Chemical Computation

- Slime Mold
- Belouzov-Zhabotinski
- Reaction diffusion systems
- Gray Scott
  - http://www.swiss.ai.mit.edu/projects/amorphous/GrayScott/

Good talk on Reaction-diffusion Systems

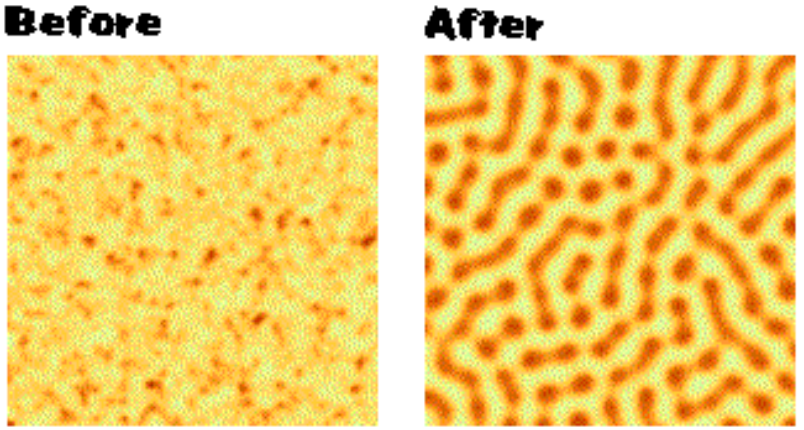
Brusselator Scheme

A -> X

B + X -> Y + waste

2X + Y -> 3X

X -> waste

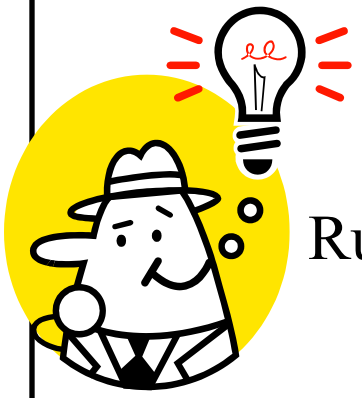


Diffusion

Structures maintained by diffusion of Energy

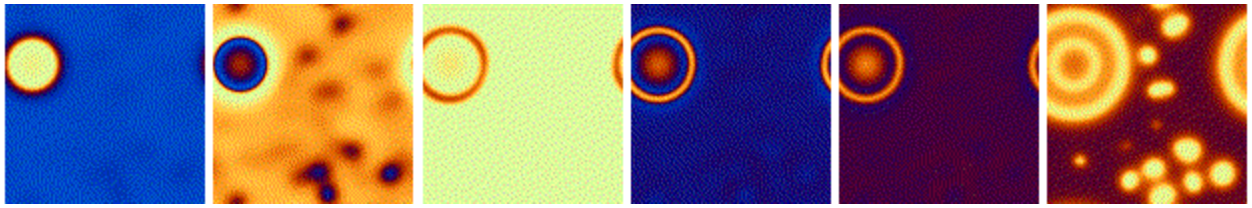
Reactions necessary to maintain structures

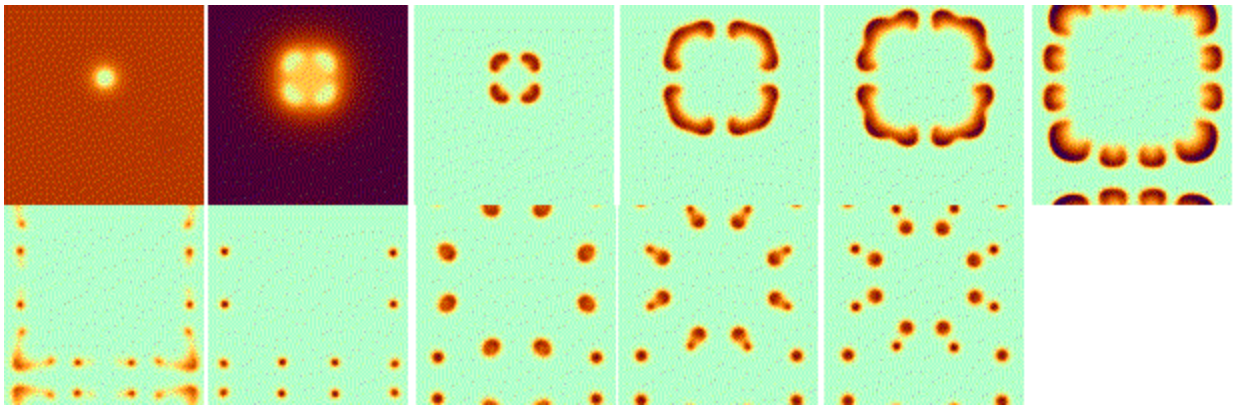
A,B have fixed concentrations, rate constant 1.0



Run demo

Self Organization





Emergent Behaviour and Mobile Agents

Tony White
Bernie Pagurek

Overview

- Why emergent, why not rational?
- Characteristics of emergent problem solving?
- SynthECA
  - architecture
  - examples
- Conclusions

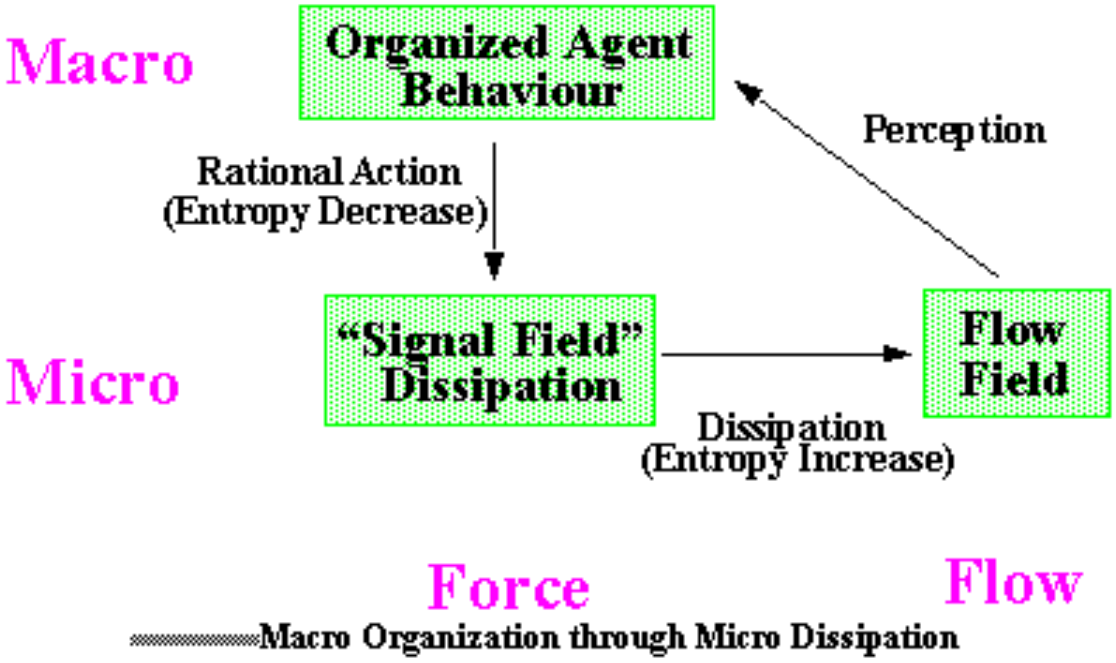
Problems of Rationality

- Knowledge of self (and sometimes society)
  - BDI
  - Interrap, TouringMachine etc.
- Grounded in symbolic view of world
  - closed world assumptions
  - limitations of first order predicate logic
- Frame problem
- Tend to be brittle

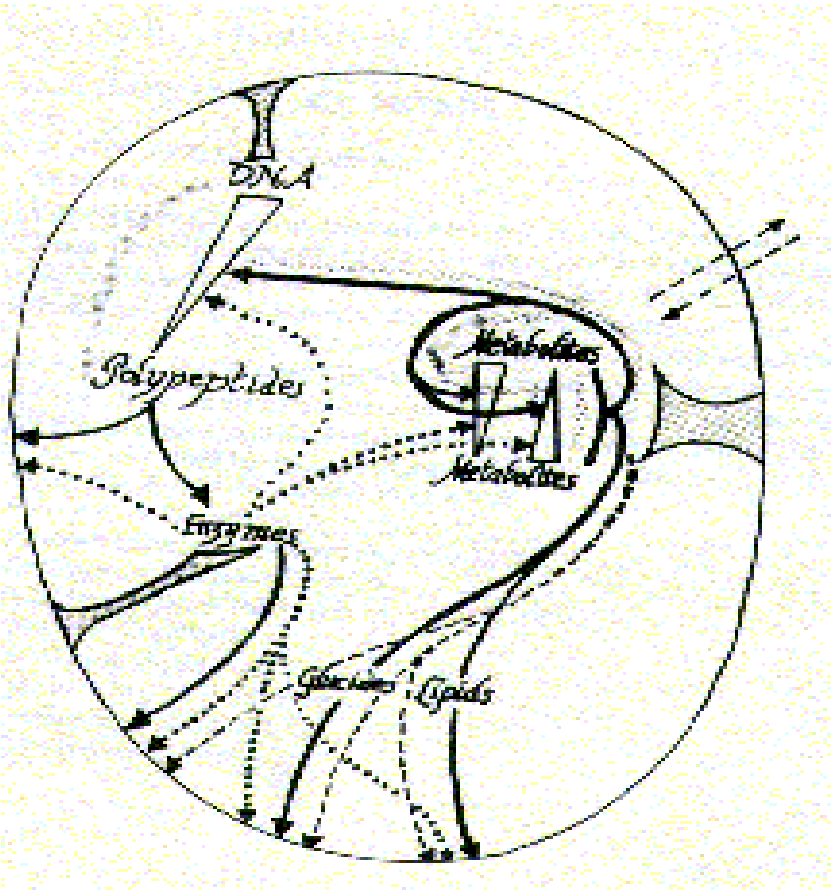
Characteristics of Emergent Problem Solving

- Aggregation
  - agents of small "mass", i.e., simple, reactive
- Non-linearity
  - interactions are not additive
- Flows
  - gradients, feedback
- Diversity
  - multiple agent classes more effective
  - stochastic behaviour

Emergent Organization: Forces at Play



Cellular Inspiration



A Cellular Autopoietic Network

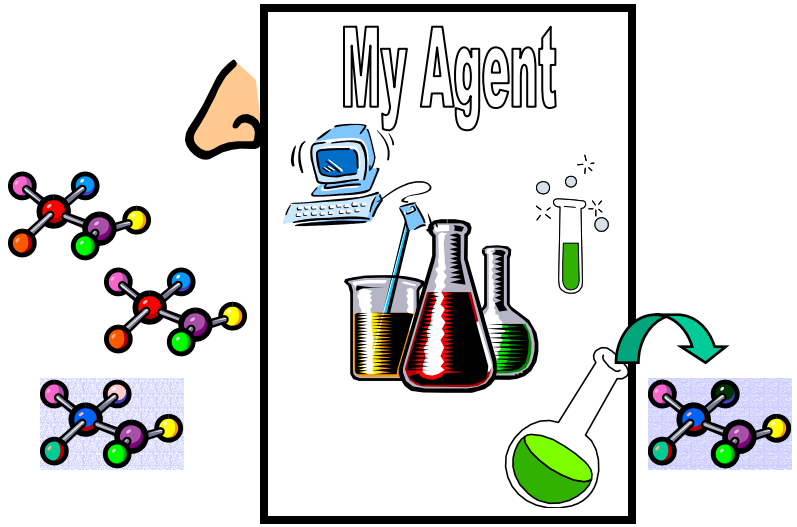
The cellular figure to the left is taken from Maturana and Varela and is used by them to introduce autopoietic concepts. The essence of this figure is autopoiesis and the relationships within a cell that make self creation possible. The reader cannot fail but to notice the striking similarities between it and the agent architecture described in section 3.5. The cell chemistry, reaction pathways and chemical exchange with the cell environment are all present.

SynthECA

- Based upon the Chemical Abstract Machine (CHAM)
- Draws on societies of agents
- Problem solving is distributed, emergent
  - no global interactions
  - robust to failure of individual agents
- British Telecom interested in ecological problem solving.

SynthECA operation

- Swarm agents:
  - arrive at a node,
  - sense environment,
  - undertake local activity,
  - modify environment,
  - use sensory input to make migration decision.

