Journal of Network Intelligence ©2021 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **6**, Number **3**, August **2021**

Optimal Operation with Parallel Compact Bee Colony Algorithm for Cascade Hydropower Plants

Xianbao Cheng

School of Electronics and Information Engineering
Beibu Gulf University
No.12, Binhai Avenue, Qinzhou, Guangxi, P.R.C
luyu1233@foxmail.com

Yu Jiang

School of Electronics and Information Engineering
Beibu Gulf University
No.12, Binhai Avenue, Qinzhou, Guangxi, P.R.C
63131758@qq.com

Dan Li

School of Electronics and Information Engineering
Beibu Gulf University
No.12, Binhai Avenue, Qinzhou, Guangxi, P.R.C
757527741@qq.com

Zhaomin Zhu

School of Electronics and Information Engineering
Beibu Gulf University
No.12, Binhai Avenue, Qinzhou, Guangxi, P.R.C
zhuzhaomin@gmail.com

Ning Wu*

School of Electronics and Information Engineering
Beibu Gulf University
No.12, Binhai Avenue, Qinzhou, Guangxi, P.R.C
1323677063@qq.com

Received March 2021; Revised May 2021

Abstract. *The optimal operation of cascade hydropower plants is a high-dimensional, multi constrained and nonlinear problem. In order to improve the efficiency of water resource utilization, an accurate model is required to reflect the actual optimal scheduling problem. This research proposed an improved artificial bee colony algorithm based on parallel compact technology for the power generation problem of cascade hydropower plants. The improved algorithm is tested on several selected functions, and it achieves better convergence speed and calculation accuracy with less memory compared with other optimization algorithms.*
Keywords: cascaded hydropower, artificial bee colony, parallel compact technology, management optimization

---

1. **Introduction.** Intelligent algorithms are usually used to solve all kinds of non-linear and non-differentiable mathematical problems as well as practical problems. However, there is no perfect optimization algorithm so far, and various approaches are being improved to solve the increasingly complex problems.

The Artificial Bee Colony (ABC) algorithm is a type of swarm intelligence model, which simulates the behavior of bee colony foraging and can be used to effectively solve complex optimization problems [1-4]. The ABC scheme includes four elements such as food source, employed bees, onlookers and scouts. A food source is a feasible solution of the optimization problem, and the quality of a food source represents the fitness of this solution. In the ABC algorithm, the number of employed bees is equal to the number of food sources. The task of the employed bee is to find the food source information and share it with the onlookers with a certain probability which is the selection strategy calculated by roulette according to the fitness of the food source. Onlookers select food sources according to the nectar amount told by the employed bees in the search area of the hive and scouts search for new food sources near the hive. During the search for new food sources the onlookers make greedy selection based on the information given by the employed bees. If a food source has not been updated after a certain number of iterations, the employed bee for this food source will become a scout and starts to look for a new food source.

The ABC scheme has been used in many applications, such as telecommunication, signal and image processing, neural networks, data mining, control, NP class problems, traffic problems, multi-objective problems and etc. [5-13]. In many application scenarios, it is often necessary to process and store a large amount of data in order to calculate each of the food sources, and high-performance computing is also required. However, high power computing environment is not always available in real world applications especially in portable applications. Compact Evolutionary Algorithm (CEA) takes up less memory and only one initial variable is required, so the computation demand can be significantly reduced. Compact Genetic Algorithm (CGA) [14] only optimizes one target variable and the optimization accuracy is similar to the original genetic algorithm. It was shown that two strategies can be combined for the particle swarm to achieve better performance [15]. A compact version of the ABC scheme has also been proposed to reduce the memory consumption and increase the convergence speed [16]. However, the compact ABC scheme only generates one solution in each of the iterations and therefore the stability of solution cannot be guaranteed. This paper proposes an improved Compact ABC scheme with multi group parallel compact artificial bee colony. In this scheme, the optimization process is divided into several groups each of which communicates through a certain strategy, and after a number of iterations, their solution is fused to a single one. Experimental results will be shown in the sections below.

