G-Networks and the Modeling of Adversarial Agents

Yu Wang

Dept. of Electrical & Electronic Engineering
Imperial College London, London, SW7 2BT, UK
yu.wang3@imperial.ac.uk

Abstract. As a result of the structure and content transformation of an evolving society, many large scale autonomous systems emerged in diverse areas such as biology, ecology or finance. Inspired by the desire to better understand and make the best out of these systems, we propose an approach which builds stochastic mathematical models, in particular G-networks models, that allow the efficient representation of systems of agents and offer the possibility to analyze their behavior using mathematics. This approach is capable of modelling the system at different abstraction levels, both in terms of the number of agents and the size of the geographical location. We demonstrate our approach with some urban military planning scenarios and the results suggest that this approach has tackled the problem in modelling autonomous systems at low computational cost. Apart from offering the numerical estimates of the outcome, the approach helps us identify the characteristics that impact the system most and allows us to compare alternative strategies.

Keywords: mathematical modelling, G-Networks, military strategy and planning, multi-agent systems.

1 Introduction

As a society evolves, its structure and content transform accordingly to reflect and address its needs. As a result, more and more large scale autonomous systems occur in various forms in the surrounding world, from diverse areas of study such as biology, ecology, finance or transportation. Large scale systems have been traditionally characterized by a large number of variables, nonlinearities and uncertainties. As an example taken from biology, a human body, where organs, containing billions of cells, perform different functions that contribute towards the operating of the body can be seen as a large scale system. Inspired by the desire to better understand and utilize the environment, we study such systems and hope to gain insights, predict the future and control them partially if not fully.

There have been many attempts to model large scale systems, such as building differential equations or with simulations [1-5]. However the sheer complexity and diversity of large scale systems make them difficult to be described and modelled, and it is even more difficult to provide numerical predictions of the underlying processes of such systems. To tackle these problems, we propose to use a stochastic approach, in particular G-networks [6-10], to model the individuals of the same nature collectively. In doing so, the computational complexity is greatly reduced. Another innovative aspect of our approach is that it is able to model systems with multiple geographical

S. Kollias et al. (Eds.): ICANN 2006, Part I, LNCS 4131, pp. 330 – 339, 2006.

locations at different levels of abstraction. With the approach, we aim to provide insights into systems in terms of their performance and behaviours, to identify the parameters which strongly influence them, and to evaluate how well an individual's task can be achieved and, therefore compare the effects of alternative strategies.

As presented in [13-17], our approach has many application areas, such as military planning, systems biology and computer networking. In this paper, we use strategic military planning in urban scenarios as an example to demonstrate our approach. We describe the systems of interest, the mathematical model, and the chosen scenarios. We illustrate two methods that we use in dealing with the situations where complete or incomplete world knowledge is available. To validate our model, we compare its results with those obtained from a simulator [11, 12] that was built in our group. We will briefly discuss the differences between these two approaches, and analyze the discrepancy between the results. The paper concludes with a discussion of potential extensions to the model.

2 System Descriptions

As an example of a potential application, we consider a closed system containing N distinct geographic locations and a set of C agent classes. Locations may have obstacles that an agent cannot physically enter, such as stretches of water, trees, buildings and so on. Obstacles depend of course on the physical objects that the agents represent (e.g. land and air vehicles will have different kinds of obstacles). At the moment, we do not take the local minimum problem into consideration and we assume all obstacles in the terrain are convex in shape.

In these systems, agents take actions to achieve their goals, which are reflected by their motions and behaviors. A goal, be it stationary or motion-oriented, attracts agents to move towards it. A stationary goal is a specific location, whereas a motion-oriented goal refers to the situation where an agent itself is the goal (target) of others. Agents either protect or destroy their motion-oriented goals. To achieve this, agents might need to cooperate or compete with others in the system. The difference in nature of the goals results in agents belonging to different adversary teams. Teams in the same group collaborate with each other to achieve their goals, while those in adversarial groups would exhibits competing behaviors to prevent others accomplishing their goals.

