The Improvement of Particle Swarm Optimization: a Case Study of Optimal Operation in Goupitan Reservoir

Haichen Li1, Tao Qin1, Weiping Wang1, Xiaohui Lei1, and Wenhui Wu2

1State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing, 100038, China
2Guangzhou Municipal Institute of Water Affairs Investigation and Design, Guangzhou, 510610, China
374779007@qq.com

Abstract. Due to the weakness in holding diversity and reaching global optimum, the standard particle swarm optimization has not performed well in reservoir optimal operation. To solve this problem, this paper introduces downhill simplex method to work together with the standard particle swarm optimization. The application of this approach in Goupitan reservoir optimal operation proves that the improved method had better accuracy and higher reliability with small investment.

1. Introduction

The multidimensional non-linearity of optimal dispatching problems on reservoir makes its resolution with traditional algorithms easy to be caught in curse of dimensionality. Since 1990s, along with the development of computer technology and artificial intelligence technology, intelligent algorithms have begun to be gradually applied to solve the reservoir optimal dispatching problems due to its characteristics of non-differentiability, fast solution, etc. However, there are some problems with intelligent algorithms due to their large randomness as well as weak mathematical theoretical basis. Intelligent optimization algorithm refers to meta-heuristic algorithms based on population, such as particle swarm algorithm [1], big bang algorithm [2], big flock algorithm [3], etc. Most of these algorithms refer to making some simple and idealized constraints based on the abstraction of natural phenomena or biological laws, behaviors, characteristics, etc., and establishing corresponding mathematical operators to describe the reasoning of intelligent population, which is the way multiple individuals collaboratively work together to find optimal solution in order to achieve the resolution of optimization problems.

New intelligent algorithms have always been proposed, including the early proposed genetic algorithm [4], particle swarm algorithm [1], differential algorithm [5] and bacterial foraging algorithm [6], as well as the recently proposed firefly algorithm [7], cuckoo algorithm [8], big bang crunch algorithm [2], water cycle algorithm [9], spider optimization algorithm [10] and so on.. Although each kind of intelligent algorithms has its own characteristics and focus, they all share the common ground of randomness and optimal selectivity, where randomness corresponds to population diversity and can strengthen particle-searching capability, while optimal selectivity corresponds to global convergence.

Researches on improving intelligent algorithms have also been ongoing. For example, as an intelligent optimization algorithm, particle swarm optimization algorithm shows great advantage in
resolution accuracy and efficiency, and it can solve some problems. However, there are some problems in the application of standard particle swarm algorithm in dispatching reservoir optimal, which is specifically manifested in the fact that the diversity is unknown when particles are randomly generated. When updating all particles in the population, the location information of individual optimal value and global optimal value are used for reference, which is of strong convergence. This restricts particle search capability to a certain extent and leads to great loss in population diversity, which eventually causes algorithm prematurity and fall into local optimum.

Currently, the improvement works on swarm intelligence algorithm are mainly divided into 3 categories. The first is to prove the global convergence of the optimal solution based on mathematical theories [11-14]. The second is to improve the calculation parameters in the algorithm to avoid falling into local optimum [15-19]. And the third is to improve the search method of the algorithm, i.e. improvement of the iterative updating formula in particle swarm algorithm [20, 21]. Based on the improvement of algorithm search method by dividing the original population into several simplex groups, in this paper the population in the group is continuously carrying particle substitution according to the downhill simplex method. Meanwhile, the optimal value of the group serves as the local optimal value for constant carrying of particle update, in order to improve solution accuracy and convergence speed of standard particle swarm algorithm, and provide a certain degree of reference for solving of the problems on reservoir optimal dispatching.

2. Improvement of particle swarm algorithm

2.1. Basic ideas for particle swarm algorithm improvement

Particle Swarm Optimization (PSO) algorithm is a type of swarm intelligence optimization algorithms which was proposed by Dr. Kennedy and Dr. Eberhart in 1995 [1]. Also, the downhill simplex method is the new simplex search method designed by Nelder and Mead based on the works of Spendley and others. The method is a local search algorithm for solving unconstrained optimization problems, and does not need any derivative information of the target function.