The optimization of power generation efficiency and total output for a cascade hydropower plant is a high-dimensional, multi constrained and nonlinear problem, and there has seen much literature on this topic. The difficulty of the optimization lies in how to find the optimal solution quickly and accurately under multiple constraints. For example, the IB-RBCO algorithm can be applied to solve the UC and economic load distribution problems to obtain the maximum power generation [17]. We can also use the new mutation strategy to obtain a wider search range and accelerate the convergence by using Preventing Individual Repeated Failure Evolution (PIRFE) strategy to improve the efficiency [18]. The application of ant colony algorithm and slime mold algorithm in the optimization of cascade hydropower plants has also shown good performance [19, 20]. However, previous proposed ABC strategies rarely considered both data storage and computational load problems at the same time, and therefore in this study a parallel and compact ABC approach is proposed to improve the efficiency and stability as well as reliability. Since most of the computers used in power stations are outdated with very limited memory size, efficient computation cannot be guaranteed, and therefore a parallel scheme can be introduced to distribute the computation task to several computers. The optimization of power generation requires high efficiency using poor computational environment, and the performance of a parallel and compact ABC scheme is suitable for this application.

The rest of the paper is organized as followed. The second section reviews the related strategies, the third section demonstrates the theory of the proposed algorithm, the fourth section tests the performance of the theory, the fifth section gives an application example of cascade hydropower plant, and finally a conclusion is drawn in section six.

2. **Related work.**

2.1. **Artificial Bee Colony Algorithm.**

The Bee Colony self-organization model was firstly proposed by Seeley in 1995 [21], and later improved by Tereshko [22, 23]. Based on these schemes, Karaboga proposed the ABC algorithm in 2005 [24, 25]. The main steps of the ABC algorithm are as follows [25].

1.Initialization

The initialization of the bee colony is to determine the numbers of employed bees and onlookers respectively, and they are usually equal number of SN.The number of food source is equal to that of employed bees $\overrightarrow{x_m}, (m = 1; \cdots, \mathbf{S}N)$ and every food source $\overrightarrow{x_m}$ is a solution vector. Therefore, there are n variables in $\vec{x}_m\,(x_{mi}, \bar{i} = 1, \cdots, n)$.The initialization is then realized using $\mathrm{x_i} = l_\mathrm{i} + \mathrm{rand}(0,1) \times (u_i - l_i)$,where u_i and l_i are the upper and lower limit respectively.

2.Employed bee search

The employed bees will search the location near the food source, and if a new food source is found, the fitness value will be compared with the food sources recorded in memory. Then the formula $x'_{id} = x_{id} + \phi_{id}\,(x_{id} - x_{kd})$ is applied to determine the food source in the neighborhood area, where x_{id} is a randomly selected food source, i is a randomly selected location index and ϕ_{id} is a random number between [-1, 1]. When a new food source is found, its fitness value will be calculated and a better solution between the two is determined by the greedy algorithm, which is transformed into a minimization problem such that:

$$f_m\,(\overrightarrow{x_m}) = \begin{cases} \frac{1}{1+f_m(x_m)} + x & \text{if} \quad f_m\,(x_m) \geq 0 \\ 1 + \mathrm{abs}\,(f_m\,(\overrightarrow{x_m})) & \text{if} \quad f_m\,(\overrightarrow{x_m}) < 0 \end{cases} \tag{1}$$

where $f_m\,(\overrightarrow{x_m})$ is the objective function.

3.Onlookers search

The employed bees will pass the food source information to the waiting onlookers who will randomly choose a food source based on the roulette rule, such that the probability for $f_m\,(\overrightarrow{x_m})$ being selected is given by $p_m = \frac{fit_m(\overrightarrow{x_m})}{\sum_{m=1}^{SN} fit_m(x_m)}$. The food source location is then updated by $x'_{id} = x_{id} + \phi_{id}\,(x_{id} - x_{kd})$, and the fitness value is calculated for the greedy algorithm to make a choice.

4.Scouts search

If an employed bee does not find a more adaptable food source within a given number of attempts, it will be turned into a scout bee on the spot and starts search for a new solution randomly, and the current solution will be abandoned.

