

Analysis of Parameter Settings for the Bat Algorithm Using Variance Evolution

Xin-She Yang*[0000-0001-8231-5556] and Mehmet
Karamanoglu[0000-0002-5049-2993]

Faculty of Science and Technology, Middlesex University,
The Burroughs, London NW4 4BT, United Kingdom.
x.yang@mdx.ac.uk

Abstract. Parameter settings in evolutionary algorithms and metaheuristics are important because such parameter values can influence the performance of algorithms under evaluation. For a given algorithm, there are many different numerical experiments to show that the algorithm can work well in practice; however, in most cases there is no theoretical analysis of parameter settings. In this work, we show that theoretical analysis using the theory of dynamical systems and evolution of population variance can give some good results in terms of parameter ranges for the bat algorithm. We also show that results from numerical experiments are consistent with theoretical bounds. Such analyses can provide good insights from different perspectives about the algorithmic characteristics such as variance evolution, transition between exploration and exploitation as well as convergence behaviour.

Keywords: Bat algorithm · Variance evolution · Nature-inspired computing · Optimization.

1 Introduction

There are many algorithms and techniques for solving optimization problems in engineering and industrial applications. Most algorithms have algorithm-dependent parameters that may require a certain degree of tuning or parametric studies. As real-world design problems can be highly nonlinear and multimodal, subject to multiple constraints [6, 30], some traditional algorithms such as gradient-based algorithms cannot deal well with such nonlinear optimization problems. Nature-inspired algorithms are a class of algorithms that can have some advantages over traditional algorithms because they use some randomization components, which makes it more likely for algorithms to find the global optimal solutions. However, most nature-inspired metaheuristic algorithms tend to have more parameters than traditional algorithms [18, 25, 23, 30], and their performance can be influenced largely by their parameter settings. Therefore, parameter tuning is important to metaheuristic algorithms.

* Corresponding Author

Parameter tuning and parameter settings are important [11, 17, 16]. There are different ways for tuning parameters in algorithms, such as the Monte Carlo based methods [17] and others [11]. These methods can provide some great insights into the parameter value ranges and the best parameter values for a given algorithm for solving a set of optimization problems. Apart from parameter tuning methods, there are ways for analyzing algorithms to gain insights into their parameter settings, including the use of dynamical system theory [10] and the analysis of population variance [32]. In this paper, we will use the bat algorithm (BA) as the main focus to show that the evolution of population variance can obtain the same parameter ranges as other methods. In addition, the comparison of actual variance with theoretical predictions gives constant results, and the order of transition time from exploration to exploitation for a given problem is also consistent with theoretical analysis.

This paper is organized as follows. Section 2 briefly outlines the bat algorithm and parameter settings, whereas Section 3 presents the analysis of the variance evolution in the swarm population. Section 4 uses three benchmark functions to validate and compare the variance variations during iterations. Section 5 shows the parameter settings on acceptance probability of new solutions. Finally, Section 6 provides a summary with some discussion about future work.

2 Bat Algorithm and its Parameter Settings

The bat algorithm (BA) is a swarm intelligence based algorithms [29], which has been extended and applied to many applications [31, 12, 21]. For a swarm of n virtual bats at locations x_i^t with flying velocities v_i^t at iteration t , the main algorithmic equations of the BA can be rewritten as

$$\omega_i = \omega_{\min} + (\omega - \omega_{\min})\beta, \quad (1)$$

$$v_i^{t+1} = \rho v_i^t + (g - x_i^t)\omega_i, \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \quad (3)$$

where g is the best solution found so far at iteration t among all the population x_i^t ($i = 1, 2, \dots, n$). Here, ω_i is the frequency of bat i in the range from ω_{\min} to ω (maximum possible frequency), and β is a random number drawn from a uniform distribution $[0, 1]$. In addition, there are variations of pulse emission rate and loudness in the standard BA [29]. However, we will focus on the algorithmic equations only; therefore, there are two main parameters to be considered here: inertia weight ρ and the maximum frequency range ω .

To analyze the characteristics and iteration behaviour, fixed point theory based methods can usually work well for deterministic, gradient-based algorithms under certain assumptions. However, such methods do not work well for metaheuristic algorithms because randomization and random variables are used in metaheuristic algorithms [19, 14, 34]. Researchers have studied metaheuristic algorithms, such as particle swarm optimization (PSO), from different perspectives. For example, PSO has been analyzed using simplified dynamical system

framework [10]. Genetic algorithms, differential evolution and other algorithms have been analyzed using Markov chains and other methods to gain insights about parameters and convergence [1–3, 5, 24, 27, 8, 26, 22].

