Cat Swarm Optimization

Shu-Chuan Chu[1], Pei-wei Tsai[2], and Jeng-Shyang Pan[2]

[1] Department of Information Management,
Cheng Shiu University
[2] Department of Electronic Engineering,
National Kaohsiung University of Applied Sciences

Abstract. In this paper, we present a new algorithm of swarm intelligence, namely, Cat Swarm Optimization (CSO). CSO is generated by observing the behaviors of cats, and composed of two sub-models, i.e., tracing mode and seeking mode, which model upon the behaviors of cats. Experimental results using six test functions demonstrate that CSO has much better performance than Particle Swarm Optimization (PSO).

1 Introduction

In the field of optimization, many algorithms were being proposed recent years, e.g. Genetic Algorithm (GA) [1-2], Ant Colony Optimization (ACO) [6-7], Particle Swarm Optimization (PSO) [3-5], and Simulated Annealing (SA) [8-9] etc. Some of these optimization algorithms were developed based on swarm intelligence. Cat Swarm Optimization (CSO), the algorithm we proposed in this paper, is motivated from PSO [3] and ACO [6].

According to the literatures, PSO with weighting factor [4] usually finds the better solution faster than the pure PSO, but according to the experimental results, Cat Swarm Optimization (CSO) presents even much better performance.

Via observing the behavior of creatures, we may get some idea for solving the optimization problems. By studying the behavior of ants achieves ACO, and with examining the movements of the flocking gulls realizes PSO. Through inspecting the behavior of cat, we present Cat Swarm Optimization (CSO) algorithm.

2 Behaviors of Cats

According to the classification of biology, there are about thirty-two different species of creatures in feline, e.g. lion, tiger, leopard, cat etc. Though they have different living environments, there are still many behaviors simultaneously exist in most of felines.

In spite of the hunting skill is not innate for felines, it can be trained to acquire. For the wild felines, the hunting skill ensures the survival of their races, but for the indoor cats, it exhibits the natural instinct of strongly curious about any moving things. Though all cats have the strong curiosity, they are, in most times, inactive. If you spend some time to observe the existence of cats, you may easily find that the cats spend most of the time when they are awake on resting.

Q. Yang and G. Webb (Eds.): PRICAI 2006, LNAI 4099, pp. 854 – 858, 2006.

The alertness of cats are very high, they always stay alert even if they are resting. Thus, you can simply find that the cats usually looks lazy, lying somewhere, but opening their eyes hugely looking around. On that moment, they are observing the environment. They seem to be lazy, but actually they are smart and deliberate.

Of course, if you examine the behaviors of cats carefully, there would be much more than the two remarkable properties, which we discussed in the above.

3 Proposed Algorithm

In our proposed Cat Swarm Optimization, we first model the major two behaviors of cats into two sub-models, namely, seeking mode and tracking mode. By the way of mingling with these two modes with a user-defined proportion, CSO can present better performance.

3.1 The Solution Set in the Model -- Cat

No matter what kind of optimization algorithm, the solution set must be represented via some way. For example, GA uses chromosome to represent the solution set; ACO uses ant as the agent, and the paths made by the ants depict the solution sets; PSO uses the positions of particles to delineate the solution sets. In our proposed algorithm, we use cats and the model of behaviors of cats to solve the optimization problems, i.e. we use cats to portray the solution sets.

In CSO, we first decide how many cats we would like to use, then we apply the cats into CSO to solve the problems.

Every cat has its own position composed of M dimensions, velocities for each dimension, a fitness value, which represents the accommodation of the cat to the fitness function, and a flag to identify whether the cat is in seeking mode or tracing mode. The final solution would be the best position in one of the cats due to CSO keeps the best solution till it reaches the end of iterations.

3.2 Seeking Mode

This sub-model is used to model the situation of the cat, which is resting, looking around and seeking the next position to move to. In seeking mode, we define four essential factors: seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC).

SMP is used to define the size of seeking memory for each cat, which indicates the points sought by the cat. The cat would pick a point from the memory pool according to the rules described later.

SRD declares the mutative ratio for the selected dimensions. In seeking mode, if a dimension is selected to mutate, the difference between the new value and the old one will not out of the range, which is defined by SRD.

CDC discloses how many dimensions will be varied. These factors are all playing important roles in the seeking mode.

SPC is a Boolean variable, which decides whether the point, where the cat is already standing, will be one of the candidates to move to. No matter the value of SPC is true or false; the value of SMP will not be influenced. How the seeking mode works can be described in 5 steps as follows:

Step1: Make j copies of the present position of cat_k, where j = SMP. If the value of SPC is true, let j = (SMP-1), then retain the present position as one of the candidates.
Step2: For each copy, according to CDC, randomly plus or minus SRD percents of the present values and replace the old ones.
Step3: Calculate the fitness values (FS) of all candidate points.
Step4: If all FS are not exactly equal, calculate the selecting probability of each candidate point by equation (1), otherwise set all the selecting probability of each candidate point be 1.
Step5: Randomly pick the point to move to from the candidate points, and replace the position of cat_k.

$$P_i = \frac{|FS_i - FS_b|}{FS_{\max} - FS_{\min}}, \text{ where } 0 < i < j \tag{1}$$

If the goal of the fitness function is to find the minimum solution, $FS_b = FS_{max}$, otherwise $FS_b = FS_{min}$.

3.3 Tracing Mode

Tracing mode is the sub-model for modeling the case of the cat in tracing some targets.

