CMS (2006) 3:207–223
DOI 10.1007/s10287-006-0016-x

ORIGINAL PAPER

Numerical modelling of autonomous agent movement and conflict

Y. Wang

Published online: 13 June 2006

Abstract The world that we live in is filled with large scale agent systems, from diverse fields such as biology, ecology or finance. Inspired by the desire to better understand and make the best out of these systems, we propose to build stochastic mathematical models, in particular G-networks models. With our approach, we aim to provide insights into systems in terms of their performance and behavior, to identify the parameters which strongly influence them, and to evaluate how well individual goals can be achieved. Through comparing the effects of alternatives, we hope to offer the users the possibility of choosing an option that address their requirements best. We have demonstrated our approach in the context of urban military planning and analyzed the obtained results. The results are validated against those obtained from a simulator (Gelenbe et al. in simulating the navigation and control of autonomous agents, pp 183–189, 2004a; in Enabling simulation with augmented reality, pp 290–310, 2004b) that was developed in our group and the observed discrepancies are discussed. The results suggest that the proposed approach has tackled one of the classical problems in modeling multi-agent systems and is able to predict the systems' performance at low computational cost. In addition to offering the numerical estimates of the outcome, these results help us identify which characteristics most impact the system. We conclude the paper with potential extensions of the model.

Keywords Mathematical modeling · G-Networks · Military strategy and planning · Multi-agent systems

This work was supported by a contract from General Dynamics UK Ltd. to Imperial College London under DIF DTC Project 6.8.

Y. Wang (✉)
Electrical and Electronic Engineering Department,
Imperial College London, London SW7 2BT, UK
e-mail: yu.wang3@imperial.ac.uk

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 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 in biology, a human body can be seen as a large scale system, where organs, containing billions of cells, perform different functions that contribute towards the operating of the body. Inspired by the desire to better understand and make the best out of the environment, we study such systems and hope to gain insights, predict the future and control it partially if not fully.

There have been many attempts to model large scale systems, such as building differential equations or with simulations (Amin and Mikler 2002; Burmeister 1996;Cysneiros and Yu 2003;Huang et al. 1998; Kinny 1996) . However the sheer complexity and diversity of large scale systems make them difficult to be described and modeled, and it is even more difficult to provide numerical predictions of the underlying processes of such systems. The complexity of a system usually increases along with the number of individuals within it. To tackle these problems, we propose a stochastic approach, in particular G-networks (Gelenbe 1989a,b, 1993a,b, Gelenbe and labed 1998;fourneau et al. 1996) to model the large scale systems mathematically.

Differing from the existing methods, our approach reduces the computational complexity by handling individuals of the same nature collectively. Another innovative aspect of the approach is that it is able to model systems with multiple geographical locations at different levels of abstraction. With our approach, we aim to provide insights into systems in terms of their performance and behaviors, 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.

To validate our model, we compare its results with those obtained from a simulator (Gelenbe et al. 2004a, b) that was built in our group. Designed with different objectives in mind, the simulator offers a visual representation of the underlying processes of a system whereas the mathematical model examines the system at a higher level of abstraction and provides an estimated numerical outcome of the system. Despite the differences, they can still be used to model the same scenarios. We will briefly discuss the differences between these two approaches, and analyze the discrepancy between the results.

Our approach has many application areas, such as military planning, systems biology and computer networking (Gelenbe and wang 2004; Gelenbe et al. 2004,2005).In this paper, we use strategic military planning in urban scenarios as an example to demonstrate our approach. We describe the systems that we are interested in modeling, the general mathematical model, and the chosen scenarios. We then compare the obtained results with the simulation statistics as a way to validate our model. The paper concludes with a discussion of potential extensions to the model.

2 System description

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 this stage, we do not take the local minimum problem into consideration and we assume all obstacles in the terrain are convex in shape.

In the 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 that, agents might need to cooperate or compete with others in the system. The difference in nature of the goals results 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.

Motions in the system is a result of the forces exercised by the goal and agents. There are three types of forces in the system: attractive forces, repulsive forces and attractive–repulsive forces. 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. An attractive–repulsive force makes a group of agents stay together and keeps a distance at the mean time. So if the agents are too close, repulsive forces are applied, otherwise, attractive forces are applied.