In order to improve the local search capability of the PSO algorithm, this paper embeds the downhill simplex method into PSO algorithm to introduce the particle swarm optimization algorithm based on downhill simplex method (SDPSO). The original intention of the improved algorithm is to combine the advantages of both, to enhance the capability of particle swarm algorithm to escape from local optimization, and to effectively avoid the shortcomings of PSO algorithm on fast loss of diversity, in order to greatly improve algorithm global convergence. According to the basic principle of the above improved particle swarm algorithm, the flow chart of solving optimization problems is shown in figure 1, and the specific steps are as follows:

1. Initialization of parameters. Various parameter values, iterations, particle population size, number of simplex vertices (num) is determined, where the number of simplices L refers to the ratio of population size and number of simplex vertices, and the particle group is distributed randomly into simplex.

2. Calculation of particle fitness value and selection of local and global optimal values. The particle local optimal value within the simplex generally is the particle maximum fitness value within the simplex.

3. Update of particle speed and location. According to the particle fitness value, the particle $X_{\text{worst}}^k(l)$ with worst performance in each simplex is substituted by $X_{\text{new}}^k(l)$, the speeds and locations of other particles is still updated in accordance with the improved formula of standard particle swarm algorithm.

$$X_{\text{centroid}}^k(l) = \frac{\sum_{\text{num}} X_{\text{new}}^k(l)}{\text{num}}$$  \hspace{1cm} (1)

$$X_{\text{new}}^k(l) = X_{\text{worst}}^k(l) \pm (X_{\text{centroid}}^k(l) - X_{\text{worst}}^k(l))$$  \hspace{1cm} (2)

$$X^{k+1} = X^k + \text{speed}(k + 1)$$  \hspace{1cm} (3)
speed\((k + 1) = w \times \text{speed}(k) + C_1(x_{\text{global optimum}}^k - X^k) + C_2(x_{\text{local optimum}}^k - X^k) \) (4)

In the above formula, \(k\) refers to iteration number, and \(l\) refers to simplex number.

(4) Step (2) and (3) are repeated until the maximum iteration and iteration termination conditions are reached.

(5) Optimal solution output.

![Figure 1. Flow Chart of the Solving of Optimization Problems of Improved Particle Swarm Algorithm](image)

2.2. Test of standard function

To verify the effectiveness of improved particle swarm algorithm, the improved particle swarm algorithm (IPSO) is used for optimization problems of standard function, and the parameter setting is shown in Table 1.

| Algorithm | Parameter value          |
|-----------|--------------------------|
| PSO       | \(c_1 = c_2 = 1.5, w = 1\) |
| IPSO      | \(sdnum = 10, c_1 = c_2 = 1.5, w = 1\) |

This paper uses the most common 10 standard test functions [23, 24], as shown in Table 2. In addition, the basic particle swarm algorithm (PSO) was deemed as a contrasting algorithm. In all the test functions, the maximum iteration and independent running times, the population scale and other common parameters of both the particle swarm algorithm and the improved particle swarm algorithm remain consistent, wherein, the iteration and independent running times of the algorithm is 10,000 and 200, respectively.
Table 2. Table of standard test function

| Question | Boundary | Optimal value |
|----------|----------|---------------|
| F1 = $x_1^2 + 1000000 \sum_{i=2}^{N} x_i^2$ | [-100,100] | 0 |
| F2 = $1000000 x_1^2 + \sum_{i=2}^{N} x_i^2$ | [-100,100] | 0 |
| F3 = $\left( \sum_{i=1}^{N} x_i^2 \right)^{2} - \left( \sum_{i=1}^{N} x_i \right)^{2} + 0.5 \sum_{i=1}^{N} x_i^2 + \sum_{i=1}^{N} x_i^2 + 0.5$ | [-100,100] | 0.5 |
| F4 = $-20 \exp \left( -0.2 \left( \frac{1}{N} \sum_{i=1}^{N} x_i^2 \right) \right) + 20 + e$ | [-32,32] | 0 |
| F5 = $\frac{1}{4000} \sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$ | [-600,600] | 0 |
| F6 = $\sum_{i=1}^{N} x_i^2$ | [-100,100] | 0 |
| F7 = $\sum_{i=1}^{N} \left( x_i + 0.5 \right)^2$ | [-100,100] | 0 |
| F8 = $\sum_{i=1}^{N} \left| x_i \right| + \prod_{i=1}^{N} \left| x_i \right|$ | [-1,1] | 0 |
| F9 = $\sum_{i=1}^{N} x_i^2 + \sum_{i=1}^{N} 0.5 x_i^2$ | [-5,10] | 0 |
| F10 = $\sum_{i=1}^{N} \left( i x_i^2 \right)$ | [-5,12,5,12] | 0 |

The statistical results of solving the 10-optimization functions with the algorithm are listed in table 3. From the table, it can be seen that except the basic particle swarm algorithm, which can obtain the equivalent optimized results as the improved particle swarm algorithm for the 7th function, the optimal results obtained with the improved particle swarm algorithm are better than the standard particle swarm algorithm, which shows the advantages for improving the particle swarm algorithm.