Motion in the system is a result of forces exercised by the goal and agents. There are three types of forces in our system: attractive, repulsive and long-range attractive and short-range repulsive. A straightforward example of an attractive force would be the force that a stationary goal (i.e. the destination) applies on an agent. A repulsive force can be interpreted as the tension of moving away. For example, if agent D's aim is to destroy agent B, then agent B applies an attractive force on D. On the other hand, agent D exercises a repulsive force on B. A long-range attractive and short-range repulsive force makes a group of agents stay together and keeps their distance at the same time. Thus if the agents are too close, repulsive forces are applied, otherwise, attractive forces are applied.

3 The Mathematical Model

Let $i = g(t, c, k)$ denote the location at time t of agent k belonging to team c. The location i may be a single variable or a vector, depending on the geographic representation that is being used, e.g. the (x, y, z) coordinates of a location on a map, where z may represent elevation when this is relevant. Thus in a two-dimensional coordinate space, a location will be denoted by $i=(x(i),y(i))$ and a neighboring location will be denoted by some $j=i+d$ where $d \in \{(0,0),(\pm1,0),(0,\pm1),(\pm1,\pm1)\}$. It is assumed that each agent has an initial location denoted by $S(c, k)$, and it may have (though this is not necessary) a final destination $D(c, k)$. We also assume that agents may die as a result of adversarial effects, or for other reasons, in which case they are relocated to "heaven" denoted by H. For the purpose of this model, we assume that there is just one heaven for everyone.

The mathematical model we develop aims at being able to compute, over a large number of experiments, quantities such as the probability $q(i, c, k)$ that agent k of team c is in location i. Such probabilities will also be used to compute indirectly the average time it may take certain agents to reach specific locations. The approach we take is based on constructing an ergodic model, i.e. one which has a stationary probability distribution, so that:

$$q(i,c,k) = \lim_{t\to\infty} prob[g(t,c,k) = i] \tag{1}$$

3.1 Agent Parameters

In the model, the motion of an agent is a result of two parameters:

- Its speed or rate of movement, $r(i, c, k)$ which will depend on its location (reflecting the geographic characteristics of its location)
- The direction of this movement which will be denoted by the probability $p(i, j, c, k)$ that agent (c, k) moves from i to j.

In this study, we assume that locations i and j are adjacent, in which case $j = i + d$ where d is one of the nine cardinal directions from i, including $d=(0, 0)$ which means that the agent has not moved at all.

The direction of motion for any one of the agents is determined by:

- the objectives or final destinations of the agents, when applicable, as expressed by a force of attraction
- the interaction between agents, as expressed by forces of attraction and repulsion
- the physical obstacles in the terrain, or the specific terrain related difficulties which may discourage or motion in a particular direction

In addition to motion, interaction between the agents is exhibited by their ability to destroy the others. Each agent (c, k) has a set of enemies, $E(c, k)$, that it tries to destroy, a shooting range $R(c, k)$ within which it is able to destroy an adversary, and a firing rate $f(c, k)$. Of course, these parameters may be identical in certain cases for all agents belonging to the same class or adversary team. In our model, the concept of enemy need not be reciprocal, i.e. $(c,k)\in E(c',k')$ does not necessarily imply $(c',k')\in E(c,k)$.

3.2 Forces

We use the cumulative-force exercised on an agent to determine its motion probabilities *p(i, j, c, k),* which define the direction of motion. Let *Forces(c', k', c, k)* be the force exercised on agent *(c, k)* by agent *(c', k').* A positive coefficient implies that agent *(c, k)* is attracted to agent *(c', k')*, whereas a negative coefficient implies that agent *(c, k)* is repulsed by agent *(c', k').* The strength of an inter-agent force varies with the distance of the two agents. The destination of an agent, *D(c, k)*, if one exists, also exercises an attractive force *G(i, d, c, k)*, which may also vary across the terrain.

The net force *v(i, d, c, k)* exerted on agent *(c, k)* at location *i* in direction *d* is computed as follows, where the function $dist(i,j)>0$ represents the way that the force component changes with the distance between agents. The set *L(i, d)* represents all locations at direction *d* from *i* in the terrain and *d* is defined as $d=\{direction(j-i)\}$.

$$v(i,d,c,k)=\sum_{all(c',k')}\sum_{j\in L(i,d)}\frac{Forces(c',k',c,k)q(j,c',k')}{dist(i,j)}+G(i,d,c,k) \tag{2}$$

