Particle Swarm Based Collective Searching Model for Adaptive Environment

Xiaohui Cui, Robert M. Patton, Jim Treadwell, and Thomas E. Potok

Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA cuix@ornl.gov

Summary. This report presents a pilot study of an integration of particle swarm algorithm, social knowledge adaptation and multi-agent approaches for modeling the collective search behavior of self-organized groups in an adaptive environment. The objective of this research is to apply the particle swarm metaphor as a model of social group adaptation for the dynamic environment and to provide insight and understanding of social group knowledge discovering and strategic searching. A new adaptive environment model, which dynamically reacts to the group collective searching behaviors, is proposed in this research. The simulations in the research indicate that effective communication between groups is not the necessary requirement for whole self-organized groups to achieve the efficient collective searching behavior in the adaptive environment. One possible application of this research is building scientific understanding of the insurgency in the count-Insurgent warfare.

1 Introduction

The real world is a complex system. The self-organized social groups (human community or animal colony) in the complex system search for a high profit strategy as well as adapt to the changing environment. At the same time, the changes of the environment will be impacted by the collective behaviors that emerge from the social groups when these collective behaviors are effective enough to alter the environment. The central control model and the hierarchical model are no longer suitable to provide insight and understanding of the self-organized groups' knowledge discovering and strategic searching in such complex system.

The research of some social insects, such as ants, indicate that these social insects have a new kind of social collective behavior model to help them quickly respond and adapt to the dynamic environment and survive for millions of years. Swarm Intelligence is the research field that attempts to design computational algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies [1]. Compared to the traditional algorithms, the Swarm Intelligence algorithms are flexible, robust, de-centralized, and self-organized. Swarm Intelligence provides a basis to explore collective (or distributed) problem solving without centralized control or the provision of a global model. Particle swarm algorithm [2] is one of the major research results from Swarm Intelligence. Since 2004, researchers have successfully applied the particle swarm model in the simulation of the social behavior in animals [5, 6] and strategic adaptation in organizations [6, 7]. However, in terms of self-organized group's collective strategy searching model for dynamic and adaptive environment, there does not appear to be any mature or widely used methodology.

In this research, a modified adaptive particle swarm model is used to model the self-organized group's collective strategic searching behavior in an adaptive environment. Different from randomly changing environment model used in many research, a new adaptive environment model, which dynamically reacts to the group's collective searching behaviors, is proposed in this research. The objective of this research is to apply the particle swarm metaphor as a model of human social group adaptation for the dynamic environment and to provide insight and understanding of social group's knowledge discovering and strategic searching in changing environment.

This paper is organized as follows: Section 2 provides an introduction to the canonical particle swarm optimization algorithm. Section 3 describes the particle swarm strategic searching behavior model, the dynamic and adaptive strategy profit landscape model and a modified adaptive particle swarm algorithm for dynamic environment. Section 4 explains the implementation of self-organized group's collective strategic searching simulation. Result discussion and conclusion are presented in Section 5 and 6.

2 Particle Swarm Algorithm

The particle swarm algorithm was originally developed by Eberhart and Kennedy in 1995 [2], inspired by the social behavior of the bird flock and social interactions of the human society. In the particle swarm algorithm, birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space.

The velocity and direction of each particle moving along each dimension of the problem space are altered at each generation of movement. It is the particle's personal experience combined with its neighbors' 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 [4]:

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

$$x_{id} = x_{id} + v_{id} \tag{2}$$

where, p_{id} is the location of the particle where it experiences the best fitness value; p_{gd} is the location of the particle experienced the highest best fitness value in the whole population; x_{id} is the particle current location; c_1 and c_2 are two positive acceleration constants; d is the number of dimensions of the problem space; rand1, rand2 are random values in the range of $(0, 1)$. w is called the constriction coefficient [8, 9].