Table 3. Table of statistical results for solving standard test function in two algorithms

| Question | Algorithm | Optimal value | Average value | Worst value | Variance |
|----------|-----------|---------------|---------------|-------------|----------|
| F1       | PSO       | 0             | 1300          | 10000       | 339.7    |
|          | IPSO      | 0             | 0             | 0           | 3.65E-106|
| F2       | PSO       | 0             | 0             | 0           | 4.67E-33 |
|          | IPSO      | 0             | 0             | 0           | 3.47E-110|
| F3       | PSO       | 0.44922       | 0.28877       | 1.03866     | 0.012086 |
|          | IPSO      | 0.4999        | 0.50719       | 0.79963     | 0.00879  |
| F4       | PSO       | 0             | 0             | 0           | 1.31E-16 |
|          | IPSO      | 0             | 0             | 0           | 6.21E-17 |
| F5       | PSO       | 0             | 0             | 0           | 5.37E-08 |
|          | IPSO      | 0             | 0             | 0           | 0        |
| F6       | PSO       | 0             | 0             | 0           | 1.90E-35 |
|          | IPSO      | 0             | 0             | 0           | 1.27E-107|
| F7       | PSO       | 0             | 0             | 0           | 0        |
|          | IPSO      | 0             | 0             | 0           | 0        |
| F8       | PSO       | 0             | 0             | 0           | 4.79E-60 |
|          | IPSO      | 0             | 0             | 0           | 8.66E-112|
| F9       | PSO       | 0             | 0             | 0           | 6.58E-11 |
|          | IPSO      | 0             | 0             | 0           | 2.41E-68 |
| F10      | PSO       | 0             | 0             | 0           | 2.13E-37 |
|          | IPSO      | 0             | 0             | 0           | 3.42E-118|

3. Reservoir optimal dispatching model
Objective function:

\[ E = \max \sum_{t=1}^{n} N_t T \]  

where, \( E \) refers to reservoir power output, \( n \) refers to calculated number of time interval; \( t \) refers to calculated time interval number; \( N_t \) refers to reservoir output during \( t \) time interval; and \( T \) refers to duration within \( t \) time interval.

Constraint conditions:

Water balance constraint:

\[ V_{t+1} = V_t + (q_t - Q_t)T \]  

Water level constraint:

\[ Z_t^{\text{min}} \leq Z_t \leq Z_t^{\text{max}} \]  

Minimum output constraint:

\[ N_t^{\text{min}} \leq N_t \leq N_t^{\text{max}} \]  

Discharge volume constraint:

\[ Q_t^{\text{min}} \leq Q_t \leq Q_t^{\text{max}} \]  

Where, \( V_{t+1} \) refers to the reservoir impoundage at the end of \( t \) time interval; while \( V_t \) refers to the reservoir impoundage at the beginning of \( t \) time interval; \( q_t \) refers to the reservoir impoundage at the beginning of \( t \) time interval; \( Z_t^{\text{min}} \) refers to minimum reservoir water level required to be assured; \( Z_t \) refers to period water level; \( Z_t^{\text{max}} \) refers to reservoir allowable maximum water level; \( N_t^{\text{min}} \) and \( N_t^{\text{max}} \) refer to reservoir minimum and maximum allowable output respectively; \( Q_t^{\text{min}} \) and \( Q_t^{\text{max}} \) refer to minimum and maximum discharge volume respectively; \( Q_t \) refers to period discharge volume.