Let *O(i)* be the set of neighbors of *i* which do not contain obstacles. In the process of obtaining motion probabilities, we introduce an adjusting factor to assist renormalizing *v(i, d, c, k)* to positive values. The adjusting factor is set in a way that it has a trivial impact on the accuracy of the motion probabilities. Let *V(i, c, k)* be the sum (numerical) of forces exerted on an agent from all the directions. It can be represented as:

$$V(i,c,k)=\sum_{d\in O(i)}|v(i,d,c,k)| \tag{3}$$

The motion probability, *p(i, j, c, k)*, is defined in equation (4), which also allows us to take *d= (0,0)*, i.e. the probability of staying in the current location. This of course raises the issue of certain agents getting "stuck" in a place from which they will not move away until conditions related to other agents have changed.

$$p(i,j,c,k)=\begin{cases}\dfrac{v(i,d,c,k)}{V(i,c,k)+factor} & if\ \ d\notin O(i)\\ 0 & if\ \ d\in O(i), d\neq(0,0)\\ \dfrac{factor}{V(i,c,k)+factor} & if\ \ d=(0,0)\end{cases} \tag{4}$$

3.3 Conditions Under Which the Simulation Game Ends

We consider that the simulation game ends when some subset of agents, for instance any one of the agents of some class *c*, reaches some pre-selected set of positions, which may include their destinations. Alternatively, the game may also end when some agents reach heaven (i.e. when they are killed). To formalize the terminating conditions, we define a final state set *F(c, k)* for the agent *(c, k)* as a subset:

$$F(c,k) \subseteq \{locations \quad j\} \bigcup \{H\} \tag{5}$$

It is also possible that $F(c,k) = \phi$, in which case this means that this particular agent does not influence the time at which the simulation ends. The terminating condition *F* is then simply:

$$F = \cup_{all(c,k)} F(c,k) \tag{6}$$

and the interpretation we give to it is that:

$$Simulation \quad Ends \quad At \quad t \Leftrightarrow if \quad \exists (c,k),\, g(c,t,k) \in F(c,k),\, forF \ (c,k) \neq \phi \tag{7}$$

When a game attains its terminating condition, after some random time of average value 1 (this value is chosen for the purpose of normalization), each agent *(c, k)* (including the agents that made it to heaven), will move instantaneously to its initial location *S(c, k)*, and the game will start again at rate *Rstart*. For the purpose of this mathematical model, this cycle repeats itself indefinitely. This allows us to compute ensemble averages that are of interest. We assume that in the class of games, either some agent of some designated class(es) reach their destination, or all agents of designated class(es) reach heaven. Thus, we exclude situations where all agents become blocked and cannot move any further, or enter cycles of behavior which exclude the agents' demise, or that impair their ability to attain their destinations.

The terminating probability *T* is defined as the stationary probability that the model is in the terminating state:

$$T = \lim_{t \to \infty} prob \ [\vee_{all(c,k)} g(t,c,k) \in F(c,k)] \tag{8}$$

Similarly we define:

$$T(c,k) = \lim_{t \to \infty} prob[\vee_{all(c,k) \neq (c,k)} g(t,c',k') \in F(c',k')] \tag{9}$$

Thus *T(c, k)* is the stationary probability that the game is in the terminating state, given that agent *(c, k)* is already in a final state. Suppose now that we wish to compute the expected time τ it will take some specific agent *(c, k)* to reach some specific location γ. In that case we would set $F(c,k) = \{\gamma\}$, and the terminating probability, *T*, becomes $q(\gamma,c,k)$. We then have:

$$\tau = \frac{1}{q(\gamma,c,k)} \tag{10}$$

3.4 Model Equations

The equations that describe the overall long-term behavior of the system are obtained heuristically based on the equations satisfied by the stationary probability distributions of G-networks [7]. We heuristically, but plausibly, choose to relate the *q(i, c, k)* to each other and to the agents' parameters via the following equations:

$$q(i,c,k)=\begin{cases}\dfrac{Rstart+Neighbors(i,c,k)}{r(i,c,k)(1-p(i,i,c,k))+Killed(i,c,k)+MutaRate(i,c,k)} & ,if\ \ i\in S(c,k)\\ \dfrac{Neighbors(i,c,k)+MutaIn(i,c,k)}{r(i,c,k)(1-p(i,i,c,k))} & ,if\ \ i\in D(c,k)\\ \dfrac{\sum_{(c,k)\in E(c',k')}\sum_{i,j\neq H}q(i,c,k)Killed(i,c,k)}{Rstart} & ,if\ \ i=H\\ \dfrac{Neighbors(i,c,k)+MutaIn(i,c,k)}{r(i,c,k)(1-p(i,i,c,k)+Killed(i,c,k)+MutaRate(i,c,k)} & ,otherwise\end{cases} \tag{11}$$