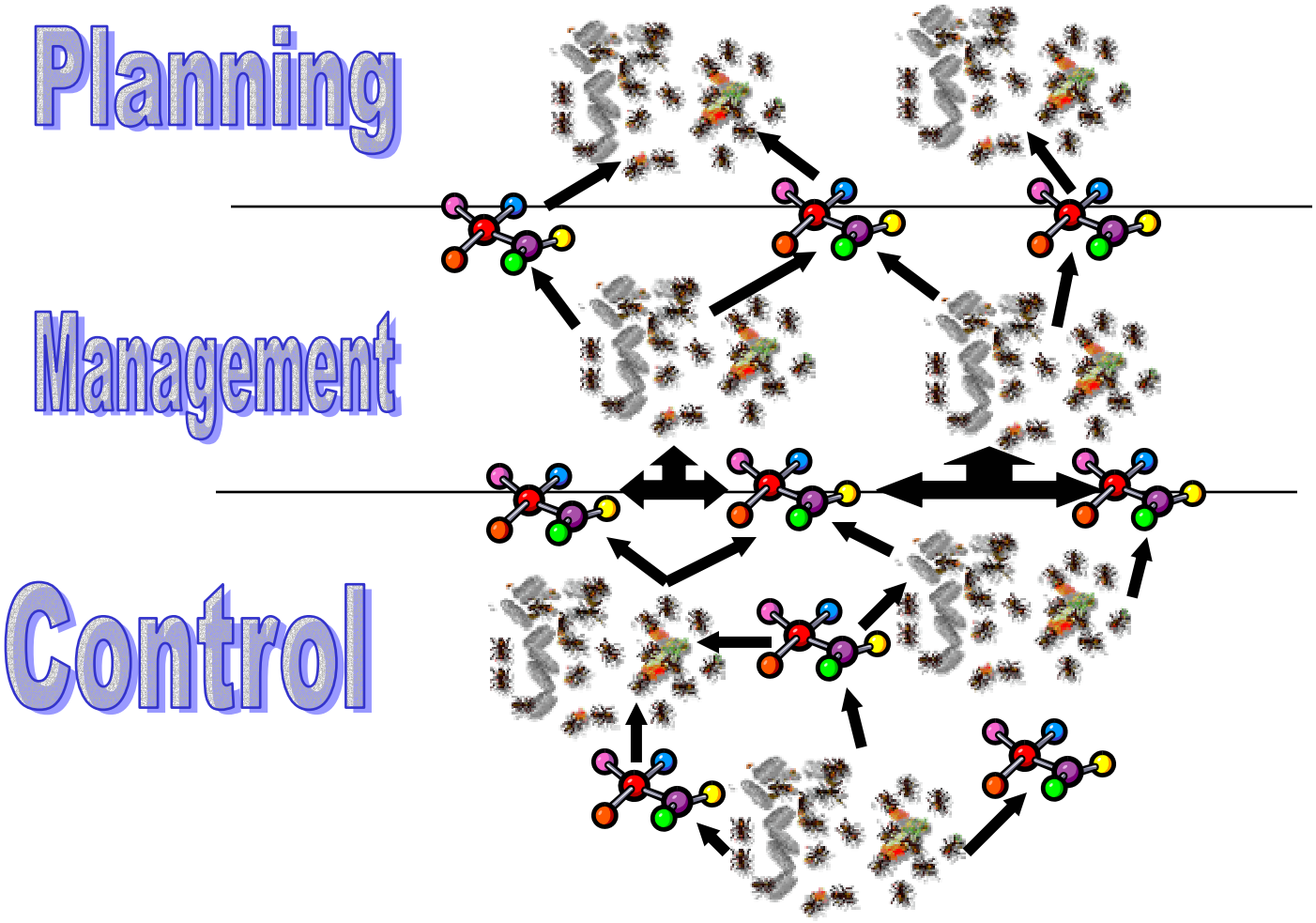


SynthECA Formalism

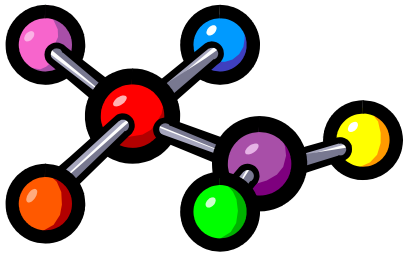
A=(E,R,C,MDF,m)

- Agents (A) have an architecture consisting of five components:
  - emitters (E),
  - receptors (R),
  - chemistry (C),
  - a migration decision function (MDF),
  - memory (m)

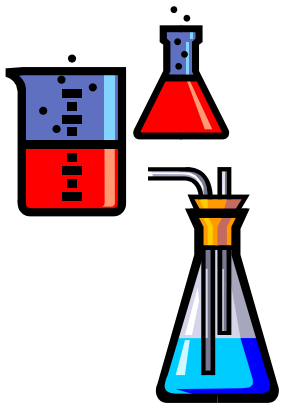
Multi-swarm Architecture



Chemicals



- Chemicals are digitally-encoded using a {1, 0, #} alphabet.
- The # symbol unifies with 1 or 0.
- Chemicals have two attributes:
  - encoding
  - concentration
- Chemicals participate in reactions.



Chemistry

- The set of chemical reactions (C) that can operate on sensed and locally stored chemicals.

Chemical interactions

$$X \rightarrow \text{'}nothing\text{'}$$

$$X + Y \rightarrow Y$$

$$X + Y \rightarrow Z$$

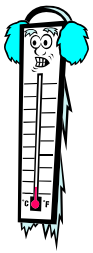
$$X + Y \rightarrow X + Z$$

$$X + Y \rightarrow W + Z$$

Examples of chemical reactions

Catalytic breakdown of 011

$$011 + 1\#0 \rightarrow 1\#0$$



$$T+Y \rightarrow Y$$

$$T+Y \rightarrow Z$$

$$X+T \rightarrow X+Z$$

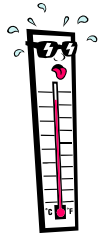
$$X+T \rightarrow W+Z$$

Endothermic reactions

$$X+Y \rightarrow T$$

$$X+Y \rightarrow X+T$$

$$X+Y \rightarrow W+T$$



Exothermic reactions

Migration Decision Function

The MDF is used to determine the next node to visit in the network

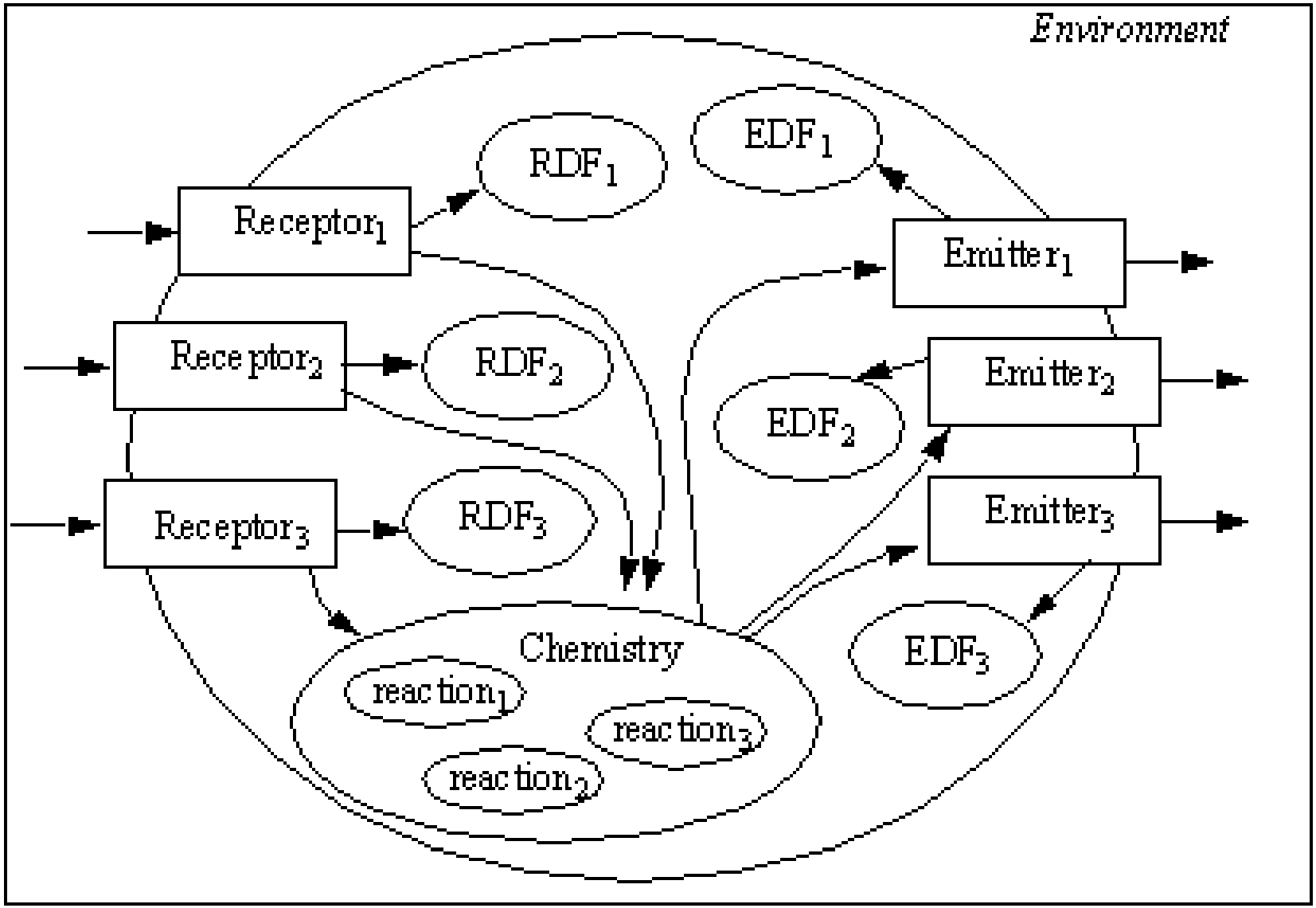
$$p_{ij}^{k}(t) = \Pi_p[T_{ijkp}(t)]^{-\alpha_{kp}}[C(i,j)]^{-\beta} / N_k(i,j,t)$$

$$N_k(i,j,t) = \Sigma_{j \text{ in } A(i)} \Pi_p [T_{ijkp}(t)]^{-\alpha_{kp}}[C(i,j)]^{-\beta}$$

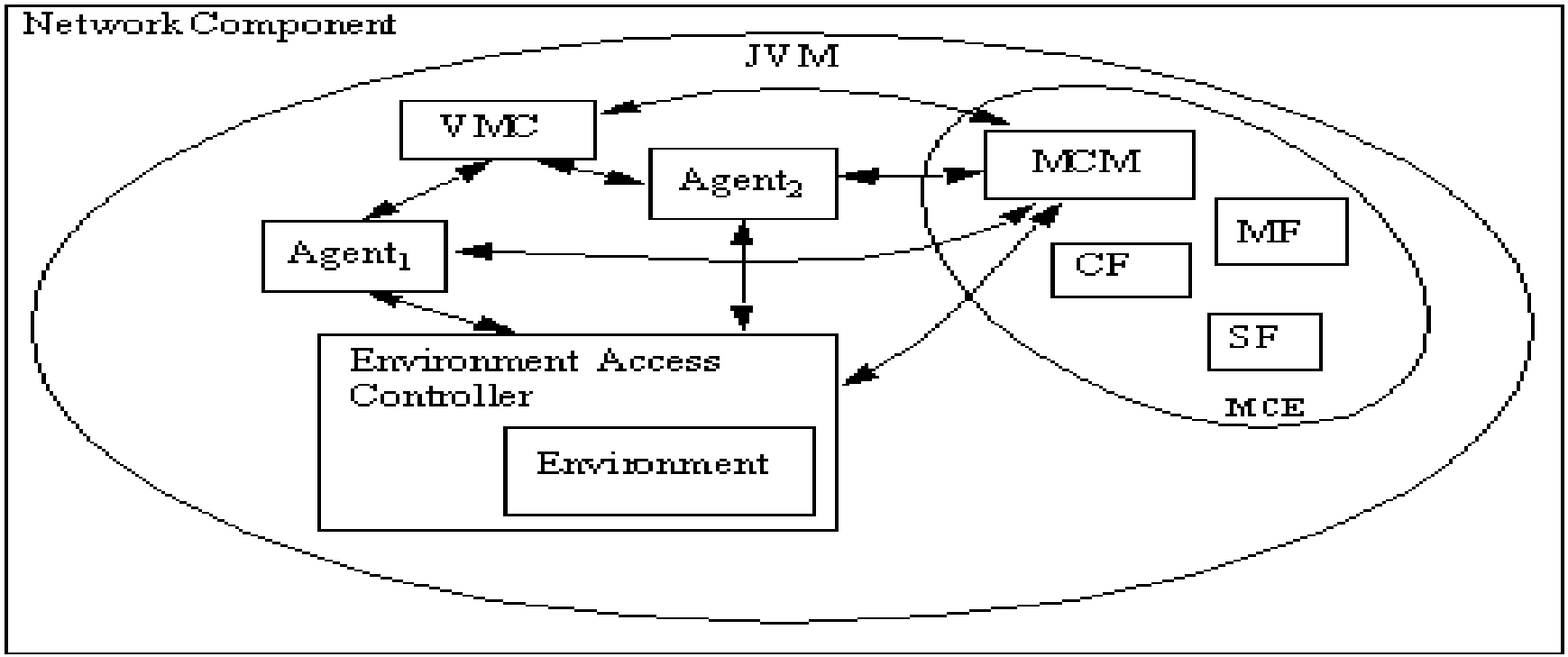
α_{kp}, β are control parameters

$N_k(i,j,t)$ is a normalization term

A(i) is the set of available egress links



Agent Architecture and Interactions



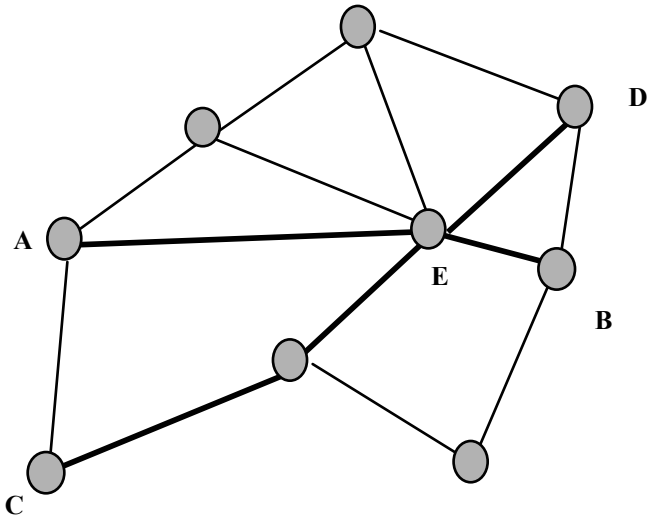
Legend.

