS, mobile ad hoc and mesh networks, autonomic view . . .

$\Downarrow$

Ant Colony Routing (ACR)

- *General framework for the design of (autonomic) routing/control systems based on the generalization of ACO ideas: collection of general strategies*

- From colony of ants to *societies of learning agents*

ACR: general architecture, ideas, applications

- *Node managers:* non-mobile reinforcement learning agents that control/monitor local activities, proactively and/or on-demand generate:
  - *Active perceptions:* ant-like mobile agents, gather non-local information
  - *Effectors:* mobile specialized agents
- First steps toward the implementation of the ACR view:
  - *AntNet+SELA [Di Caro & Vasilakos, 2000] QoS in ATM*
  - *AntHocNet [Di Caro, Ducatelle & Gambardella, 2004] Best-effort routing in mobile ad hoc networks*

The end

Thanks for listening!

Few general references

- E. Bonabeau, M. Dorigo and G. Theraulaz, "Swarm Intelligence: From Natural to Artifi cial Systems", Oxford University Press, 1999
- J. Kennedy and R.C. Eberhart, "Swarm Intelligence", Morgan Kaufmann, 2001
- S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz and E. Bonabeau, "Self-Organization in Biological Systems", Princeton University Press, 2001
- M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization", Artifi cial Life, Vol. 5(2), pages 137-172, 1999
- D.H. Wolpert and K. Tumer, "An introduction to collective intelligence", in Handbook of Agent Technology, ed. J. Bradshaw, AAAI/MIT Press, 2001
- M. Dorigo and G. Di Caro, ""The Ant Colony Optimization Meta-Heuristic", in "New Ideas in Optimization", D. Corne, M. Dorigo and F. Glover editors, McGraw-Hill, pages 11–32, 1999
- M. Dorigo and T. Stuetzle, "Ant Colony Optimization", MIT Press, Cambridge, MA, 2004
- G. Di Caro, "Ant Colony Optimization and its application to adaptive routing in telecommunication networks", Ph.D. thesis, Faculte' des Sciences Appliquees, Universite' Libre de Bruxelles, Brussels, Belgium, 2004
- F. Glover and G. Kochenberger, "Handbook of Metaheuristics", Kluwer Academic Publishers, International Series in Operations Research and Management Science, Vol. 57, 2003
- G. Di Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks", Journal of Artifi cial Intelligence Research (JAIR), Vol. 9, pages 317-365, 1998
- R. Schoonderwoerd, O. Holland, J. Bruten and L. Rothkrantz, "Ant-based Load Balancing in Telecommunications Networks", Adaptive Behavior, Vol. 5(2), pages 169-207, 1996
- G. Di Caro, F. Ducatelle and L.M. Gambardella, "AntHocNet: an adaptive Nature-inspired algorithm for routing in mobile ad hoc networks", European Transactions on Telecommunications, 2005 (to appear)
- F. Ducatelle, G. Di Caro and L.M. Gambardella, "Using ant agents to combine reactive and proactive strategies for routing in mobile ad hoc networks", International Journal of Computational Intelligence and Applications, 2005 (to appear)