$$Neighbors(i,c,k)=\sum_{d\notin O9i),d\neq(0,0)}q(i+d,c,k)r(i+d,c,k)p(i+d,i,c,k)$$

$$Killed\ (i,c,k)=\sum_{(c,k)\in E(c',k'),j}q(j,c',k')1[|\ j-i\ |\leq R(c',k')]f(c',k')$$

$$MutaIn(i,c,k)=\sum_{(c',k'),d\notin O(i),d\neq(0,0)}q(i,c',k')r(i+d,c',k')p(i,i+d,c',k')$$

In addition, we use the normalizing condition that the states that the sum of the probabilities that any given agent is at any one of the locations (including “heaven”) is one. Thus for any *(c, k)* we have:

$$\sum_i q(i,c,k)=1 \tag{12}$$

You might have noticed that the *Rstart* rate is not defined in the above equation. Our approach is versatile in the sense that it provides insights into various aspects that are of interest based on one set of system equations. Therefore, the condition under which the process repeats itself is defined accordingly. For example, with the same model, we can examine the impact that the initial locations have on the average time that agents take to complete their tasks, the average time of a specific agent achieving its goal or the average percentage of a team achieving its goal.

3.5 Scenarios and Obtained Results

We have experimented with scenarios where agents of the same class travel towards their goals in the system. Results show that the model converges quickly and the inter-agent forces have impacts on the agents’ performance depending on their strength. The detail of those scenarios and corresponding results can be found in [18].

After our initial success with a single agent class, we incorporate collaborating and competing behaviors into the model by introducing multiple agent classes, in other words, adversary teams. We demonstrate such models with a system containing three agents of different adversary teams: civilian (agent 1), terrorist (agent 2) and police (agent 3). The civilian’s aim is to reach its destination alive with the help of the police. The police fights against the terrorist so that it will not kill the civilian. The terrorist attacks anybody who prevents it from killing the civilian. Both the police and the terrorist are only capable of attacking their enemies within a certain range. The collaborating and competing behaviors are not necessarily symmetrical, as illustrated in this scenario. The scenario repeats itself either when agent 1 is dead or arrives at its goal. Therefore the *Rstart* Rate has the numerical value of *Max[q(H, c, k), q(D (c, k), c, k)].*

The detail of the scenario is as follows: in a 15×15 terrain, the agents' initial locations are (7,14) for the civilian, (5,2) for the terrorist and (13,2) for the police. The civilian has a stationary destination, location (14,14), whereas the police and the terrorist have motion-oriented goals. The nature of the terrorist attracts it towards the civilian and keeps it away from the police. The civilian travels towards its goal and avoids being killed by the terrorist. The police is attracted by the terrorist more than the civilian, simply because its task is to prevent the civilian from being killed.