Eq.1 requires each particle to record its current coordinate x_{id}, its velocity V_{id} that indicates the speed of its movement along the dimensions in a problem space, its personal best fitness value location vector P_{id} and the whole population's best fitness value location vector P_{gd}. The best fitness values are updated at each generation based on Eq.3, where the symbol f denotes the fitness function; $P_i(t)$ denotes the best fitness coordination; and t denotes the generation step.

$$f(P_i(t+1)) = \begin{cases} f(P_i(t)), & \text{if } f(X_i(t+1)) \leq f(P_i(t)) \\ f(X_i(t+1)), & \text{if } f(X_i(t+1)) > f(P_i(t)) \end{cases} \tag{3}$$

The P_{id} and P_{gd} and their coordinate fitness values $f(P_{id})$ and $f(P_{gd})$ can be considered as each individual particle's experience or knowledge and Eq.3 is the particle's knowledge updating mechanism.

3 Particle Swarm Based Collective Searching Behavior Model

In this proposed particle swarm based collective searching model, different self-organized group members seek efficient strategy configurations that can generate the highest profit in a dynamic and adaptive environment. The environment can be modeled as an adaptive profit landscape. The landscape will dynamically change as the group members search for the highest profit strategy configuration. In addition, the change of the landscape is impacted by the location of the group members. This demands that the groups not only find a highly profitable strategy in a short time, but also track the trajectory of the profitable strategy in the dynamic environment. The fitness value for assessing the performance of the self-organized groups' strategy searching is the summary value of each group member's profit in each simulation iteration instead of the highest profit one single group member can find. The group members do not have any prior-knowledge about the profit landscape. The objective of each group member is to find the strategy in the landscape that can generate greatest profit. The particle swarm based collective searching behavior model includes two important elements: the dynamic and adaptive profit landscape and the individual behavior model integrated with adaptive particle swarm algorithm.

3.1 Dynamic and Adaptive Fitness Landscape

In this model, the strategic searching in the dynamic and adaptive environment is considered as an attempt to uncover and track the highest fitness values on a dynamic fitness 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 [10], is used to construct the dynamic landscape. This DF1 test function generator has been widely used as the generator of dynamic environments [11, 12, 13, 14]. The DF1 generator is capable of generating a given number of cone shape peaks in a given number of dimensions. For a two dimensional space, the fitness value evaluation function in DF1 is defined as:

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

where N denotes the number of peaks in the environment. The (xi, yi) represents each cone's location. R_i and H_i represent the cone's height and slope.

The dynamic environment is simulated with the movement of the cones and the change of the height of the cone-shaped peaks. Different movement functions generate different types of dynamic environments. In this research, the environment change rate is controlled through the logic function [10]:

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

where A is a constant and Y_i is the value at the time-step i. The 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 (x_i, y_i). The Y value represents the moving velocity of the cone location.

In real-world applications, the evaluated fitness value cannot always be calculated precisely. Most of the time, the fitness value will be polluted by some degree of noise. To simulate this kind of noise pollution in the fitness evaluation, a noise polluted fitness value is generated with the following approach. At each iteration, the fitness value $f(x)$ can only be obtained in the form of $f^n(x)$, where $f^n(x)$ is the approximation of $f(x)$ and contains a small amount of noise n. The function can be represented as [14]:

$$f^n(x) = f(x) \times (1 + \eta);\quad \eta \sim N(0, \sigma^2) \tag{6}$$

where η illustrate the noise and is a Gaussian distributed random variable with zero mean and variance σ^2. Therefore, at each time, the particle will get a $f^n(x)$ evaluation value instead of $f(x)$. Another dynamic mechanism of the fitness landscape is the fitness value of the strategic configuration will gradually decrease with an increasing number of the searching group members that adopt similar strategic configurations.

$$f_i(x,y) = f_{i-1}(x,y) \times (\frac{1}{e^{(N-1)}}) \tag{7}$$

where f is the landscape fitness value of strategic configuration (x, y) at the iteration i. N denotes the number of group member that adopts similar strategic configurations.

