

Ecology Based Decentralized Agent Management System

Maxim D. Peysakhov, Vincent A. Cicirello, and William C. Regli

Department of Computer Science, Drexel University,
Philadelphia PA 19104

Abstract. The problem of maintaining a desired number of mobile agents on a network is not trivial, especially if we want a completely decentralized solution. Decentralized control makes a system more robust and less susceptible to partial failures. The problem is exacerbated on wireless ad hoc networks where host mobility can result in significant changes in the network size and topology. In this paper we propose an ecology-inspired approach to the management of the number of agents. The approach associates agents with living organisms and tasks with food. Agents procreate or die based on the abundance of uncompleted tasks (food). We performed a series of experiments investigating properties of such systems and analyzed their stability under various conditions. We concluded that the ecology based metaphor can be successfully applied to the management of agent populations on wireless ad hoc networks.

1 Introduction

In a typical agent based system, a number of mobile agents cooperate to achieve a desired goal. The efficiency of the agent system in reaching the goal, and the completeness of the result depends on the number of agents in the system. Too few agents will not achieve the full potential of parallelism and will lead to decreased system efficiency. Too many agents can overburden the system with unnecessary overhead, and may also result in significant delays. The task of finding the optimal number of agents required to achieve the desired effect is difficult and problem-specific. In this paper, we propose an ecosystem-inspired approach to this problem. Similar to a real ecosystem, our solution exhibits properties of emergent stability, decentralized control, and resilience to possible disturbances. In our work, we propose to solve the technical problem of agent management using an ecological metaphor.

In Section 2 we describe the current state of research in the fields of simulated ecosystems and multi-agent control and stability. Section 3 introduces the problem of managing the number of agents populating a physical network and also explains a proposed solution. Lastly, Section 4 demonstrates the initial experimental results and conclusions.

2 Related Work

2.1 Simulated Ecology

The majority of ecology-inspired systems are used to answer some question about real world ecosystems and its properties. For example, the RAM system has been used to study mosquito control [23]. There are two major approaches to simulating an ecosystem [6]. One is a species-based view of the system, where large classes of individuals interact in the simulation (i.e., modeling the dynamics of interaction of species rather than the interaction of individuals). Evolutionary game theory (e.g., [1] [18] [17]) and dynamical systems (e.g., [9] [15] [14]) are two approaches that often take the species-based view. The second approach is to simulate individuals and their interactions, a bottom up approach to construction of the ecological simulator.

We are most interested in *individual-based* simulations, since they are usually built with software agents. An example of an individual-based approach to ecosystems is a simulated habitat populated with synthetic organisms (agents) [19]. Often such systems are used to study the evolution (and co-evolution) of different species and testing their interactions and emergent behavior. Genetic Algorithms [8] and Genetic Programming [10] engines can be used in conjunction with synthetic ecosystems to allow species to evolve over time. Some of the most well known examples of synthetic ecosystems of this type are Evolve 1, 2 and 3 [4] [5] [21], “Artorg world” [3] and LAGER [19].

With this approach, global trends in the behavior of the system may *emerge* as a result of the low-level interactions of individual agents. The emergent behavior observed in an ecosystem may not be obvious given the individual behaviors of agents.

2.2 Agent System Stability

Service Replication. An increasing number of researchers are investigating the problems of reliability, robustness, and stability of multi-agent systems (*MAS*). Most approaches toward improving system robustness revolve around the replication of agents and/or services on the MAS network. This direction has been taken by [7], [12], [16] and several others. Existing approaches focus on the methodology of agent/service replication.

Probabilistic Models. Another approach is the application of probabilistic models to the prediction of agent system stability and robustness. This research assumes some uncertainty in agent behavior or the agent’s environment, and proposes mechanisms for estimating, evaluating and hopefully improving stability of agent systems. One of the first researchers to analyze probabilistic survivability in an MAS is Kraus in [11]. In that paper Kraus proposed a probabilistic model of MAS survivability based on two assumptions: (1) global state of the network is known at all times; and (2) the probabilities of host or connection failure are known. An alternative approach was proposed in [20, 2], where agents reason about the state of the network and security (insecurity) of their actions.