For the bat algorithm, Chen et al. used both Markov chain framework and dynamical system [9] and obtained the appropriate ranges of the two parameters in the BA. They rewrote the BA as a dynamical system

$$v_i^{t+1} = \rho v_i^t + (g - x_i^t)\omega_i = -\omega x_i^t + \rho v_i^t + \omega g, \quad (4)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} = x_i^t + \rho v_i^t - \omega x_i^t + \omega g, \quad (5)$$

which can be expressed as a matrix form

$$Y^{t+1} = AY^t + Wg, \quad (6)$$

where

$$Y^t = \begin{pmatrix} x_i^t \\ v_i^t \end{pmatrix}, \quad A = \begin{pmatrix} 1 - \omega & \rho \\ -\omega & \rho \end{pmatrix}, \quad W = \begin{pmatrix} \omega \\ \omega \end{pmatrix}. \quad (7)$$

The stability theory of dynamical system requires both eigenvalues λ of A must satisfy $|\lambda| \leq 1$. Their analysis [9] concludes that

$$-1 \leq \rho \leq 1, \quad 0 \leq \omega \leq 2\rho + 2. \quad (8)$$

Though such analysis is insightful, however, it does not provide direct insight into how the population in the bat algorithm evolves during iterations. A different way of analyzing the population is to see how the population variance changes over time. For example, the population variance of differential evolution (DE) has been analyzed by Zaharie [32, 33], which provides some insights about the influence of the mutation probability and variations of population variance in the DE.

For the BA, we can use a similar framework to analyze its variance evolution. In order to carry out such analysis and for the ease of notations for late analysis, we will drop the superscripts and use $z_i = x_i^{t+1}$, $y_i = v_i^{t+1}$, $x_i = x_i^t$, and $\omega_{\min} = 0$. Thus, we can rewrite the algorithmic equations (2) and (3) as

$$y_i = \rho v_i + (g - x_i)\omega_i, \quad (9)$$

$$z_i = x_i + y_i = x_i + \rho v_i + (g - x_i)\omega_i, \quad (10)$$

where $\omega_i = \omega\beta$. This set of equations will be used for analysis in the next section.

3 Evolution of Variance

For a given random variable X with a mean of μ and a variance of σ^2 , the mean of an arbitrary function $\phi(X)$ can be estimated by the Taylor expansion method [15, 28] or the delta method in general [4, 20].

$$\mathbb{E}[\phi(X)] = \phi(\mu) + \frac{\phi''(\mu)}{2}\sigma^2, \quad (11)$$

where ϕ'' is the second derivative of ϕ . The variance of $\phi(X)$ can be approximately by

$$\text{Var}[\phi(X)] = [\phi'(\mu)]^2 \sigma^2 + \frac{[\phi''(\mu)]^2}{2} \sigma^4. \quad (12)$$

3.1 Variance Evolution of the Bat Algorithm

Now we can carry out the theoretical analysis of the BA using a similar analysis for differential evolution [32, 33]. For bat i , the updating equation (10) becomes

$$z_i = x_i + \rho v_i + (g - x_i)\omega\beta, \quad (13)$$

where β obeys a uniform distribution with a mean of β_0 . When $\beta \sim U(0, 1)$, we have $\beta_0 = 1/2$.

Similarly, the corresponding equation for bat j is

$$z_j = x_j + \rho v_j + (g - x_j)\omega\beta. \quad (14)$$

The differences of the preceding two equations give

$$(z_i - z_j) = (x_i - x_j) + \rho(v_i - v_j) - (x_i - x_j)\omega\beta, \quad (15)$$

which can be written compactly as

$$(z_i - z_j) = (1 - \omega\beta)(x_i - x_j) + \rho(v_i - v_j). \quad (16)$$

By squaring this equation, it becomes

$$\begin{aligned} (z_i - z_j)^2 &= (1 - \omega\beta)^2 (x_i - x_j)^2 \\ &+ \rho^2 (v_i - v_j)^2 + 2(1 - \omega\beta)\rho(x_i - x_j)(v_i - v_j). \end{aligned} \quad (17)$$

Though the population evolves with time, it is difficult to show if all bats or agents are completely independent in the statistical sense. However, as an approximation and for simplicity, we assume here that $(x_i - x_j)$ and $(v_i - v_j)$ are independent for a given population of size n . Thus, the expectation of the preceding equation becomes

$$\frac{2n}{(n-1)}\sigma_z^2 = (1 - \omega\beta_0)^2 \frac{2n}{(n-1)}\sigma_x^2 + \rho^2 \frac{2n}{(n-1)}\sigma_v^2. \quad (18)$$

where we have used

$$\mathbb{E}[(x_i - x_j)^2] = \frac{2n}{(n-1)}\sigma_x^2, \quad (19)$$

$$\mathbb{E}[(1 - \omega\beta)g(Z)] = \mathbb{E}[(1 - \omega\beta)] \mathbb{E}[g(Z)] \quad (20)$$

assuming β and $g(Z)$ are independent [4, 32].