2.2. **Compact ABC.**

Population-based artificial intelligence algorithms usually start from initializing m vectors with n-dimensional elements, and therefore $m \times n$ memory units are required to store these vectors. The compact algorithm uses the probability model to represent the distribution of all vectors based on a Probability Vector (PV) as the probability model of all solutions [26]. The data in each dimension can be represented by a normal distribution function, so that the memory units required are reduced to $2 \times n$. In the compact ABC scheme, the distribution of each bee is described by a Probability Distribution Function (PDF), and usually we assume that it follows the Gaussian distribution with a mean value of μ and a standard deviation of σ [27]. However, this is not a standard Gaussian distribution but has been truncated, so the optimization problem of ABC is transformed to the normalization of two truncated Gaussian curves. PV is a matrix vector of $2 \times n$-element, which is defined as $PV^t = [\mu^t, \delta^t]$, where t is the time step and the amplitude of the PDF is normalized so as to maintain the total probability to one. The solution x_i is determined by $PV(\mu_i, \sigma)$, and the corresponding PDF is given as:

$$PDF_{\mu_i,\sigma_i}(x) = \frac{e^{-\frac{(x-\mu_i)^2}{2\delta_i^2}}\sqrt{\frac{2}{\pi}}}{\delta_i\left(\operatorname{erf}\left(\frac{u_i+1}{\sqrt{2}\delta_i}\right) - \operatorname{erf}\left(\frac{u_i^{-1}}{\sqrt{2}\delta_i}\right)\right)} \tag{2}$$

where $erf()$ is an error function [28]. The Cumulative Distribution Function (CDF) is defined with Chebyshev polynomials such that [29]

$$CDF = \int_0^1 \frac{e^{-\frac{(x-\mu_i)^2}{2\delta_i^2}}}{\delta_i\left(\operatorname{erf}\left(\frac{u_i+1}{\sqrt{2}\delta_i}\right) - \operatorname{erf}\left(\frac{u_i-1}{\sqrt{2}\delta_i}\right)\right)}dx \tag{3}$$

The value of x_i is then obtained through the inverse function of the CDF by generating an evenly distributed random number between 0 and 1.

The initial values of μ and σ are 0 and 10 respectively. In the process of evolution, the position of x_i is updated to reach a new solution x_{i+1}, and then their fitness values are compared to give rise to a winner. The values of μ and σ in the PV are updated such that,

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}(\ winner_i - loser\ _i) \tag{4}$$

$$\delta_i^{t+1} = \sqrt{\left(\delta_i^t\right)^2 + \left(\mu_i^t\right)^2 - \left(\mu_i^{t+1}\right)^2 + \frac{(winner_i^2 - loser_l^2)}{Np}} \tag{5}$$

$$[winner, loser] = complete(x_{best}, x^{t+1}) \tag{6}$$

3. **The optimal operation of cascade hydropower plants.**

In recent years there have seen more and more attentions focused on the optimization of renewable energy such as solar, wind and hydropower et al. In traditional multi-level hydropower plants only single objective optimization has been considered to focus on the optimization of in one aspect taking account of other conditions as constraints. The corresponding optimization schemes include linear programming [30], nonlinear programming [31], dynamic programming [32] and et al. However, for the optimal operation of cascade hydropower plants, many different aspects are required to be taken into account such as the power generation efficiency, capacity and income. In the annual non-flooding season, the objective function for maximizing the power generation capacity can be written as,

$$E_1 = \max\left\{\sum_{i=1}^{N}\sum_{t=1}^{T}\eta_i \times Q_{i,t} \times H_{i,t}\right\} \times \Delta t \tag{7}$$

The objective function for maximizing the peak load regulation capacity is given as,

$$maxE_2 = \max\left\{\min\sum_{t=1}^{T} N(i,t)\right\} \tag{8}$$

where $E1$ is the maximum generating capacity of a cascade hydropower plant, $E2$ is the maximum guaranteed target output of each level, N is the number of cascade hydropower plants,η_i is the output coefficient of each level, i is the level number $(i = 1, 2, ..., n)$, Q is the power generation flow, T is the total number of dispatching periods, t labels each dispatching period $(t = 1, 2, ..., N)$, Δt is the length of dispatching period, and $N(i,t)$ represents the average output of the ith level of the hydropower plant in the T period. Other constraints will include:

(1) Water balance constraint, given as

$$V_{i,t+1} = V_{i,t} + [I_{i,t} - Q_{i,t} - S_{i,t}] \times \Delta t \tag{9}$$

where $V_(i,t), I_(i,t), Q_(i,t), S_(i,t)$ represent the water storage, inflow, loss and abandoned respectively for the ith level hydropower plant at time t.