3 The mathematical model

3.1 Model description

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 $d \in \{(0,0), (\pm 1, 0), (0, \pm 1), (\pm 1, \pm 1)\}$. 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 destinations. The approach we take is based on constructing an ergodic model, i.e. one which has a stationary probability distribution, so that:

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

3.2 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), and
- 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 eight 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.

3.3 Forces

We use the net-force exercised on an agent to determine the motion probabilities $p(i,j,c,k)$, which defines the direction of motion. Let $\text{Forces}(c',k',c,k)$ 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 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 $\text{dist}(i,j) \geq 0$ 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 $d = \text{direction}(j - i)$.

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

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 re-normalizing $v(i, d, c, k)$ 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 as the following:

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

Note that the expression (4) 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.

In addition to motion, interaction between the agents is exhibited by their ability of destroying 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. Note that in our model, the concept of enemy need not be reciprocal, i.e. $(c, k) \in E(c', k') \nRightarrow (c', k') \in E(c, k)$.

3.4 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, reach some pre-selected set of positions, which can 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:

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

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

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

and the interpretation we give to it is that:

$$\textit{At Time t Simulation Ends} \Leftrightarrow \text{if } \exists\ (c,k), g(t,c,k) \in F(c,k), \text{for } F(c,k) \neq \emptyset. \tag{7}$$

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. 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 of interest, 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:

$$T = \lim_{t \longrightarrow \infty} \text{Prob}[\vee_{\text{all } (c,k)} g(t,c,k) \in F(c,k)]. \tag{8}$$

Similarly we define:

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

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

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

3.5 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 (Gelenbe 1989a,1993a). We heuristically, but plausibly, choose to relate the $q(i, c, k)$ to each other and to the agents' parameters via the following equations:

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

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

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

In addition, we use the normalizing condition to state 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 *Restart 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.

3.6 Modeling social potential field

There have been some previous work (Liu et al. 2003; Yun and Tan 1997) on potential field based motion planning and approaches to avoid local minima. In the work of Yun and Tan (1997), the authors proposed to use a wall-following approach when a robot falls into a local minimum. With the approach, in the circumstance of being tracked in a local minima, a robot switches from the potential field guide control mode to the wall-following control mode, in which it follows the contour of the wall. This approach does not always work at the

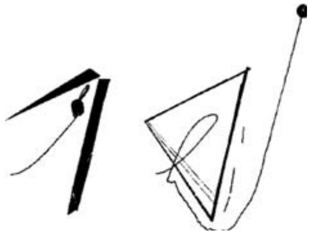

Fig. 1 An efficient approach of avoiding local minima (taken from Liu et al. (2000))

first attempt, because a robot may fall into the same local minima again. It is not very efficient in the sense that a robot may not travel towards its goal while following the contour of a wall.

The work of Liu et al. (2003) proposed a conditional-weighted potential field based control algorithm for target tracking. The motion of the robots is determined by a vector sum of the local forces imposed on them as well as collaboration factors. In addition to the collaboration behaviors, our approach is able to model competition as well as multiple geographical locations. In the work of Liu et al. (2000), the author proposed a simple yet efficient method of escaping the local minima by adding subgoals. With the new set of subgoals, the robots are able to avoid concave shaped obstacles (See Figure 1). We plan to integrate this method into our approach in a later stage so that obstacles of all shapes can be included into the system.

4 Scenarios and obtained results

To validate our approach, we start with modeling scenarios that have been studied in our group with the simulator (Gelenbe et al. 2004a,b). By comparing the results of the two approaches, we can assess the correctness of the model and identify the flaw if there is any.

4.1 Systems with a single agent class

As an initial step to examine our approach, we experiment with systems that have a single agent class. Agents travel towards their goals in the system and this process repeats itself when the goals are achieved. The system equations of such a system can be derived from the general model by removing the Killed(i, c, k) term and the heaven location. In this case, the *Restart Rate* takes the numerical value of $q(D(c, k), c, k)$.

