Firefly optimized particle filter algorithm based on adaptive differential evolution

Ziwei Meng\textsuperscript{1,2,*}, Jinjie Mei\textsuperscript{1}, Chengjun Yu\textsuperscript{3}

\textsuperscript{1}Communication Teaching and Research department Air Force Early Warning Academy, Wuhan, Hubei,430019, China
\textsuperscript{2}31683 Unit of the Chinese People's Liberation Army, Lanzhou, Gansu, 730300, China
\textsuperscript{3}93534 Troop,Tianjin,310700,China
\*Corresponding author’s e-mail: mzw@freemail.kjld.mtn mzw054526@126.com

Abstract The improved particle filter algorithm based on fireflies reduces the particle diversity in the later iterations, which is easy to fall into the problem of local optimization. To solve the problem, after firefly position updated, we can integrate the differential evolution algorithm into it. The process can be described like this: firstly, the mutation process is guided by the particle weight adaptively; the crossover process is selecting individuals randomly according to the crossover probability; finally, the selection process takes the observation probability density function as the fitness value, and retains individuals with high fitness. After adding differential evolution, the diversity of individuals has increased and jumped out the process of local optimization. The overall quality of the particle swarm has been improved. Experiments show that the tracking accuracy of the improved algorithm has been improved, as well as the global optimization capabilities.

1. Introduction
Particle filter is a classic algorithm used to solve nonlinear system estimation and the noise can make non-Gaussian assumptions. After many calculations, the weights of a large number of particles show gradual decrease trade and the value of system state estimation is small but the calculation is large. The resampling process sacrifices particle diversity. Discarding small weight particles and sampling large-weight particles multiple times, which will reduce the particle diversity inevitably.

Regarding particles as individual organisms, by simulating the movement characteristics of species in nature, we can use swarm intelligence optimization ideas to guide the movement of particles to optimize their distribution, making it closer to the real state of the system. The combination of swarm intelligence optimization ideas and particle filtering as a new research direction has attracted the attention of experts and scholars continuously. Literature [1] regards the fitness value in the genetic algorithm as the particle weight, retains the high fitness particles, and implements adaptive crossover and mutation operations on the poor fitness particles to make them evolve. The new particle set consists of high fitness particles and The combination of evolutionary particles increases the diversity of particles to a certain extent. Literature [2] is based on the standard fruit fly optimization algorithm. It optimizes the particle distribution through the crossover operation in the genetic algorithm, and uses the Cauchy jitter to mutate adaptively to make it jump out of the extreme value, which solves the problem of particle precocity effectively. Literature [3] combines the improved butterfly algorithm with particle filtering. It introduces the latest measurement information into the butterfly fragrance formula, then lets the
attraction radius parameter to control the search step, finally uses a smaller number of particles to achieve more accurate state estimation. The bats\cite{4} can adjust the pulse, frequency, and loudness to find prey, which can bring into particle filter to guide the particles to search for the optimal solution in space. In order to avoid falling into a local extreme value, the levy distribution is added to improve the estimation accuracy of the system state. The above-mentioned swarm intelligence optimization algorithms are all dedicated to solving the problem of particle diversity reduction and balancing global and local optima.

Based on the above research, this paper proposes an adaptive differential optimization firefly algorithm, which is applied to the particle filter process to solve the problem that the decrease of sample diversity in the basic firefly algorithm lead to the local optimal solution. The last of this paper, a nonlinear system is used to verify the effectiveness of the improved algorithm.

2. Particle filter algorithm

The essence of particle filter\cite{5} algorithm is to sample many weighted particles from the posterior probability density function to form a particle set. The particle set is used to approximate the posterior probability distribution of the target stage. This process mainly converts the integral into a summation method, namely:

\[
p(x_k | z_{i,k}) \approx \sum_{i=1}^{N} w^i_k \delta (x_k - x^i_k)
\]

In the formula, \(x^i_k\) is the particle \(i\) sampled from the prior distribution at time \(k\), and \(w^i_k\) is the weight of the particle \(i\) at time \(k\) obtained by recurrence from the observation function. \(N\) is the number of particles.