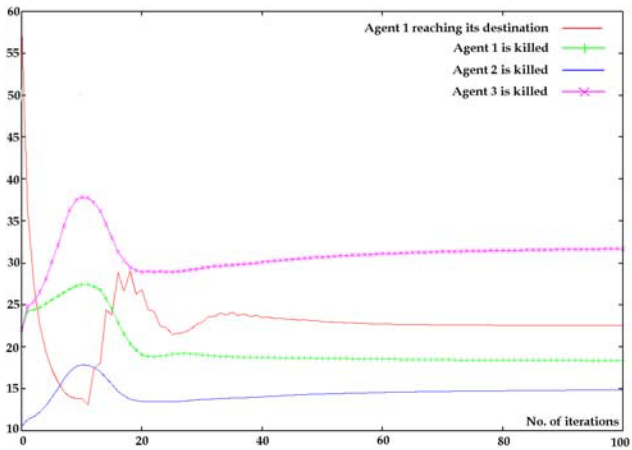

Fig. 1. Estimated time of events occurrence

Technically, the police and the terrorist are identically equipped except the police fires twice as frequently as the terrorist. Due to this nature, we expect the terrorist stands a higher chance of being killed by the police than killing the police. The result (See Fig 1) shows that, as a result of receiving help from the police, the civilian stands a higher chance of reaching the goal (55.5%) than being killed (44.5%). The reason that the police does not have a very big impact on the situation is that it takes times for the police to approach the terrorist. The result indicates that, on average, it takes 32 steps for the terrorist to kill the police and 15 steps for the police to kill the terrorist. This is inline with our predictions. As one might have noticed, for this scenario, the algorithm converges at around 50 iterations, which is shorter comparing with obtaining the average statistics from the simulator.

As mentioned before, the police and the terrorist have the same technical characteristics except their shooting rates. So if they have identical settings, it should take the same amount of effort to kill each other. We therefore assign them with the same shooting rate and see how that affects the outcome. Results confirm that, the estimated time of the police and the terrorist killing each other is the same.

In military planning, one often has to consider the trade-off between protection and agility. Weakening the protecting capacity of the police gives the terrorist more chances to launch attacks; however, it offers advantages such as agility and low consumption. Depending on the nature of the task, the police forces may have different focuses. With our approach, we can easily compare alternative strategies and select one that addresses our needs best.

Now we illustrate how mutation is modelling with an example of a system consisting of two agents. We restrain the two agents from interaction, so that mutation is the dominating factor of the outcome. The agents locate at (2,2) and (8,8) respectively. Agent 1 plans to go to (4,4) and agent 2 travels towards (14, 14). The two agents are identical apart from the fact that agent 1 has the mutating ability. With mutation rate *a*, agent 1 mutates and exhibits the same behaviour as agent 2. Thus after mutating, agent 1 will start pursuing agent 2's goal.

The following table (See Fig 2) shows how the estimated time of un-mutated agent 1, mutated agent 1 and agent 2 reaching their destinations. For example, with mutation rate 0.6, on average, 73% of the time agent 1 carries out an alternative task, which requires 6.35 unit time to complete, whereas 27% of the time, agent 1 preserves its nature and arrives its goal at 10.66 unit time. This feature is significant in task planning, where by altering parameters, one could "foresee" how fast multiple goals can be achieved with a reasonable overhead, if such need rises during the mission. In this case, if the team has to be split to achieve different tasks without too much overhead, mutation rate 0.4 can be a good choice.

Mutation-Rate	Non-Mutated	Mutated	A1 Goal(NM)	A2 Goal	A1 Goal(M)
0.1	0.819	0.18	2.645	6.8	50.53
0.2	0.6224	0.376	3.66	6.8	17.58
0.3	0.436	0.5624	5.47	6.8	8.68
0.4	0.33	0.663	7.5	6.8	6.62
0.5	0.296	0.703	9.05	6.8	6.47
0.6	0.2693	0.7352	10.66	6.8	6.35
0.7	0.23	0.7626	12.44	6.8	6.29
0.8	0.2143	0.7849	14.18	6.8	6.21
0.9	0.19556	0.8037	15	6.8	6.14
1	0.1812	0.8181	18.1	6.8	6.12

Fig. 2. Estimated time of an agent reaching its goal (with different mutating rate)

During the experiments, we have also discovered that our approach can be used to model systems at different levels of abstraction. For example, 1 agent in the mathematical model does not necessarily represent a single agent; instead, it can represent a certain number of agents, say 150. From another perspective, we can also estimate the outcome of a larger terrain by modelling a smaller terrain with similar natures/characteristics, as presented in [14]. The approach is also validated again a simulator that was developed in our group. Results [14] show that, despite the magnitude discrepancy, the mathematical model is inline with the statistics collected in the simulator.

So far, the features that our approach offers are desirable. However, it is assumed so far is that we can calculate the probability *q(i, c, k)* (and therefore the estimated time for events). This is not always the case for large scale autonomous systems which are known for their uncertainties and complexities. We have also proposed a method [17,19] to overcome situations where a complete knowledge of the system is not available. This method estimates the agents' motion probability using historical observation records.