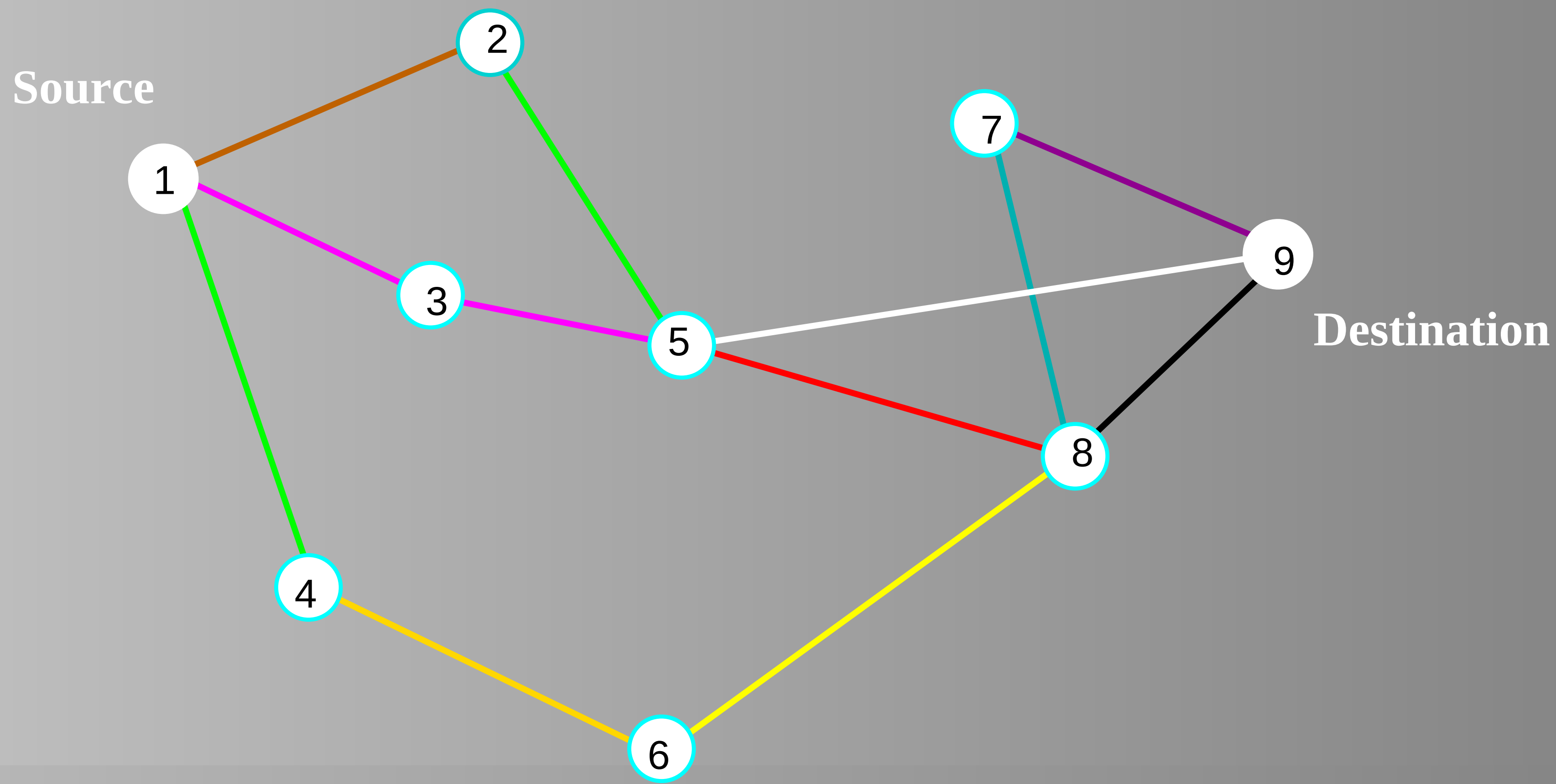
Combinatorial problems

- *Instance of a combinatorial optimization problem:*
  - *Finite set* S of feasible solutions
  - Each with an associated *real-valued cost* J
  - Find the solution $s^* \in S$ with *minimal cost*
- Costs can change over time $\rightarrow$ dynamic problems
- Different modalities of solution: centralized, offline, distributed, online

- In practice, a compact formulation $\langle C, \Omega, J \rangle$ is used:
  - C is a finite set of elements $\rightarrow$ *solution components*
  - Ω is a set of relations among C's elements $\rightarrow$ *constraints*
  - *The feasible solutions are subsets of components that satisfy the constraints* Ω