3. Firefly algorithm

Firefly Algorithm\cite{6} (FA) is a heuristic swarm intelligence optimization algorithm. It was proposed by Cambridge scholar XIN-She Yang based on the glowing behavior of fireflies.

Assuming that there are \(N\) fireflies in the space, the position of the fireflies represents the solution of the objective function, which can be expressed as a dimensional vector \(x_i = (x_{i1}, x_{i2}, ..., x_{id})\), \(i = 1, 2, ..., n\) and it is \(d\)-dimensional vector.

The relative fluorescence brightness of fireflies:

\[I = I_0 \times e^{-r \gamma} \]

Mutual attraction of fireflies:

\[\beta(r) = \beta_0 e^{-r \gamma} \]

Firefly location update:

\[x_i' = x_i' + \beta \left( g value_{x_k} - x_i' \right) + \alpha \left( rand - \frac{1}{2} \right) \]

Objective function:

\[I = \exp \left[-\frac{1}{2R_k} \right](z_{new} - z_{pred}(i)) \]

4. Improved firefly algorithm based on differential evolution

In order to increase the diversity of the firefly population and make the algorithm accomplish the optimal search in the whole space, after the firefly position is updated, the mutation, crossover, and selection process in the differential evolution algorithm are introduced.

1 Mutation
\begin{equation}
 v_i(g+1) = x_i(g) + F \left( x_{i2}(g) - x_{i3}(g) \right)
\end{equation}

\(x_i\) and \(v_i\) represent the original population and the mutation population, respectively. \(F\) is the scaling factor, \(F=1-\tilde{w}_i\). \(\tilde{w}_i\) is the normalized weight after sampling from the importance function at time \(k\).

When the weight is large, the influence of the difference term is small, and the individual to be mutated remains unchanged basically. When the weight is small, it moves in the space through mutation. \(g\) represents the algebra of variation, and the three individuals are generated randomly.

2 Crossover

\begin{equation}
u_{i,j}(g+1) = \begin{cases}
v_{i,j}^{x=1} & \text{rand}(j) \leq CR, or \ j = \text{rand}(j) \\
 x_{i,j}^{x=0} & \text{rand}(j) > CR, or \ j \neq \text{rand}(j)
\end{cases}
\end{equation}

\(i = 1, 2, ..., N, \ j = 1, 2, ..., D,\ \text{rand}(j)\) means randomly generated parameters between \([0,1]\) and \(CR\) is the crossover probability. \(\text{random}(j) \in [1, 2, ..., D]\) ensures that at least one dimensional vector comes from the mutated population, ensuring that a new population is added after each crossover.

3 Selection

Whether the test population can be selected as the next-generation population depends on the population fitness function. Taking the observation probability density function as the fitness function, the selection process can be expressed by the following formula:

\begin{equation}
x_i(g+1) = \begin{cases}
u_i(g+1), p(y_k | u_i(g+1)) > p(y_k | x_i(g)) & \text{otherwise}
\end{cases}
\end{equation}

The firefly population after differential mutation is richer in population diversity than its own iteration. Keeping well-adapted individuals not only improves the population quality but also speeds up the subsequent optimization effect of fireflies.

5. Algorithm steps

The steps of the modified adaptive firefly particle filter algorithm (AMFA-PF) proposed in this paper are as follows:

Step 1. Initialize the particles

At this initial moment in time, sample \(N\) particles \(\{x_{i0}, i = 1, \cdots, N\}\) from the importance density function as the initial particles:

\(x_i - q(x_i | x_{i-1}, z_i) = p(x_i | x_{i-1})\)

Step 2. Calculate weights and normalize weights

\(w'_i = w_{i-1} p(z_i | x'_i)\)

Step 3. Use the firefly algorithm for the particles at this moment to find the brightest firefly, calculate its objective function, update the position of the firefly with the position update formula, and calculate the current brightness of the firefly.

Step 4. When the number of internal iterations or the threshold is reached, the firefly population uses the formula to perform mutation, crossover and selection operations.

Step 5: Calculate and normalize the individual weights of fireflies after population mutation.

