Using Bat Algorithm with Levy Walk to Solve Directing Orbits of Chaotic Systems

Xingjuan Cai[1,2], Lei Wang[1], Zhihua Cui[2], and Qi Kang[1]

[1] Department of Control Science and Engineering, Tongji University, 201804, China
[2] Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology, 030024, China
xingjuancai@gmail.com

Abstract. Bat algorithm is a novel swarm intelligent version by simulating the bat seeking behavior. However, due to the poor exploitation, the performance is not well. Therefore, a new variant named bat algorithm with Gaussian walk is proposed in which the original uniform sample manner is replaced by one Gaussian sample. In this paper, we want to further investigate the performance with Levy sampling manner. To test its performance, three other variants of bat algorithms are employed to compare, simulation results show our modification is superior to other algorithms.

Keywords: Bat algorithm, Levy walk, chaotic system.

1 Introduction

Bat algorithm (BA)[1] is novel swarm intelligent algorithm [2] inspired by the echolocation behavior of bats with varying pulse rates of emission and loudness. Due to the simple concepts and easy implementation, it has been applied to many areas successfully. Tsai et al. [3] proposed an improved EBA to solve numerical optimization problems. G. Wang et al. [4] incorporated the mutation operator into the methodology of BA. Bora et al. [5] applied BA to solve the brushless DC wheel motor problem. Recently, Gandomi and Yang introduced chaos into BA so as to increase its global search mobility for robust global optimization [6].

However, due to the poor exploitation, BA can not achieve good performance when dealing with multi-modal problems. To solve this problem, a novel variant which is called bat algorithm with Gaussian walk (BAGW)[7] is designed, simulation results show it is effective. In this paper, we provide one attempt to replace Gaussian walk with Levy walk because Levy sample can escape from local optima with more probability, furthermore, we apply this algorithm to solve the directing orbits of chaotic systems.

The rest of this paper was organized as follows: Section 2 provides a short introduction for the bat algorithm with Levy walk, as well as in section 3, our experiments are provided.

Z. Shi et al. (Eds.): IIP 2014, IFIP AICT 432, pp. 11–19, 2014.

2 Bat Algorithm with Levy Walk

In BA with Levy walk, each bat is defined by its position $\overrightarrow{x_i}(t)$, velocity $\overrightarrow{v_i}(t)$, frequency $\overrightarrow{f_i}$, loudness $A_i(t)$, and the emission pulse rate $r_i(t)$ in a $D-$dimensional search space. The new solutions $\overrightarrow{x_i}(t+1)$ at time $t+1$ is given by

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1) \tag{1}$$

and the velocity $v_{ik}(t+1)$ is updated by

$$v_{ik}(t+1) = v_{ik}(t) + (x_{ik}(t) - p_k(t)) \cdot f_{ik} \tag{2}$$

where $\overrightarrow{p}(t)$ is the current global best location found by all bats in the past generations. Frequency f_{ik} is dominated by:

$$f_{ik} = f_{min} + (f_{max} - f_{min}) \cdot \beta \tag{3}$$

where $\beta \in [0, 1]$ is a random number drawn from a uniform distribution, two parameters f_{min} and $f_{max}$are assigned to 0.0 and 2.0 in practical implementation, respectively.

To improve the exploitation capability, there exists one local search strategy for each bat, once a solution (e.g. $\overrightarrow{x_j}(t+1)$) is selected, it will be changed as with a random walk:

$$\overrightarrow{x_j}(t+1) = \overrightarrow{p}(t) + \eta\mu \tag{4}$$

where μ is a random number sampled with Levy distribution, η is a parameter to control the size of the random influence. Generally, in the first period, η assigns a large number to enhance the escaping probability from local optimum. as well as η is a small value in the later period to improve the local search probability. From our experiments, the parameter η is decreased from 0.8 to 0.001.