3 Problem Formulation

3.1 Motivation

In a typical dynamic ad hoc network there is limited, variable bandwidth between hosts, and the memory and CPU on each host is constrained. Given this dynamic and resource constrained environment, it is impractical to prescribe any pre-computed solution.

The solution we propose for such networks is to create a system that can control the number of agents dynamically, adapting to the ever-changing environment. In order to work in the context of an agent based system, a control system should be *distributed* and *decentralized*. By distributed, we mean that the system should be able to use the underlying network to parallelize problem solving on multiple hosts. By decentralized, we mean that the system should avoid reliance on a single node, and should allow each agent to act independently. The emergent behavior resulting from the individual localized control decisions ideally will yield an optimal, or near-optimal, solution at the global level.

3.2 Approach

Large ecosystems usually have several attractive qualities (such as dynamic decentralized control, self regulation, no single point of failure, robustness, and stability) that we require for our system. We propose a solution to the problem of determining the number of agents appropriate for a task at hand that is inspired by large ecosystems:

1. Each task in our system is associated with food.
2. Agents which successfully complete a task collect the associated food points.
3. Agents consume food points over time to sustain their existence.
4. Agents that exhaust their supply of food die.
5. An abundance of food can cause a new agent to spawn.

By this analogy, tasks can be thought of as plant life growing at some rate. Agents are associated with herbivore animals that perform tasks, therefore eating all the food provided by successfully completing a task. Upon completion of a task, an agent is forced to migrate to look for more food (tasks to complete). As time passes, agents consume food according to a predefined consumption function, analogous to a metabolic rate of an animal. Agents unable to find enough food (tasks) to sustain their existence over time will exhaust their food resources and will be terminated. Large amounts of food collected by a single agent or accumulated in a single location can force a new agent to spawn at this location. Agents procreate by division similar to a cell mitosis. However, this approach makes it impossible for the system to recover from a state with no agents. Therefore, we also allow tasks the ability to spawn a servicing agent whenever a certain threshold of accumulated food supply is reached. This control metaphor allows the system to dynamically adjust to the environment, while avoiding centralized control.

3.3 Formal Model

The set H denotes the set of producers h where $h \in H$, with the production rate defined by a function $F_h(t)$ for each individual producer h . The set A defines the set of consumers a ($a \in A$), and each consumer has a predefined consumption function $f_a(t)$. The dynamic system of H producers and A consumers is considered to be in an equilibrium state over some period of time from t_1 to t_2 , if and only if the amount of food produced during that period of time is equal to the amount of food consumed during that same period of time. This relationship can be expressed as:

$$\sum_{h \in H} \int_{t_1}^{t_2} F_h(t) dt = \sum_{a \in A} \int_{t_1}^{t_2} f_a(t) dt$$

At the simplest level, these principles can be modeled by a dynamic system of homogeneous producers and homogeneous consumers with constant production and consumption rates, c and d respectively. The equations below define the equilibrium state for this simple example:

$$\begin{aligned} \sum_{h \in H} \int_{t_1}^{t_2} F_h(t) dt &= \sum_{a \in A} \int_{t_1}^{t_2} f_a(t) dt \\ \sum_{h \in H} \int_{t_1}^{t_2} c dt &= \sum_{a \in A} \int_{t_1}^{t_2} d dt \\ |H| \times c \times (t_2 - t_1) &= |A| \times d \times (t_2 - t_1) \\ |A| &= |H| \frac{c}{d} \end{aligned}$$

This is essentially a species-based analysis of our individual-based ecological control system.