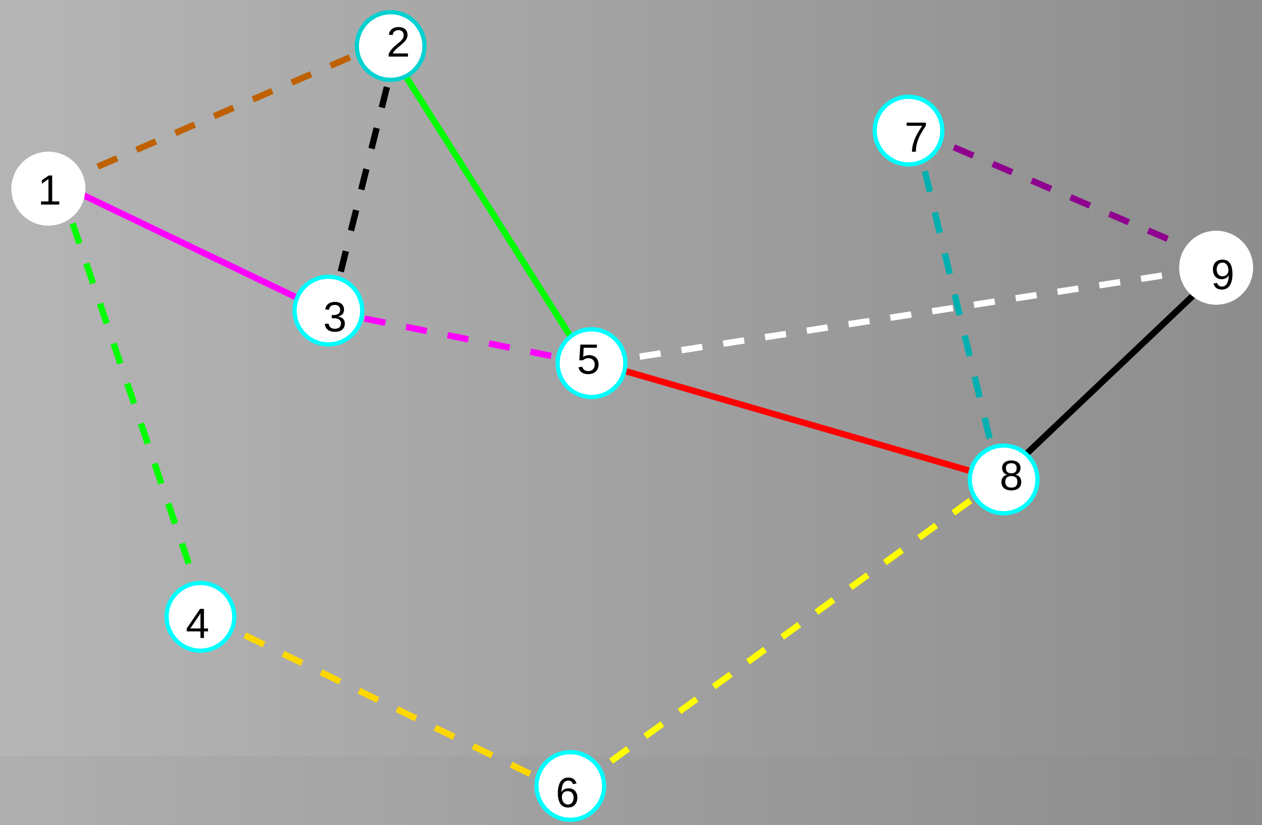
Examples of combinatorial problems (1)

- *Shortest path (SPP)*: $C = \{\text{graph nodes}\}$
- Ex. Sequential decision processes (capacited graph)