4. Case study

Goupitan reservoir is located on the main stream of Wujiang River within the territory of Yuqing County, Guizhou Province, and is the 7th level of the main stream of Wujiang River. The area of controlled catchment above the dam site of Goupitan reservoir is 43,250 km², accounting for 49% of the total catchment. The dam site average flow and runoff volume for many years is 717 m³/s and 22.6 billion m³. The reservoir normal impounded level is 630 m, with the dead water level of 590 m. what’s more, the operation of the water level for maximum power output during flood season of June to July is controlled according to flood control restrained level of 626.24 m, while the operation of the water level for maximum power output in August is controlled according to flood control restrained level of 628.12 m. The power output transmitting follows the flood control dispatching. When the Goupitan power station is under independent operation, it has annual regulating capability; and when it is under unified operation with the upstream reservoir, it has regulating capability for many years. Besides, Goupitan power station is a landmark project of “transmitting electricity from Guizhou to Guangdong” and is the backbone support of power supply point for the “West-East electricity transmission project”.

4.1. Algorithm solving steps

(a) Initialize population, individuals in the population correspond to the actual value of reservoir capacity, namely:

\[ V_{t+1} = V_{\text{dead}} + r \ast (V_{lt} + \text{inf}low_{t+1} \ast \alpha - V_{\text{dead}}) \]  

(b) Calculate population fitness function:

\[ E_t = e_t - \beta \ast \gamma \ast \min(0, N_t - N_{\text{min}}) \]  

(c) To keep diversity, update individuals in accordance with equation (1), equation (2), equation (3), and equation (4) to get new generation of population.

(d) Determine whether the convergence condition is met (convergence condition in this paper refer to iteration gene). If it is satisfied, the optimal solution is output, otherwise, it turns to step (b) for restart.

Where, \( \alpha \) refers to flow capacity conversion coefficient; \( \beta \) refers to the penalty coefficient, which is defined as 10 in this paper; and \( \gamma \) refers to output power conversion coefficient.
4.2. Results and analysis

Taking Goupitan as an example, the standard and improved particle swarm algorithm are used to calculate power output capacity for typical years under different inflow frequencies by taking a month as the time interval.

During the solving process, 100, 200, 300 are respectively used for population size. Regarding to number of iterations, after many times of experimental results analysis, this study has found out that it is the economic and reasonable to set the number of iterations as 500 whether it’s from the aspect of solving efficiency or investment. The optimizing results of annual power output capacity of Goupitan reservoir under circumstances with different population sizes are listed in table 4, and average values of all the results are 10 times of the operation results.

| Year | Annual runoff (m³/s) | Inflow frequency | Annual power output capacity (hundred million Kwh) |
|------|----------------------|------------------|---------------------------------------------------|
|      | popsize=100 | popsize=200 | popsize=300 | popsize=100 | popsize=200 | popsize=300 |
| 2007 | 367        | >99%   | 44.3 | 50.86 | 48.4 | 51.04 | 49.7 | 51.04 |
| 1967 | 436        | 90-95% | 51.09 | 59.22 | 55.43 | 60.9 | 59.14 | 61.1 |
| 1973 | 614        | 70-75% | 67.75 | 86.2  | 76.99 | 86.7 | 82.76 | 86.6 |
| 1999 | 710        | 45-50% | 80.56 | 95.51 | 89.86 | 96.78 | 93.97 | 96.78 |
| 1984 | 800        | 20-25% | 92.33 | 111.8 | 101.67 | 112.4 | 109.61 | 112.9 |
| 1965 | 929        | 5-10%  | 104.59 | 128.04 | 114.86 | 128.04 | 121.7 | 127.64 |

From the above table, taking Goupitan as an example, under different typical years and the conditions of same population size and iteration, the power output capacity under the improved particle swarm algorithm is significantly higher than the standard algorithm, and the IPSO algorithm achieves better optimization results, with higher solution accuracy. However, along with increasing of population size and iteration, the additional issuance margin of the improved particle swarm algorithm is far less than the standard particle swarm algorithm, which indicates that the improved algorithm had a higher solving efficiency, and has better optimization effect under the circumstance with smaller economic input. Regarding to the reservoirs, the drier the inflow, the harder to meet each dispatching target, but for year with low water like 2007, the annual power output capacity optimized with the improved particle swarm algorithm is still 134 million Kwh, with the increasing of 2.75% more than that generated by the standard particle swarm algorithm.

The population size has a certain effect on the optimization of algorithm. Generally, the larger the population, the closer to the global convergence the results are. However the larger the population, the more time is spent, and after the population size exceeds 200, the calculation time of improved particle swarm algorithm increases significantly with an equivalent optimization effect, and even there is a situation of reduction in power output capacity.