4 Experimental Results

4.1 System Setup

In order to confirm our conclusions we implemented a series of experiments using a discrete event simulation. The control flow of an ecology based agent is shown in Figure 1(a). According to this control flow diagram, an agent first decreases its internal food bank by $f_a(t)$ for each second that elapsed since the last decrement. Then, the agent completes the task and collects all food points associated with that task. Based on its current food resources, the agent may decide to die or to reproduce. Lastly, the agent migrates to another random host looking for food. We experimented with different ways for an agent to decide when to reproduce. We chose a *fuzzy threshold* method. Given the threshold value r , the probability of an agent reproducing is 0 if the amount of food is less than $r - \frac{r}{2}$. The probability of an agent reproducing is 1 if the

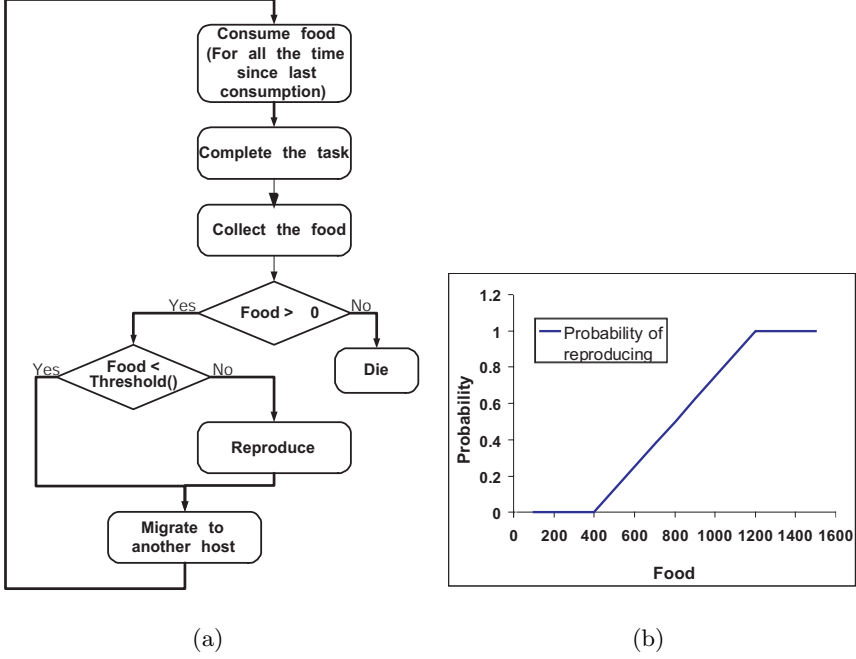


Fig. 1. Agents life cycle (a) and probability of reproduction based on the food level (b)

food level exceeds $r + \frac{r}{2}$. And the probability of reproducing grows linearly between these two points. A plot of the probability of reproducing is shown in Figure 1(b). The threshold needs to be fuzzy to avoid undesirable oscillations in the system.

If a small number of agents is desired on the network, it is possible for the system to go into an *extinction mode* — state with no agents on the network. In order to recover from this situation, we enable hosts to spawn new agents. The same fuzzy threshold rules apply to hosts as to agents. All of the experiments were performed on the completely connected network of statically placed hosts. All hosts grow food at the rate 1 unit per iteration. All experiments start with a single agent with initial food bank of 500 units. The reproduction threshold r was set to 800 resulting in 800 ± 400 range for hosts and agents. Additional experiments were performed using the real agent system EMAA [13] over a wired local area network.

4.2 System Behavior over Time

In this section, we investigate the changes in the number of agents over time. The consumption rate is set to 5 food units per iteration for all agents. Each experiment consists of 15 trials. A single trial consists of initializing the system and running it for 90,000 iterations. The number of agents is recorded every 10 iterations. Data is averaged across all trials to obtain the plots.