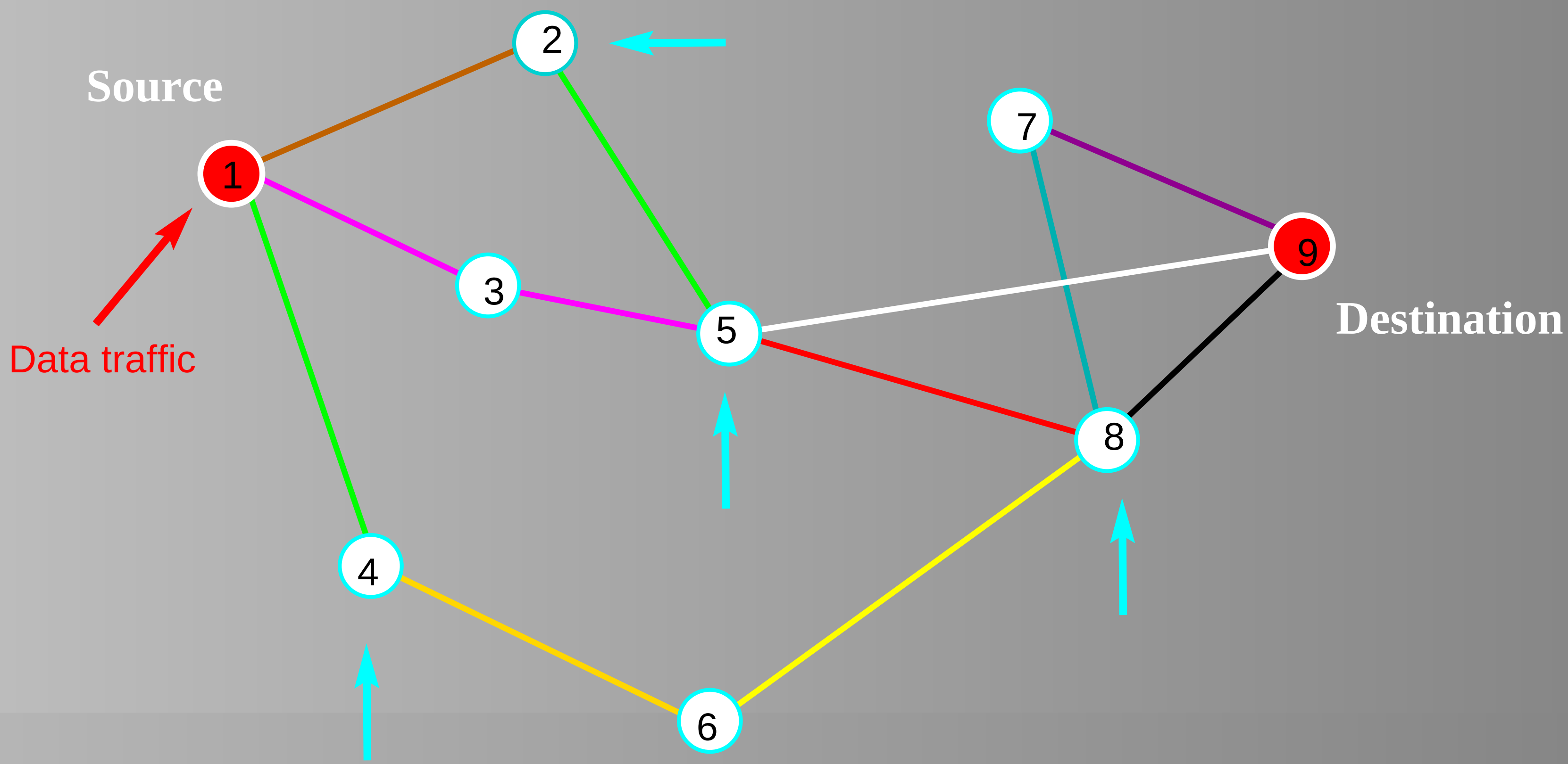
Examples of combinatorial problems (2)

- *Traveling salesman problem (TSP)*: $C = \{\text{cities to visit}\}$
- Ex. Goods delivery
- Constrained shortest path



Examples of combinatorial problems (3)

- *Data routing*: $C = \{\text{network nodes}\}$
- Shortest path + Multiple traffic flows to route simultaneously
- Telecommunication networks



These are all important but difficult problems

- SPP is the easiest (polynomial time complexity)
- TSP has $n!$ solutions, it's NP-hard (exponential worst case complexity) [centralized, offline]
- Routing is a real-world problem, depends on traffic dynamics (and on our knowledge about it), traffic requirements (QoS), network size, topology (mobile ad hoc networks) and capacity [distributed, online]
- They can be all seen in terms of solving shortest path problems. . .

