Modeling the Collective Strategic Searching of Artificial Insurgent Groups: A Particle Swarm Approach

Xiaohui Cui
Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085
cuix@ornl.gov

Thomas. E. Potok
Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085
potokte@ornl.gov

Abstract

A swarm based social adaptive model is proposed to model multiple insurgent groups' strategy searching in a dynamic changed environment. This report presents a pilot study on using the particle swarm modeling, a widely used non-linear optimal tool, to model the emergence of insurgency campaign. The objective of this research is to apply the particle swarm metaphor as a model of insurgent social adaptation for the dynamic environment and to provide insight and understanding of insurgent group strategic adaptation.

Contact:
Dr. Xiaohui Cui
Applied Software Engineering Research Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-6085

Tel: 1-865-576-9654
Fax: 1-865-576-0003
Email: cuix@ornl.gov

Key Words: Insurgency, Simulation, Particle Swarm, Strategic searching

Modeling Collective Strategy Searching of Artificial Insurgent Groups: A Particle Swarm Approach

Xiaohui Cui and Thomas E. Potok

1. Introduction

Insurgency warfare is dynamic, adaptive, and non-linear warfare. Today, insurgency becomes a significant challenge for the U.S. and other governments because of its two dominant characteristics: protractions and ambiguity. Currently, most ongoing work in the insurgency warfare research area is typically concerned with enhancing existing military capabilities for counter-insurgency rather than building scientific understanding of the insurgency. In terms of group strategy searching and adaptation, there does not appear to be any mature or widely used methodology. In this paper, we present a swarm based simulation to simulate artificial insurgent groups and analyze the insurgent groups' strategic searching and adaptation.

Swarm Intelligence provides a basis to explore collective (or distributed) problem solving without centralized control or the provision of a global model. Swarming tactics have been used successfully in wars throughout history by a variety of organizations. Terrorist groups like Al Qaeda are also utilizing the swarm tactics (Arquilla and Ronfeldt 2000). Particle swarm algorithm is a widely used non-linear optimal tool in swarm intelligence. In this research, a modified particle swarm modeling is used to model the emergence of insurgency campaign. The objective of this research is to apply the particle swarm metaphor as a model of insurgent social adaptation for the dynamically changing environment and to provide insight and understanding of insurgent group strategic searching. This paper is organized as follows: Section 2 incorporates an introduction to the canonical particle swarm optimization algorithm. Section 3 describes the agent based insurgency warfare simulation. Section 4 explains the implementation of particle swarm model for strategic searching. Section 5 contains the experimental setups and results. Result discussion and conclusion are presented in Section 6.

2. Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is an important part of swarm intelligence. It was originally developed by Eberhart and Kennedy in 1995 (Eberhart and Kennedy 1995), inspired by the social behavior of the bird flock. In the PSO algorithm, birds in a flock are symbolically represented as particles. These particles can be considered as simple agent swarm "flying" through a problem space. A problem space in PSO may have as many dimensions as needed to model the real problem space. Each particle has a location X-vector and a velocity V-vector. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative fitness value of the solution's utility.

Each particle also has memory to record the "best location" in the problem space that it has experienced so far, and the knowledge of the best location found so far by all the particles of the swarm. The "best location" means the problem solution generated on this location has the best fitness value. Particles of a swarm communicate the best location with each other and adjust their own location and velocity based on this best location. It is the particle's personal experience combined with its peers' experience that influences the movement of each particle through a problem space. For every generation, the particle's new location is computed by adding the particle's current velocity ***V-vector*** to its location ***X-vector***. Mathematically, given a multi-dimensional problem space, the ***ith*** particle changes its velocity and location according to the following equations (Clerc 1999; Clerc and Kennedy 2002):

$$v_{id} = w*(v_{id} + c_1 * rand_1 * (p_{id} - x_{id}) + c_2 * rand_2 * (p_{gd} - x_{id})) \tag{1a}$$

$$x_{id} = x_{id} + v_{id} \tag{1b}$$

where $\boldsymbol{V_{id}}$ indicates the speed of the particle moving along the dimensions in a problem space; $\boldsymbol{x_{id}}$ is the particle's current location; $\boldsymbol{p_{id}}$ (personal best) is the location of the particle experienced its personal best fitness value; $\boldsymbol{p_{gd}}$ (global best) is the location that the particle experienced the highest best fitness value in the entire population; $\boldsymbol{d}$ is the number of dimensions of the problem space; $\boldsymbol{rand_1}$, and $\boldsymbol{rand_2}$ are random

values in the range of (0,1). c_1 and c_2 are two positive acceleration constants; w is the constriction coefficient (Clerc and Kennedy 2002).