Constant Number of Hosts. Experiments were repeated for graphs of 35, 23, 15 and 8 hosts on the Figure 2 (top to bottom). Horizontal bold lines represent the targeted number of agents 7.0, 4.6, 3.0 and 1.6 respectively and the actual number of agents on the system is plotted by the thinner lines. It is easy to see that in all of the experiments, the actual number of agents oscillates close to the target value, however oscillations are somewhat higher for the network of 8 hosts. Figure 3 demonstrates the actual distribution of the numbers of agents during the experiments for 8 (a), 15 (b), 23 (c) and 35 (d) hosts with a normal distribution curve fitted to the data. Distribution is close to normal for all experiments but the one with 8 hosts. Such system behavior can be explained by the fact that the system with the small number of agents is prone to extinction of the population. Whenever the system recovers, it usually overshoots the targeted number of agents and oscillates for a while. These oscillations are repeated every time the system goes into extinction mode. More detailed analysis of this phenomenon is given in Section 4.3.

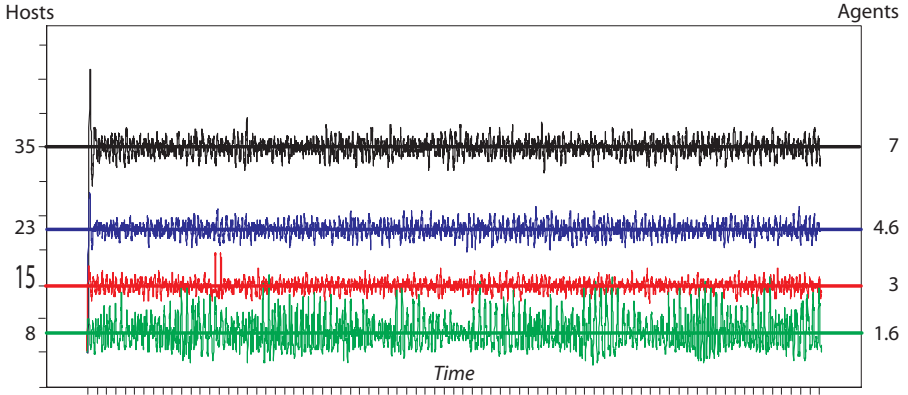


Fig. 2. System behavior over time

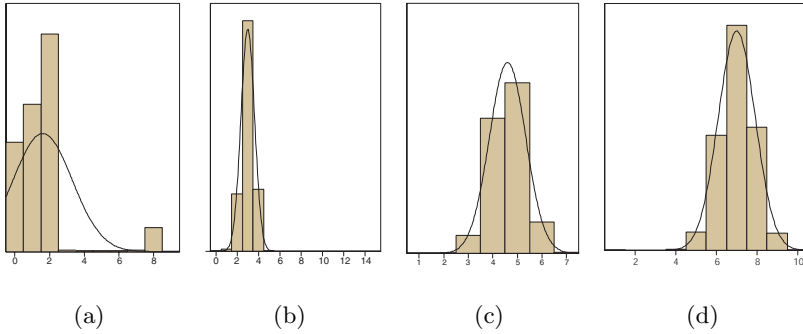


Fig. 3. Distribution of the number of agents for the network sizes of 8 (a), 15 (b), 23 (c) and 35 (d) hosts

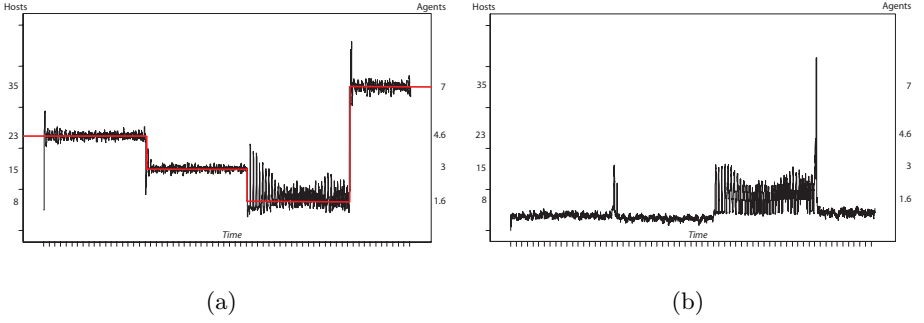


Fig. 4. System behavior over time (a) and Standard deviation (b)