JVM	Java Virtual Machine
MCE	Mobile Code Environment
VMC	Virtual Managed Component
MCM	Mobile Code Manager
CF	Communication Facilitator
MF	Migration Facilitator
SF	Security Facilitator

Agent-Environment Coupling

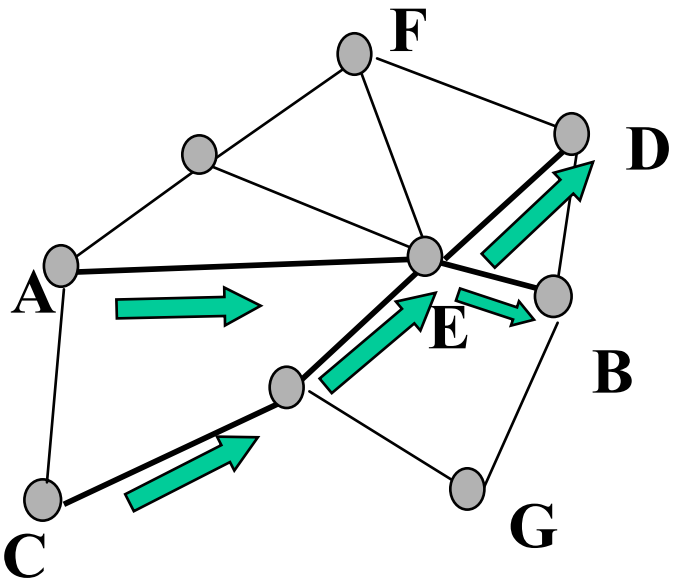
Multi-Swarm scenario

- Distributed Network Management employing delegation [Yemini, 91]
- Three interacting swarms:
  - connection finding,
  - connection monitoring,
  - connection fault diagnosis

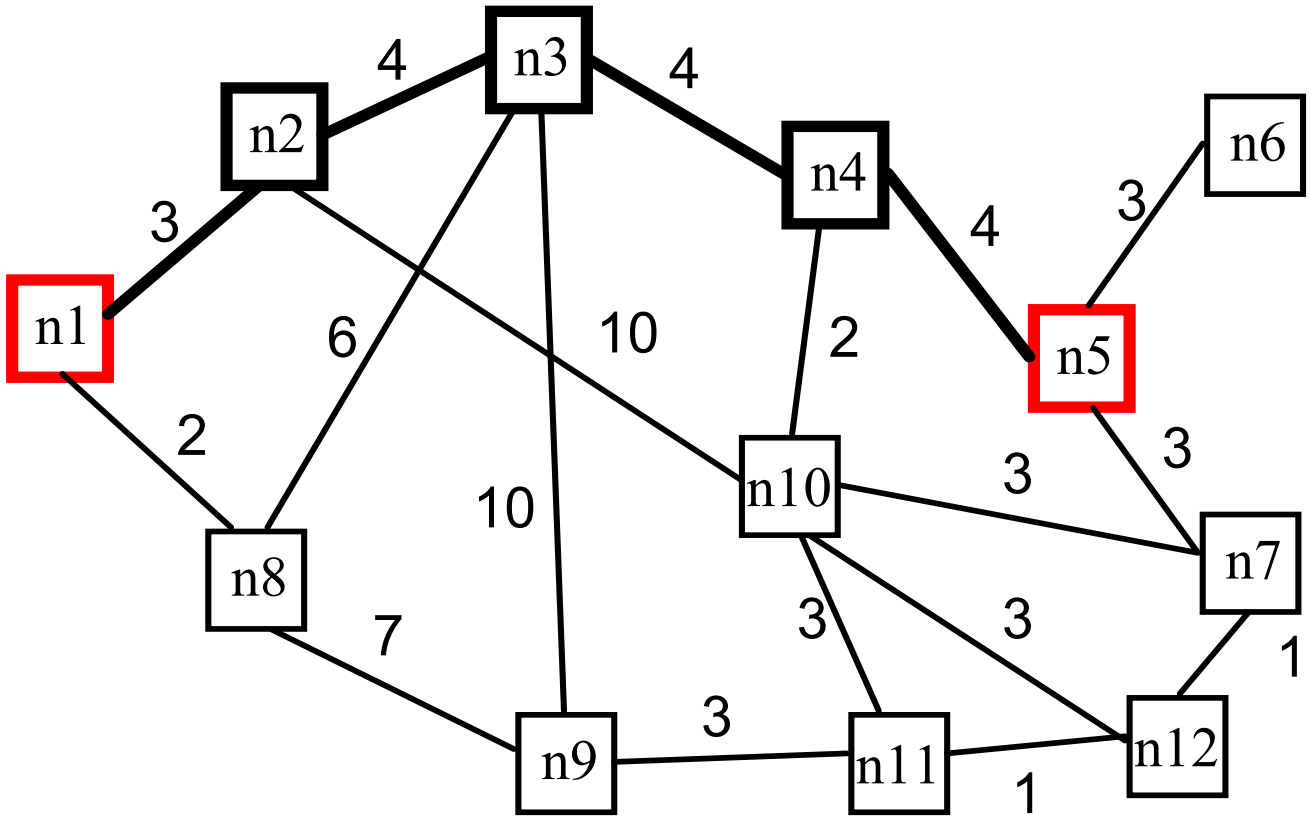


Routing Agent Architecture

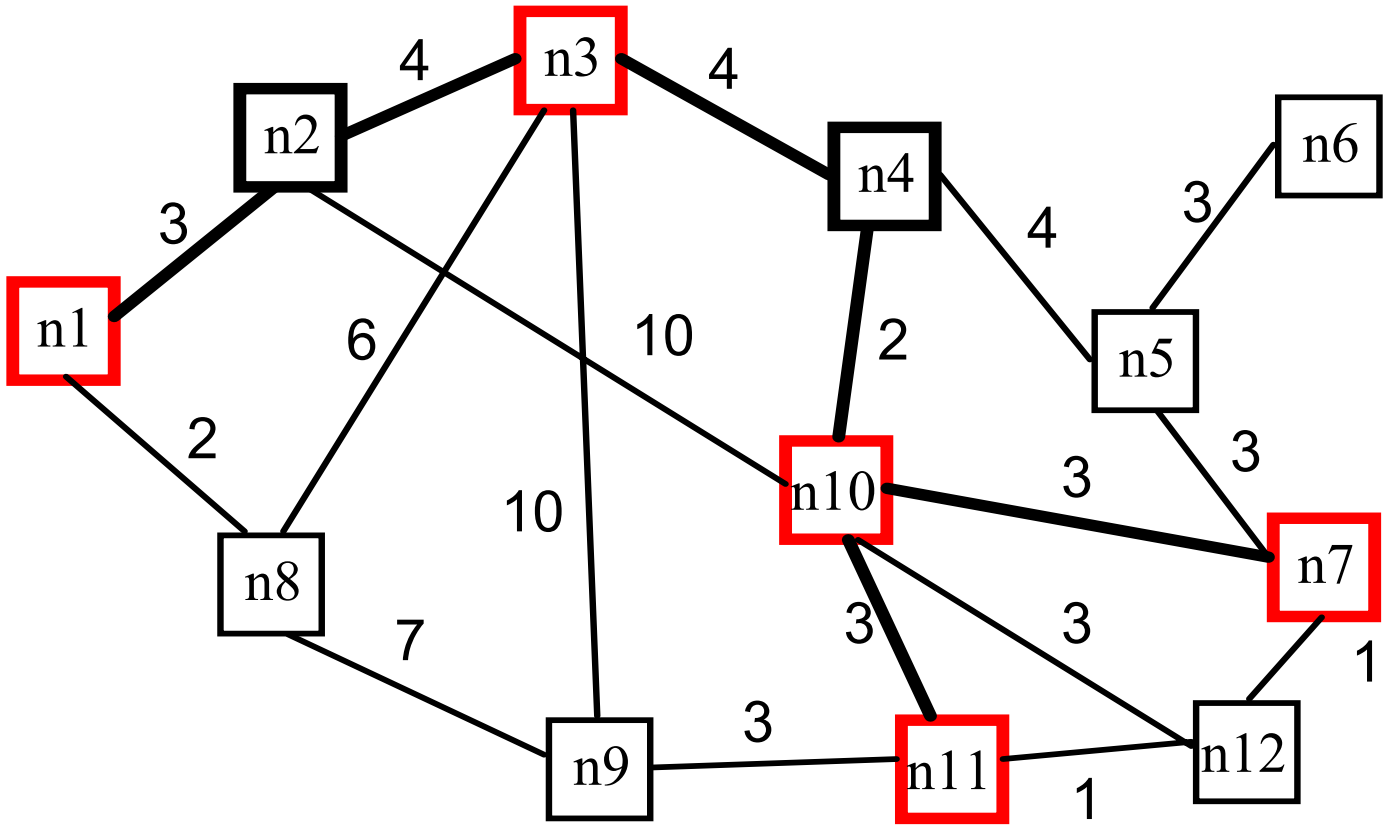
- “heavyweight” static agents that make real decisions present at the source node.
- “lightweight” mobile agents traverse network laying down chemical to reinforce path.



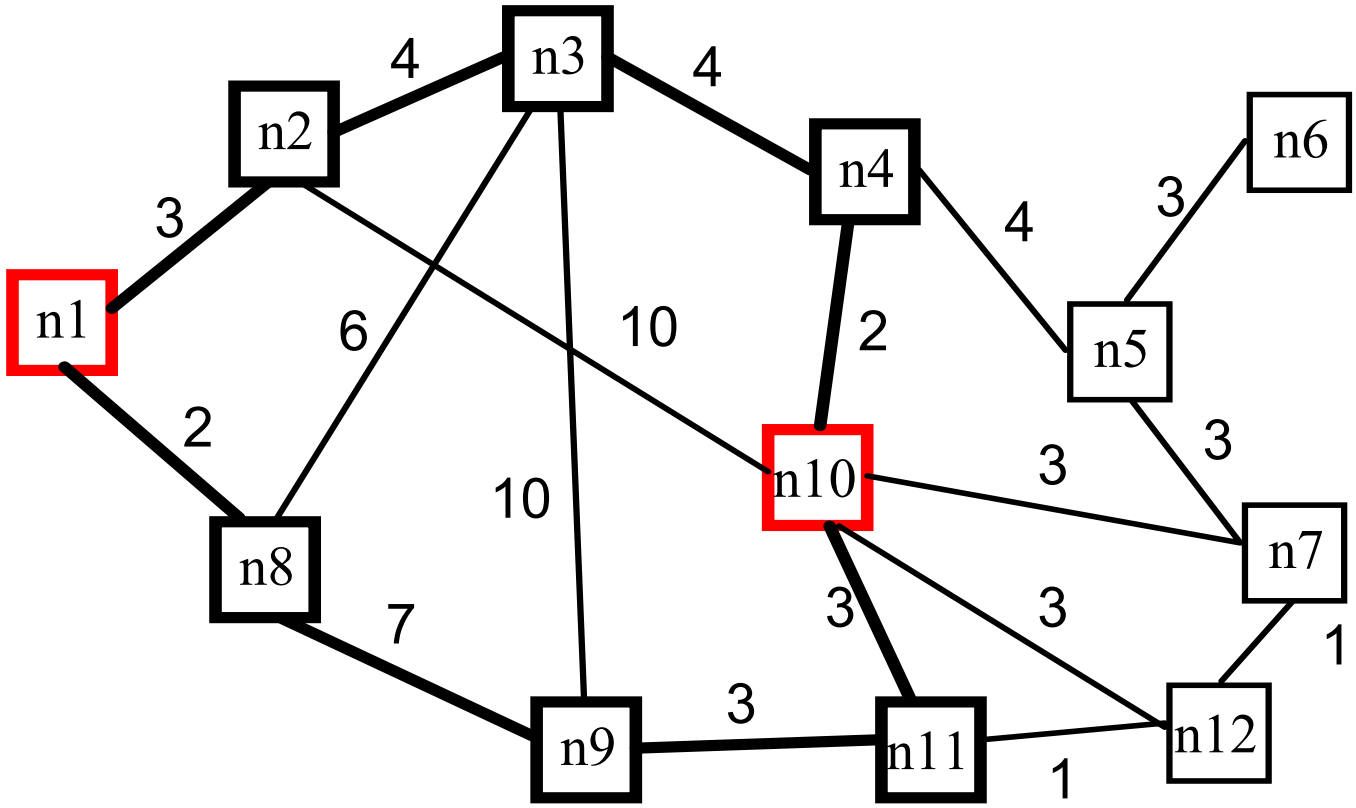
Point to point path



Point to multi-point path



Protected Path



Routing agents

- Agent types:
  - explorer
    - used for route determination
  - allocator
    - allocates resources in network when route emerged
  - deallocator
    - deallocates resources in network when removing connection

Point-2-Point Connections

- For explorer agents:
  - At each node, they choose path with probability proportional to $f(c_e, p_e)$;
  - explorers visit edges once only (achieved through use of tabu list);
  - when destination reached, ants return along the path explored laying down pheromone trail;
  - when explorers return a decision is made regarding path emergence

Path Emergence

- At source node ("nest"):
  - store paths for previous m explorer agents;
  - when p% follow same path allocator agent is sent to allocate bandwidth in network;
  - explorer agents continue to look for new (possibly better) paths.
- Applies for one or many pt-2-pt connections:
  - ants use different, non-reacting pheromones.