Eq. (1a) requires each particle to record its current coordinate x_{id}, its velocity V_{id}, its personal best fitness value location vector P_{id}, and the whole population's best fitness value location vector P_{gd}. The best fitness value X_i is updated at each generation based on Eq. (2), where the symbol $f()$ denotes the fitness function; $X_i()$ denotes the best fitness values; and t denotes the generation step.

$$X_i(t+1) = \begin{cases} X_i(t) & f(P_d(t+1)) \le X_i(t) \\ f(P_d(t+1)) & f(P_d(t+1)) > X_i(t) \end{cases} \tag{2}$$

The P_{id}, P_{gd} and their coordinate fitness values $f(P_{id})$ and $f(P_{gd})$ can be considered as the experience or knowledge of each individual particle. Eq. (2) represents the particle's knowledge updating and learning mechanism. In PSO, the knowledge of each particle will not be updated until the particle encounters a new vector location with a higher fitness value than the currently stored value in its memory. However, in a dynamic environment, the fitness value of each point in the problem space may change over time. The location with the highest fitness value ever found by a specific particle may not have the highest fitness value after several generations. Therefore, the particle needs to renew its memory whenever the real environment status does not match the particle's memorized knowledge. However, the traditional PSO lacks an updating mechanism to monitor the environmental change and renew the particles' memory when the environment changed. As a result, the particle continually uses its outdated experience/knowledge to direct its search, which inhibits the particle from following the moving path of the current optimal solution and eventually, results in the particle being easily trapped in the region of the former optimal solution. Therefore, a memory update mechanism (Cui, Hardin et al.) is used to renew particles' memory when it is necessary. In this mechanism, a new notion, an evaporation constant T, is introduced. T has a value between 0 and 1. The personal fitness value and global fitness value stored in each particle's memory will gradually evaporate (decrease) at the rate of the evaporation constant over time. The update process is formulated as:

$$X_i(t+1) = \begin{cases} X_i(t) & f(P_d(t+1)) \le X_i(t)*T \\ f(P_d(t+1)) & f(P_d(t+1)) > X_i(t)*T \end{cases} \tag{3}$$

In Eq. (3), after the value of fitness evaporates for a period of time, the fitness value, ***X-fitness***, of the current location may be higher than the evaporated fitness values and will replace the old fitness value. Although all particles have the same evaporation constant T, each particle's updating frequency may not be the same. Depending on the particle's current stored best fitness value $f(P)$ and the current fitness value $f(X)$ the particle acquired, the particle will update its best fitness value more frequently by using its current fitness value when the $f(P)$ is lower and the $f(X)$ is higher. However, when the $f(P)$ is higher and the $f(X)$ is lower in a changing environment indicates the particle's current location is farther away from the current optimal solution compared to the distance between the optimal solution and the best fitness value's position stored in the particle's memory. In this situation, the best fitness value will be kept in the particle's memory till the best fitness value became too obsolete after several generations. The fitness value update equation enables each particle to self-adapt to the changing environment.

3. Artificial Insurgent Warfare Simulation

3.1 Related Work

A relevant multi-agent based model has been built and studied by Epstein (Epstein 2002). Epstein reported a simple cellular automata (CA) model for simulating civil violence. In this idealized spatial model, a central authority seeks to use police officers to arrest (remove) active insurgency from the society for a specified jail term to suppress a decentralized rebellion. This model contains three types of agents: the general citizen, the insurgency, and the police officer. All agents possess local vision and can randomly move to an unoccupied site within its limited vision over a two-dimensional lattice. By using this simple CA simulation, Epstein showed how the complex dynamics resulting from simple assumptions can generate empirically interesting macroscopic regularities, which are too difficult to be analyzed using standard modeling approaches. The MANA model (Yiu, Gill et al. 2002), an extension of Epstein's model, introduced specific movement strategies that are aimed at correcting the purely random movement of

agents. However, the interaction between agents and cognition in both simulations are too rigid and simplistic to be psychologically plausible. The behavior of the software agent is strictly rule-based: if a particular condition appears in the agent's environment, the agent can only respond with a particular preprogrammed action. There is no direct communication between agents. Each agent does not have any capability of learning from its previous experience. This is clearly a very unrealistic representation of the social world.

