Chapter 2
Cat Swarm Optimization (CSO) Algorithm

Mahdi Bahrami, Omid Bozorg-Haddad and Xuefeng Chu

Abstract In this chapter, a brief literature review of the Cat Swarm Optimization (CSO) algorithm is presented. Then the natural process, the basic CSO algorithm iteration procedure, and the computational steps of the algorithm are detailed. Finally, a pseudo code of CSO algorithm is also presented to demonstrate the implementation of this optimization technique.

2.1 Introduction

Optimization algorithms based on the Swarm Intelligence (SI) were developed for simulating the intelligent behavior of animals. In these modeling systems, a population of organisms such as ants, bees, birds, and fish are interacting with one another and with their environment through sharing information, resulting in use of their environment and resources. One of the more recent SI-based optimization algorithms is the Cat Swarm Optimization (CSO) algorithm which is based on the behavior of cats. Developed by Chu and Tsai (2007), the CSO algorithm and its varieties have been implemented for different optimization problems. Different variations of the algorithm have been developed by researchers. Tsai et al. (2008)

M. Bahrami · O. Bozorg-Haddad (✉)
Department of Irrigation and Reclamation Engineering, Faculty of Agricultural Engineering and Technology, College of Agriculture and Natural Resources, University of Tehran, Karaj, Tehran 31587-77871, Iran
e-mail: obhaddad@ut.ac.ir

M. Bahrami
e-mail: m.bahrami9264@ut.ac.ir

X. Chu
Department of Civil and Environmental Engineering, North Dakota State University, Dept 2470, Fargo, ND 58108-6050, USA
e-mail: xuefeng.chu@ndsu.edu

O. Bozorg-Haddad (ed.), *Advanced Optimization by Nature-Inspired Algorithms*, Studies in Computational Intelligence 720, DOI 10.1007/978-981-10-5221-7_2

presented a parallel structure of the algorithm (i.e., parallel CSO or PCSO). They further developed an enhanced version of their PCSO (EPCSO) by incorporating the Taguchi method into the tracing mode process of the algorithm (Tsai et al. 2012). The binary version of CSO (BCSO) was developed by Sharafi et al. (2013) and applied to a number of benchmark optimization problems and the zero–one knapsack problem. The chaotic cat swarm algorithm (CCSA) was developed by Yang et al. (2013a). Using different chaotic maps, the seeking mode step of the algorithm was improved. Based on the concept of homotopy, Yang et al. (2013b), proposed the homotopy-inspired cat swarm algorithm (HCSA) in order to improve the search efficiency. Lin et al. (2014a) proposed a method to improve CSO and presented the Harmonious-CSO (HCSO). Lin et al. (2014b) introduced a modified CSO (MCSO) algorithm capable of improving the search efficiency within the problem space. The basic CSO algorithm was also integrated with a local search procedure as well as the feature selection of support vector machines (SVMs). This method changed the concept of cat alert surroundings in the seeking mode of CSO algorithm. By dynamically adjusting the mixture ratio (MR) parameter of the CSO algorithm, Wang (2015) enhanced CSO algorithm with an adaptive parameter control. A hybrid cat swarm optimization method was developed by Ojha and Naidu (2015) through adding the invasive weed optimization (IWO) algorithm to the tracing mode of the CSO algorithm.