Explorer agent algorithm

1. **Initialize**

   set t:= 0

   For every edge (i,j) set an initial value $T_{ij}(t)$ for trail intensity. Place m ants on the source node. [Generate new explorers at freq. e_{fj}]

2. **Set s:= 1 { tabu list index)**

   for k:= 1 to m do

   Place starting town of the kth ant in $tabu_k(s)$.

3. **Repeat until dest'n reached:**

   Set s := s + 1

   for k:=1 to m do

   Choose the node j to move to with probability p_{ij}^k (t)

   Move the kth ant to node j.

   Update explorer route cost:

   $r_k = r_k + c(i,j)$

   if ($r_k > r_{max}$)

   kill $explorer_k$

   Insert town j in $tabu_k(s)$.

   At destination go to 4.

4. **While s > 1**

   traverse edge (i,j)

   $T(i,j) = T(i,j) + p_e$

   s := s - 1

5. **At source node do:**

   if ($path_e$ = pathBuffer * d)

   create and send allocator

   if t > Tmax

   create and send allocator

Evaporation occurs concurrently with exploration

Point-2-Multipoint Connections

- For j destinations, consider as j pt-2-pt connections with:
  - same pheromone, i.e. all explorers communicate;
  - j allocator agents only allocate bandwidth once;
  - allocator send decision made when % of all j explorer ants agree on spanning tree.

Routing Function

Transition probability (mdf):

$p_{ij}^{k}(t) = [T_{ijk}(t)]^{-\alpha}[C(i,j)]^{-\beta} / N_k(i,j,t)$

$N_k(i,j,t) = \Sigma_{j \text{ in (S-Tabu(k))}} [T_{ijk}(t)]^{-\alpha}[C(i,j)]^{-\beta}$

α, β are control parameters that determine the sensitivity of the algorithm to link cost and pheromone.

C(i,j) a ***function*** that depends upon the ***type of traffic***, the ***length*** and ***utilization*** of the link.

Allocator agents

- Allocator agents can fail:
  - bandwidth already allocated by time allocator is sent;
  - allocator agent backtracks to source rolling back resource allocation and decreases pheromone levels;
  - decision to re-send allocator made at a later time (a backoff period is observed);
  - explorer ants continue to search for routes.

Routing Results

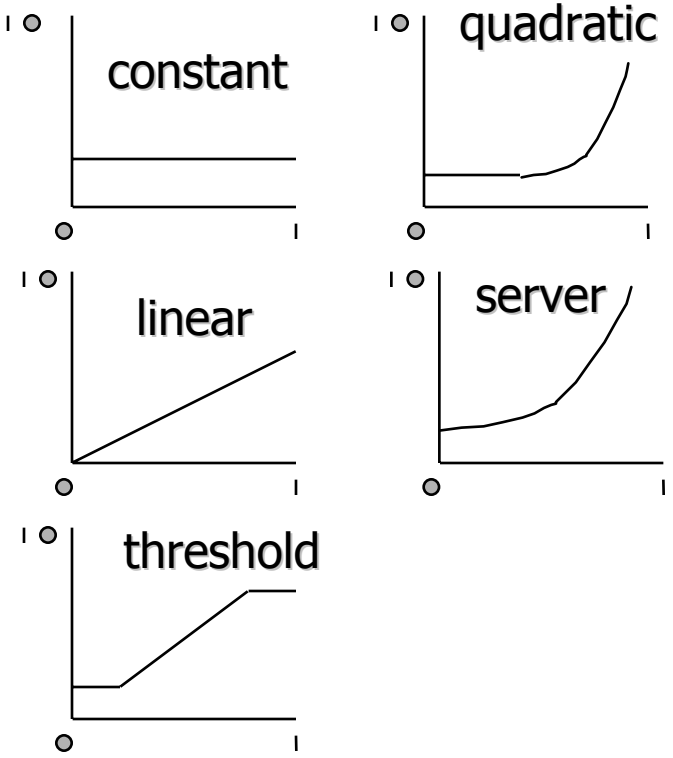
- Shortest paths emerged quickly
- Mixed pt-2-pt, pt-2-mpt routes emerged
- Routing responds to changes in environment:
  - node failure;
  - link failure;
  - link cost changes

Parameter Sensitivity

- Bad solutions and stagnation
  - For high values of α the algorithm enters stagnation behavior very quickly without finding very good solutions.
- Bad solutions and no stagnation
  - α too low, insufficient importance associated with trail.
- Good solutions
  - α , β in the central area (1,1), (1,2), (1,5), (0.5, 5)

Routing Results

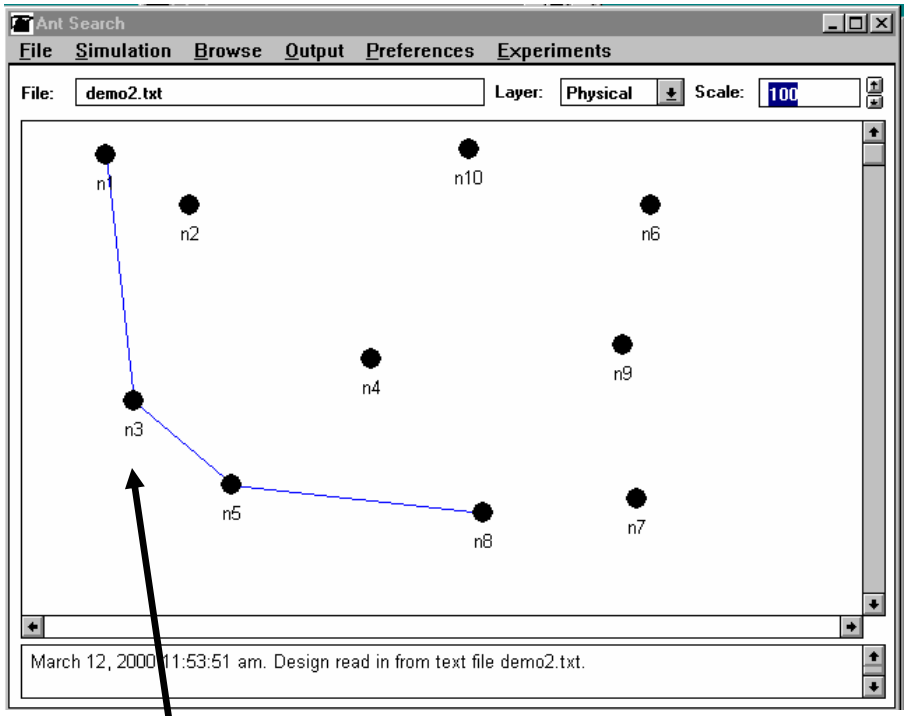
- Link cost functions: C(i,j)



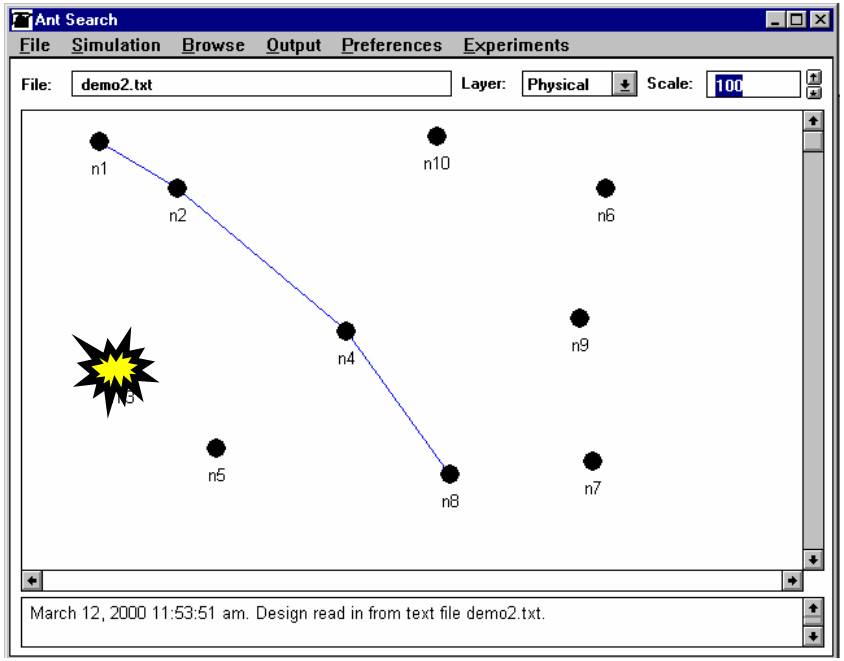
Five functions studied experimentally:

- constant
- linear
- linear threshold
- quadratic
- server (1/1-u)

At high occupancy (> 50%) server appeared to give the best results. At low occupancy (<25%) function relatively unimportant.



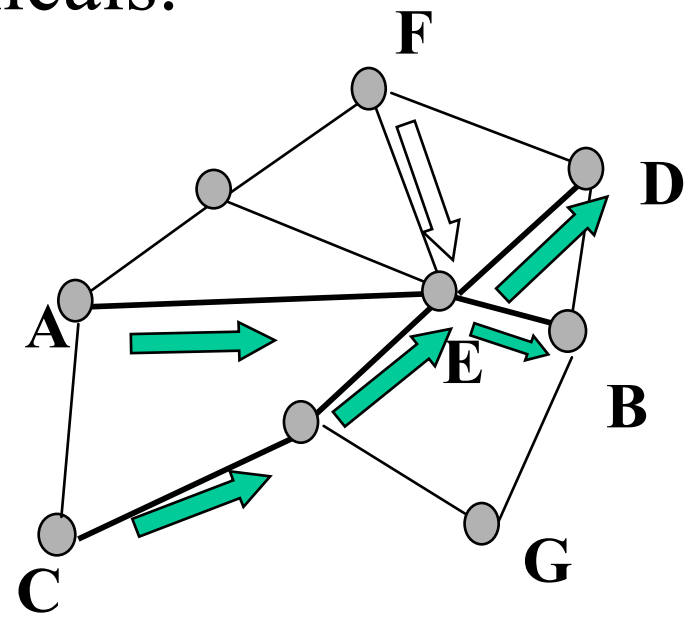
Fail n3, route is recomputed



Multi-priority routing

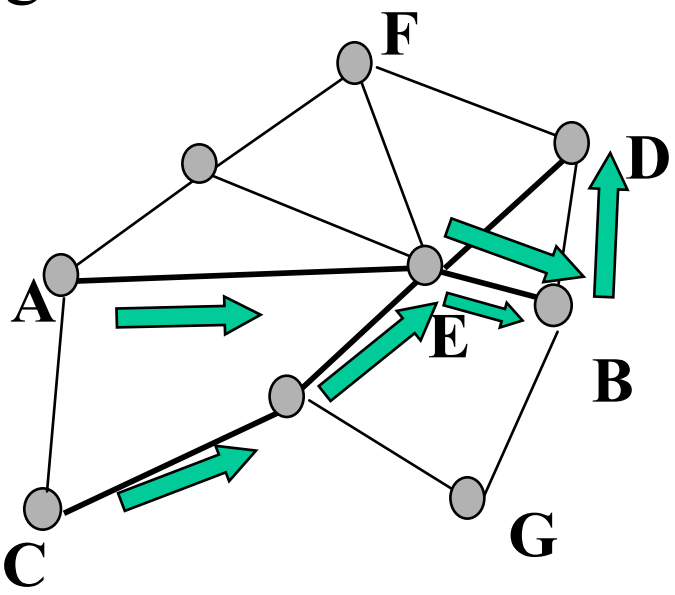
- AB (10), CD (11) standard routing agents.
- FG (00) is a high priority routing agent.
- FG evaporates 10,11 chemicals.

  1# → ‘nothing’



Preferred application routing

- Two (or more) applications should use common path elements:
  - use same chemical in computing route or,
  - use wild cards in receptor; e.g. 0#1 to detect 001 and 011.



Application coalescence/avoidance

- Force applications to go along particular paths.