Based on the above two points, for Goupitan, the optimization effect of improved particle swarm algorithm is better when the population size is 200 and iteration is 500. In the following sections, the optimization results obtained before and after algorithm improvement are analyzed by taking 2007 as an example. The optimization result variations under different algorithms along with increase in iterations are shown in figure 2. To better understand the solution results of the model, comparisons are made for reservoir dispatching decision process under different solving algorithms from the aspects of reservoir level, period output and reservoir in-out flow, and the calculation results of the two algorithms are listed in table 5.
From figure 2, it can be seen that it is easy for the standard particle swarm algorithm to fall into local optimum, and therefore, it does not possess the ability to jump out of local optimum. On the other hand, the improved particle swarm algorithm maintains particle diversity by retaining particle substitute and update in the simplex, thus increasing the algorithm robustness, restraining the “prematurity” phenomena of standard particle swarm algorithm. Therefore, the improved particle swarm algorithm possesses the ability to jump out of local optimum, and as a result, it improves the optimization results.

From table 5, based on period output, it can be seen that 2007 was of low water, but the standard and improved particle swarm algorithms are able to meet requirements of minimum output in the corresponding periods, which ensures that reservoir operation is reliable. From the reservoir level rising process, falling process, and reservoir outflow, the drainage process of standard particle swarm algorithm is a bit more homogeneous, while the improved particle swarm algorithm maintains generating power at high water level after flood season of August, where the power output capacity of 5.104 billion Kwh for the whole year is significantly higher than 4.970 billion Kwh obtained from the optimization with standard particle swarm algorithm. Therefore, from the economic point of view, the improved particle swarm algorithm achieves better power output efficiency.
Based on the above analysis, the improved particle swarm algorithm not only has fast convergence speed, but it could meet requirements of achieving better power output efficiency with minimum input, and thus, it integratedly has good solution accuracy and solution reliability.

5. Conclusion and suggestion
This paper embeds the idea of downhill simplex method to propose the improved particle swarm algorithm based on standard particle swarm algorithm, and applies it in power output dispatching of Goupitan reservoir. The application results showed that under different points of time in a year, the improved algorithm not only has fast convergence speed, but also meets the requirements of achieving better power output efficiency with minimum input. Therefore, this algorithm integratedly has good solution accuracy, solution reliability and economic feasibility, and it provides the possibility of successfully solving optimal dispatching problems of complex and multi-dimensional reservoirs.

Acknowledgment
This work was supported by the National Key Research and Development Project of China, under grant No. 2016YFC0402208, No. 2016YFC0401903, and No. 2016YFC0402204. The authors also thank anonymous reviewers for their helpful comments and suggestions.

References
[1] Eberhart R and Kennedy J 1995 Proc. of the Sixth International Symposium on Micro Machine and Human Science(Nagoya: Nagoya Municipal Industrial Research Institute) pp 39–43
[2] Kaveh A and Talatahari S 2010 Acta Mechanica 213 267-89.
[3] Meng X B, Gao X Z, Lu L, et al 2016 Journal of Experimental & Theoretical Artificial Intelligence 28 673-87
[4] Coello C A C. and Montes E M 2002 Advanced Engineering Informatics 16 193-203
[5] Das S and Suganthan P N 2011 IEEE Transactions on Evolutionary Computation 15 4-31
[6] Wang Q L, Gao X Z and Wang C H 2010 International Journal of Innovative Computing, Information and Control 6 1-10
[7] Gandomi A H, Yang X S and Alavi A H 2011 Computers and Structures 89 2325-36
[8] Gandomi A H, Yang X S and Alavi A H 2013 Engineering with Computers 29 17-35
[9] Eskandar H, Sadollah A, Bahreininejad A, et al. 2012 Computers and Structures 151-66
[10] Cuevas E and Cienfuegos M 2014 Expert Systems with Applications 41 412-25
[11] Shang G, Kezong T, Xinzi J and Jingyu Y 2006 Science Technology and Engineering 6 1625-7
[12] Shouyi Y and Suqiong K 2010 Control Theory & Applications 27 843-8
[13] Zhiyong L, Liang M and Huizhen Z 2013 Mathematics in Practice and Theory 43 182-90
[14] Jianfeng Q 2014 Improved method and convergence theory of artificial bee colony algorithm (Anhui:Anhui university)
[15] Shuanghu Z, Qiang H, Hongshou W and Juxiang Y 2007 Journal of Hydroelectric Engineering 26 1-5
[16] Spendley W, Hext G R, Himsworth F R 1962 Technometrics 4 441-61