When the population size n is sufficiently large ($n \gg 1$), we have approximately $n/(n-1) \approx 1$ and

$$\sigma_z^2 = (1 - \omega\beta_0)^2 \sigma_x^2 + \rho^2 \sigma_v^2, \quad (21)$$

It should be sufficiently realistic to assume that $\sigma_v^2 = \sigma_x^2$ for the previous population at an earlier iteration because the variance of velocities is closely linked to the variance of positions. Then, equation (21) becomes

$$\sigma_z^2 = [(1 - \omega\beta_0)^2 + \rho^2]\sigma_x^2. \quad (22)$$

This is the change of variances in a single iteration. For $t \geq 1$ iterations, the new variance at iteration t becomes

$$\sigma_{z,t}^2 = \Lambda^t \sigma_0^2, \quad (23)$$

where σ_0^2 is the the variance of the initial population. The factor is given by

$$\Lambda = (1 - \omega\beta_0)^2 + \rho^2 \leq 1. \quad (24)$$

The reduction of variance during the iteration requires that

$$\Lambda \leq 1. \quad (25)$$

At one extreme $\rho^2 = 1$ from $\rho = \pm 1$, it requires that

$$1 - \omega\beta_0 = 0, \quad (26)$$

or

$$\omega = \frac{1}{\beta_0} = 2, \quad (27)$$

for a uniform distribution in $[0, 1]$ with $\beta_0 = 1/2$. At the other extreme $\rho = 0$, equation (24) becomes

$$-1 \leq 1 - \omega\beta_0 \leq +1, \quad (28)$$

which gives

$$0 \leq \omega \leq \frac{2}{\beta_0} = 4. \quad (29)$$

This conclusion is consistent with the results, given in Eq. (8). That is

$$-1 \leq \rho \leq +1, \quad 0 \leq \omega \leq 4. \quad (30)$$

This shows that both the variance evolution and dynamical systems can obtain the same parameter value ranges for the BA.

3.2 Transition from Exploration to Exploitation

As the iterations and search for optimality continue, the overall population variance should gradually be reduced. Once the variances become sufficiently small, the diversity of the whole population may be limited, indicating potentially converged states.

There is a transition time constant $\tau = t_*$ when the population variance reduces to about $1/100 = 1\%$ of its initial variance. That is

$$\frac{\sigma_{z,t}^2}{\sigma_0^2} = \Lambda^{t_*} = [(1 - \omega\beta_0)^2 + \rho^2]^{t_*} = 0.01. \quad (31)$$

Taking the logarithm of both sides, we have

$$\tau = t_* = \frac{\ln 0.01}{\ln[(1 - \omega\beta_0)^2 + \rho^2]}. \quad (32)$$

In case of $\beta_0 = 1/2$ and $\omega = 2$, we have

$$\tau \approx \left\lceil \frac{4.6}{-\ln[(1 - 2 \times 1/2)^2 + \rho^2]} \right\rceil = \left\lceil \frac{2.3}{-\ln \rho} \right\rceil. \quad (33)$$

For $\rho = 0.9$, we have $\tau \approx 22$ after rounding up to the nearest integer. In practice, the transition can occur between τ to 2τ (or about 22 to 44 iterations for $\rho = 0.9$).

It is worth pointing out that the choice from the initial variance to $1/100$ of the initial value is rather arbitrary. If a different value such as $1/1000$ is used, the time constant τ only changes by a factor of $\ln 0.001 / \ln 0.01 = 1.5$, and this factor does not affect the results significantly. Thus, the transition time constant should be viewed as an approximate timescale for the transition from exploration to exploitation, which can be estimated by $O(-1/\ln \rho) = O(1/\ln(1/\rho))$.

Even the variance does not change much after this transition, the objective values continue to reduce (or improve) for minimization problems because the moves are mostly local refinements. However, if the transition occurs too early, it may lead to premature convergence of the population. Therefore, ρ cannot be too small. In practice, $\rho \in [0.7, 0.97]$ is the appropriate range, as indicated from our empirical parameter studies.