3.2 The individual behavior model

The particle swarm algorithm is used to control the group member's search behavior in the fitness landscape. Under the particle swarm metaphor, each member is represented as a search particle. The particle moves through the profit landscape discussed in the previous section to search for a function optimum. Each particle has two associated properties, a current strategic configuration position x in the profit landscape and a velocity v. Each particle has a memory of its best strategy configuration location (*pbest*) where the strategy configuration can generate the highest fitness value, which is equal to the highest benefit gained by the individual. Each particle also knows the global best location (*gbest*) found by all other neighbor particles that belong to the same group. The *gbest* of different groups will be exchanged between different groups.

3.3 Distributed adaptive particle swarm algorithm for dynamic environment

In canonical particle swarm algorithm, particles' knowledge will not be updated until the particle encounters a new vector location with a higher fitness value than the value currently stored in its memory. However, in the dynamic environment discussed in the previous section, the fitness value of each point in the profit landscape may change over time. The strategic configuration location vector with the highest fitness value ever found by a specific particle may not have the highest fitness value after several iterations. It requires the particle to renew its memory whenever the real environment status does not match the particle's memorized knowledge. However, the traditional particle swarm algorithm lacks an updating mechanism to monitor the change of the environment and renew the particles' memory when the environment has changed. As a result, the particle continually uses the outdated experience/knowledge to direct its search, which inhibits the particle from following the moving path of the current optimal solution and eventually, causes the particle to be easily trapped in the region of the former optimal solution.

In this research, we adopt a modified particle swarm algorithm [15], the distributed adaptive particle swarm algorithm approach as each group member's searching behavior. In the distributed adaptive particle swarm algorithm approach, there is no specially designed particle to monitor the change of the environment. Like the traditional particle swarm algorithm, each particle uses the Eq.1 to determine its next velocity. The only difference is each particle will compare the fitness value of its current location with that of its previous location. If the current fitness value doesn't have any improvement compared to the previous value, the particle will use Eq.8 for the fitness value update. Eq.8 is slightly different compare to the traditional fitness value update function in Eq.3.

$$f(P_i(t+1)) = \begin{cases} f(p_i(t)) \times \rho, & \text{if } f(X_i(t+1)) \leq f(P_i(t)) \times \rho \\ f(X_i(t+1)), & \text{if } f(X_i(t+1)) > f(P_i(t)) \times \rho \end{cases} \quad (8)$$

In Eq.8, a new notion, the evaporation constant ρ, is introduced. ρ has a value between 0 and 1. The personal fitness value that is stored in each particle's memory and the global fitness value of the particle swarm will gradually evaporate (decrease) at the rate of the evaporation constant ρ over time.

If the particle continuously fails to improve its current fitness value by using its previous search experience, the particle's personal best fitness value as well as the global best fitness value will gradually decrease. Eventually, the personal and global best fitness value will be lower than the fitness value of the particle's current location and the best fitness value will be replaced by the particle's current fitness value. Although all particles have the same evaporation constant ρ, each particle's updating frequency may not be the same. The updating frequency depends on the particle's previous personal best fitness value $f(P)$ and the current fitness value $f(X)$ that the particle acquired. The particle will update its best fitness value more frequently by using the 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, it indicates the particle's current location is far 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. Usually the new environment (after changing) is closely related to the previous environment from which it evolves. It would be beneficial to use the knowledge/experience about the previous search space to help search for the new optimal. In this situation, the particle will keep the best fitness value in its memory until the best fitness value becomes obsolete. The fitness value update equation enables each particle to self-adapt to the changing environment.

4 Agent Based Collective Searching Simulation

The implementations of the particle swarm collective searching behavior model and the adaptive profit landscape model simulations are carried out under the Netlogo agent modeling environment [16]. Each agent in the Netlogo environment represents one particle in the model. The agents use Eq.8 to update their best fitness value. There are 300 agents randomly distributed in an environment that consists of a 100x100 rectangular grid. The grid represents all the possible strategic configurations the agent may adopt for their profit. A dynamic profit landscape is generated as discussed in section 3.1 and mirrored on the grid. The two dimensional visual grid is shown in Fig.2.