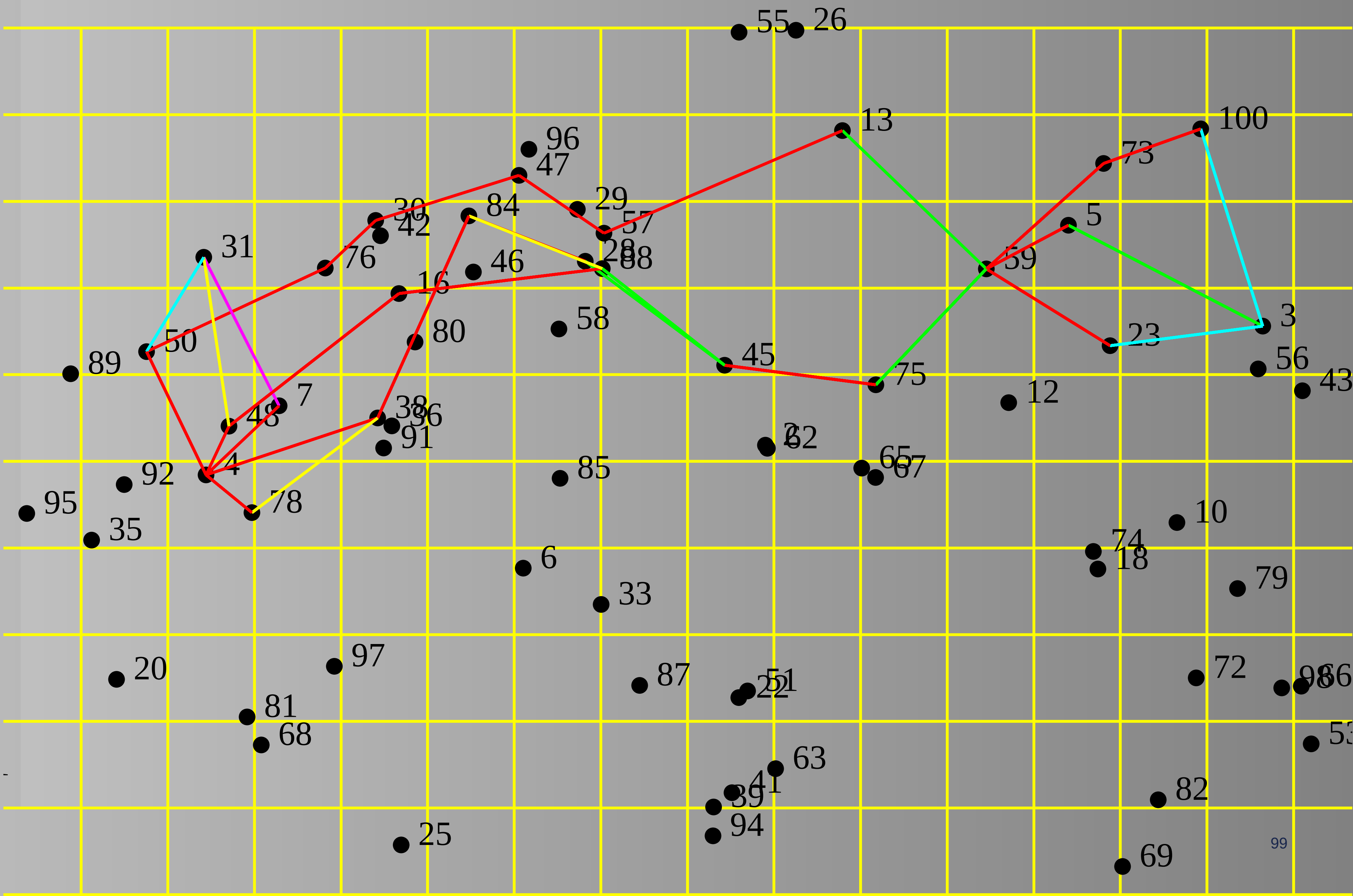
Then, how do we deal with these problems?

- *SPP:* very efficient algorithms are available (label setting / correcting methods)

- *TSP:* for NP-hard problems optimal algorithms are computationally inefficient or totally unfeasible
  $\longrightarrow$ *Heuristic algorithms for good solutions in practice*