  Send agent into network with chemistry:

  $X + Y \rightarrow Z$

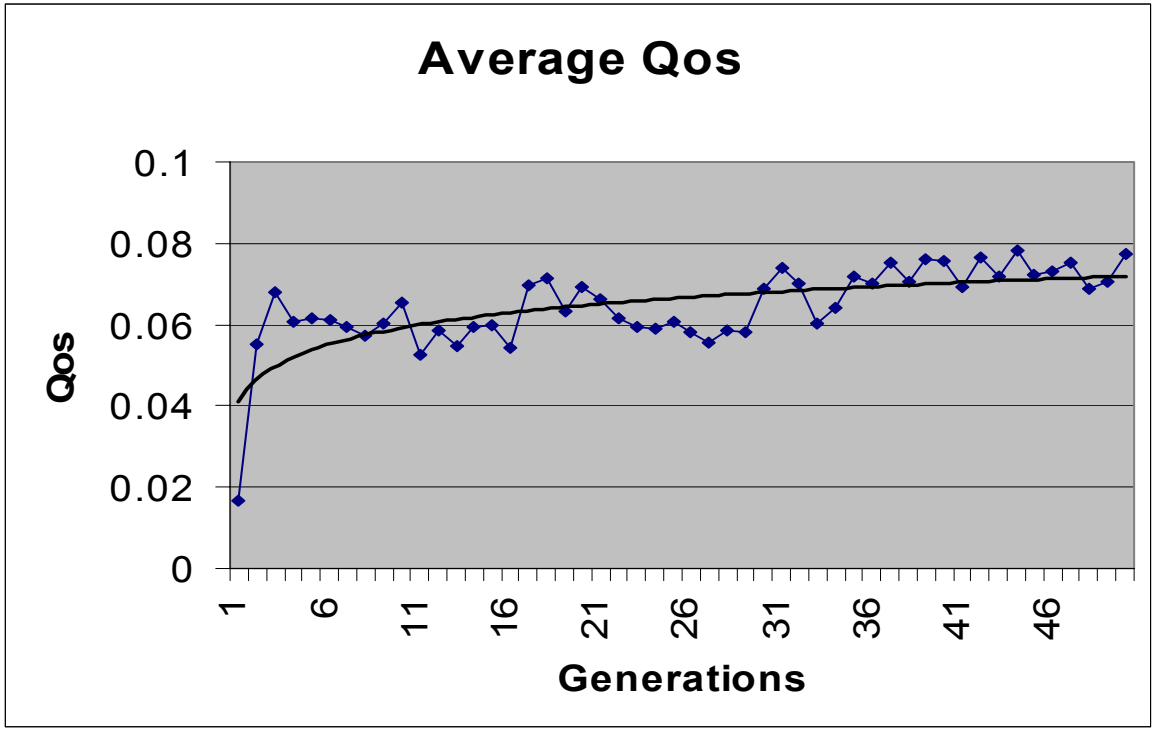
  Agents generating X or Y can also sense Z.

- Force applications to avoid another.

  Agent uses catalytic conversion:

  $X + Y \rightarrow Y$

Application Oriented Routing



Adapt encodings for pheromones if applications sharing resource have high QoS

Service Dependency Model

- Services (S) depend upon resources (R_i)

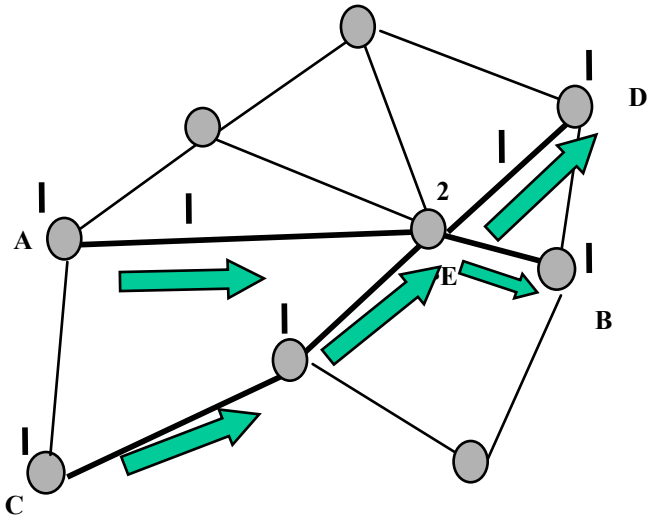
$$S \mapsto \{(R_i, p_i)\}$$

For Example:

A PVC depends upon the nodes and links that it passes through. Those links, in turn, depend upon lower level links in the virtual network.

Connection monitoring agents

- Connection's quality of service monitored.
- Changes in QoS cause Connection monitoring agents to be sent into network, laying down q-chemical.



Diagnosis

- Diagnosis agent is ‘hill climbing’ in space of q-chemical.
- Very quickly finds high q-chemical concentrations and initiates diagnostic activity.
- An example of distributed diagnosis through constructive chemical interference.

Overlay chemical agents

- Want to separate sensing from reasoning.
- Have agents that interact with actual components, generating chemical messages.
- Have “pure” agents that use only chemical concentrations for movement and problem solving.

Learning

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

Updates to Q values for states are positive or negative depending upon improvement or degradation in QoS. If QoS improves, Q value increases and vice versa.