Eight white circuits represent the maximum profit values. The brighter the white circuit, the higher the profit value is. The agents are represented as the color dots in the grid. Different colors indicate different groups of agents. The searching of highly profitable strategic configuration is presented as the movement of agent in the two dimensional grid. The movement of each agent is controlled by Eq.1 and Eq.2, in which c1 and c2 are set to 1.49, V_{max} is set to 5 and the w value is set to 0.72 as recommended in canonical particle swarm algorithm [8].

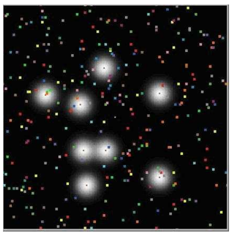


Fig. 1. The initial environment and agent groups

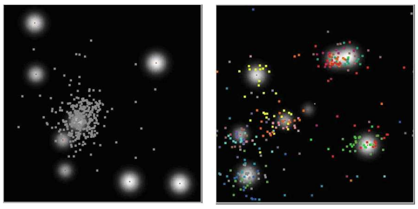
In the canonical particle swarm algorithm, each particle is fully aware of what happens to its neighbors. When one particle discovers a related good solution in the search landscape, all particles which are neighbors of this particle will be affected and change their moving direction in the next iteration. However, this is not true in the real world. The information exchange between different self-organized groups is not as efficient as that within the same group. Because of the dynamic topology or competition, some groups may not be able to share their newest high profit strategy to other agent groups. The information about other groups is usually non-accurate or delayed.

In this simulation, it is assumed that agents 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 pollute the value of the information to reduce the information's accuracy. The delayed time-step for information exchange between agent groups is pre-set as 20 time-steps. There is a 20% possibility that the information, including the location of the best fitness value and the fitness value itself, is incorrect. Two different agent group topology scenarios, scenario a and scenario b, are simulated in this study. In scenario a, 300 agents belong to one single group. In scenario b, the 300 agents are evenly distributed into 20 different groups with 15 agents in each group. Each simulation will be run for 200 iterations.

5 Results

The final distribution maps after 200 iterations are presented in Fig.2. As shown in Fig.2(a), for scenario a, all agents belong to the same group. These agents can freely exchange information about their strategic configuration and strategy performance. Every agent wants to adopt the strategic configuration that can generate the highest profit (fitness value). This will cause all agents to swarm around the highest profit peak in the profit landscape. However, because of the dynamic adaptation character of the landscape, the fitness value of the strategies around the highest peak will gradually reduce when the number of agents around it increases. In this scenario, all agents can find the highest fitness value strategy in a short time and nearly all agents will swarm around the trajectory of the highest fitness value in the dynamic environment.

For scenario b, as shown in Fig.2(b), limited communication between agent groups causes some agents to not receive the newest information about the best strategy configuration that other agents have found. Consequently, agents are distributed relatively evenly around different fitness peaks.



(a) (b)

Fig. 2. The collective searching results after 200 iterations for (a) 1 group, 300 agents scenario, (b) 20 groups, 15 agents per group scenario

In each simulation, the summary of profit (fitness value) of all agents at each iteration is recorded and used as the evaluation of the performance of the whole agent groups. The results are shown as profit (fitness value) vs time-steps chart in Fig.3. Initially, scenario a has a higher fitness value than the scenario b, because in scenario a, with the help of distributed adaptive particle swarm model, all agents can quickly aggregate around the highest peak in the strategic configuration profit landscape. However, the fitness value in the landscape will adaptively change according to Eq.4. The congregation of the agents around the highest fitness value will cause a quick decrease of the fitness value of the nearby landscape and eventually cause the summary of profit to quickly reduce. As shown in Fig.3, the profit of scenario a reduces quickly from the peak and remains low until around 200 iterations.