Changing Number of Hosts. During this test we observed the system’s ability to react to rapid unplanned changes in the number of hosts. Experiment was setup identically to the one described in Section 4.2 except that the number of hosts was changed every 30,000 iterations without reinitializing the system. The number of hosts was changed from 23 to 15 to 8 and back to 35 hosts. We feel that such drastic changes in the number of hosts approximate the process of islanding and merging in wireless mobile networks of lightweight devices carried on foot by police or military units. Whenever the hosts were shut down all of the agents on these hosts and agents traveling to these hosts were also shut down. Whenever brought back on line, hosts initially had no food or agents on them. That type of change introduces a high level of disturbance into the system. The number of agents over time is plotted in Figure 4(a). The bold red line represents the target number of agents at any given moment. The black thinner line shows the actual number of agents. One can see that the actual number of agents follows closely the target number in all segments of the plot except for the one that corresponds to 8 hosts.

The standard deviation of the number of agents is plotted in Figure 4(b). Standard deviation peaks when we change the number of agents on the network due to the highly disruptive nature for the agent community of shutting down (or starting up) several hosts. Also standard deviation is higher at the segment corresponding to 8 hosts. We believe that such high standard deviation is caused by temporary extinction of agents and the oscillations that occur during recovery from it.

4.3 Dependency Between the Number of Agents and the Number of Hosts

In this Section, a single trial consisted of 100,000 iterations of the simulator. The number of agents is recorded every iteration and averaged across the trial to obtain a single data point. Trials were repeated for networks of sizes 3 to 35 hosts (odd numbers of hosts only). Experiments are plotted in Figure 5(a) with consumption rates set to be 3 times, 5 times and 7 times the production rate from top to bottom. Although all 3 graphs appear to be linear, they are composed

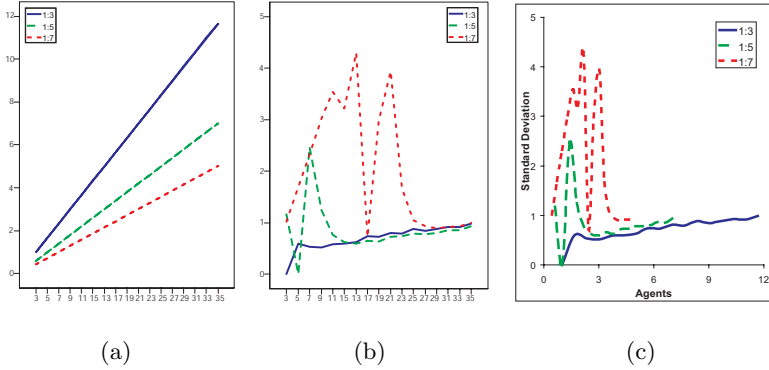


Fig. 5. Dependency between number of agents and number of hosts (a), between number of hosts and standard deviation (b), number of agents and standard deviation (c)

of 16 independently obtained data points. The experiment confirms that the system does what it is designed to do, namely maintain the given average ratio of hosts to agents, despite dynamic changes in the number of hosts. Figures 5(b) and 5(c) show the standard deviation of the number of agents in terms of the number of hosts and the number of agents respectively for all 3 experiments. Styles and colors of the plots correspond to the ones in figure 5(a). Although each is unique, the overall shape of the plots is similar. After the initial hump associated with the extinction mode and recovery from it, the plots level off in the area of 3 – 4 agents and then increase slightly. The linear increase can be explained by the linear increase in the number of agents. The only disturbance to that scheme is the point with target value of exactly 1 agent. For such systems it is possible to sustain a single agent for the duration of the whole experiment without ever going into the extinction mode resulting in *no* variance in the data.