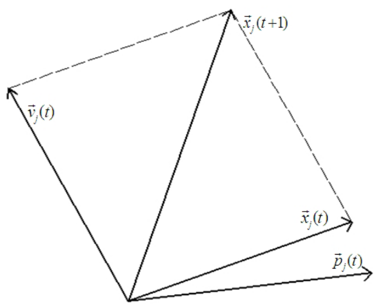

Fig. 1. Problem Illustration

For those bats whose positions are changed with Eq.(4) in iteration t, then in iteration $t+1$, the position of them will be updated with Eq.(2). In this velocity update equation, $x_{ik}(t) - p_k(t)$ is closely to zero because $\overrightarrow{x_i}(t)$ is located in the neighbor of $\overrightarrow{p}(t)$, therefore, if $\overrightarrow{v_i}(t)$ is large, then this local search pattern will be destroyed (please refer to Fig.1). Thus, in our modification, if the position $\overrightarrow{x_i}(t)$ is updated with Eq.(4), then the velocity update equation of bat i is changed as follows:

$$v_{ik}(t+1) = (x_{ik}(t) - p_k(t)) \cdot f_{ik} \tag{5}$$

while for other bats, the velocities are still updated by Eq.(2).

For bat j, the loudness $A_i(t+1)$ and the rate $r_i(t+1)$ of pulse emission are updated as follows:

$$A_i(t+1) = \alpha A_i(t) \tag{6}$$

$$r_i(t+1) = r_i(0)[1 - exp(-\gamma t)] \tag{7}$$

In this paper, $r_i(0) = 0.3$, $\gamma = 0.9$, $\alpha = 0.95$, and for each bat i, $A_i(0) = 0.9$. All of them are coming from experiments.

Algorithm 1. Bat Algorithm with Gaussian Walk

Objective function $f(\overrightarrow{x})$ $\overrightarrow{x} = (x_1, x_2, ..., x_D)^T$;
Initialize the parameters for each bat: position $\overrightarrow{x_i}(0)$, velocity $\overrightarrow{v_i}(0)$, loudness $\overrightarrow{A_i}(0)$, rate $\overrightarrow{r_i}(0)$ and repulse frequency $\overrightarrow{f_i}(0)$;
t=0;
while t<Largest iterations **do**
  Update the velocity for each bat with Eq.(2) or Eq.(5);
  Update the position for each bat with Eq.(1);
  if $rand < \overrightarrow{r_i}(t)$ **then**
    Re-update the position and velocity of bat j around the selected best solution $\overrightarrow{p}(t)$ with Eq.(4);
    Accept the new solution $\overrightarrow{x_i}(t+1)$ and velocity $\overrightarrow{v_i}(t)$;
  end if
  if $rand < \overrightarrow{A_i}(t) \& f(\overrightarrow{x_i}(t)) < f(\overrightarrow{p}(t))$ **then**
    Increase $\overrightarrow{r_i}(t)$ with Eq.(7) and reduce $\overrightarrow{A_i}(t)$ with Eq.(6);
  end if
  Rank the bats and find the current best $\overrightarrow{p}(t)$;
  t=t+1;
end while
Output the best solution $\overrightarrow{p}(t)$;

---

3 Application to Direct Orbits of Chaotic Systems

Directing orbits of chaotic systems is a multi-modal numerical optimization problem [8][9]. Consider the following discrete chaotic dynamical system:

$$\vec{x}(t+1) = \vec{f}(\vec{x}(t)), \quad t = 1, 2, ..., N \tag{8}$$

where state $\vec{x}(t) \in R^n$, $\vec{f} : R^n \to R^n$ is continuously differentiable.

Let $\vec{x}_0 \in R^n$ be an initial state of the system. If small perturbation $\vec{u}(t) \in R^n$ is added to the chaotic system, then

$$\vec{x}(t+1) = \vec{f}(\vec{x}(t)) + \vec{u}(t), \quad t = 0, 2, ..., N-1 \tag{9}$$

where $||\vec{u}(t)|| \leq \mu$, μ is a positive real constant.

The goal is to determine suitable $\vec{u}(t)$ so as to make $\vec{x}(N)$ in the $\epsilon-$neighborhood of the target $\vec{x}(t)$, i.e., $||\vec{x}(N) - \vec{x}(t)|| < \epsilon$, where a local controller is effective for chaos control.