4 Numerical Experiments

To validate the above theoretical results and compare with the actual variations of variances for the BA, we will use three test benchmarks: the sphere function, the Rosenbrock function and the spring design benchmark. The sphere function is convex and separable, whereas the Rosenbrock function is non-convex and non-separable. The spring design problem is a nonlinear constrained design problem.

4.1 Sphere Function

For the D -dimensional sphere function

$$f_1(\mathbf{x}) = \sum_{i=1}^D x_i^2, \quad x_i \in \mathbb{R}, \quad (34)$$

we can use the BA to find its minimum $f_{\min} = 0$ at $(0, 0, \dots, 0)$. For a population of size $n = 20$, the number of iteration is set to be $t_{\max} = 100$. For $D = 3$ and different values of $\rho = 0.7, 0.8, 0.9$ with $\omega = 2$, the theoretical prediction of the population variances with the actual population variance in simulation are shown in Fig. 1.

The transition from a quick variance reduction to a slower rate occurs between $\tau = 22$ to $2\tau = 44$ iterations, as seen in Fig. 1. After this transition period, even the variance does not change, the objective values continue to reduce for minimization problems, which shows that the search moves are mostly local refinements at later iterations.

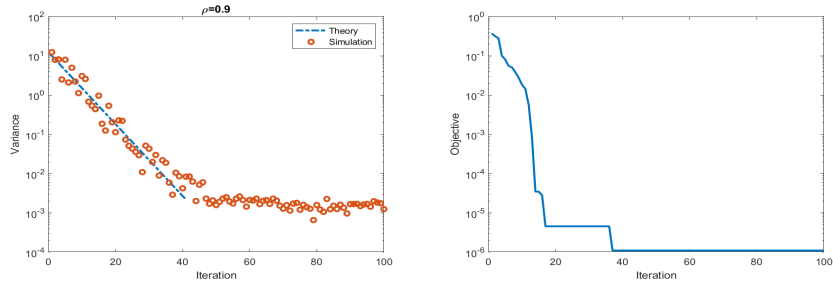


Fig. 1. Variation of population variance of the bat algorithm with $\rho = 0.9$ for the sphere function where the dashed line is the prediction from theory and the data are obtained from simulation (left). The variations of the objective values f_{\min} are shown on the right.

Similarly, variances for $\rho = 0.8$ and $\rho = 0.7$ are also shown in Fig. 2. As we can see, the predictions are quite accurate at the earlier iterations for $\rho = 0.9$ and $\rho = 0.8$. When ρ gets smaller, the prediction is less accurate.

In addition, at later iterations, the predictions are very different, which may indicate that the BA becomes more exploitation focused. In this case, the search move and update of solutions are mainly local, and the acceptance rate of new moves becomes low. Thus both, the variances and the objective values become slow varying as see in the flat part of the curves.

4.2 Rosenbrock Function

The sphere function is a convex function, and the evolution of its variances fits well to the theoretical trends for different values of ρ . Now let us test the theoretical results further using a non-convex function

$$f_2(\mathbf{x}) = (1 - x_1)^2 + 100 \sum_{i=2}^D (x_i - x_{i-1}^2)^2, \quad -100 \leq x_i \leq 100. \quad (35)$$

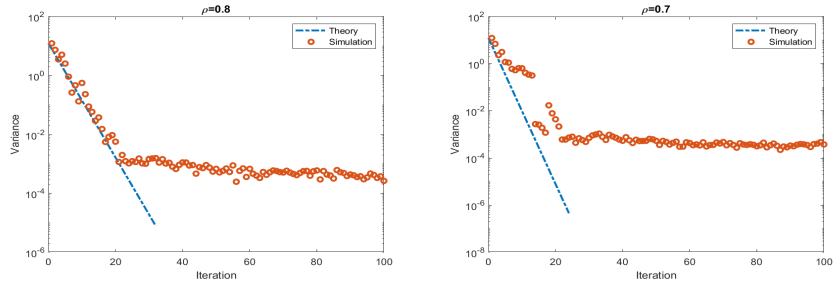


Fig. 2. Evolution of population variance at different stages of iterations for $\rho = 0.8$ (left) and $\rho = 0.7$ (right).

