Self-stabilizing population of mobile agents

Z. Gołębiewski
University of Wroclaw

M. Kutyłowski, F. Zagórski
Wrocław University of Technology

T. Łuczak
Adam Mickiewicz University, Poznań

IPDPS'2008, 4/15/2008, Miami

Model

We consider a network:

- consisting of n nodes
- fully connected:
  a node can send a message directly to another node

.. and mobile agents in such a network.

Mobile agent - definition

mobile agent is a unit that can migrate through the system.

activities an agent:

1. can migrate to an arbitrary chosen node,
2. can reproduce itself, i.e. generate its copies at the node where it resides,
3. can kill other agents or become killed (e.g. by another agent or the system)

Time

- Time is divided into synchronous rounds.
- Each round consist of 2 phases:

move phase an agent can migrate to another node,

evolution phase an agent can

1. reproduce or become killed,
2. perform internal tasks

Application of agent systems

worms an agent is a worm that tries to infect as many nodes as possible:

- it tries to replicate through the system (but no more than one worm in a node)
- it tries to behave so that it is hard to catch all copies of a worm

Application of agent systems

monitoring agents: the agents perform some supervision and protect a system consisting of many PCs, they:

- should survive in the system even if a substantial number of PCs is taken over by the adversary
- should be hard to remove even if a (malicious) administrator wants them to switch off for a moment

Main goals

Design an algorithm that

- keeps number of agents in the system around pre-defined level $\alpha = \alpha(n)$,
- agents cannot leave any information in nodes,
- agents can communicate only with the agents residing in the same node.

Previous work

- K. Amin and A. Mikler, *Dynamic agent population in agent-based distance vector routing*, ISDA 2002.
- T. White and B. Pagurek and D. Deugo, *Management of Mobile Agent Systems using Social Insect Metaphors*, SRDS 2002.

Similar algorithms of controlling agents population, but:

- agents leave traces at host nodes
- no analytic results, only simulations

Our Algorithm

Algorithm executed by an agent:

Move: Pick a node uniformly at random; move to this node (agents' choices are independent)

Our Algorithm

Algorithm executed by an agent:

Move: Pick a node uniformly at random; move to this node (agents' choices are independent)

Evolution:

if (there is exactly one agent in the node)
then with probability p it creates a new agent in this node,
else fight!
exactly one of the agents survives (other agents in this node are killed).

Intuitions

There are two opposite mechanisms integrated in the protocol:

- if the number of agents is low, then the number of agents is increasing
  (agents replicate, killing occurs rarely since they do not meet frequently),

Intuitions

There are two opposite mechanisms integrated in the protocol:

- if the number of agents is low, then the number of agents is increasing
  (agents replicate, killing occurs rarely since they do not meet frequently),
- if the number of agents is high, then the number of agents is reduced
  (agents meet frequently, killings outnumber replications)

Algorithm analysis

The analysis is based on the labeled combinatorial structures and their exponential multivariate generating functions (EMGF). They allow us to compute easily:

- the expected number of the number of born and killed agents in a network, (possible with the " approach)
- the variance of the number of born and killed agents in a network - handling with dependencies between agents!

Basic definitions and notation

- Let $\mathcal{Z}$ be the atomic class, i.e. $\mathcal{Z} = \{①\}$, and ① be a labeled atom of size 1 (an atom corresponds to the agent).

Basic definitions and notation

- Let $\mathcal{Z}$ be the atomic class, i.e. $\mathcal{Z} = \{①\}$, and ① be a labeled atom of size 1 (an atom corresponds to the agent).
- The EGF of the atomic class $\mathcal{Z}$ is $Z(z) = z$.

Basic definitions and notation

- Let $\mathcal{Z}$ be the atomic class, i.e. $\mathcal{Z} = \{ \textcircled{1} \}$, and $\textcircled{1}$ be a labeled atom of size 1 (an atom corresponds to the agent).
- The EGF of the atomic class $\mathcal{Z}$ is $Z(z) = z$.
- Let $\mathcal{P}_k\{\mathcal{Z}\}$ denote the class of all sets of size k of the classes $\mathcal{Z}$.

Basic definitions and notation

- Let $\mathcal{Z}$ be the atomic class, i.e. $\mathcal{Z} = \{①\}$, and ① be a labeled atom of size 1 (an atom corresponds to the agent).
- The EGF of the atomic class $\mathcal{Z}$ is $Z(z) = z$.
- Let $\mathcal{P}_k\{\mathcal{Z}\}$ denote the class of all sets of size k of the classes $\mathcal{Z}$.
- The EGF of the class $\mathcal{P}_k\{\mathcal{Z}\}$ is $P_k(z) = \frac{1}{k!}(Z(z))^k$.

Basic definitions and notation

- Let $\mathcal{P} = \sum_k \mathcal{P}_k\{\mathcal{Z}\}$ corresponds to all possible sets of the labeled atoms.