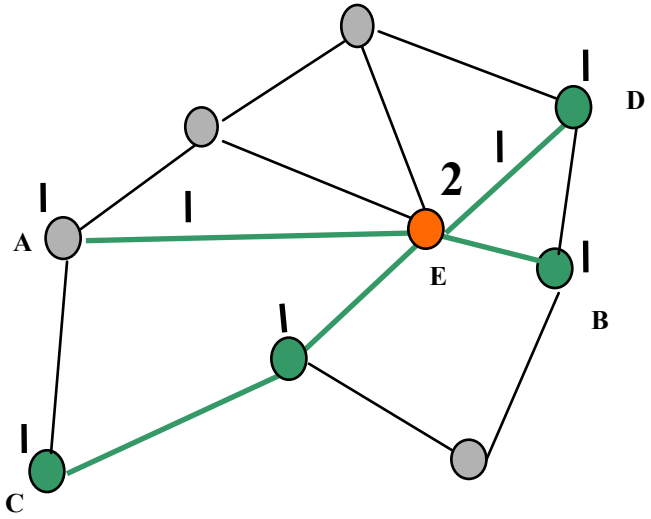
Connection diagnosis agents

- Examples of *netlets*
- Sense q-chemical concentrations

Mdf

$$p_{ij}(t) = Q_{ij}(t) / N(t)$$

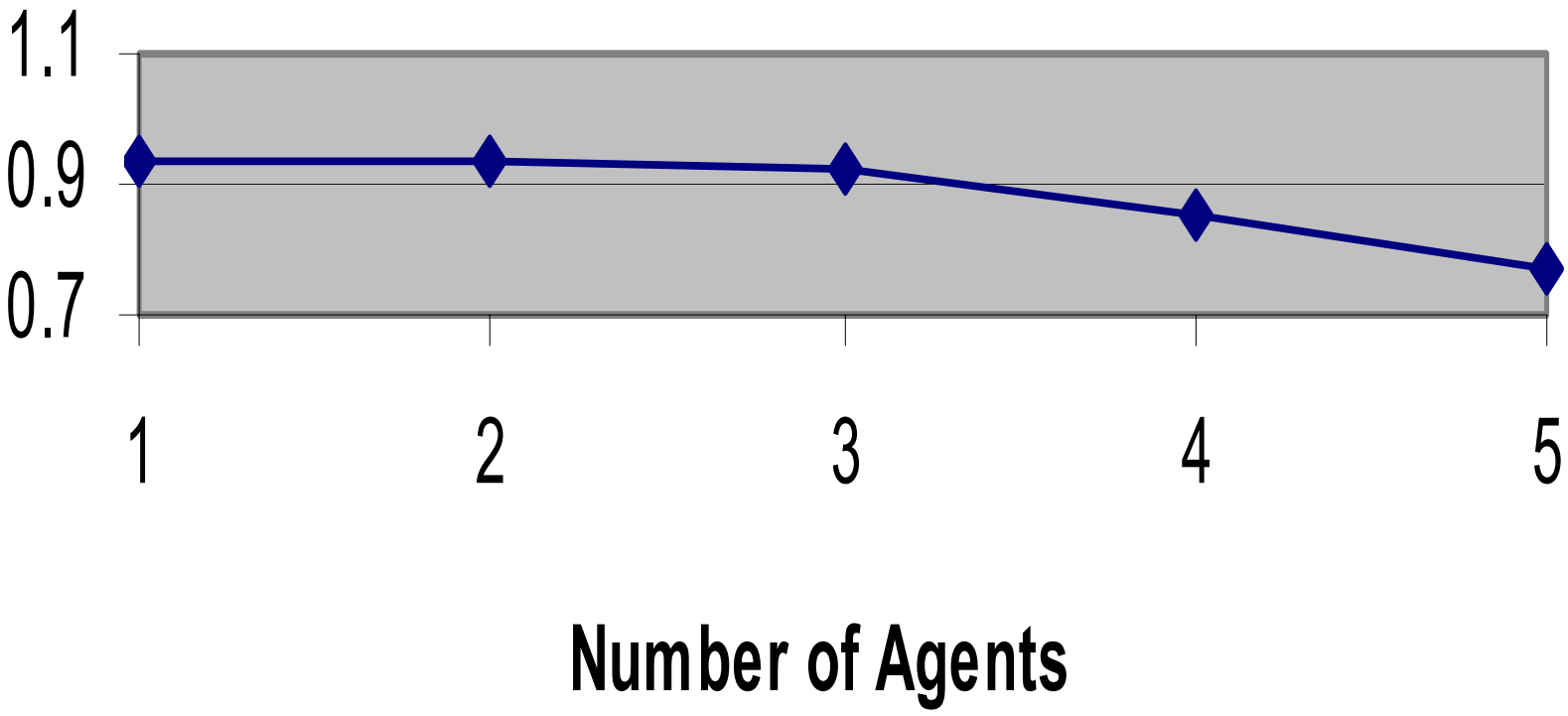
$$N(t) = \Sigma_j Q_{ij}(t)$$

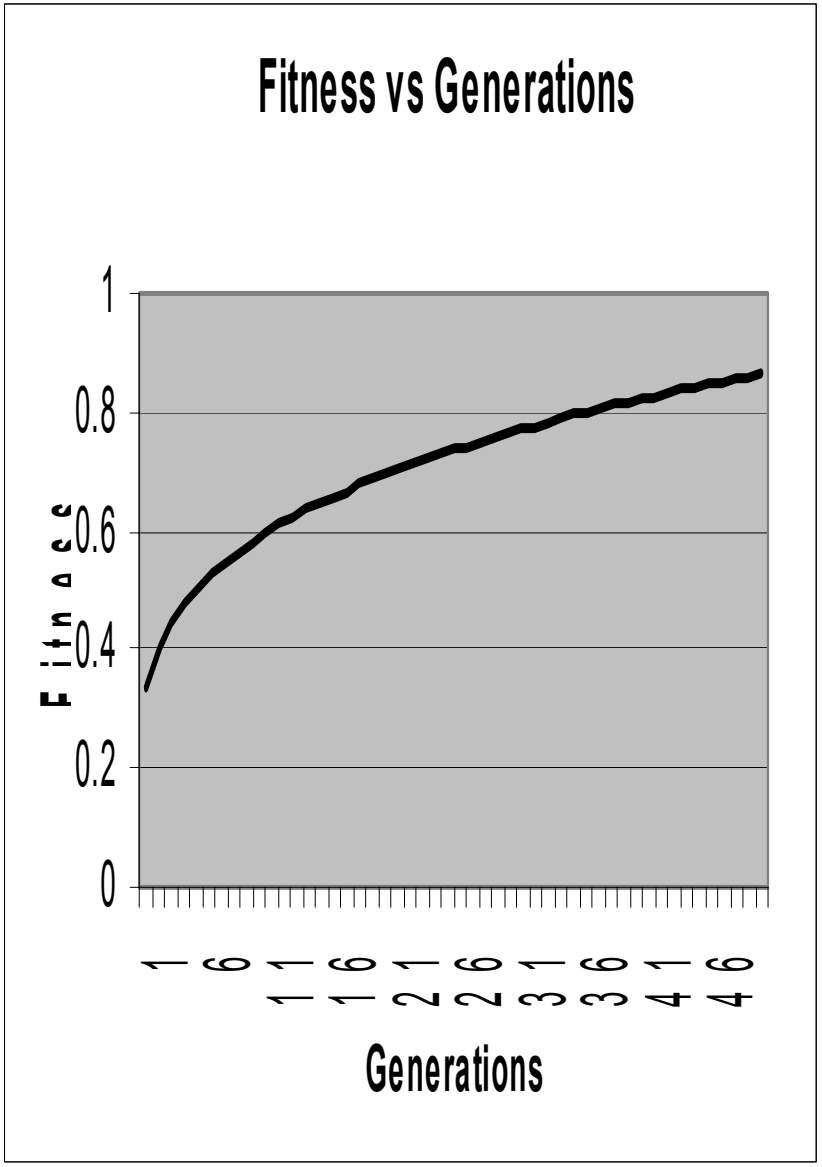


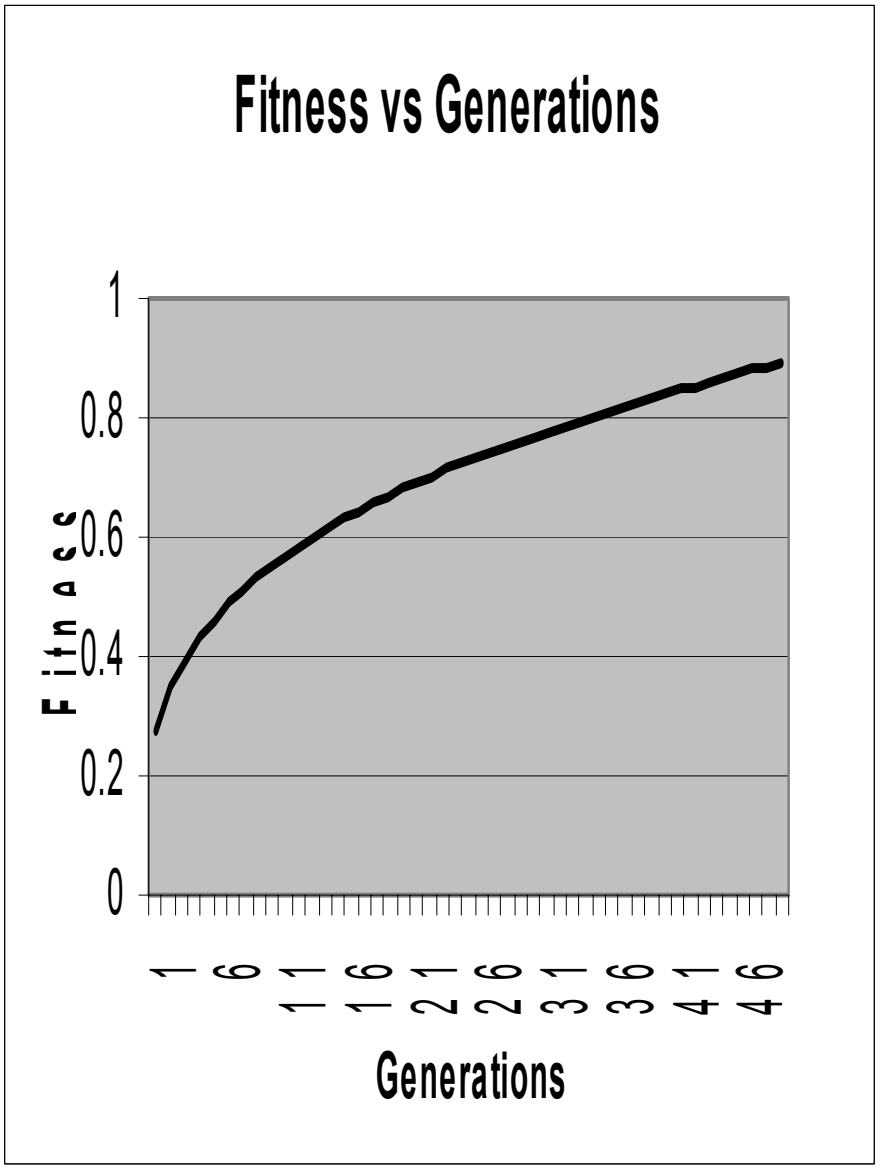
Problem Agents

- PVC *Quality Of Service* agent
  - q-chemical sensing
- *Chronic Failure* agent
  - c-chemical sensing
- *Overload* agent
  - driven by output of condition sensing agents

Converged Performance vs Agent Number




Fitness vs Generations
Fitness
1
0.8
0.6
0.4
0.2
0
1
6
11
16
21
26
31
36
41
46
Generations


Fitness vs Generations
Fitness
1
0.8
0.6
0.4
0.2
0
1
6
11
16
21
26
31
36
41
46
Generations

Unreliable components

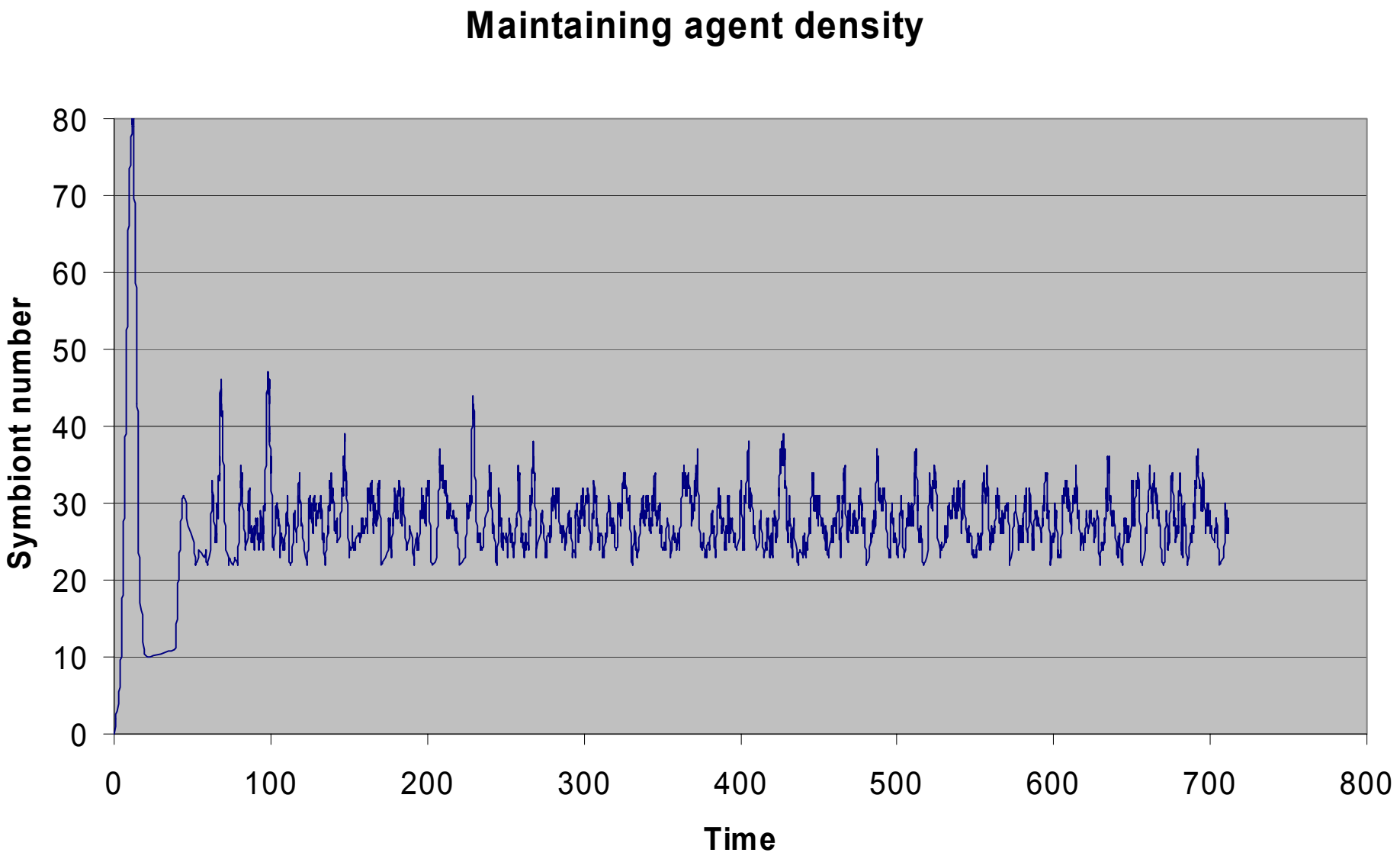
- Install a broad spectrum evaporator agent (or have it move around network).

  1##### → ‘nothing’

- Unreliable chemical + congestion chemical used to generate another which is used by an agent to infer that a component is unreliable in situations of congestion.

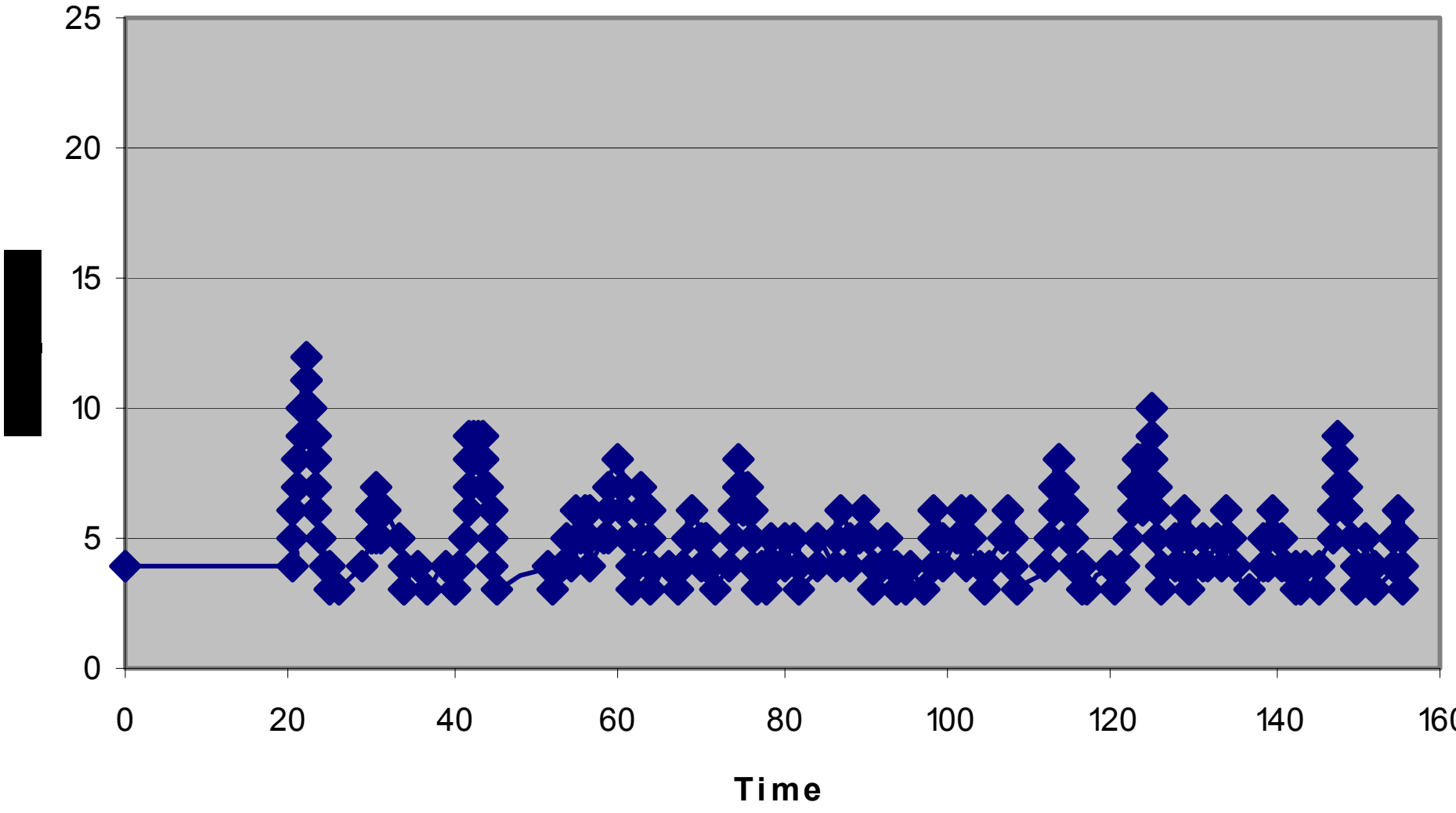
Symbiont Agent Classes

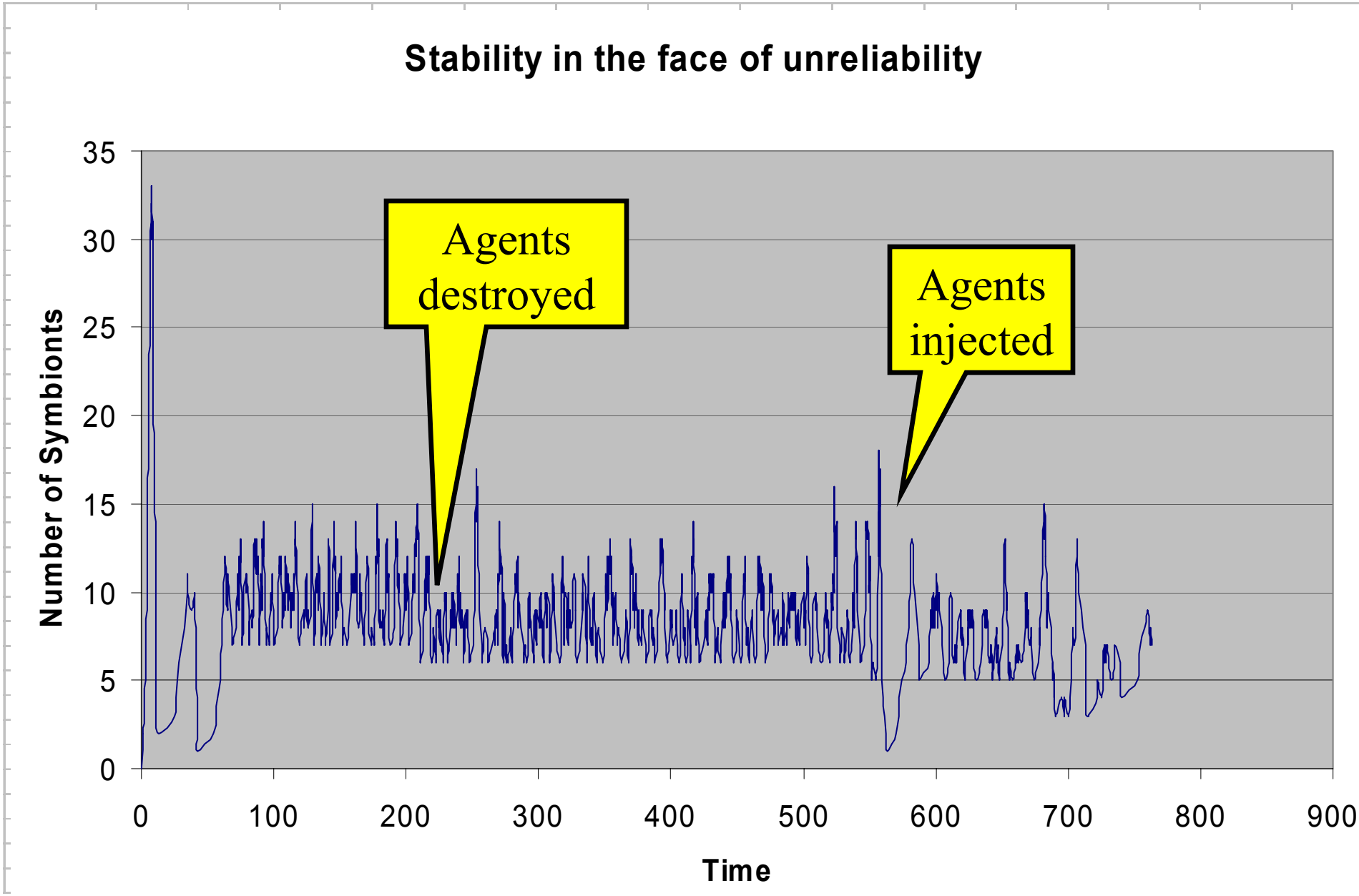
- Network is unreliable, we may lose agents
  - need to regenerate agents
- $A_1=(E_1,R_1,C_1,MDF_1,m_1)$,$A_2=(E_2,R_2,C_2,MDF_2, m_2)$, share chemicals in their emitters and receptors; i.e. $E_1 \cap R_2 \neq \emptyset$ and $E_2 \cap R_1 \neq \emptyset$.
- A_1 generates new A_2 if visit frequency too low; A_1 dies if visit frequency too high.