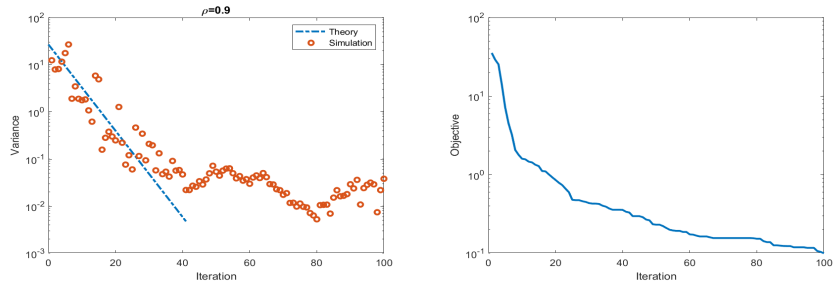


Fig. 3. The population variance for $\rho = 0.9$ for the Rosenbrock function (left) and the variations of objective values (right).

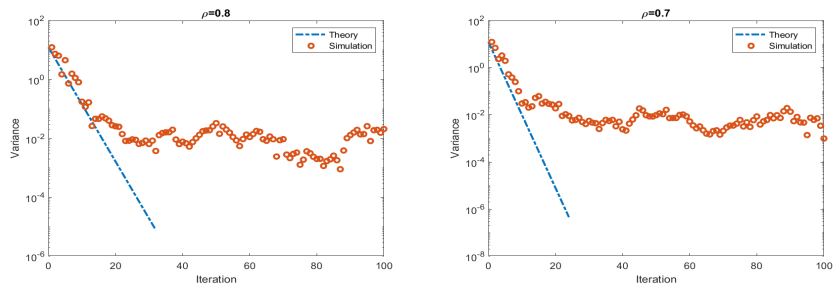


Fig. 4. The evolution of f_2 variances for $\rho = 0.8$ (left) and $\rho = 0.7$ (right).

Using the same parameter settings of the BA ($n = 20$, $t_{\max} = 100$, $D = 3$, and $\rho = 0.7, 0.8, 0.9$), the optimal solutions are found by the BA. The variations of the variances are shown in Figs. 3 and 4 where we can see that the actual variations of variances fit reasonably well to the theoretical trend on the log scale at the initial stage of the iterations. As the iteration continues, the search process becomes more focused on the local exploitation, thus the variances becomes slow varying, but objective values continue to reduce due to local refinement.

The transition from exploration to exploitation also occurs at about 22 to 44 iterations for $\rho = 0.9$. This means that the transition seems to be independent of the benchmarks in the simulation.

4.3 Spring Design

The spring design problem is to manufacture a compression and tension spring from a wire of diameter w , which will form a spring with a mean coil diameter D and the number m of active coils. The objective is to minimize the overall weight or mass

$$\min f(w, D, m) = (2 + m)w^2D, \quad (36)$$

subject to four constraints

$$g_1 = 1 - \frac{mD^3}{71785w^4} \leq 0, \quad (37)$$

$$g_2 = \frac{(4D^2 - wD)}{12566(Dw^3 - w^4)} + \frac{1}{5108w^2} - 1 \leq 0, \quad (38)$$

$$g_3 = 1 - \frac{140.45w}{mD^2} \leq 0, \quad (39)$$

$$g_4 = \frac{(w + D)}{1.5} - 1 \leq 0. \quad (40)$$

$$(41)$$

These constraints are related to deflection, shear stress, frequency and physical dimension constraints. In the simple domain of

$$0.05 \leq w \leq 1.00, \quad 0.25 \leq D \leq 1.30, \quad 2 \leq m \leq 15, \quad (42)$$

the best solution in the current literature [7, 13] is

$$f_{\min} = 0.01266522, \quad w = 0.05169, \quad D = 0.35673, \quad m = 11.2885. \quad (43)$$

With the same parameter settings ($t_{\max} = 100$, $n = 20$, $\rho = 0.7$), we have run the BA to find its optimal solutions. The obtained objective values and variances are plotted in Fig. 5.

From Fig. 5, we can see that the variances reduce quite rapidly at the initial stage of iterations, and the transition from exploration to exploitation seems to occur between 35 to 60 iterations. Though the variances do not change much from iteration 60 onwards, the objective continues to improve/reduce gradually, which means that late iterations are mainly local exploitative moves. This characteristic is consistent with the theoretical results.