Generally, assuming that $\vec{u}(t)$ acts only on the first component of $\vec{f}$, then the problem can be re-formulated as follows:

min $||\vec{x}(N) - \vec{x}(t)||$ by choosing suitable $\vec{u}(t)$, t=0,2,...,N-1

S.t.

$$\begin{cases} x_1(t+1) = f_1(\vec{x}(t)) + \vec{u}(t) \\ x_j(t+1) = \vec{f}_j(\vec{x}(t)) \qquad \text{j=2,3,...,n} \end{cases} \tag{10}$$

$$|u(t)| \leq \mu \tag{11}$$

$$\vec{x}(0) = \vec{x}_0 \tag{12}$$

As a typical discrete chaotic system, Hénon Map is employed as an example in this paper. Hénon Map can be described as follows:

$$\begin{cases} x_1(t+1) = -px_1^2(t) + x_2(t) + 1 \\ x_2(t+1) = qx_1(t) \end{cases} \tag{13}$$

where $p = 1.4$, $q = 0.3$.

The target $\vec{x}(t)$ is set to be a fixed point of the system $(0.63135, 0.18941)^T$, $\vec{x}_0 = (0, 0)^T$, and $\vec{u}(t)$ is only added to $\vec{x}_1$ with the bound $\mu = 0.01$, 0.02 and 0.03. The population is 20, and the largest generation is 1000.

To test the performance of BAGW, the standard version of bat algorithm (SBA)[1], bat algorithm with chaotic frequency (CBA)[6], bat algorithm with mutation operator (BAM)[4] and our modified bat algorithm (MBA) are used to compare.

Under the different values of N, Table 1-3 and Figure 2-10 list the comparison of SBA, CBA, BAM and MBA. It is obviously MBA is better than other two algorithms significantly.

Table 1. Statistics Performance with μ=0.01

N	Algorithm	Mean Value	Standard Deviation
7	SBA	1.3507e-02	2.4871e-04
	CBA	1.3574e-02	2.2013e-04
	BAM	1.3500e-02	1.7893e-04
	MBA	1.3432e-02	1.6884e-04
8	SBA	1.4627e-03	2.4469e-04
	CBA	1.4435e-03	3.9140e-04
	BAM	1.2050e-03	3.4073e-04
	MBA	1.2487e-03	2.3760e-04
9	SBA	9.1685e-04	4.4396e-04
	CBA	7.3079e-04	3.3920e-04
	BAM	6.8183e-04	3.5905e-04
	MBA	7.0345e-04	5.1549e-04

Table 2. Statistics Performance with μ=0.02

N	Algorithm	Mean Value	Standard Deviation
7	SBA	1.1719e-02	3.0371e-04
	CBA	1.1785e-02	3.3121e-04
	BAM	1.1674e-02	3.0044e-04
	MBA	1.1618e-02	3.4935e-04
8	SBA	6.7194e-04	3.7888e-04
	CBA	5.3853e-04	2.1505e-04
	BAM	4.3811e-04	3.2369e-04
	MBA	4.2412e-04	2.8488e-04
9	SBA	7.8986e-04	3.0203e-04
	CBA	6.9046e-04	3.2074e-04
	BAM	6.2232e-04	3.4635e-04
	MBA	5.4411e-04	3.8880e-04

Table 3. Statistics Performance with μ=0.03

N	Algorithm	Mean Value	Standard Deviation
7	SBA	1.0274e-02	5.4125e-04
	CBA	1.0065e-02	4.3152e-04
	BAM	1.0060e-02	3.3370e-04
	MBA	1.0016e-02	4.0748e-04
8	SBA	8.0668e-04	4.1823e-04
	CBA	6.7150e-04	3.1692e-04
	BAM	6.3732e-04	3.8370e-04
	MBA	5.3046e-04	3.4043e-04
9	SBA	8.4733e-04	4.9613e-04
	CBA	1.0249e-03	4.8139e-04
	BAM	8.6568e-04	5.3757e-04
	MBA	7.1709e-04	3.9151e-04