4.4 Dependency Between the Number of Agents and the Link Quality

This set of experiments was set up exactly as the one described in Section 4.3 except that the changing parameter was link quality. A link of 100% quality implies that no artificial delay is introduced and migration only takes one iteration. Link of 0% quality means that maximum possible delay is introduced and migration takes 16 iterations (in simulator time). Figure 6(a) shows the target number of agents, actual number of agents and standard deviation of the number of agents changing based on the link quality. Although the actual number of agents slightly increases with decrease in link quality, it remains within 10% of the target value. Standard deviation however increases significantly as the speed of communication decreases. Some improvement of standard deviation at extremely low speeds can be explained by *consistently poor* performance of the system. Figure 6 (b) and (c) show the actual distribution of the number of agents for 10% and 90% respectively. The distribution for the higher link speeds

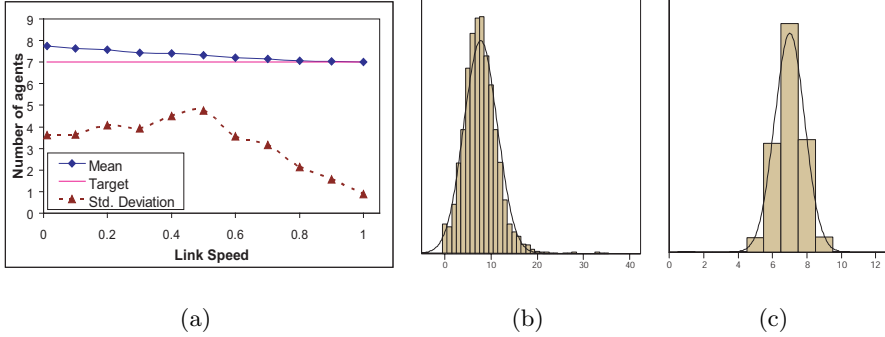


Fig. 6. Dependency between number of agents and standard deviation from link speed (a), and distributions for 10% (b) and 90 % (c) link speed

is more compact and closer to normal. Such behavior of the system can be explained by the fact that at the lower values, agents cannot move from one host to another fast enough to collect enough food to sustain their existence. This causes extinction of agents and forces the system to re-stabilize after it recovers from the state with no agents.

5 Future Work and Conclusions

5.1 Future Work

In the future we are planning several extensions to this work.

1. We are planning more extensive set of live experiments utilizing the Secure Wireless Agent Testbed (SWAT) [22].
2. We would also like to create a more detailed mathematical model of such systems to be able to predict and control the emergent behavior of an agent system. This model should be used for parameter fine tuning, something that was done manually during current experiments.
3. We are also planning to introduce an on-line system for tuning such parameters as consumption and production rates, thresholds and fuzzy intervals, etc. Some of the techniques we are planning to try include machine learning, swarm based techniques and genetic algorithms.
4. It would be interesting to expand the model from plant — herbivore system to plant — herbivore — carnivore. That extension will allow us to create more complicated food chains resulting in more elaborate control over the populations of different types of agents.

All of these techniques promise to improve on the current research and provide a more stable decentralized ways to control the number of agents on a wireless ad hoc network.

5.2 Conclusions

This paper developed an ecology-based model for managing the number of agents on ad hoc wireless networks. We have discovered that an ecosystem based model can provide decentralized distributed robust control of agents in dynamic and uncertain network environments. Our approach involves a novel exploitation of properties of ad hoc networks, enabling mobile agents to automatically adapt to changes that affect their communication and migration. The capability to dynamically adjust to the state of their network provides new possibilities for stable MAS.