We first experiment with a 100×100 terrain with no obstacles and only 1 agent. The agent has to travel diagonally across the terrain to achieve its goal. Without any disturbance, we estimate that it will take the agent a little over 100 steps to complete the task. The result (See Figure 2) is inline with our prediction.

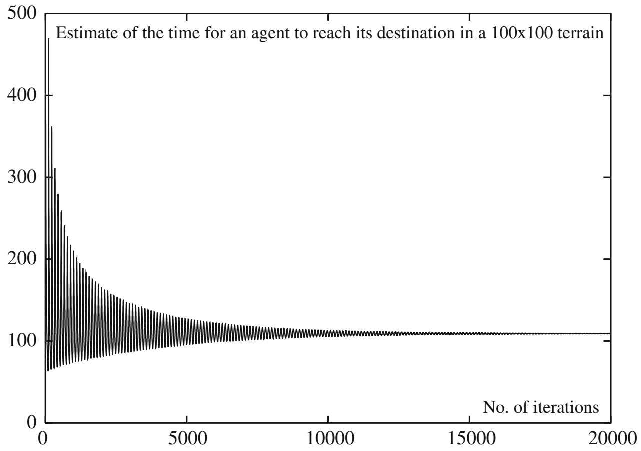

Fig. 2 A single agent in a 100 × 100 terrain

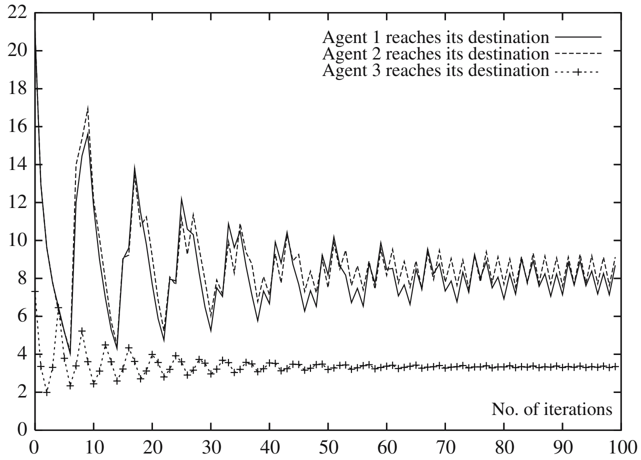

Fig. 3 Multiple agents of the same class in a 8 × 8 terrain

We explore the effect of inter-agent forces by introducing multiple agents of the same class into the system. A 8 × 8 terrain is used so that the effects can be easily identified. In this scenario, agent 1 travels from location $(2, 2)$ to $(7, 7)$, agent 2 goes from location $(7, 7)$ to $(2, 2)$, and agent 3's task is to travel from location $(5, 5)$ to $(3, 3)$. The inter-agent forces modeled here are long range attractive and short range repulsive.

Without any disturbance, it should take the agents around $5, 5$, and 2 steps to reach their goals, respectively. We predict agent 3 to be least affected since it will reach its goal before others approaching it. Agent 1 and 2 have to pass by each other during their journeys. So they attract each other before encountering, repulse when they are close-by and then attract each other again. Therefore we predict they will spend more time in the area that they encounter. As predicted, the result (See Figure 3) suggests that the inter-agent forces have trivial effect on agent 3. Comparing with the scenario where inter-agent forces do not exist, it takes agent 1 and 2 longer to reach their goal. From the results, we can draw the conclusion that the approach provides valid results and the algorithm converges in the case where systems contain a single class of agents.

4.2 Systems with multiple agent classes

After the initial success, 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, and they are the civilian (agent 1), the terrorist (agent 2) and the 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 set of system equations is the same as stated in (11). The process of such a scenario repeats itself either when agent 1 is dead or arrives at its goal. Therefore the *Restart Rate* has the numerical value of $\mathrm{Max}[q(H,c,k), q(D(c,k),c,k)]$.

We experiment with a scenario where the terrain size is 15×15. 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 at the mean time. The police is attracted by the terrorist more than the civilian, simply because its task is to prevent the civilian from being killed.