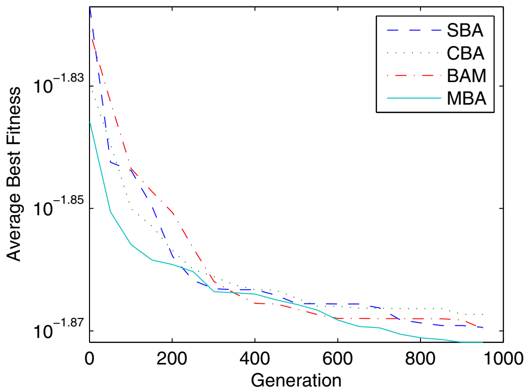

Fig. 2. Dynamic Performance with Error=0.01 and N=7

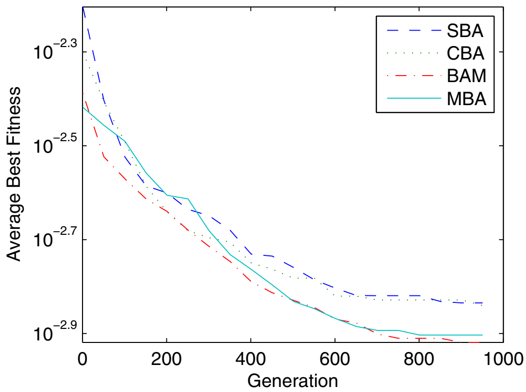

Fig. 3. Dynamic Performance with Error=0.01 and N=8

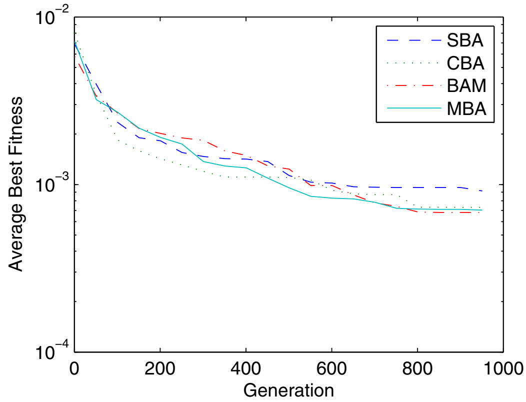

Fig. 4. Dynamic Performance with Error=0.01 and N=9

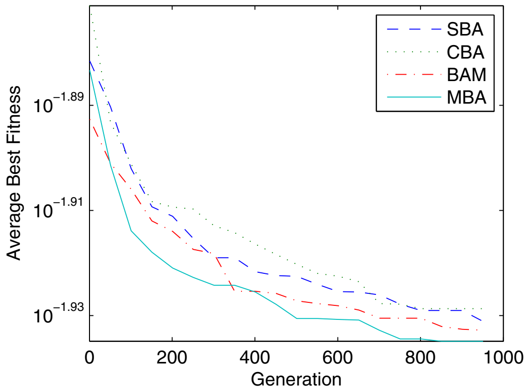

Fig. 5. Dynamic Performance with Error=0.02 and N=7

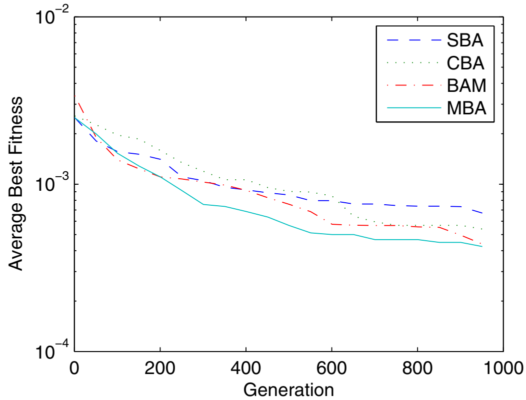

Fig. 6. Dynamic Performance with Error=0.02 and N=8

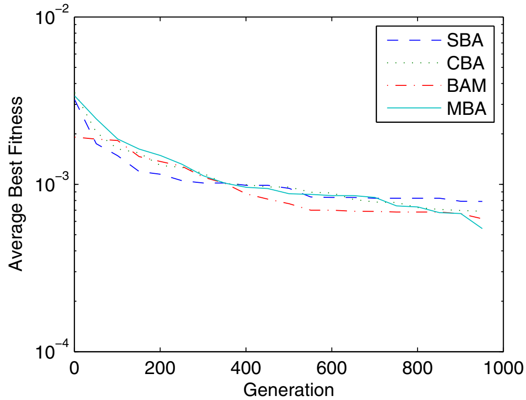

Fig. 7. Dynamic Performance with Error=0.02 and N=9

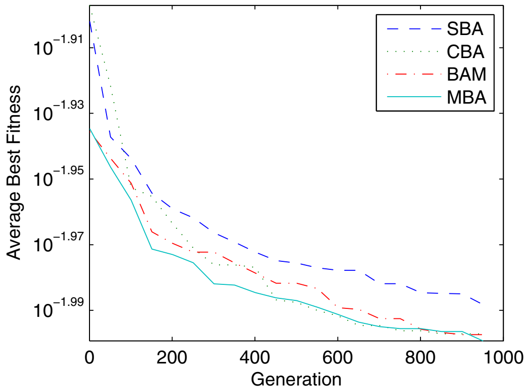

Fig. 8. Dynamic Performance with Error=0.03 and N=7

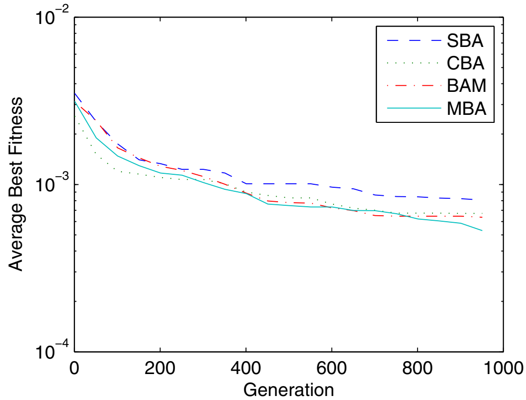

Fig. 9. Dynamic Performance with Error=0.03 and N=8

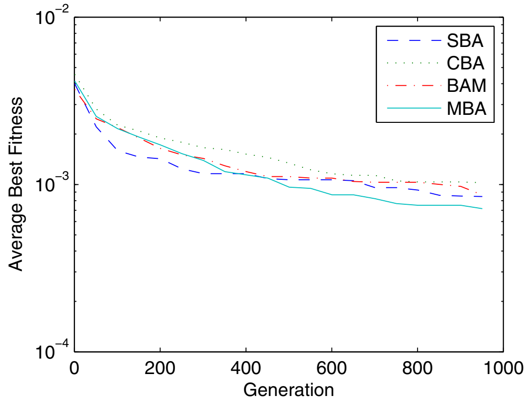

Fig. 10. Dynamic Performance with Error=0.03 and N=9

4 Conclusion

In this paper, a new variant of BA, bat algorithm with Levy walk, is applied to solve the directing orbits of chaotic systems. Simulation results show it is effective. Furthermore research topics include other applications.

Acknowledgment. This paper is supported by Natural Science Foundation of China (61003053, 71371142, 61005090), the Fundamental Research Funds for the Central Universities, the Research Fund of State Key Lab. of Management and Control for Complex Systems and the Program for the Innovative Talents of Higher Learning Institutions of Shanxi.