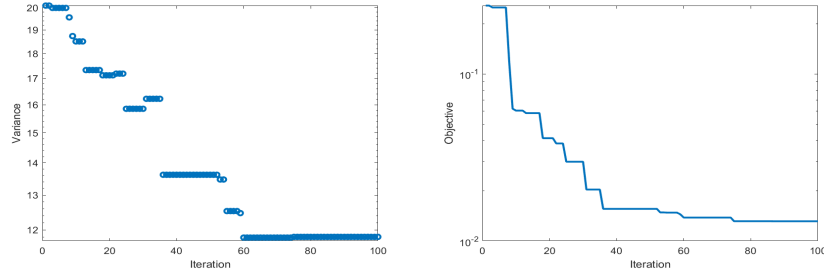


Fig. 5. The evolution of variances and spring design objective values for $\rho = 0.7$.

5 Probability of Acceptance

In the above analysis, we have considered only two parameters ρ and ω . In the BA, there is a switching probability between two branches based on the decision about the new solution. If a new solution is better, it will be accepted. If the solution is partially improved (i.e., not the new best solution but better than some of the old solutions in the population), it will be accepted by comparing with a random number. In essence, there is a probability p of acceptance for a given new solution.

In the rest of this paper, let us explore the possible influence of parameter settings on the effective acceptability p of new moves or solutions. For a new search move or a new solution, it will be accepted with a probability p and rejected with a probability of $1-p$. Alternatively, the solution in the new iteration will be the new solution with probability of p , while keeping the old solution x_i with a probability of $1-p$. Thus, equation (10) can be rewritten as

$$\begin{aligned} z_i &= (1-p) \underbrace{x_i}_{\text{old}} + p \underbrace{[x_i + \rho v_i + (g - x_i)\omega_i]}_{\text{new solution}} \\ &= (1-p)x_i + px_i + \rho p v_i + p g \omega \beta - p \omega \beta x_i, \end{aligned} \quad (44)$$

where we have used $\omega_i = \omega\beta$ with β being a uniform distribution.

In our derivations, we will need to use the properties of the variance of two uncorrelated random variables U and V with means μ_U and μ_V and variances σ_U^2 and σ_V^2 , respectively. The variance of their product can be calculated by

$$\sigma_{UV}^2 = \text{Var}(UV) = (\sigma_U^2 + \mu_U^2)(\sigma_V^2 + \mu_V^2) - \mu_U^2 \mu_V^2, \quad (45)$$

with the standard properties

$$\begin{aligned} \text{Var}(aU \pm bV) &= a^2 \sigma_U^2 + b^2 \sigma_V^2 \pm 2ab \text{cov}(U, V) \\ &= a^2 \sigma_U^2 + b^2 \sigma_V^2, \quad a, b \in \mathbb{R}, \end{aligned} \quad (46)$$

where we have assumed that U and V are uncorrelated. That is, $\text{cov}(U, V) = 0$ in the current context for the swarm population. Again as pointed out earlier, this assumption is an approximation.

Let σ_x^2 be the variance of x_i , σ_v^2 be the variance of v_i , and σ_z^2 be the variance of z_i . By taking the variance of Eq. (44), we have

$$\begin{aligned} \sigma_z^2 &= (1-p)^2\sigma_x^2 + p^2\sigma_x^2 \\ &+ p^2\rho^2\sigma_v^2 + p^2\omega^2\text{Var}(g\beta) + \omega^2p^2\text{Var}(x_i\beta). \end{aligned} \quad (47)$$

Here, we can assume that g and β are uncorrelated, and x_i and β are also uncorrelated.

The mean and variance of a uniform distribution $\beta \in [0, 1]$ are $\mu_\beta = 1/2$ and $\sigma_\beta^2 = 1/12$, respectively. The mean of g is g , and the variance of g is σ_g^2 , which can be considered as the same as σ_x^2 . Similarly, the mean and variance of x_i are μ_x and σ_x^2 , respectively. As the current best solution is usually obtained from x_i , we can approximate it as $g = \mu_x$. Thus, we have

$$\text{Var}(g\beta) = (\sigma_x^2 + g^2) \left[\frac{1}{12} + \left(\frac{1}{2}\right)^2 \right] - g^2 \left(\frac{1}{2}\right)^2 = \frac{\sigma_x^2}{12} + \frac{g^2}{12} + \frac{\sigma_x^2}{4}, \quad (48)$$

and

$$\begin{aligned} \text{Var}(x_i\beta) &= (\sigma_x^2 + \mu_x^2) \left[\frac{1}{12} + \left(\frac{1}{2}\right)^2 \right] - \mu_x^2 \left(\frac{1}{2}\right)^2 \\ &= \frac{\sigma_x^2}{12} + \frac{g^2}{12} + \frac{\sigma_x^2}{4}. \end{aligned} \quad (49)$$

Now Eq. (47) becomes

$$\sigma_z^2 = (1-p)^2\sigma_x^2 + p^2\sigma_x^2 + p^2\rho^2\sigma_v^2 + p^2\omega^2 \left[\frac{2\sigma_x^2}{3} + \frac{g^2}{12} \right]. \quad (50)$$

For simplicity, we can also assume that σ_v^2 is the same as that for x_i in the previous iteration (from Eq. (3)).