Several other authors have used CSO algorithm in different fields of research on optimization problems. Lin and Chien (2009) constructed the CSO algorithm + SVM model for data classification through integrating cat swam optimization into the SVM classifier. Pradhan and Panda (2012) proposed a new multiobjective evolutionary algorithm (MOEA) by extending CSO algorithm. The MOEA identified the non-dominated solutions along the search process using the concept of Pareto dominance and used an external archive for storing them. Xu and Hu (2012) presented a CSO-based method for a resource-constrained project scheduling problem (RCPSP). Saha et al. (2013) applied CSO algorithm to determine the optimal impulse response coefficients of FIR low pass, high pass, bandpass, and band stop filters to meet the respective ideal frequency response characteristics. So and Jenkins (2013) used CSO for Infinite Impulse Response (IIR) system identification on a few benchmarked IIR plants. Kumar et al. (2014) optimized the placement and sizing of multiple distributed generators using CSO. Mohamadeen et al. (2014) compared the binary CSO with the binary PSO in selecting the best transformer tests that were utilized to classify transformer health, and thus to improve the reliability of identifying the transformer condition within the power system. Guo et al. (2015) proposed an improved cat swarm optimization algorithm and redefined some basic CSO concepts and operations according to the assembly sequence planning (ASP) characteristics. Bilgaiyan et al. (2015) used the cat swarm-based multi-objective optimization approach to schedule workflows in a cloud computing environment which showed better performance, compared with the multi-objective particle swarm optimization (MOPSO) technique. Amara et al. (2015) solved the problem of wind power system design reliability optimization using CSO, under the performance and cost constraints. Meziane et al. (2015) optimized the electric power distribution of a solar system by determining the optimal topology among various alternatives using CSO. The results showed a better performance than the binary CSO. Ram et al. (2015) studied a 9-ring time-modulated concentric circular antenna array (TMCCAA) with isotropic elements based on CSO, for reduction of side lobe level and improvement in the directivity. Crawford et al. (2016) solved a bi-objective set covering problem using the binary cat swarm optimization algorithm. In order to achieve higher overall system reliability for a large-scale primary distribution network, Majumder and Eldho (2016) examined the effectiveness of CSO for groundwater management problems, by coupling it with the analytic element method (AEM) and the reverse particle tracking (RPT) approach. The AEM-CSO model was applied to a hypothetical unconfined aquifer considering two different objectives: maximization of the total pumping of groundwater from the aquifer and minimization of the total pumping costs. Mohapatra et al. (2016) used kernel ridge regression and a modified CSO-based gene selection system for classification of microarray medical datasets.

2.2 Natural Process of the Cat Swarm Optimization Algorithm

Despite spending most of their time in resting, cats have high alertness and curiosity about their surroundings and moving objects in their environment. This behavior helps cats in finding preys and hunting them down. Compared to the time dedicated to their resting, they spend too little time on chasing preys to conserve their energy. Inspired by this hunting pattern, Chu and Tsai (2007) developed CSO with two modes: "seeking mode" for when cats are resting and "tracing mode" for when they are chasing their prey. In CSO, a population of cats are created and randomly distributed in the M-dimensional solution space, with each cat representing a solution. This population is divided into two subgroups. The cats in the first subgroup are resting and keeping an eye on their surroundings (i.e., seeking mode), while the cats in the second subgroup start moving around and chasing their preys (i.e., tracing mode). The mixture of these two modes helps CSO to move toward the global solution in the M-dimensional solution space. Since the cats spend too little time in the tracing mode, the number of the cats in the tracing subgroup should be small. This number is defined by using the mixture ratio (MR) which has a small value. After sorting the cats into these two modes, new positions and fitness functions will be available, from which the cat with the best solution will be saved in the memory. These steps are repeated until the stopping criteria are satisfied.

Table 2.1 Characteristics of the CSO algorithm

General algorithm	Cat swarm optimization
Decision variable	Cat's position in each dimension
Solution	Cat's position
Old solution	Old position of cat
New solution	New position of cat
Best solution	Any cat with the best fitness
Fitness function	Distance between cat and prey
Initial solution	Random positions of cats
Selection	–
Process of generating new solution	Seeking and tracing a prey

Following Chu and Tsai (2007), the computational procedures of CSO can be described as follows:

Step 1: Create the initial population of cats and disperse them into the M-dimensional solution space ($X_{i,d}$) and randomly assign each cat a velocity in range of the maximum velocity value ($\upsilon_{i,d}$).
Step 2: According to the value of MR, assign each cat a flag to sort them into the seeking or tracing mode process.
Step 3: Evaluate the fitness value of each cat and save the cat with the best fitness function. The position of the best cat (X_{best}) represents the best solution so far.
Step 4: Based on their flags, apply the cats into the seeking or tracing mode process as described below.
Step 5: If the termination criteria are satisfied, terminate the process. Otherwise repeat steps 2 through 5.

Table 2.1 lists the characteristics of the CSO and Fig. 2.1 illustrates the detailed computational steps of the CSO algorithm.