Step 6: Status output
6. Simulation experiment and analysis

6.1 Experimental parameter settings

Table 1. Algorithm parameter settings

| Parameter description     | Parameter size |
|---------------------------|----------------|
| Number of fireflies $N$  | 100            |
| The maximum number of iterations $k$ | 30             |
| Firefly attraction $\beta_0$ | 0.8            |
| Light absorption coefficient $\gamma$ | 1              |
| Randomness parameter $\alpha$ | 0.5            |
| Crossover probability $CR$ | 0.9            |

The calculation formula of the root mean square error RMSE at time $k$:

$$RMSE = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (x_i^* - \hat{x}_i^*)^2}$$

This paper selects a univariate growth model as a nonlinear system to test the effective performance of several algorithms. The state equation and measurement equation are:

Equation of state:

$$x(t) = 0.5x(t - 1) + \frac{25x(t-1)}{1 + [x(t-1)]} + 8\cos[1.2(t-1)] + w(t)$$

Measurement equation:

$$z(t) = \frac{x(t)^2}{20} + v(t)$$

In the formula, $w(t)$ and $v(t)$ are process noise and measurement noise respectively, both of which are Gaussian noise with zero mean value. Using particle filtering and improved method estimates the system, in which the noise variance $R=2$ is measured, and the filter state value and error value are observed by changing the value of the particle number $N$ and process noise variance $Q$.

6.2 Analysis of experimental results

1. Number of particles $N = 20$, variance $Q = 2$
2. Number of particles $N = 50$, variance $Q = 2$

![Figure 3. Filter estimated value](image1)

![Figure 4. Absolute value of error](image2)

Table 2. Root Mean Square Error comparison

| Algorithm parameter | PF     | FA-PF  | DEFA-PF |
|---------------------|--------|--------|---------|
| $N = 20, Q = 2$     | 6.4512 | 4.6872 | 2.6722  |
| $N = 50, Q = 2$     | 5.5836 | 4.4732 | 2.3411  |
| $N = 20, Q = 10$    | 7.7255 | 5.4732 | 1.6422  |
| $N = 50, Q = 10$    | 6.7634 | 4.7625 | 0.8214  |

Table 3. Algorithm running time comparison

| Algorithm parameter | PF     | FA-PF  | DEFA-PF |
|---------------------|--------|--------|---------|
| $N = 20, Q = 2$     | 0.1275 | 0.1438 | 0.1577  |
| $N = 50, Q = 2$     | 0.1475 | 0.1613 | 0.1764  |
| $N = 20, Q = 10$    | 0.1296 | 0.1476 | 0.1585  |
| $N = 50, Q = 10$    | 0.1514 | 0.1652 | 0.1796  |

The fireflies improved by the DE algorithm have better tracking results when used in particle filtering. The main reason is that the DE algorithm makes the fireflies population mutate, which increases the diversity of individuals, and the individuals with better fitness are retained. After multiple iterations later, it is closer to the real state of the system. With the increase in the number of particles and the complexity of the algorithm, the running time is also getting longer.

7. Conclusion

In this paper, the particle filter is improved with the intelligent optimization algorithm. The improved algorithm increases the diversity of particles, makes the particle optimization process jump out of the local optimal solution, improves the accuracy of system state estimation, and can be applied to specific problems such as target tracking, which has certain application value.

References

[1] DU Z, Deng X. (2017) Particle filter based on adaptive genetic algorithm. Journal of Chengdu University of Technology.2017(05):1617-9727.

[2] Han K, Zhang H . (2018) Improved particle filter based on Drosophila optimization algorithm and its application in target tracking. Journal of Hunan University.45(10):1674-2974.

[3] Zhang W, Guo M, He Y, Sun X, Zhu D (2019) An improved butterfly algorithm to optimize particle filter algorithm. Journal of Xi'an University of Science and Technology. 39(01): 1672 -9315.

[4] Chen Z, Tian M, Wu P. (2017) Research on Particle Filtering Method Based on Bat Algorithm. Journal of Physics.66(05):500-502.

[5] Zan M, Zhou H, Han D.(2019) Overview of Particle Filter Target Tracking Algorithms.55(5):8-17

Lin Q, Wang X. (2017) Target tracking based on improved particle filter algorithm. Information Technology.2017(10):88-92.

[6] Tian M, Bo Y, Chen Z. (2016) Firefly algorithm intelligently optimizes particle filtering. Acta Automatica Sinica, 42(1): 89-97