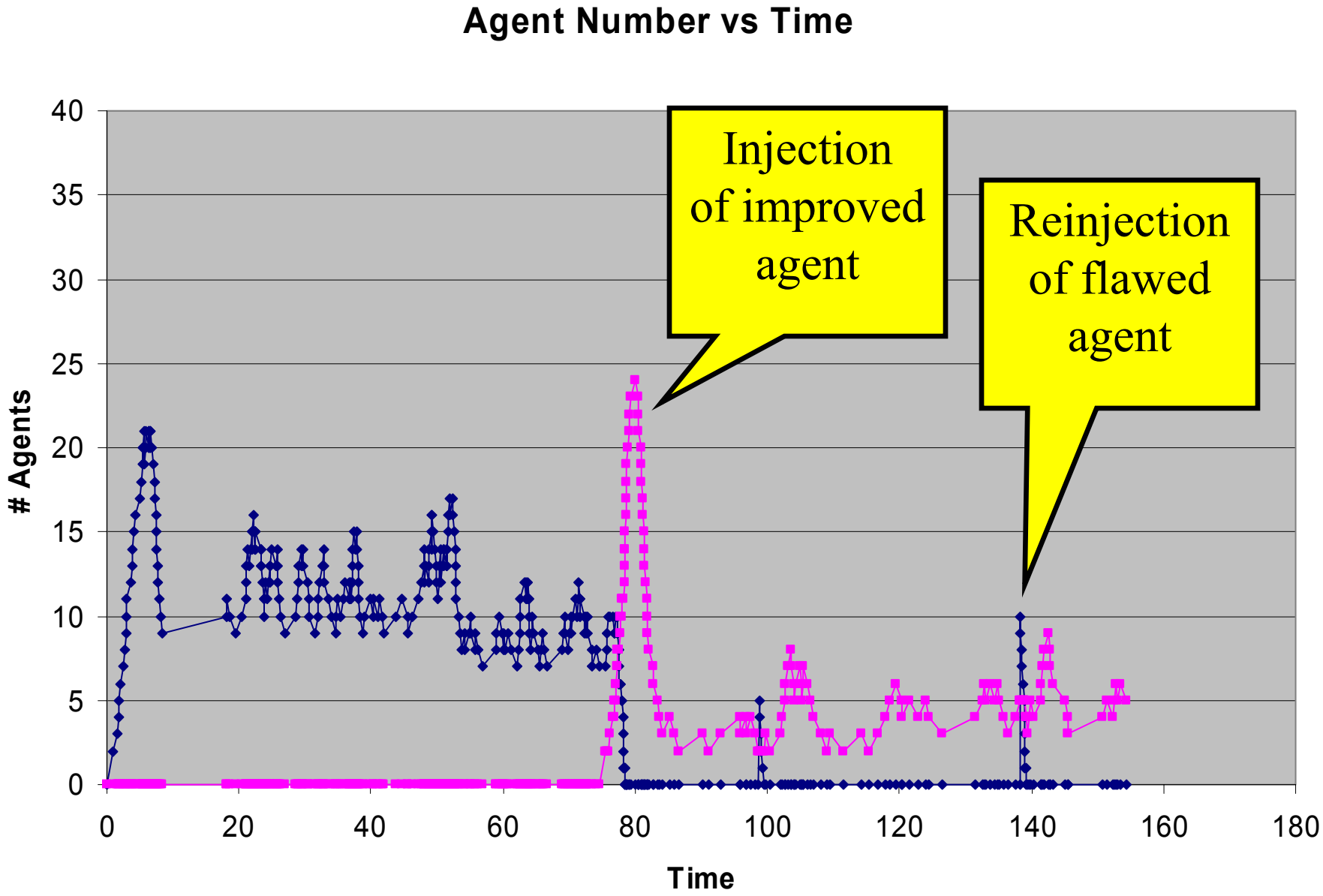
Think of dynamics in terms of Lotka-Volterra

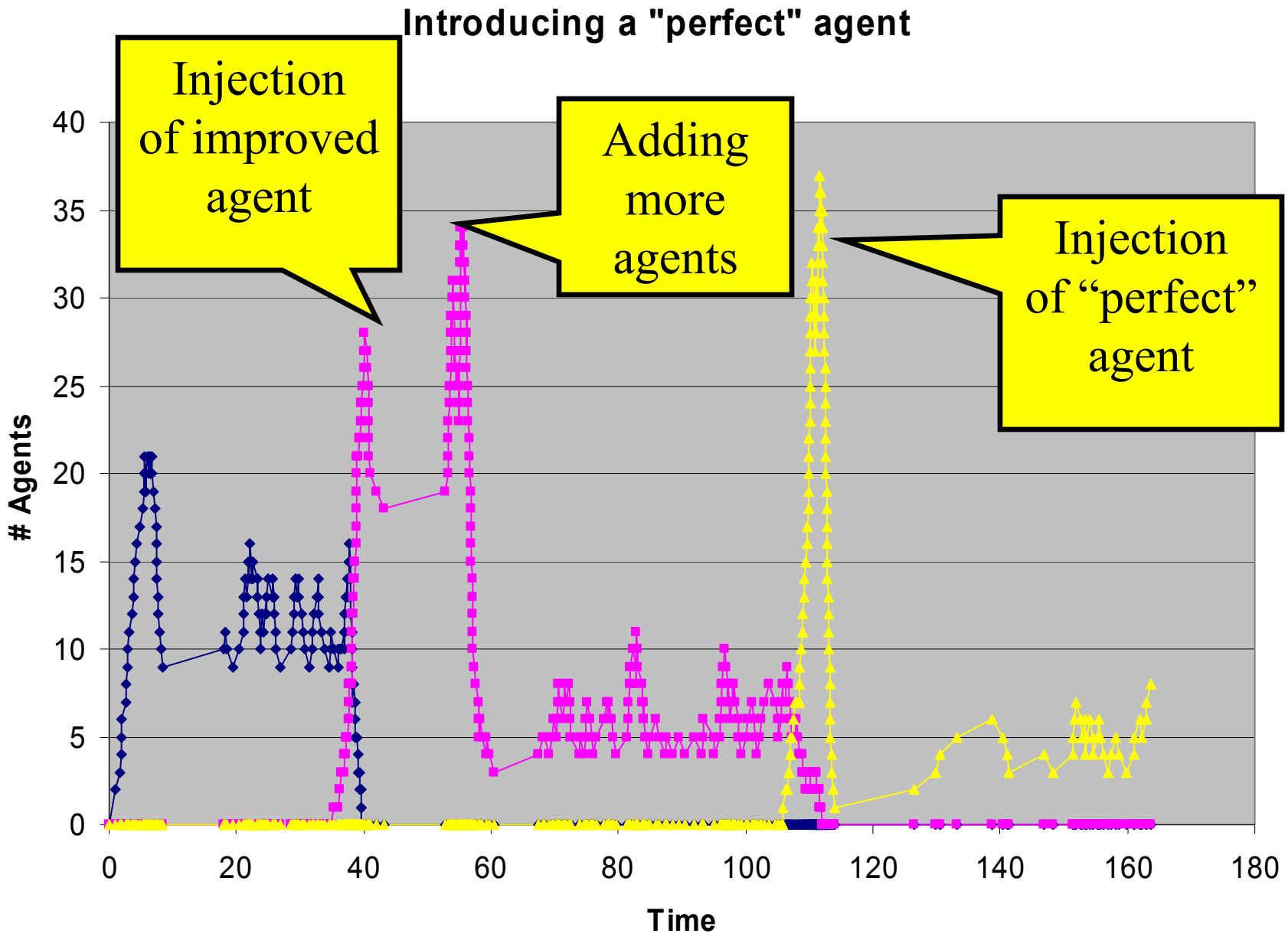
Figure 3. Agent # vs Time

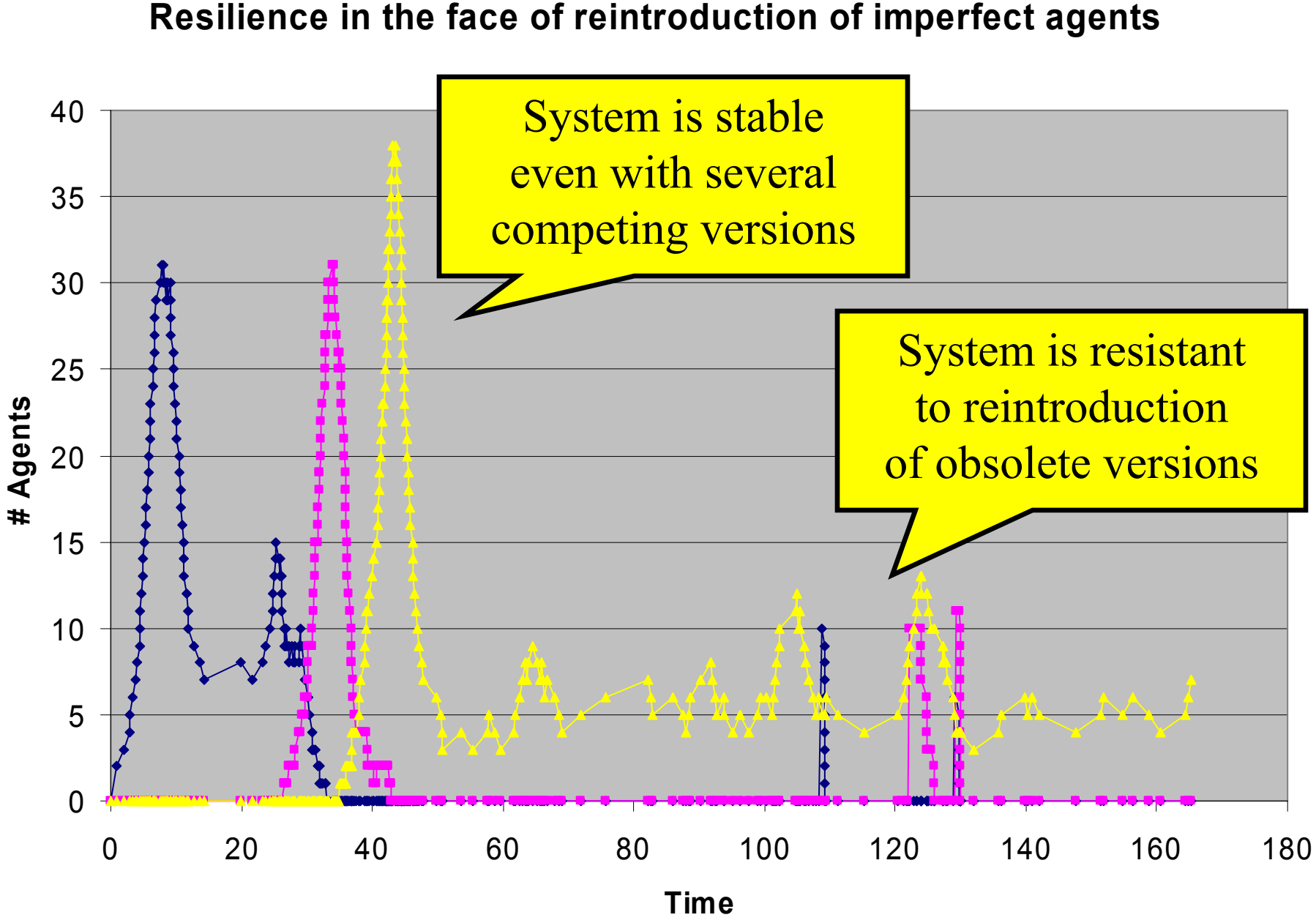




Think of dynamics in terms of Lotka-Volterra

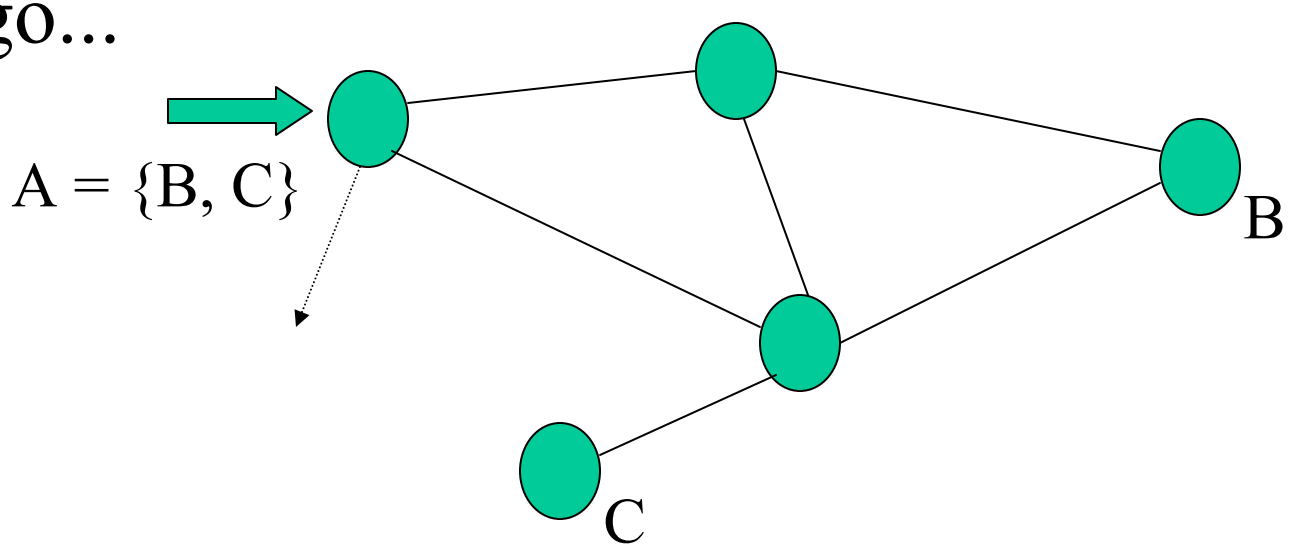

Agent Number vs Time
Agents
Time
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100
120
140
160
180
Injection of improved agent
Reinjection of flawed agent