With the above assumptions and notations, the preceding equation becomes

$$\begin{aligned} \sigma_z^2 &= \left[(1-p)^2 + \frac{2p^2\omega^2}{3} + p^2 \right] \sigma_x^2 + p^2\rho^2\sigma_x^2 \\ &= \left[(1-2p+2p^2 + \frac{2p^2\omega^2}{3} + p^2\rho^2) \right] \sigma_x^2 + p^2\omega^2\frac{g^2}{12}. \end{aligned} \quad (51)$$

For many test functions or design benchmarks, the best solutions are either close to $x_* = 0$ or a known vector constant. Thus, we can consider the final term on the right-hand side of the above equation as a constant or zero in most cases.

Thus, the reduction of the variance and the convergence of the algorithm requires that

$$Q = 1 - 2p + 2p^2 + \frac{2p^2\omega^2}{3} + p^2\rho^2 \leq 1, \quad (52)$$

which means that

$$p \leq \frac{2}{2 + 2\omega^2/3 + \rho^2} = \frac{1}{1 + \omega^2/3 + \rho^2/2}, \quad (53)$$

where we have used the fact that $p > 0$. It is easy to see that this result implies that $0 < p \leq 1$ is always true.

For the special case of $\rho = 1$ and $\omega = 4$, we have

$$\rho \leq \frac{6}{41} \approx 0.146. \quad (54)$$

This is the maximum theoretical probability of accepting new solutions, but the true rate of acceptance of new moves can be lower in practice. The comparison of actual probability of acceptance is summarized in Table 1. As we can see, the theoretical values provide an upper bound for the true acceptance probability. Obviously, the true acceptance rate can depend on the ways of generating new solutions, the diversity of population, feasibility of the new solutions and the locality of the new solutions.

Table 1. Comparison of the actual acceptance probability with the theoretical limit.

Function	ρ	Maximum probability from theory	Actual Rate
f_1	0.9	0.365	0.140
	0.8	0.377	0.102
	0.7	0.388	0.091
f_2	0.9	0.365	0.307
	0.8	0.377	0.224
	0.7	0.388	0.258
Spring	0.9	0.365	0.076
	0.8	0.377	0.089
	0.7	0.388	0.115

6 Conclusion and Discussion

We have analyzed the evolution of variance for the bat algorithm in this work and have shown that the parameter ranges are consistent with the results from other theoretical analysis using dynamical system theory. In addition, the change in the variance reduction can also indicate a transition in the search mechanism from global exploration to local exploitation. The characteristics of variance evolution can be predicted well at the earlier stage of the iterations. This shows that the theoretical insights and results are correct.

We also have estimated the upper limit of the acceptance probability of new solutions in the population using variance analysis, and the results from numerical experiments are largely consistent with theoretical indications. The difference between the theoretical bounds and the actual acceptance rates means that the theory tends to over-estimate the probability. Thus, there is room for improvement to provide a tight bound. This can form a research topic for further studies.

Furthermore, the current work indicates that the transition timescale seems to be independent of the problems to be solved. In practice, there may be other factors that could affect the behaviour of a swarm population, such as population size, modality of the problem, and nonlinearity of the constraints as well as the initialization of the swarm population. This work can form the basis for further research.

Disclosure of Interests. The authors confirm that there are no relevant financial or non-financial competing interests to report.

No funding was received for this research work.

No datasets were received/used. All results are simulated and reproducible from the proposed method.