2.2.1 *Seeking Mode (Resting)*

During this mode the cat is resting while keeping an eye on its environment. In case of sensing a prey or danger, the cat decides its next move. If the cat decides to move, it does that slowly and cautiously. Just like while resting, in the seeking mode the cat observes into the M-dimensional solution space in order to decide its next move. In this situation, the cat is aware of its own situation, its environment, and the choices it can make for its movement. These are represented in the CSO algorithm by using four parameters: seeking memory pool (SMP), seeking range of

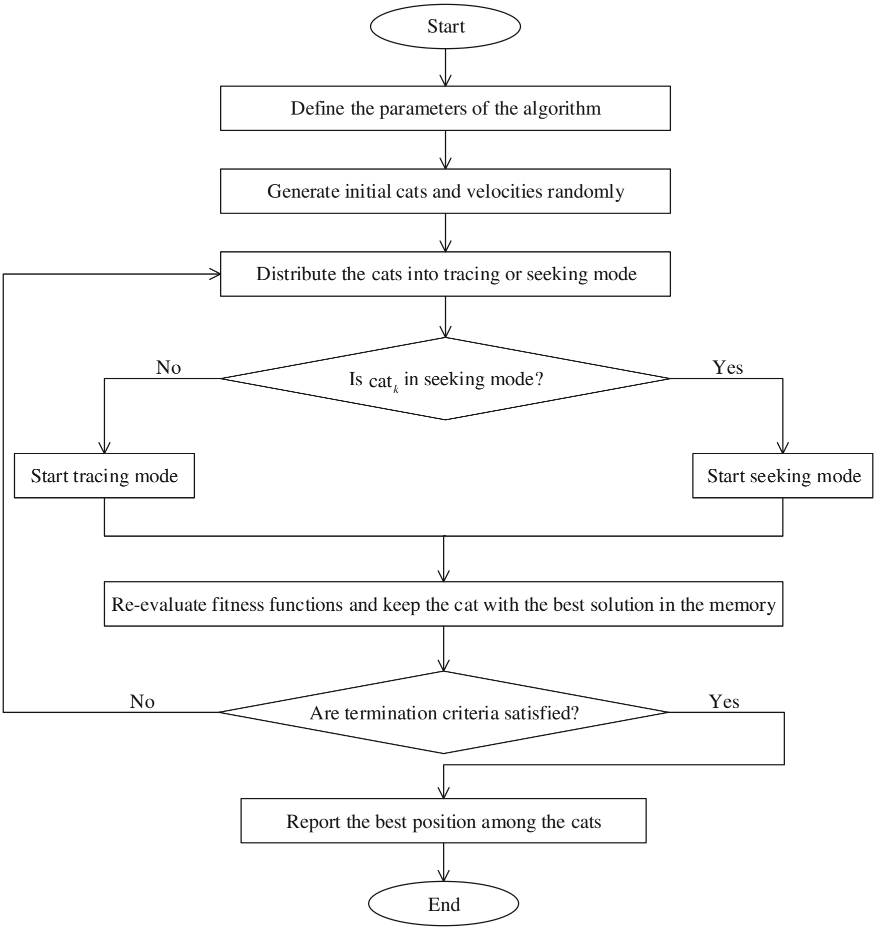

Fig. 2.1 Flowchart of the CSO algorithm

the selected dimension (SRD), counts of dimension to change (CDC), and self-position consideration (SPC) (Chu and Tsai 2007). SMP is the number of the copies made of each cat in the seeking process. SRD is the maximum difference between the new and old values in the dimension selected for mutation. CDC tells how many dimensions will be mutated. All these parameters define the seeking process of the algorithm. SPC is the Boolean variable which indicates the current position of the cat as a candidate position for movement. SPC cannot affect the value of SMP.

Following Chu and Tsai (2007), the process of the seeking mode is described below.

Step 1: Make SMP copies of each cat_i. If the value of SPC is true, SMP-1 copies are made and the current position of the cat remains as one of the copies.

Step 2: For each copy, according to CDC calculate a new position by using Eq. (2.1) (Majumder and Eldho 2016)

$$X_{cn} = (1 \pm \mathrm{SRD} \times R) \times X_c \tag{2.1}$$

in which

X_c current position;
X_{cn} new position; and
R a random number, which varies between 0 and 1.

Step 3: Compute the fitness values (FS) for new positions. If all FS values are exactly equal, set the selecting probability to 1 for all candidate points. Otherwise calculate the selecting probability of each candidate point by using Eq. (2.2).

Step 4: Using the roulette wheel, randomly pick the point to move to from the candidate points, and replace the position of cat_i.

$$P_i = \frac{|FS_i - FS_b|}{|FS_{\max} - FS_{\min}|}, \quad \text{where } 0 < i < j \tag{2.2}$$

where

P_i probability of current candidate cat_i;
FS_i fitness value of the cat_i;
$FS_{\max}$ maximum value of fitness function;
$FS_{\min}$ minimum value of fitness function; and
$FS_b = FS_{\max}$ for minimization problems and
$FS_b = FS_{\min}$ for maximization problems.