Technically, the police and the terrorist are identically equipped except the police fires twice as frequent 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 Figure 4) 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 help does not have a very big impact on the situation because 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 the police to kill the terrorist. This is inline with the prediction that was made earlier. As you 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 expect the shooting rate. 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. The overlapping curves (See Figure 5) indicate that our hypothesis is correct. Weak-

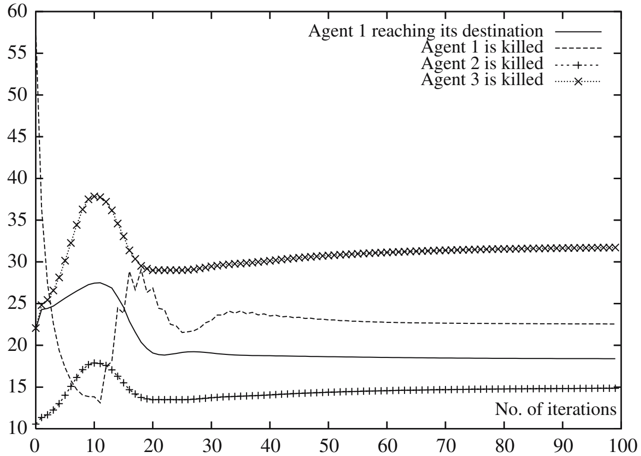

Fig. 4 A system with three adversary teams taking place in a 15 × 15 terrain

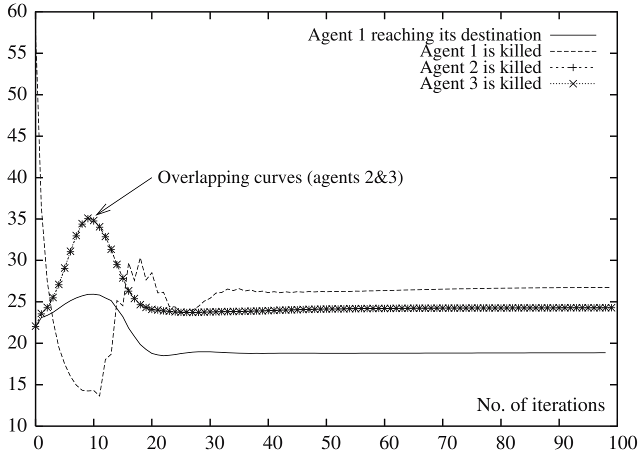

Fig. 5 Same system with a different setting

ening the protecting capacity of the police gives the terrorist more chances to launch attacks. After the change, it takes the civilian around 26.5 steps to reach its goals comparing with 22.5 steps in the previous scenario. The estimated time of the civilian being killed remains at 18 steps, since the attacking capacity of the terrorist does not change.

5 Validating the mathematical model

The simulator and the mathematical model are designed with different objectives in mind. The former simulates scenarios and offers visual representation of a process as it happens, whereas the latter uses stochastic mathematical model

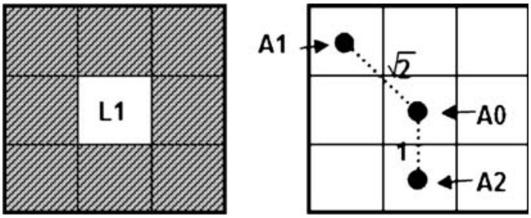

Fig. 6 Different ways of measuring distance

to estimate the average outcome of a process as if it is repeated many times. Of course, both of the simulator and the mathematical model can be used to predict outcome of the same scenario but many things are done differently, as briefed below.

When a process being carried out, the simulator repeats at a pre-set time or time step, whereas a process in the mathematical model repeats when certain agents achieving their goals. A terrain in the simulator is continuous and infinite. The grid is merely a visual representation which offers some idea of the progress. Agents have attributes such as mess and size. In the mathematical model, to make the computation easier, we assume that a terrain is finite and agents do not travel beyond its boundary. It is worth mentioning that in the simulator, agents do not always reach their goals. Using the previous scenario as an example, the following situation might occur during the simulation runs: the police kills the terrorist and approaches to the civilian. The police ends up in between the civilian and its goal. Because mess and size are considered, the police now becomes an obstacle which prevents the civilian from reaching its goal. When the civilian looks for alternative paths, it is followed by the police (as a result of inter-agent forces). So the police and the terrorist are stuck. To deal with such situations, we assume that an agent is reached the destination when it is within a certain range. This problem does not exist in the mathematical model, since we do not consider such attributes.