A new insurgency warfare model that can provide a better understanding of the insurgent communication and learning activity is needed. We propose a particle swarm social adaptive model to simulate insurgency warfare. This report presents a pilot study of an integration of particle swarm social knowledge adaptation and multi-agent approaches for modeling the collaboration of insurgent groups and their strategic searching. In the following sections, detailed description of each component of the agent based insurgency warfare simulation will be presented.

3.2 Agents

In an agent based simulation, the most important object is the "agent." Agents represent people or groups of people need to be simulated. Agent interaction represents the process of social interaction. Two types of agents are specified in this insurgent warfare simulation – the insurgent and the authority. There can be multiple insurgent groups. The insurgent agent can be affiliated with different groups. All agents behave, act, and react in accordance with the environment they have detected.

3.3 Simulation Scenario

Different groups of insurgent agents seek efficient attacking strategies to strike the authority. The insurgent agents do not have any prior-knowledge about the strategies. The objective of each insurgent agent is to find out the action strategy that can generate the greatest profit for the insurgent group as well as the highest damage to the authority. In contrast, the authority seeks to reduce the insurgent profits of some particular strategies when the strategies are adopted by the insurgent agents. The insurgent agent that attacks the authority may receive a feedback on the results of the current and historic attack strategy. The profit of each strategy changes randomly in each time step. After a strategy is adopted by many insurgents, the profit of this strategy each insurgent agent gained will gradually reduce. The more insurgent agents that adopt the strategy, the lower the profit each insurgent agent can gain from this strategy.

3.4 Insurgent Information Exchange Rule

In the real world, the information exchange between insurgent groups is not as efficient as that within the same insurgent group. Because of the competition, some insurgent groups may not want to share their newest attacking strategy to other insurgent groups. The security reason also forces insurgent groups to avoid too many inter-insurgent group communications. Other insurgent groups have to acquire the intelligence information through other methods. For instance, they can learn other groups' new attacking strategy through newspaper, internet, and other public broadcast methods. The newest and the most efficient strategy learned from other groups are usually non-accurate or delayed. In this simulation, insurgents belonging to the same group can exchange information without any restriction. But the information exchanged between different groups will be delayed for a pre-defined number of time-steps and some noise will be added to the value of the information to reduce the information's accuracy.

3.5 Strategic Landscape

To simulate the movement of the strategies and the dynamic change of the fitness value of different strategic configurations, a test function generator, DF1, proposed by Morrison and De Jong (Morrison and De Jong 1999), is used to construct the dynamic changed landscape of insurgent strategic configurations. The DF1 generator is capable of generating a given number of cone shape peaks in a given number of dimensions. For a two dimensional strategic configuration space, the fitness value evaluation function in DF1 is defined as following:

$$f(X,Y) = MAX[H_i - R_i * \sqrt{(X - x_i)^2 + (Y - y_i)^2}] \quad (i=1,...N) \tag{4}$$

where $\boldsymbol{N}$ denotes the number of peaks in the environment. The $(\boldsymbol{x_i}, \boldsymbol{y_i})$ represents each cone's location. $\boldsymbol{R_i}$ and $\boldsymbol{H_i}$ represent the cone's height and slope. By using the DF1 generator, the landscape of the insurgent

strategic configuration space is shown in Fig. 1. The strategic landscape consists of eight different sized cone-shaped peaks randomly located in the strategic configuration space.

The dynamic changing environment is simulated as the movement of the cones in the strategic configuration space and the change of the height of the cone-shaped peaks. Different movement functions generate different types of dynamic changing environments. In this research, the environment changing rate is controlled through the logic function (Morrison and De Jong 1999):

$$Y_i = A * Y_{i-1} * (1 - Y_{i-1}) \tag{5}$$

where $\boldsymbol{A}$ is a constant and $\boldsymbol{Y_i}$ is the value at the time-step $\boldsymbol{i}$. The $\boldsymbol{Y}$ value produced on each time-step will be used to control the changing step sizes of the dynamic environment. In this research, the dynamic environment is simulated by the movement of the cone's location $\boldsymbol{(x_i, y_i)}$. The $\boldsymbol{Y}$ value represents the moving velocity of the cone location.