References

1. J. M. Alexander. Evolutionary game theory. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, Summer 2003. <http://plato.stanford.edu/archives/sum2003/entries/game-evolutionary/>.
2. Donovan Artz, Max Peysakhov, and William C. Regli. Network meta-reasoning for information assurance in mobile agent systems. In *Eighteenth International Joint Conference on Artificial Intelligence*, pages 1455–57, Aug 2003.
3. A. Assad and N. Packard. Emergent colonization in an artificial ecology. In F. Varela and P. Bourguine, editors, *Toward A Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life.*, pages 143–152. MIT Press, 1992.
4. M. Conrad and H.H. Pattee. Evolution experiments with an artificial ecosystem. *J. Theoret. Biol.*, 28:393–409, 1970.
5. M. Conrad and M. Strizich. EVOLVE II: A computer model of an evolving ecosystem. *BioSystems*, 17:245–258, 1985.
6. Gary William Flake. *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*. MIT Press, July 1998.
7. Felix C. Gartner. Fundamentals of fault-tolerant distributed computing in asynchronous environments. *ACM Computing Surveys*, 31(1):1–26, 1999.
8. D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Pub Co, December 1989.
9. S. Goldenstein, E. Large, and D. Metaxas. Non-linear dynamical system approach to behavior modeling. *The Visual Computer*, 15:349–364, 1999.
10. J. R. Koza, F. H. Bennett III, F. H. Bennett, D. Andre, and M. A. Keane. *Genetic Programming III: Automatic Programming and Automatic Circuit Synthesis*. Morgan Kaufmann Publishers, 1999.
11. S. Kraus, V.S. Subrahmanian, and N. Cihan Tacs. Probabilistically survivable mass. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2003)*, pages 789–795, 2003.
12. Sanjeev Kumar, Philip R. Cohen, and Hector J. Levesque. The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000)*, pages 159–166, July 2000.
13. R.P. Lentini, G. P. Rao, J. N. Thies, and J. Kay. Emaa: An extendable mobile agent architecture. In *AAAI Workshop on Software Tools for Developing Agents*, July 1998.

14. K. Lerman and A. Galstyan. A general methodology for mathematical analysis of multi-agent systems. Technical Report ISI-TR-529, USC Information Sciences, 2002.
15. K. Lerman and A. Galstyan. Macroscopic analysis of adaptive task allocation in robots. Submitted to IROS-03, 2003.
16. O. Marin, P. Sens, J. Briot, and Z. Guessoum. Towards adaptive fault tolerance for distributed multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 737 – 744, 2002.
17. J. Maynard-Smith. *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, 1982.
18. J. Maynard-Smith and G. Price. The logic of animal conflict. *Nature*, 146:15–18, 1973.
19. R. L. Olson and A. A. Sequeira. An emergent computational approach to the study of ecosystem dynamics. *Ecological Modeling*, 79:95–120, 1995.
20. M. Peysakhov, D. Artz, E. Sultanik, and W. C. Regli. Network awareness for mobile agents on ad hoc networks. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-2004)*, July 2004.
21. M. Rizki and M. Conrad. EVOLVE III: A discrete events model of an evolutionary ecosystem. *BioSystems*, 18:121–133, 1985.
22. Evan Sultanik, Donovan Artz, Gustave Anderson, Moshe Kam, William Regli, Max Peysakhov, Jonathan Sevy, Nadya Belov, Nicholas Morizio, and Andrew Mroczkowski. Secure mobile agents on ad hoc wireless networks. In *The Fifteenth Innovative Applications of Artificial Intelligence Conference*, pages 129–36. American Association for Artificial Intelligence, Aug 2003.
23. Taylor, E. Charles, Turner, Scott, and Seth R. Goldman. Ram: Artificial life for the exploration of complex biological systems. In C.G. Langton, editor, *Artificial Life: SFI Studies in the Sciences of Complexity.*, pages 275–295. Addison-Wesley, Redwood City, CA, 1989.