2.2.2 *Tracing Mode (Movement)*

The tracing mode simulates the cat chasing a prey. After finding a prey while resting (seeking mode), the cat decides its movement speed and direction based on the prey's position and speed. In CSO, the velocity of cat k in dimension d is given by

$$v_{k,d} = v_{k,d} + r_1 \times c_1 (X_{\text{best},d} - X_{k,d}) \tag{2.3}$$

in which, $v_{k,d}$ = velocity of cat k in dimension d; $X_{\text{best},d}$ = position of the cat with the best solution; $X_{k,d}$ = position of the cat_{k}; c_1 = a constant; and r_1 = a random value in the range of [0,1]. Using this velocity, the cat moves in the M-dimensional decision space and reports every new position it takes. If the velocity of the cat is greater than the maximum velocity, its velocity is set to the maximum velocity. The new position of each cat is calculated by

$$X_{k,d,\text{new}} = X_{k,d,\text{old}} + v_{k,d} \tag{2.4}$$

in which

$X_{k,d,\text{new}}$ new position of cat k in dimension d; and
$X_{k,d,\text{old}}$ current position of cat k in dimension d.

2.3 Termination Criteria

The termination criterion determines when the algorithm is terminated. Selecting a good termination criterion has an important role to ensure a correct convergence of the algorithm. The number of iterations, the amount of improvement, and the running time are common termination criteria for the CSO.

2.4 Performance of the CSO Algorithm

Chu and Tsai (2007) used six test functions to evaluate the CSO performance and compared the results with the particle swarm optimization (PSO) algorithm and the PSO with weighting factor (PSO-WF). According to the results CSO outperformed PSO and PSO-WF in finding the global best solutions.

2.5 Pseudo Code of the CSO Algorithm

```
Begin
    Input parameters of the algorithm and the initial data
    Initialize the cat population X_i (i = 1, 2, ... , n), υ, and SPC
    While (the stop criterion is not satisfied or I < I_max)
        Calculate the fitness function values for all cats and sort them
        X_g= cat with the best solution
            For i = 1: N
                If SPC = 1
                    Start seeking mode
                    Else
                    Start tracing mode
                End if
            End for i
    End while
    Post-processing the results and visualization
End
```

2.6 Conclusion

This chapter described cat swarm optimization (CSO) which is a new swarm-based algorithm. CSO consists of two modes, seeking mode and tracing mode which simulate the resting and hunting behaviors of cats. Each cat has a position in the M-dimensional solution space. The cats' movement toward the optimum solution is based on a flag that sorts them into the seeking or tracing mode, the first one being a slow movement around their environment and the latter being a fast movement toward the global best.

A literature review of CSO was presented, showing the success of the algorithm for different optimization problems, along with different variations of the code developed by other researchers. The flowchart of the CSO along with the pseudo code was also presented in order to make different parts of the algorithm easier to understand. These sources are a good reference point for further exploration of the CSO algorithm.

References

Amara, M., Bouanane, A., Meziane, R., & Zeblah, A. (2015). *Hybrid wind gas reliability optimization using cat swarm approach under performance and cost constraints*. 3rd International Renewable and Sustainable Energy Conference (IRSEC), Marrakech and Ouarzazate, Morocco, 10–13 December.

Bilgaiyan, S., Sagnika, S., & Das, M. (2015). A multi-objective cat swarm optimization algorithm for workflow scheduling in cloud computing environment. Intelligent Computing, Communication and Devices (pp. 73–84). New Delhi, India: Springer.

Chu, S. C., & Tsai, P. W. (2007). Computational intelligence based on the behavior of cats. *International Journal of Innovative Computing, Information and Control, 3*(1), 163–173.

Crawford, B., Soto, R., Caballero, H., Olguín, E., & Misra, S. (2016). *Solving biobjective set covering problem using binary cat swarm optimization algorithm*. The 16th International Conference on Computational Science and Its Applications, Beijing, China, 4–7 July.

Guo, J., Sun, Z., Tang, H., Yin, L., & Zhang, Z. (2015). Improved cat swarm optimization algorithm for assembly sequence planning. *Open Automation and Control Systems Journal, 7,* 792–799.

Kumar, D., Samantaray, S. R., Kamwa, I., & Sahoo, N. C. (2014). Reliability-constrained based optimal placement and sizing of multiple distributed generators in power distribution network using cat swarm optimization. *Electric Power Components and Systems, 42*(2), 149–164.