Another dynamic mechanism of the strategic landscape is the fitness value of the strategic configuration will gradually decrease with the increasing number of the insurgents that adopt similar strategic configurations.

$$f_i(x, y) = f_{i-1}(x, y) * (\frac{1}{e^{(N-1)}}) \tag{6}$$

where $\boldsymbol{f}$ is the landscape fitness value of strategic configuration $\boldsymbol{(x,y)}$ at the iteration $\boldsymbol{i}$. $\boldsymbol{N}$ denotes the number of insurgent that adopts similar strategic configurations.

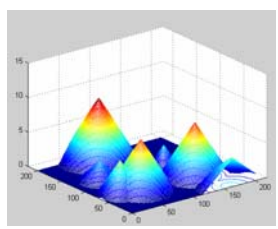


Fig 1. The strategic landscape.

4. Particle Swarm strategic searching Model

Although the PSO algorithm has been widely used as a optimization tool since it was first published in 1995, the initial research target of the PSO was to develop a human social model and the algorithm itself represents an abstract model of human knowledge social adaptation behavior (Kennedy 1997; Kennedy, Eberhart et al. 2001).

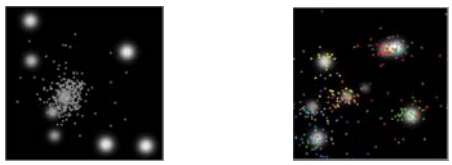
In the present proposed PSO model, the PSO algorithm is used to control the insurgent strategic searching behavior in the virtual strategy space. Under the particle swarm metaphor, each insurgent is represented as a strategic searching particle. The particle moves through a strategic virtual searching space to search for a functional optimum. Each insurgent particle has two associated properties, a current strategic configuration position $\boldsymbol{x}$ in the virtual strategy space and a velocity $\boldsymbol{v}$. Each particle has a memory of its best strategy configuration location ***(pbest)*** where the strategy configuration can generate the biggest fitness value, which is equal to the biggest loss to the authority. Each particle also knows the global best location (***gbest***) found by all other neighbor particles that belong to the same insurgent group. The ***gbest*** of different groups will be exchanged between different groups. However, these exchanges are always noisy, delayed, and some times even in error. When the delayed and noisy ***gbest*** value from other groups arrives, the ***gbest*** value from other groups will replace the ***gbest*** value within the group, if the ***gbest*** value from other groups is greater than the current ***gbest*** value. At each step of the algorithm, an insurgent particle moves from its current position to a new location based on a velocity vector. The velocity vector is influenced by the particle's previous velocity, its current location, and its ***pbest*** and ***gbest*** value. Therefore, at each step, the size and direction of each particle's movement is a function of its own history (experience) and the social influence of its peers.

5. Experimental Setups and Results

Simulations are carried out under the Netlogo (Tisue 2004) agent modeling environment. The insurgent particles use Eq. (3) to update their best fitness value. There are 300 insurgent particles randomly distributed in an environment that consists of a 100x100 rectangular grid. The grid represents all the possible strategic configurations the insurgent may adopt. A dynamic changing strategic landscape (strategic configuration fitness value) is generated and mirrored on the grid. The two dimensional visual grid is shown in Fig.2. Eight white circuits represent the insurgent attacking strategic fitness values. The brighter the white circuit, the higher the fitness value is. The insurgent particles are represented as the color dots in the grid. Different color indicates different insurgent group members. In the initial environment, all 300 insurgents are equally belonging to twenty insurgent groups, which means each group has 15 insurgent members.

Fig 2. The initial environment.

The searching of insurgent strategic configuration is presented as the movement of insurgent particle in the two dimensional grid. The movement of each insurgent particle is controlled by Eq. (1a) and Eq. (1b), in which c_1 and c_2 are set to 1.49 and V_{max} is set to 5. The w value is set to 0.72. The delayed time-step for information exchange between insurgent groups is pre-set as 20 time-steps. There is a 20% possibility that the information, including the location of the best fitness value strategic configuration and the fitness value itself, is incorrect. Two different insurgent group scenarios are simulated in this study. In the first scenario, 300 insurgents are all belong to one single insurgent group. In the second scenario, the 300 insurgents are evenly distributed into twenty different insurgent groups with 15 insurgents in each group. Each simulation will be run for 200 time-steps. The final two scenario insurgent strategic configuration distribution maps after 200 time-steps are presented in Fig.3. As shown in Fig. 3, for scenario one, all insurgents belong to the same group and they can freely exchange information about their strategic configuration and strategy performance. Every insurgent wants to adopt the strategic configuration that can generate the highest fitness value. This will cause all insurgents to swarm around the highest peak in the strategic landscape. For scenario two, as shown in Fig. 3b, limited communication between agent groups cause all insurgent agents to be relatively evenly distributed around all fitness peaks.