At unit time, an agent moves from any location to one of its neighbors. By saying this, we assume that the distance from $L1$ to any of the shaded locations is 1 (See Figure 6). However this is not how distance is measured while calculating the strength of inter-agent forces. We use the actual distance between agents, for example, the distance between $A1$ and $A0$ is $\sqrt{2}$ whereas $A2$ and $A0$ are 1 unit length away from each other (See Figure 6).

Social potential field (SPF) is the factor that results the motion in both approaches. However, it is used differently. The simulator models the agents according to physics, and each agent is associated with a current speed, a maximum speed and an acceleration rate. Under the net force, an agent might accelerate to its maximum speed and travel at that pace or decelerate to avoid collision. Due to the maximum speed, therefore the speed does not always reflect the inter-agent forces. In our approach, uniform motion is assumed and speed is not modeled explicitly. An agent is set to move from any location to one of its neighbors at unit time and its speed is therefore determined by

Results (time steps)		Potential interests			
		The terrorist dies	The police dies	The civilian dies	The civilian reaches its goal
SRP = 0.2 SRR = 0.2	Simulation	200	221	193	332
	M. Model	24	24	18.5	27
SRP = 0.4 SRR = 0.2	Simulation	160	143	177	331
	M. Model	15	32	17	22.5

Fig. 7 Results of the two approaches

the actual geographical size assigned to a location and the time unit. In the mathematical model, forces determine the motion probabilities,whereas in the simulator, motion is not only determined by the force but also the navigation mode that a agent is operating under. The navigation mode reflects the trade off between the risk of being killed and approaching the goal.

In both cases, the agents attack others by shooting. Agents in the simulator fires, with a pre-defined shooting rate, at the nearest enemy that its in the range. The outcome of the shooting is determined by a probability, that varies linearly from 0 to 1 as the distance to the enemy changes within the shooting range. In our approach, each agent has a probability of launching a successful attack at each step and it fires at the nearest enemy in its shooting range. The shooting range is also dealt differently. In the simulator an agent's attacking area is circle whereas in the mathematical model, the victim is picked within a square area. Due to the different natures of shooting, there will be discrepancies between the outcomes of the two approaches.

To compare these two approaches, we simulated the 3–agent scenarios that was presented earlier and the results are in the table below (See Figure 7). As you might have noticed that the numerical value of the results is quite different, this is because the simulation results are in time steps whereas the results of the mathematical model are in unit time. The first two roles are for the case that the police and the terrorist have the same shooting rate (0.2), and the latter two are obtained from the situation where the the police (0.4) fires twice as frequent as the terrorist (0.2).

Despite the magnitude discrepancy, the results indicates that outcomes are similar in both approaches. Our model suggested that it takes same amount of effort for the police or the terrorist to kill the other person, whereas the simulation result says that on average the police is killed at around 221 time steps and is able to kill the terrorist at around 200 time steps. The outcome of the shooting is determined by a random variable, it is therefore hard to guarantee the outcome reflects the technical characteristics of the agents as they are.

The result indicated that despite the protecting capacity of the police, it takes the civilian the same time to reach its goal. This does not sound right. However, statistics show that, with the improvement of the protecting capacity of the police, the civilian reaches its destination 363 times. Comparing with 312 times before, it is an obvious improvement. The mathematical model suggests that the civilian dies 1.088 times more than before the adjustment. We obtained very similar results, 1.0933 times, from the simulator.

Data shows that the police kills the terrorist 37.5% faster in the simulation and 20% faster in the mathematical model. As you might have noticed, after improving its attacking capacity, the police dies much faster. This is because the terrorist launches counter-attack while the police fires at it. On average, the police might be unlucky and therefore dies much faster. However, out of 500 simulation runs, it dies 138 times which is a big improvement comparing with 207 times before the adjustment. After the comparison, we can say that our approach is correct for the set of scenarios that we've tested. Of course, validation and verification is a ongoing process and it will be carried out in the future when we improve on the model.