Once a cat goes into tracing mode, it moves according to its' own velocities for every dimension. The action of tracing mode can be described in 3 steps as follows:

Step1: Update the velocities for every dimension ($v_{k,d}$) according to equation (2).
Step2: Check if the velocities are in the range of maximum velocity. In case the new velocity is over-range, set it be equal to the limit.
Step3: Update the position of cat_k according to equation (3).

$$v_{k,d} = v_{k,d} + r_1 \times c_1 \times \left(x_{best,d} - x_{k,d}\right), \text{ where } d = 1,2,\ldots,M \tag{2}$$

$x_{best,d}$ is the position of the cat, who has the best fitness value; $x_{k,d}$ is the position of cat_k. c_1 is a constant and r_1 is a random value in the range of [0,1].

$$x_{k,d} = x_{k,d} + v_{k,d} \tag{3}$$

3.4 Cat Swarm Optimization

As we described in the above subsection, CSO includes two sub-models, the seeking mode and the tracing mode. To combine the two modes into the algorithm, we define a mixture ratio (MR) of joining seeking mode together with tracing mode.

By observing the behaviors of cat, we notice that cat spends mot of the time when they are awake on resting. While they are resting, they move their position carefully and slowly, sometimes even stay in the original position. Somehow, for applying this behavior into CSO, we use seeking mode to represent it.

The behavior of running after targets of cat is applied to tracing mode. Therefore, it is very clear that MR should be a tiny value in order to guarantee that the cats spend most of the time in seeking mode, just like the real world.

The process of CSO can be described in 6 steps as follows:

Step1: Create N cats in the process.

Step2: Randomly sprinkle the cats into the M-dimensional solution space and randomly select values, which are in-range of the maximum velocity, to the velocities of each cat. Then haphazardly pick number of cats and set them into tracing mode according to MR, and the others set into seeking mode.

Step3: Evaluate the fitness value of each cat by applying the positions of cats into the fitness function, which represents the criteria of our goal, and keep the best cat into memory. Note that we only need to remember the position of the best cat (x_{best}) due to it represents the best solution so far.

Step4: Move the cats according to their flags, if cat_k is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process. The process steps are presented above.

Step5: Re-pick number of cats and set them into tracing mode according to MR, then set the other cats into seeking mode.

Step6: Check the termination condition, if satisfied, terminate the program, and otherwise repeat step3 to step5.

4 Experimental Results

We applied CSO, PSO and PSO with weighting factor into six test functions to compare the performance. All the experiments demonstrate the proposed Cat Swarm Optimization (CSO) is superior to PSO and PSO with weighting factor. Due to the space limit of this paper, only the experimental results of test function one shown in Fig. 1.

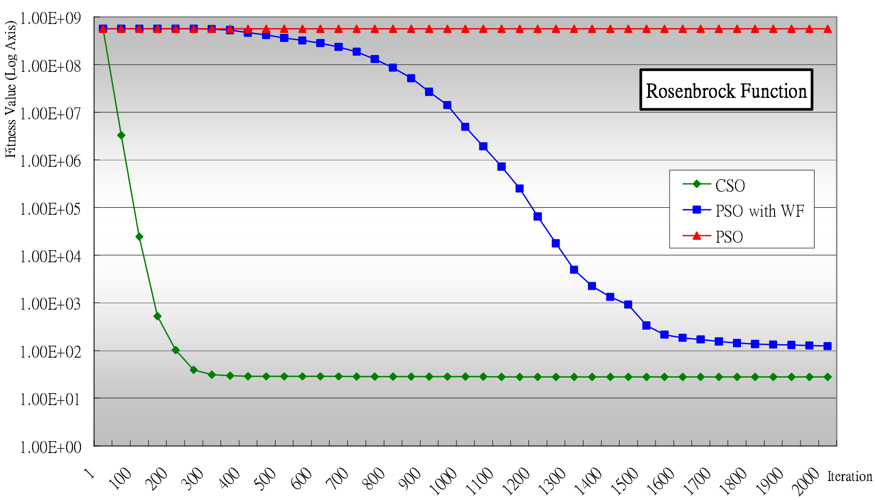

Fig. 1. The experimental result of test function 1

References

1. Goldberg, D.E.: Genetic Algorithm in Search. Optimization and Machine Learning. Addison-Wesley Publishing Company (1989)
2. Pan, J. S., McInnes, F. R., Jack, M. A. : Application of Parallel Genetic Algorithm and Property of Multiple Global Optima to VQ Codevector Index Assignment. Electronics Letters 32(4) (1996) 296-297
3. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. Sixth International Symposium on Micro Machine and Human Science (1995) 39-43
4. Shi, Y., Eberhart, R.: Empirical study of particle swarm optimization. Congress on Evolutionary Computation. (1999) 1945-1950
5. Chang, J. F., Chu, S. C., Roddick, J. F., Pan, J. S. : A Parallel Particle Swarm Optimization Algorithm with Communication Strategies. Journal of Information Science and Engineering 21(4) (2005) 809-818
6. Dorigo, M., Gambardella, L. M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. on Evolutionary Computation. 26 (1) (1997) 53-66
7. Chu, S. C., Roddick, J. F., Pan, J. S.: Ant colony system with communication strategies. Information Sciences 167 (2004) 63-76
8. Kirkpatrick, S., Gelatt, Jr. C.D., Vecchi, M.P.: Optimization by simulated annealing. Science (1983) 671-680
9. Huang, H. C., Pan, J. S., Lu, Z. M., Sun, S. H., Hang, H.M.: Vector quantization based on generic simulated annealing. Signal Processing 81(7) (2001) 1513-1523