For scenario b, the even distribution of agents around all fitness peaks makes the fitness value of the nearby landscape not decrease as quickly as

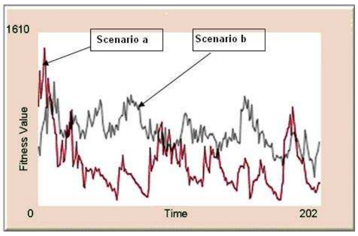


Fig. 3. Comparison of the agents summary profit value at each iteration for scenario (a) 1 group, 300 agents and (b) 20 groups, 15 agents per group.

scenario *a* and maintains a higher group fitness value than scenario a in nearly the whole simulation. The approach of scenario b also helps agents to quickly track the movement of the fitness peaks.

6 Conclusion

Most reported applications of the optimization algorithms and searching behavior models only discuss the scenarios in the static environment or the randomly changed environment. The performance evaluation of various approaches is mainly based on how fast an approach can find the optimal point in the benchmark problems. However, the real world is rarely static and its changes are not random. Most of time, the changes in the world are impacted by the collective actions of the social groups in the world. In this paper, a modified particle swarm strategic searching model is developed to simulate the complex interactions and the collective strategic searching of the self-organized groups in an adaptive environment.

We construct a novel agent based simulation model to examine the collective searching behavior of different group form scenarios. Results from the simulation have shown that effective communication is not the necessary requirements for self organized groups to attain higher profit in a dynamic and adaptive environment. Further research will discover the impact of different group architectures on the total groups' fitness value. An application that integrating the particle swarm model in an agent-based self-organizing social dynamic model will be implemented for simulating an insurgent group's social interactions and adaptations in a complex insurgency warfare system.

7 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.

References

1. Bonabeau E., Dorigo M., and Theraulaz G.: Swarm intelligence from natural to artificial systems. Oxford University Press, New York, NY (1999)
2. Eberhart R. and Kennedy J.: A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan (1995) 39–43
3. Kennedy J.: The particle swarm: social adaptation of knowledge. In Proceedings of International Conference on Evolutionary Computation, Indianapolis, IN, USA (1997) 303–308
4. Kennedy J., Eberhart R. C., Shi Y.: Swarm intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
5. Cecilia D. C., Riccardo P., and Paolo D. C.: Modelling Group-Foraging Behaviour with Particle Swarms. Lecture Notes in Computer Science, vol. 4193/2006, (2006) 661–670
6. Anthony B., Arlindo S., Tiago S., MichaelO. N., Robin M. , and Ernesto C.: A Particle Swarm Model of Organizational Adaptation. In Genetic and Evolutionary Computation (GECCO), Seattle, WA, USA (2004) 12–23
7. Silva A. S., Tiago F., Michael O. N., Robin M., Ernesto C.: Investigating Strategic Inertia Using OrgSwarm. Informatica, vol. 29, (2005) 125–141
8. Clerc M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA (1999) 1951–1957
9. Clerc M. and Kennedy J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, vol. 6 (2002) 58–73
10. Morrison R. W. and DeJong K. A.: A test problem generator for non-stationary environments. In Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA (1999) 2047-2053
11. Angeline P. J.: Tracking extrema in dynamic environments. In Angeline, Reynolds, McDonnell and Eberhart (Eds.), Proc. of the 6th Int. Conf. on Evolutionary Programming, LNCS, Vol. 1213 , Springer, (1997) 335–345
12. Blackwell T. and Branke J.: Multi-swarm optimization in dynamic environments. Applications of Evolutionary Computing, LNCS, Vol 3005, Springer, (2004) 489–500
13. Eberhart R. C. and Shi Y.: Tracking and optimizing dynamic systems with particle swarms. In Proceedings of Congress on Evolutionary Computation, Seoul, South Korea (2001) 94–100
14. Parsopoulos K. E. and Vrahatis M. N.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing (2002) 1 235–306
15. Cui X., Hardin C. T., Ragade R. K., Potok T. E., and Elmaghraby A. S.: Tracking non-stationary optimal solution by particle swarm optimizer. in Proceedings OF 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing, Towson, MD, USA (2005) 133–138
16. Tisue S.: NetLogo: A Simple Environment for Modeling Complexity. In International Conference on Complex Systems, Boston, MA (2004)