6 Modeling systems at different levels of abstraction

There have been many attempts in modeling large scale agent systems. However the sheer complexity and diversity of large scale systems make them difficult to be described and modeled, and it is even more difficult to provide numerical predictions of the underlying processes of such systems. By dealing with entities of similar natures collectively, the approach models large scale systems at a low computation cost. After a careful study of the approach, we predict that we can obtain estimates of a system's outcome without actually modeling it.

In a system, an agent's goal is reflected by its behavior. At any location, the ratio of the net force on a direction to the total net force exerted on an agent determine its motion probability of that direction. Equation (2) suggested that, for the inter-agent forces, the manner that they vary in a terrain is the same and is proportional to the terrain size. When the terrain size changes, the ratio of the net force of a direction to the total net force remains the same. That is to say, the motion probability of an agent at any location does not vary with the terrain size. Therefore, the time that it takes an agent to achieve its task should be proportional to a terrain's size. Of course, this prediction is made under the condition that the characteristics of the terrains are similar and the relative initial locations in the terrain are similar.

Using a system with two agents, A and B as an example, we experiment to see if the prediction is correct. Without any attacking capacity, agent A travels towards its destination. Agent B, on the other hand, aims to prevent A from arriving there and launches attacks during A's journey. Experimenting the scenario with various terrain size, we obtained the following results (See Figures 8,9). The results suggest that that the outcomes are roughly proportional to the terrain size. In addition to the terrain size, we have also experimented with different shooting range. We observed that shooting range does not affect our prediction.

With the approach, we can predict the estimates for a 32×32 terrain by modeling a 8×8 terrain. The result can also model the number of agents in a system at a different levels of abstraction. For example, we can use 1 agent to

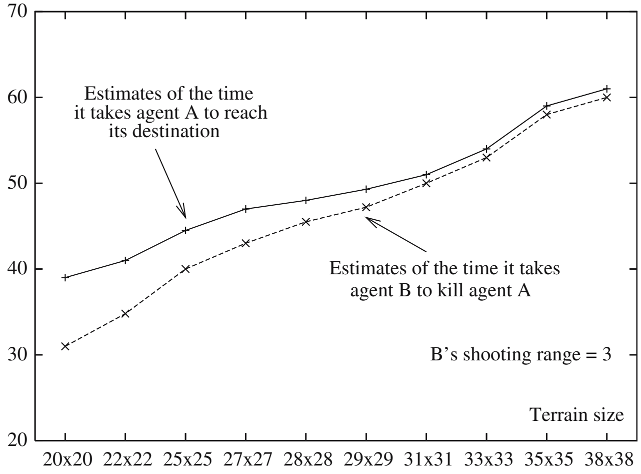

Fig. 8 Outcomes are proportional to the terrain size under certain conditions

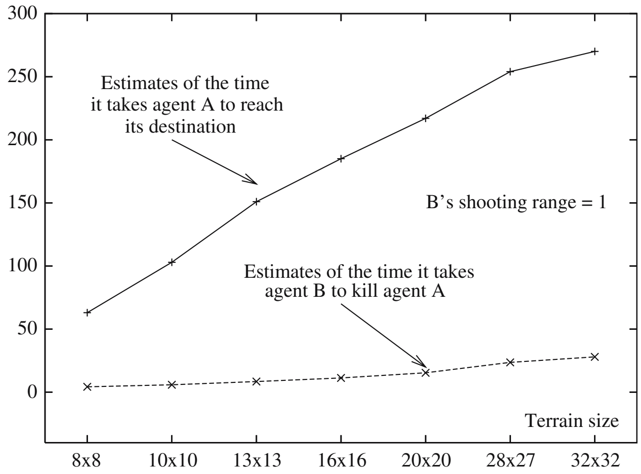

Fig. 9 Experiments with a different shooting range

symbolize a team of agents to save time and computation resources. We will test this thought thoroughly in the future study.

7 Potential extensions of the model

To advance our approach, we propose to modify the model in a few ways. We plan to incorporate behaviors such as mutation and reproduction into the model so that it can be applied in fields such as system biology. At the moment, the awareness of the agents to their surrounding environment is reflected by the SPF settings. The awareness is not necessarily mutual and it can not be changed once the process starts. We plan to modify the model so that it deals with the situation where agent awareness changes during the process. In doing so, we are able to model a change of priority or interest.