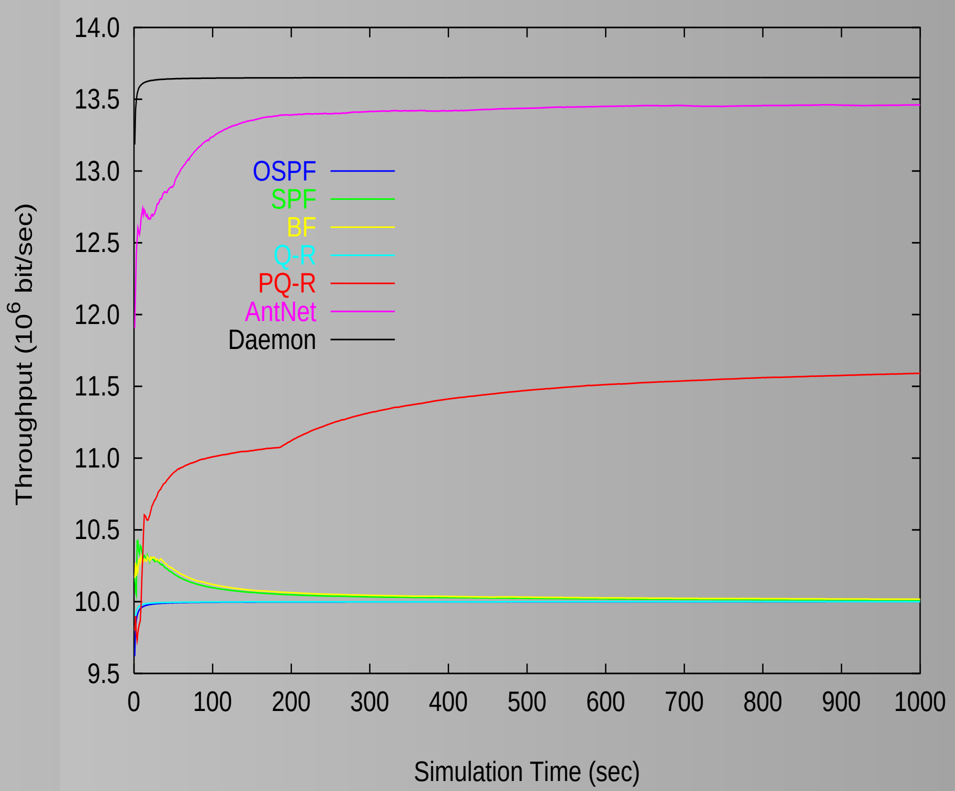
- *Routing:* there are optimal distributed algorithms for shortest paths and traffic flows allocation, but:
  - non-stationarities in traffic and topology, uncertainties, QoS constraints are the norm not the exception!
  - *Optimized solutions require full adaptivity and fully distributed behaviors* ($\longrightarrow$ Traffic Engineering)

AntHocNet: pheromone distribution ↑

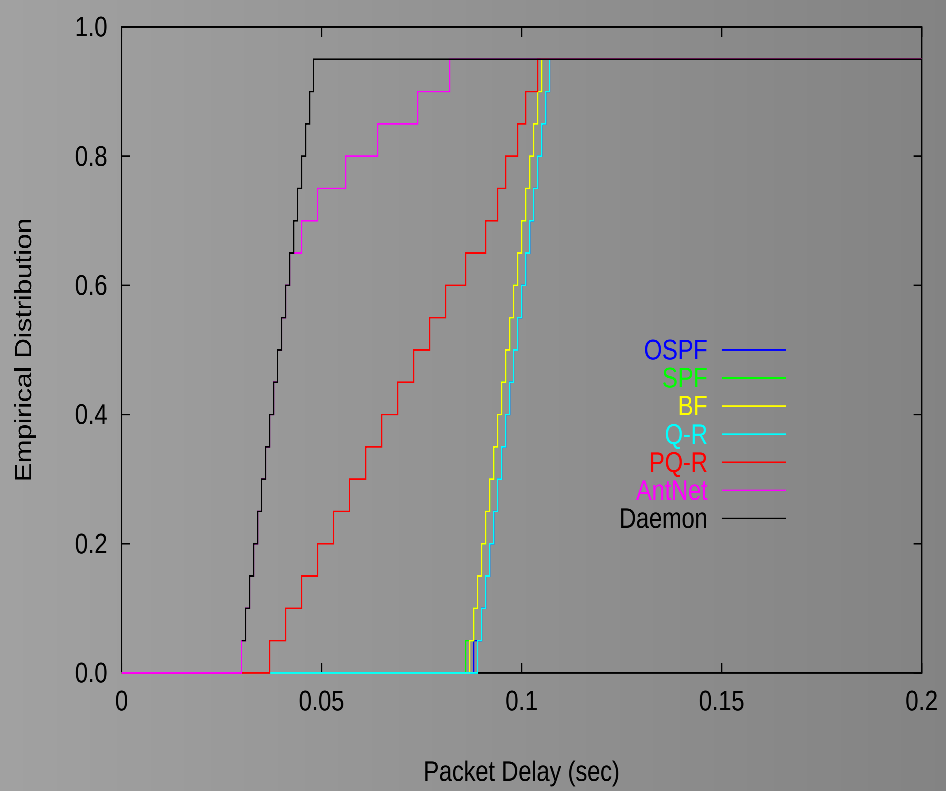


SimpleNet - CBR Load

Throughput



Delay distribution



(CBR = 0.0003)

AntNet Power Vs. Routing Overhead