References

1. Aytug, H., Bhattacharyya, S., Koehler, G.J.: A markov chain analysis of genetic algorithms with power of 2 cardinality alphabets. *European Journal of Operational Research* **96**(1), 195–201 (1996)
2. Aytug, H., Koehler, G.J.: New stopping criterion for genetic algorithms. *European Journal of Operational Research* **126**(2), 662–674 (2000)
3. Beltrami, E.J.: *Mathematics for Dynamic Modeling*. Academic Press, San Diego, USA, second edn. (1998)
4. Benaroya, H., Han, S.M.: *Probability Models in Engineering and Science*. CRC Press, Boca Raton (2005)
5. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**, 281–305 (2012)
6. Boyd, S.P., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge UK (2004)
7. Cagnina, L.C., Esquivel, S.C., Coello Coello, A.C.: Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica* **32**(2), 319–326 (2008)
8. Chatterjee, A., Siarry, P.: Nonlinear inertia variation for dynamic adaptation in particle swarm optimization. *Comp. Oper. Research* **33**(3), 859–871 (2006)
9. Chen, S., Peng, G.H., He, X.S., Yang, X.S.: Global convergence analysis of the bat algorithm using a markovian framework and dynamic system theory. *Expert Systems with Applications* **114**(1), 173–182 (2018)
10. Clerc, M., Kennedy, J.: The particle swarm: explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* **6**(1), 58–73 (2002)
11. Eiben, A.E., Smit, S.K.: Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation* **1**(1), 19–31 (2011)
12. Gandom, A.H., Yang, X.S.: Chaotic bat algorithm. *Journal of Computational Science* **5**(2), 224–232 (2014)
13. Gandomi, A.H., Yang, X.S., Alavi, A.H., Talatahari, S.: Bat algorithm for constrained optimization tasks. *Neural Computing and Applications* **22**(6), 1239–1255 (2013)
14. Granas, A., Dugundji, J.: *Fixed Point Theory*. Springer-Verlag, New York (2003)

15. Grinstead, C.M., Snell, J.L.: Introduction to Probability. Americal Mathematical Society, Providence, Rhode Island, second edn. (1997)
16. Hekmatinia, A., Shanghooshabad, A.M., Motevali, M.M., Almasi, M.: Tuning parameter via a new rapid, accurate and parameter-less method using meta-learning. *International Journal of Data Mining, Modelling and Management* **11**(4), 366–390 (2019)
17. Joy, G., Huyck, C., Yang, X.S.: Parameter tuning of the firefly algorithm by three tuning methods: Standard monte carlo, quasi-monte carlo and latin hypercube sampling methods. *Journal of Computational Science* **87**, Article 102588 (2025)
18. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. pp. 1942–1948. IEEE, Piscataway, NJ, USA (1995)
19. Khamsi, M.A., Kirk, W.A.: An Introduction to Metric Space and Fixed Point Theory. John Wiley & Sons, New York (2001)
20. Oehlert, G.W.: A note on the delta method. *The American Statistician* **46**(1), 27–29 (1992)
21. Osaba, E., Yang, X.S., Diaz, F., Lopez-Garcia, P., Carballedo, R.: An improved discrete bat algorithm for symmetric and assymmetric travelling salesman problems. *Engineering Applications of Artificial Intelligence* **48**(1), 59–71 (2016)
22. Pan, F., Li, X., Zhou, Q., Li, W., Gao, Q.: Analysis of standard particle swarm optimization algorithm based on markov chain. *Acta Automatica Sinica* **39**(4), 381–389 (2013)
23. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm, technical note. Tech. rep., Cardiff University, Manufacturing Engineering Center, Cardiff (2005)
24. Rudolph, G.: Convergence analysis of canonical genetic algorithms. *IEEE Trans. Neural Networks* **5**(1), 96–101 (1994)
25. Storn, R., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization. *Journal of Global Optimization* **11**(4), 341–359 (1997)
26. Sudholt, D., Witt, C.: Runtime analysis of a binary particle swarm optimizer. *Theoretical Computer Science* **411**(21), 2084–2100 (2010)
27. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters* **85**(6), 317–325 (2003)
28. Wolter, K.M.: Introduction to Variance Estimation. Springer-Verlag, New York (1985)
29. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Cruz, C., González, J.R., Pelta, D.A., Terrazas, G. (eds.) *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)*. Studies in Computational Intelligence, vol. 284, pp. 65–74. Springer, Berlin, Germany (2010)
30. Yang, X.S.: Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computational Science* **46**, 101104 (2020)
31. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. *Engineering Computation* **29**(5), 464–483 (2012)
32. Zaharie, D.: Influence of crossover on the behavior of the differential evolution algorithm. *Applied Soft Computing* **9**(3), 1126–1138 (2009)
33. Zaharie, D., Micota, F.: Revisiting the analysis of the population variance in differential evolution algorithms. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 1811–1818. Donostia, Spain (2017). <https://doi.org/10.1109/CEC.2017.7969521>
34. Zdenek, D.: Optimal Quadratic Programming Algorithms: With Applications to Variational Inequalities. Springer, Heidelberg (2009)