Lin, K. C., & Chien, H. Y. (2009). *CSO-based feature selection and parameter optimization for support vector machine*. Joint Conferences on Pervasive Computing (JCPC), Taipei, Taiwan, 3–5 December.

Lin, K. C., Zhang, K. Y., & Hung, J. C. (2014a). *Feature selection of support vector machine based on harmonious cat swarm optimization*. Ubi-Media Computing and Workshops (UMEDIA), Ulaanbaatar, Mongolia, 12–14 July.

Lin, K. C., Huang, Y. H., Hung, J. C., & Lin, Y. T. (2014b). Modified cat swarm optimization algorithm for feature selection of support vector machines. *Frontier and Innovation in Future Computing and Communications*, 329–336.

Majumder, P., & Eldho, T. I. (2016). A new groundwater management model by coupling analytic element method and reverse particle tracking with cat swarm optimization. *Water Resources Management, 30*(6), 1953–1972.

Meziane, R., Boufala, S., Amara, M., & Hamzi, A. (2015). *Cat swarm algorithm constructive method for hybrid solar gas power system reconfiguration*. 3rd International Renewable and Sustainable Energy Conference (IRSEC), Marrakech and Ouarzazate, Morocco, 10–13 December.

Mohamadeen, K. I., Sharkawy, R. M., & Salama, M. M. (2014). *Binary cat swarm optimization versus binary particle swarm optimization for transformer health index determination*. 2nd International Conference on Engineering and Technology, Cairo, Egypt, 19–20 April.

Mohapatra, P., Chakravarty, S., & Dash, P. K. (2016). Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system. *Swarm and Evolutionary Computation, 28,* 144–160.

Naidu, Y. R., & Ojha, A. K. (2015). A hybrid version of invasive weed optimization with quadratic approximation. *Soft Computing, 19*(12), 3581–3598.

Pradhan, P. M., & Panda, G. (2012). Solving multiobjective problems using cat swarm optimization. *Expert Systems with Applications, 39*(3), 2956–2964.
Ram, G., Mandal, D., Kar, R., & Ghoshal, S. P. (2015). Cat swarm optimization as applied to time-modulated concentric circular antenna array: Analysis and comparison with other stochastic optimization methods. IEEE Transactions on Antennas and Propagation, 63(9), 4180–4183.
Saha, S. K., Ghoshal, S. P., Kar, R., & Mandal, D. (2013). Cat swarm optimization algorithm for optimal linear phase FIR filter design. *ISA Transactions, 52*(6), 781–794.
Sharafi, Y., Khanesar, M. A., & Teshnehlab, M. (2013). *Discrete binary cat swarm optimization algorithm.* In Computer, Control & Communication (IC4). 3rd IEEE International Conference on Computer, Control & Communication (IC4), Karachi, Pakistan, 25–26 September.
So, J., & Jenkins, W. K. (2013). *Comparison of cat swarm optimization with particle swarm optimization for IIR system identification*. Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 6–9 November.
Tsai, P. W., Pan, J. S., Chen, S. M., Liao, B. Y., & Hao, S. P. (2008). *Parallel cat swarm optimization*. International Conference on Machine Learning and Cybernetics, Kunming, China, 12–15 July.
Tsai, P. W., Pan, J. S., Chen, S. M., & Liao, B. Y. (2012). Enhanced parallel cat swarm optimization based on the Taguchi method. *Expert Systems with Applications, 39*(7), 6309–6319.
Wang, J. (2015). A new cat swarm optimization with adaptive parameter control. *Genetic and Evolutionary Computing*, 69–78.
Xu, L., & Hu, W. B. (2012). Cat swarm optimization-based schemes for resource-constrained project scheduling. *Applied Mechanics and Materials, 220,* 251–258.
Yang, S. D., Yi, Y. L., & Shan, Z. Y. (2013a). Chaotic cat swarm algorithms for global numerical optimization. *Advanced Materials Research, 602,* 1782–1786.
Yang, S. D., Yi, Y. L., & Lu, Y. P. (2013b). Homotopy-inspired cat swarm algorithm for global optimization. *Advanced Materials Research, 602,* 1793–1797.

Springer

http://www.springer.com/978-981-10-5220-0

Advanced Optimization by Nature-Inspired Algorithms
Bozorg-Haddad, O. (Ed.)
2018, XV, 159 p. 34 illus., 4 illus. in color., Hardcover
ISBN: 978-981-10-5220-0