Basic definitions and notation

- Let $\mathcal{P} = \sum_k \mathcal{P}_k\{\mathcal{Z}\}$ corresponds to all possible sets of the labeled atoms.
- The EGF of the class $\mathcal{P}$ satisfies

$$P(z) = \sum_k P_k(z) = \sum_k \frac{1}{k!}(Z(z))^k = \sum_k \frac{z^k}{k!} = e^z.$$

Basic facts

The key fact concerning multivariate generating functions is that the moment of order 1 of a parameter χ_1 is given by the formula

$$E_{\mathcal{P}^{(n)}}[\chi_1] = \frac{[z^k]\partial_{u_1} P^{(n)}(z,1,1)}{[z^k]P^{(n)}(z,1,1)} \tag{1}$$

where $[z^k]S(z)$ extracts the coefficient of z^k in the power series $S(z)$, and $\partial_{u_1} := \frac{\partial}{\partial u_1}$.

Basic facts

Similarly, the moment of order 2 of a parameter χ_1 equals

$$E_{\mathcal{P}^{(n)}}[\chi_1^2] = \frac{[z^k]\partial_{u_1}^2 P^{(n)}(z,1,1)}{[z^k]P^{(n)}(z,1,1)} + \frac{[z^k]\partial_{u_1} P^{(n)}(z,1,1)}{[z^k]P^{(n)}(z,1,1)} \qquad (2)$$

where $\partial_{u_1}^2 := \frac{\partial^2}{\partial u_1^2}$.

Step 1

- The class $\mathcal{P}^{(n)}$ that describes all nodes in a network is defined as a product of n classes $\mathcal{P}$.
- The EGF of the class $\mathcal{P}^{(n)}$ is given by formula $P^{(n)}(z) = (P(z))^n$.

Step 2

We define parameters χ_1 and χ_2.

- χ_1 associates the number of nodes with exactly one agent to an arrangement of agents in the nodes,
- Since $\frac{z^1}{1!} = z$ describes the nodes with exactly one agent, we will multiply it by a formal variable u_1 that marks χ_1.

Step 2

- χ_2 associates the number of killed agents to an arrangement of agents in the nodes.
- Since $\frac{z^k}{k!}$ describes the node with exactly k agents we will multiply it by a formal variable u_2^{k-1} that marks χ_2.
- $\sum_{k\geq 2} \frac{u_2^{k-1} z^k}{k!}$ describes the nodes with more than one agent.

Step 3

Therefore we get the following exponential multivariate generating function (EMGF)

$$P^{(n)}(z, u_1, u_2) = \left(1 + u_1 z + \left(\sum_{k \geq 2} \frac{u_2^{k-1} z^k}{k!}\right)\right)^n = \dots$$

$$= (1 + u_1 z + \frac{1}{u_2}(e^{u_2 z} - u_2 z - 1))^n.$$

Number of agents born in a round

Lemma

Let k be the number of agents in a network at the beginning of a round. Then,

$$E[\chi_1] = k\left(1-\frac{1}{n}\right)^{k-1} \tag{3}$$

$$Var[\chi_1] = k\left(\left(1-\frac{1}{n}\right)^{k-1} - k\left(1-\frac{1}{n}\right)^{2(k-1)}\right) + \frac{k(k-1)(n-1)(n-2)^{k-2}}{n^{k-1}}. \tag{4}$$

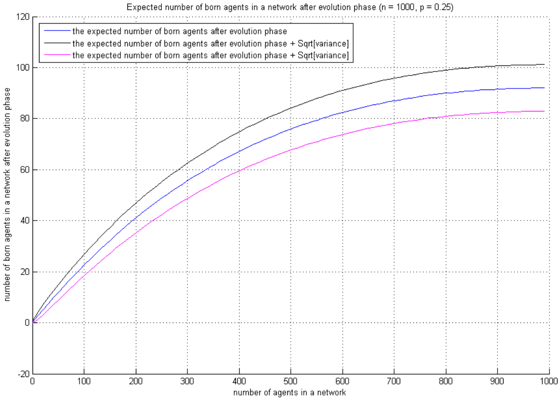
Number of agents born in a round

Corollary

Let B denote the number of agents born in a round such that there are k agents in the network immediately before the round. Then

$$E[B] = p \cdot k \left(1 - \frac{1}{n}\right)^{k-1} \tag{5}$$

$$\begin{aligned} Var[B] &= pk\left(1-\frac{1}{n}\right)^{k-1} - p^2k^2\left(1-\frac{1}{n}\right)^{2k-2} \\ &+ p^2k(k-1)\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)^{k-2} . \end{aligned} \tag{6}$$