Introducing a "perfect" agent
Injection of improved agent
Adding more agents
Injection of "perfect" agent
Agents
Time
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100
120
140
160
180


Resilience in the face of reintroduction of imperfect agents
System is stable even with several competing versions
System is resistant to reintroduction of obsolete versions
Agents
Time
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100
120
140
160
180

Bread Crumb Algorithm

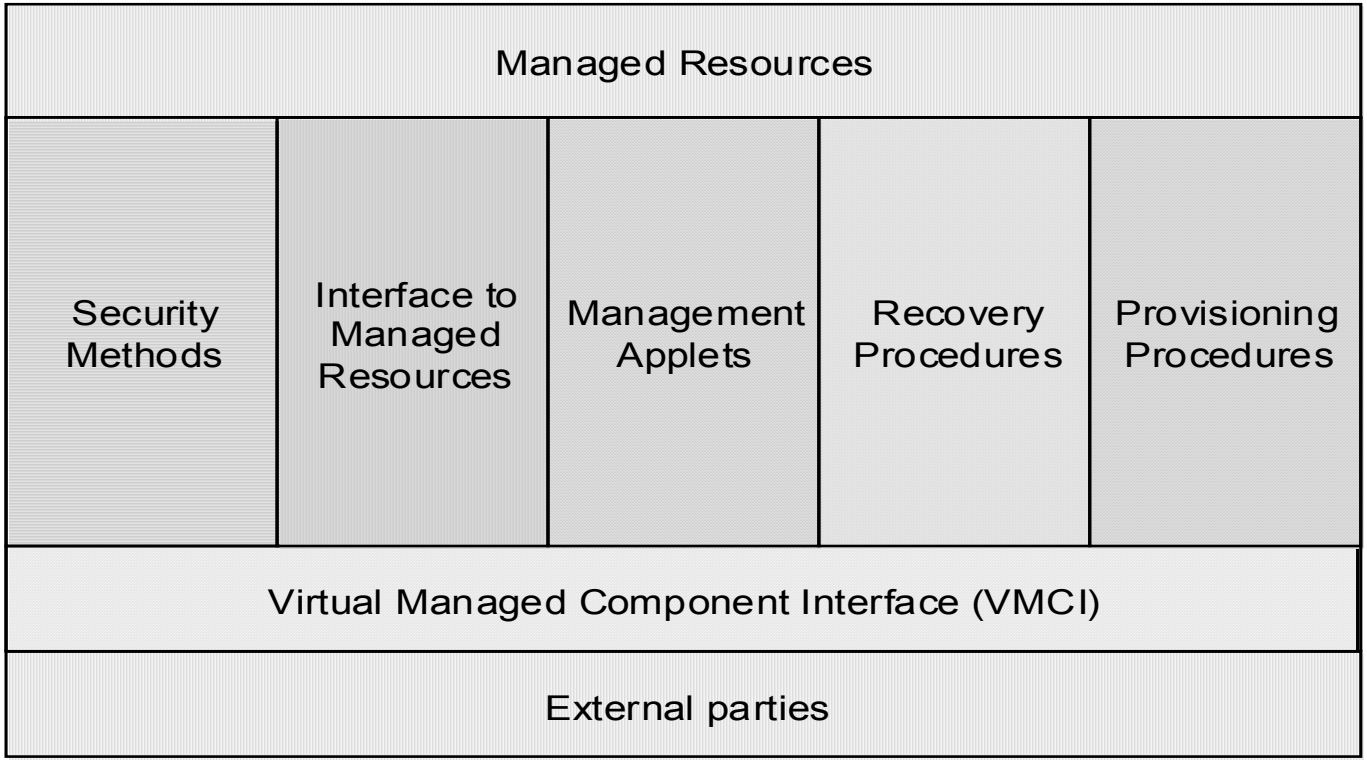
- Used to configure a service in a network; think of code mobility.
- Inject base class into network and watch it go...



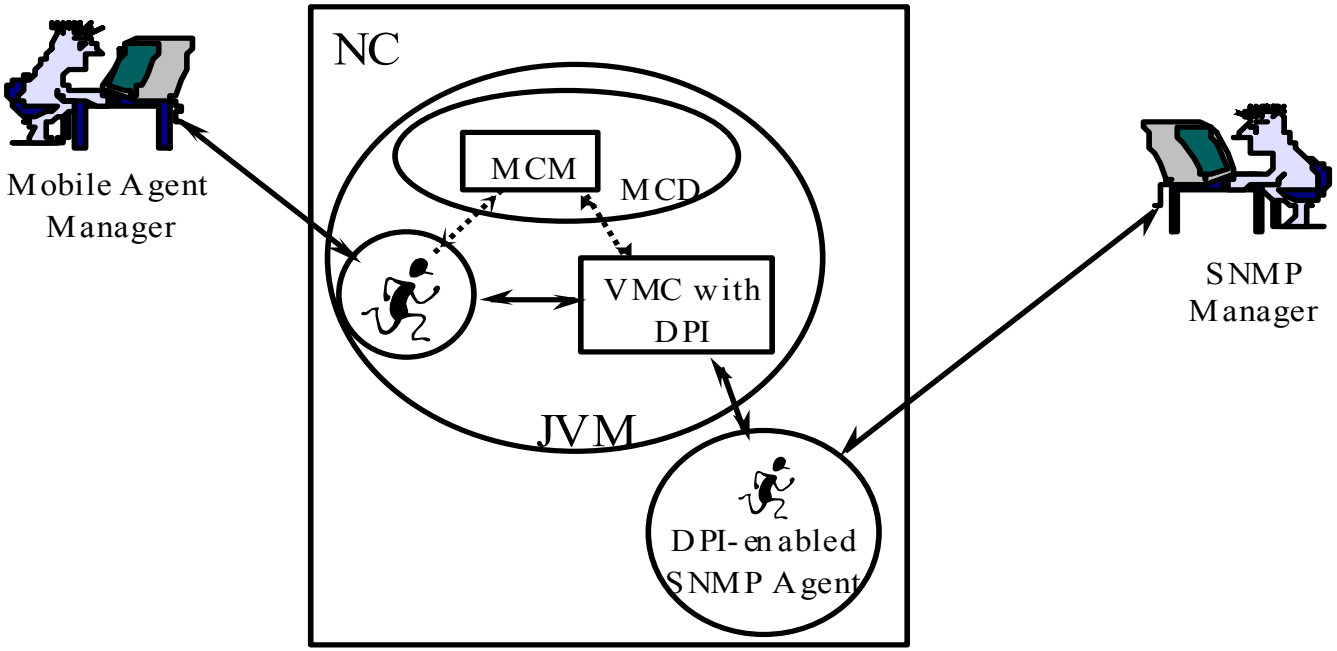
Implementation

"How it works"

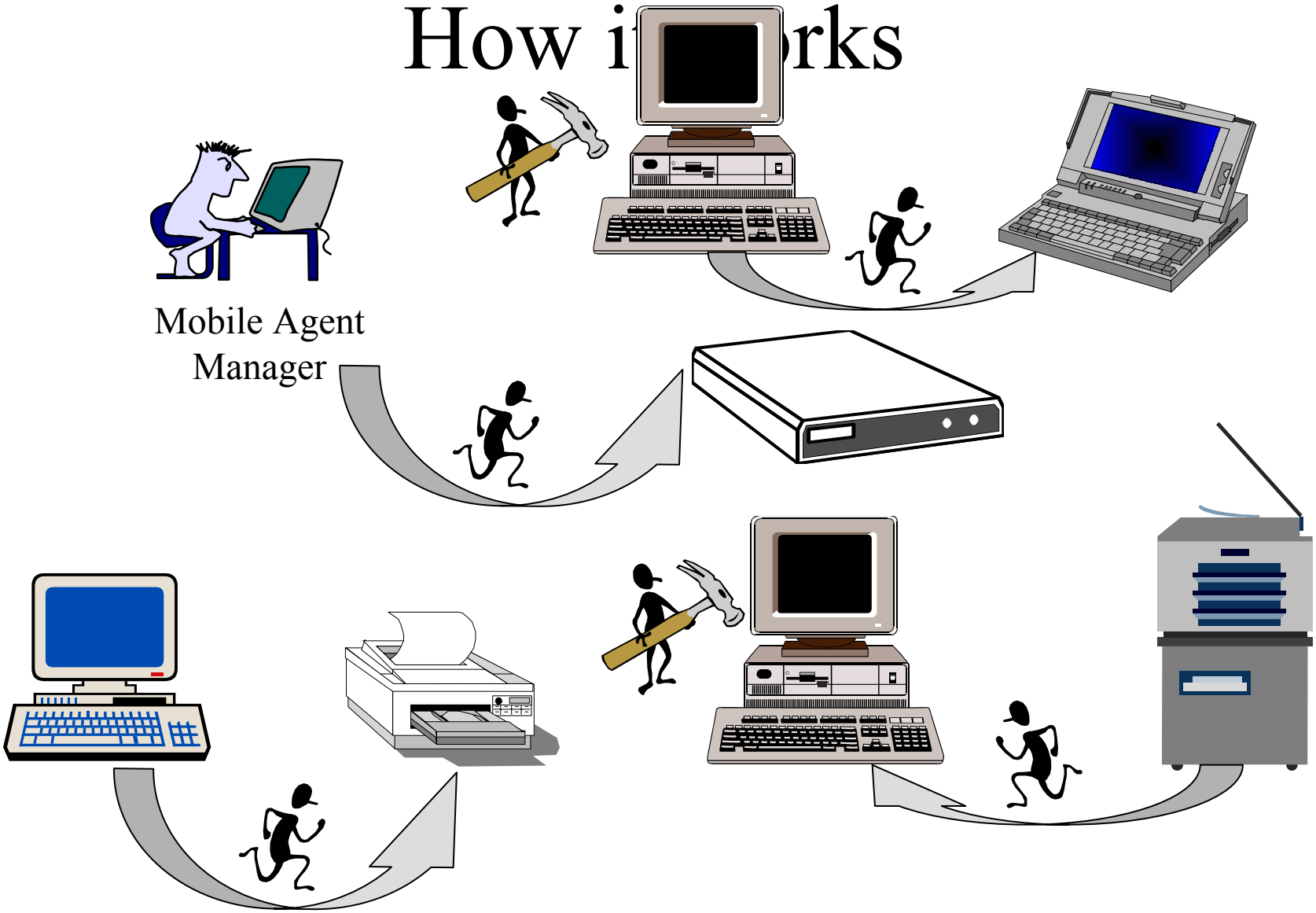
Virtual Managed Component



Sensing Agents



How it works



What the press say…

Newscientist 24th January, 1998

‘Rama Nune, a senior network designer at MCI, says the ants could eventually handle most of the day-to-day tasks of running a telephone network--everything from managing the flow of traffic to calculating everyone's phone bill. "**We are trying to exploit the autonomous intelligence of the ants and the distributed network of information they use**," says Nune.

MCI also wants to let the ants **evolve**. All ants implement an algorithm for routing information. But it is possible that the existing algorithms could be improved upon. One way to find better solutions is to let, say, a network management ant "breed" with a billing ant to produce a hybrid. In order to breed, the ants swap small segments of their programming code. Each ant will have a built-in method for judging how well it does its allotted task, using a measure known as a fitness function. By killing off unfit ants and allowing the fittest to carry on breeding, it may be possible to produce ants that do their jobs better than any human-designed ant.’

Futuristic?

- Active networks are being researched at:
  - M.I.T.
  - U. Penn.
  - CMU
  - Georgia Tech.
- Management by delegation [Yemini, 91] considered an essential design criterion for next generation network management systems.

Where next?

- Build a complex web of interacting agents, assess stability.
- Formulate a model based on reaction kinetics in order to provide analytical framework.

Good thesis work Here!