4 Conclusions

In this paper, we have presented a stochastic approach based on G-Networks, which models large scale agent systems. The approach models systems containing collaboration, competition and mutation under the condition that complete information of the system is available. We first described the systems of interest, the mathematical model and demonstrated the approach with some scenarios of military planning in urban environments. Results show that our approach identifies the parameters that strongly influence the agents' performance and allows us to compare alternative strategies at low computational cost. We then proposed an extension to the model which deals with systems where complete information is not readily available.

We plan to incorporate behaviors such as reproduction into the model so that it can be applied in fields such as system biology. After the initial success of obtaining the motion probability via observation, we are investigating how to deduce agents' intention via similar means. This undoubtedly will reduce the dependence of model to the system's knowledge.

In reality, obstacles in urban environment have an impact on an agent's behavior. For example, a building might be an obstacle for a car but not a pedestrian. Therefore we plan to change the obstacles so that they have different impact on agents' behaviors and incorporate the wall following method mentioned in [18] to deal with the local-minima problem. In doing so, we are able to model more realistic urban scenarios. Theoretical wise, we aim to study the computational constraints related to resources or time-frame, as well as conduct an extensive exploration on modelling large scale agent systems at different abstraction levels.

References

1. Amin, K.A., Mikler, A. R: Dynamic Agent population in Agent-Based Distance Vector Routing, in Proceedings of the Second international workshop on Intelligent systems design and application. (2002) 195-200
2. Burmeister, B.: Models and methodology for agent-oriented analysis and design, in K. Fischer (Eds) Working Notes of the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems. (1996)
3. Cysneiros, L.M., Yu, E.: Requirements Engineering for Large-Scale Multi Agent Systems, Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications, Vol. 2603. (2003) 39-56
4. Huang, G., Abur, A., Tsai, W.K.: A Multi-level Graded-Precision Model of Large Scale Power Systems for Fast Parallel Computation, Mathematical computing modelling, Vol. 11. (1998) 325-330
5. Kinny, D., Georgeff, M., Rao, A.: A Methodology and modelling technique for systems for BDI agents, in W. van der Velde and J. Perram (Eds). Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World MAAMAW'96, Vol. 1038. (1996) 56-71
6. Gelenbe, E.: Random neural networks with positive and negative signals and product form solution, Neural Computation, Vol. 1(4). (1989) 502-510
7. Gelenbe, E.: G-networks with instantaneous customer movement, Journal of Applied Probability, Vol. 30 (3). (1993) 742-748

8. Gelenbe, E.: G-Networks with signals and batch removal, Probability in the Engineering and Informational Sciences, Vol. 7. (1993) 335-342
9. Fourneau, J. M., Gelenbe, E., Suros, R.: G-networks with multiple classes of positive and negative customers, Theoretical Computer Science, Vol. 155. (1996) 141-156
10. Gelenbe, E., Labed, A.: G-networks with multiple classes of signals and positive customers, European Journal of Operations Research, Vol. 108(2). (1998) 293-305
11. Gelenbe, E., Hussain, K., Kaptan, V.: Simulating the navigation and control of autonomous agents, in Proceedings of the 7th International Conference on Information Fusion. (2004) 183-189
12. Gelenbe, E., Hussain, K., Kaptan, V.: Enabling Simulation with Augmented Reality, in Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems. (2004) 290-310
13. Gelenbe, E., Kaptan, V., Wang, Y.: Simulation and Modelling of Adversarial Games, in the Proceedings of the 6th European GAME-ON Conference on Simulation and AI in Computer Games. (2005) 40-44
14. Gelenbe, E., Wang, Y.: A mathematical approach for mission planning and rehearsal, to appear in the Proceedings of SPIE Defence & Security, Orlando. (2006)
15. Gelenbe, E., Wang, Y.: A Trade-off between Agility and Resilience, in Proceedings of the 13th Turkish Symposium on Artificial Intelligence and Neural Networks. (2004) 209-217
16. Gelenbe, E., Kaptan, V., Wang, Y.: Biological Metaphors for Agent Behavior, in the Proceedings of the 19th International Symposium on Computer and Information Sciences, Lecture Notes in Computer Science, Vol. 3280. (2004) 667-675
17. Gelenbe, E., Wang, Y.: Modelling Large Scale Autonomous Systems, Accepted to Fusion 2006. (2006)
18. Wang, Y.: Numerical Modelling of Autonomous Agent Movement and Conflict, to appear in Computational Management Science. (2006)