The expected number and variance of born agents, for $n = 1000$, $p = 0.25$

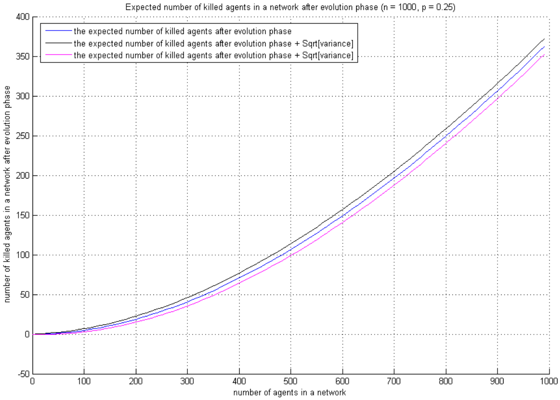
Number of agents killed in a round

Lemma

n = the number of nodes in a network,
p = the agents reproduction probability,
k = the number of agents in a network at the beginning of a round,
K = the number of agents killed in a network after the round.
Then

$$E[K] = k - n + n\left(1 - \frac{1}{n}\right)^k \tag{7}$$

$$Var[K] = n(n-1)\left(1 - \frac{2}{n}\right)^k + n\left(1 - \frac{1}{n}\right)^k - n^2\left(1 - \frac{1}{n}\right)^{2k} .$$



The expected number and variance of killed agents, for $n = 1000$, $p = 0.25$

Equilibrium condition

Definition

We say that the process considered is in the Equilibrium Point, when the expected change of the number of agents in a round equals 0,

i.e. the expected number of agents born in a round is equal to the expected number of agents killed in a round: $E[B] = E[K]$.

Equilibrium condition

Theorem

Let $n =$ the number of nodes in a network,
$p =$ the reproduction probability,
$k =$ the number of agents at the beginning of a round.
Then the Equilibrium Point

$$pk\left(1-\frac{1}{n}\right)^{k-1} = k-n+n\left(1-\frac{1}{n}\right)^{k} \tag{8}$$

is reached for

$$k \approx \frac{2p}{1+p}n\,. \tag{9}$$

Equilibrium condition

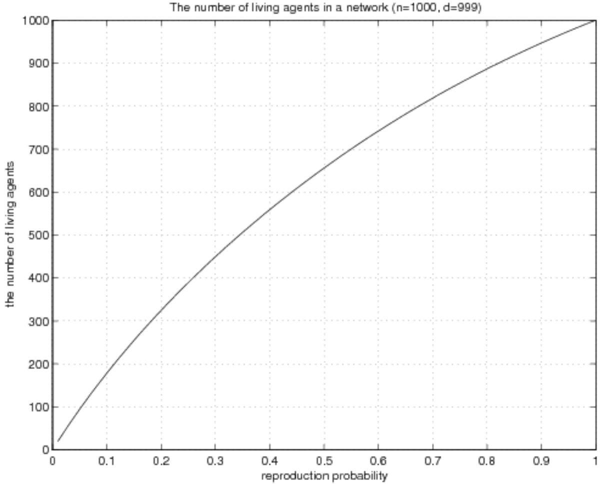
Corollary

The number of agents at the Equilibrium Point can be established on any chosen value $\alpha \cdot n$*,* $0 < \alpha < 1$ *by choosing* $p \approx \frac{\alpha}{2-\alpha}$.

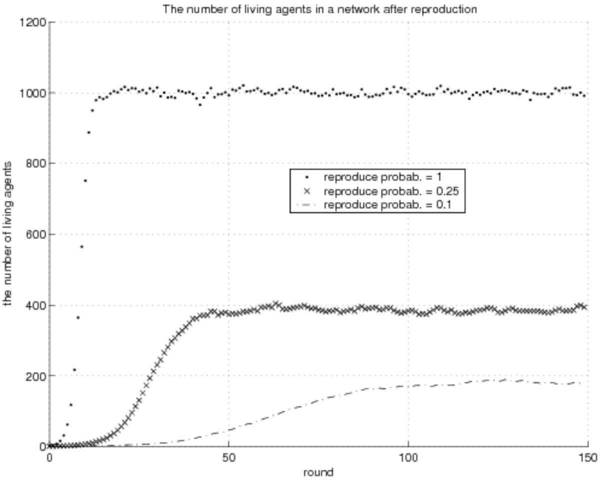
Experimental results

$n = 1000$	according to (8)	average number in simulations
$p = 0.1$	178.46	180
$p = 0.25$	386.054	385
$p = 1$	1000.58	1000

Equilibrium Point versus the average number of agents for $n = 1000$



Equilibrium Point for different reproduction probabilities, for $n = 1000$



Evolution of the number of agents, an example simulation for $n = 1000$

Final Remarks

- the rate of convergence to the Equilibrium Point has not been covered by the paper,

Final Remarks

- the rate of convergence to the Equilibrium Point has not been covered by the paper,
- computed values of the variances of variables B and K are quite low– this influences on high rate of convergence.

Final Remarks

- the rate of convergence to the Equilibrium Point has not been covered by the paper,
- computed values of the variances of variables B and K are quite low– this influences on high rate of convergence.
- Numerical experiments show that the convergence is fast regardless of the initial situation!

Thanks for your attention