(a) (b)

Fig 3. The strategic configuration of each insurgent particle for (a) one group, 300 insurgents scenario, (b) twenty groups, 15 insurgents per group scenario.

In each simulation, the summary of strategic landscape fitness value of all insurgents at every time-step is recorded and used as the evaluation of the performance of the insurgent groups. The results are shown as a fitness value vs time-steps chart in Fig. 4. At the initial stage, the scenario one has a higher fitness value than the scenario two, because in scenario one, all insurgent particles can quickly aggregate around the highest peak in the strategic configuration space. However, according to Eq. (6), the congregation of the insurgent particles will cause a quick decreasing of the fitness value of the nearby strategic configuration. For scenario two, the even distribution of insurgent particles around all fitness peaks makes the fitness value of the nearby strategic configuration decreasing not as quick as scenario one. It also helps insurgent particles to quickly track the movement of the fitness peak. Comparison of the fitness value results between these two scenarios is shown in Fig. 4.

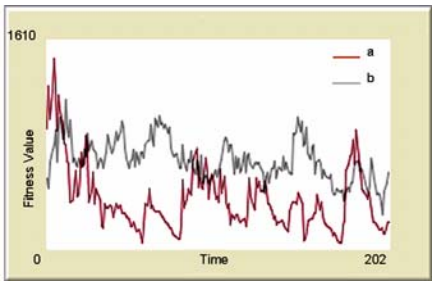


Fig 4. Comparison of the insurgent fitness summary value at each time-step for scenario (a) one group, 300 insurgents and (b) twenty groups, 15 insurgents per group.

6. Discussion and Conclusion

Discerning how insurgent groups interact, learn, and how emergent behaviors emerge from aggregate interactions in a dynamic environment is crucial for understanding insurgency. In this paper, a modified particle swarm strategic searching model is developed to simulate the complex interactions and the collective strategic searching of the insurgent groups. We construct a novel agent based simulation model to examine the impact of different group form scenarios that the insurgent groups may conduct. The objective of this research is to apply the particle swarm metaphor as a model of insurgent strategic adaptation for the dynamically changing environment and provide an insight and scientific understanding of the insurgency warfare. Results from our simulation have shown that unified leadership, planning, and effective communication are not the necessary requirements for insurgents to attain their objective.

Acknowledgments

Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725.

Reference

Arquilla, J. and D. Ronfeldt (2000). Swarming and the Future of Conflict. Santa Monica, CA, RAND.

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, IEEE.

Clerc, M. and J. Kennedy (2002). "The particle swarm-explosion, stability, and convergence in a multidimensional complex space." IEEE Transactions on Evolutionary Computation **6**(1): 58-73.

Cui, X., C. T. Hardin, et al. Tracking non-stationary optimal solution by particle swarm optimizer. Proceedings. 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, 23-25 May 2005, Towson, MD, USA, IEEE Comput. Soc.

Eberhart, R. and J. Kennedy (1995). A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, IEEE.

Epstein, J. M. (2002). "Modeling civil violence: An agent-based computational approach." PNAS **99**(90003): 7243-7250.

Kennedy, J. (1997). The particle swarm: social adaptation of knowledge, Indianapolis, IN, USA, IEEE.

Kennedy, J. F., R. C. Eberhart, et al. (2001). Swarm intelligence. San Francisco, Morgan Kaufmann Publishers.

Morrison, R. W. and K. A. De Jong (1999). A test problem generator for non-stationary environments, Washington, DC, USA, IEEE.

Tisue, S. (2004). NetLogo: A Simple Environment for Modeling Complexity. International Conference on Complex Systems, Boston, MA.

Yiu, S. Y., A. Gill, et al. (2002). Investigating Strategies for Managing Civil Violence Using the MANA Agent Based Distillation. Land Warfare Conference, Brisbane, Australia.