References

1. Yang, X., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Engineering Computation 29, 464–483 (2012)
2. Parpinelli, R.S., Lopes, H.S.: New inspirations in swarm intelligence: a survey. International Journal of Bio-inspired Computation 3(1), 1–16 (2011)
3. Tsai, P.W., Pan, J.S., Liao, B.Y., Tsai, M.J., Istanda, V.: Bat algorithm inspired algorithm for solving numerical optimization problems. Applied Mechanics and Materials 148-149, 134–137 (2011)
4. Wang, G., Guo, L., Duan, H., Liu, L., Wang, H.: A bat algorithm with mutation for UCAV path planning. The Sciencific World Journal, Article ID 418946, 15 pages (2012), doi:10.1100/2012/418946
5. Bora, T.C., Coelho, L.S., Lebensztajn, L.: Bat-inspired optimization approach for the brushless DC wheel motor problem. IEEE Transactions on Magnetics 48(2), 947–950 (2012)
6. Gandomi, A.H., Yang, X.S.: Chaotic bat algorithm. Journal of Computational Science 5, 224–232 (2014)
7. Cai, X.J., Wang, L., Kang, Q., Wu, Q.D.: Bat algorithm with Gaussian walk. International Journal of Bio-inspired Computation 6(3), 166–174 (2014)
8. Liu, B., Wang, L., Jin, Y.H.: Directing orbits of chaotic systems by particle swarm optimization. Chaos Solitons & Fractals 29, 454–461 (2006)
9. Wang, L., Li, L.L., Tang, F.: Directing orbits of chaotic dynamical systems using a hybrid optimization strategy. Physical Letters A 324, 22–25 (2004)