In reality, obstacles in urban environment have an impact on the agents' 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 Liu et al. (2000) to deal with the local-minima problem. The geographical characteristics in the model are the same in all locations throughout the terrain. We plan to alter this so that different geographical characteristics, such as rivers and mountains are reflected in a terrain. 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 modeling large scale agent systems at different abstraction levels.

Acknowledgements The author gratefully acknowledges the support and guidance of Prof. Erol Gelenbe, and Mr. Varol Kaptan for exchanging ideas and the help on the simulator.

References

Amin KA, Mikler AR (2002) Dynamic agent population in agent-based distance vector routing. In: Proceedings of the 2nd international workshop on Intelligent systems design and application, pp 195–200

Burmeister B (1996) Models and methodology for agent-oriented analysis and design. In: Fischer K. (ed) Working notes of the KI'96 workshop on agent-oriented programming and distributed systems

Cysneiros LM, Yu E (2003) Requirements engineering for large-scale multi agent systems. Software engineering for large-scale multi-agent systems: research issues and practical applications, 2603, pp 39–56

Gelenbe E. (1989a)Rseaux stochastiques ouverts avec clients ngatifs et positifs, et rseaux neuronaux. Comptes-Rendus Acad. Sciences de Paris, t. 309, Srie II, pp 979–982

Gelenbe E (1989b) Random neural networks with positive and negative signals and product form solution. Neural Comput 1(4): 502–510

Gelenbe E (1993a) G-networks with instantaneous customer movement. J Appl Probab 30(3):742–748

Gelenbe E (1993b) G-Networks with signals and batch removal. Probability in the Eng Inf Sci 7: 335–342

Fourneau JM, Gelenbe E, Suros R (1996) G-networks with multiple classes of positive and negative customers. Theor Comput Sci 155: 141–156

Gelenbe E, Labed A (1998) G-networks with multiple classes of signals and positive customers. Eur J Operat Res 108(2): 293–305

Gelenbe E, Hussain K, Kaptan V (2004a) Simulating the navigation and control of autonomous agents. In: Proceedings of the 7th international conference on information fusion, pp 183–189

Gelenbe E, Hussain K, Kaptan V (2004b) Enabling simulation with augmented reality. In: Proceedings of the international symposium on modeling, analysis and simulation of computer and telecommunication systems, pp 290–310

Gelenbe E, Wang Y (2004) A Trade-off between Agility and Resilience. In: Proceedings of the 13th Turkish symposium on artificial intelligence and neural networks, pp 209–217

Gelenbe E, Kaptan V, Wang Y (2004c) Biological metaphors for agent behaviour. In: Proceedings of the 19th international symposium on computer and information sciences. Lecture Notes in Computer Science, Vol LNCS 3280. Springer 667–675

Gelenbe E, Kaptan V, Wang Y (2005) Simulation and modelling of adversarial games. In: Proceedings of the 6th European GAME-ON conference on simulation and AI in computer games, pp 40–44

Huang G, Abur A, Tsai WK (1998) A multi-level graded-precision model of large scale power systems for fast parallel computation. Math Comput Model 11 325–330

Kinny D, Georgeff M, Rao A (1996) A methodology and modelling technique for systems for BDI agents. In: van der Velde W, Perram J (eds) Agents breaking away: proceedings of the 7th European workshop on modelling autonomous agents in a multi-agent world MAAMAW'96, (LANI vol 1038), pp 56–71

Liu Z, Ang MH, Seah WKG (2003) A potential field based approach for multi-robot tracking of multiple moving targets. Environment and Management International Conference

Liu CQ, Ang MH, Yong LS (2000) Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. In: Proceedings of the 2000 IEEE International conference on robitics and automation, pp 983–988

Reif JH, Wang HY (1995) Social potential fields: A distributed behavioral control for autonomous robots. The Algorithmic Foundations of Robotics, pp 331–345

Yun XP, Tan KC (1997) A wall-following method for escaping local minima in potential field based motion planning. In: Proceedings of 8th international conference on advanced robotics, pp 421–426