(2) Flow balance constraint, which is

$$I_{i+1,t} = Q_{i,t} + q_{i,t} \tag{10}$$

(3) Water level constraint, given as,

$$\underline{Z}_{i,t} \le Z_{i,t} \le \bar{Z}_{i,t} \tag{11}$$

where $\underline{Z}_{i,t}$ and $\bar{Z}_{i,t}$ are the lower limit and upper limit of the ith level hydropower plant at time t respectively, and it is defined that $\underline{Z}_{i,t} \le \bar{Z}_{i,t}$.

(4) Flow constraint:

$$\underline{Q}_{i,t} \le Q_{i,t} \le \bar{Q}_{i,t} \tag{12}$$

$\underline{Q}_(i,t)$ and $\bar{Q}_(i,t)$ represent the lower limit and upper limit of the abandoned flow of the ith level hydropower plant at time t respectively, and it is defined that $\underline{Q}_(i,t) \le \bar{Q}_(i,t)$

(5) Power output constraint of each level:

$$\underline{N}_{i,t} \le N_{i,t} \le \bar{N}_{i,t} \tag{13}$$

where $\underline{N}_{i,t}$ and $\bar{N}_(i,t)$ are the lower limit and upper limit of the power output of the ith level hydropower plant at time t respectively, and it is defined that $\underline{N}_(i,t) \le \bar{N}_(i,t)$

4. **The compact and parallel strategy.**

In the literature there have seen some parallel strategies suitable for the implementation of multi-core processor to improve the convergence speed and accuracy. For example, a parallel differential optimization scheme was proposed with two communication strategies [26]. A coevolutionary mutation strategy with elite population and three mutations was proposed to optimize the differential evolution algorithm [34]. Shared memory and multiprocessor are also used to improve the execution efficiency through parallel communication strategy [35]. It has been proved that the effect of parallel communication is much better than that of single search strategies. The Parallel Compact ABC algorithm

(PCABC) demonstrated in this research is based on elite parallel communication strategy and the steps are given as:

1. The optimization problem is solved from G independent groups in parallel, and the PV in each of the groups is initialized;
2. According to the ABC algorithm, each group generates an elite;
3. The offspring fitness value produced by each of the groups is compared with the elite to obtain new elite, which is used to update the PV value.
4. After α iterations, the groups communicate with each other and elect the optimal elite to replace other elites. All the PVs will also be replaced by the optimal elite PV.
5. Repeat the above steps until the termination condition is met. The pseudo code is listed as follows:

Algorithm 1 pcABC algorithm Pseudocode

```
t = 0;
g = 5;
trial = 0; limit = 10; sol = x_best = up_bound;
//Groups Initialization;
Employed Bee Phase
for i = 1 : g do
    InitializeGroup[i].PV;
    GenerateGroup[i].elitebyGroup[i].PV;
end for
while budget condition do
    //Mutation;
    Paraller for every group do;
    Generate i individuals Group[m].x;
    Generate Group[i] : x^t from PV;Caculate f(Group[i].x^t)
    if (f(Group[i].x^t))<f(sol) then
        sol = group[i].x^t
        f(sol) = f(group[i].x^t)
    else trial = trial + 1
    end if
    for i = 1 : n do
        if f(Group[i].elite) ≤ f(Group[i].x^t then
            Group[i].elite = Group[i].x^t
        end if
    end for
    //Sellection;
    [winner, loser] = copmete(Group[m].x^t, Group[m].elite)
    if Group[m].x^t == winner then
        Group[m].elite = Group[m].x^t
    end if
    // PV Update;
    for i = 1 : n do
        Equation(12) and (13);
    end for
    Find the group wyth the most adaptable solution and record the group number
  as k
    if mod(t, θ) == 0 then
        for i = 1 : n do
            Group[i].PV = Group[k].PV;
```

$Group[i].elite = Group[k].elite$;
end for
end if
$t = t + 1$
end while
Onlook Bee Phase;
$x^t = sol; Prob = equation(1)$;
if **then**$rand \leq Prob$
for $i = 1 : n$ **do**
$x^t(i) = x^t(i) + rand \times (x^t(i) - x(k)$ with random $k = l, ..., n$;
end for
end if
$Calculate(x^t)$
//Update local;
if **then**$f(x^t) lef(sol)$
$sol = x^t$
$f(sol) = f(x^t)$;
else
$trial = trial + 1$;
end if
Update Global
if **then**$f(sol) \lg f(best)$
$x_{best} = sol$;
$f_{best} = f(sol)$;
end if
Scout Bee Phase;
$\vdots$
if $trail == limit$ **then**
$initial(sol)$
end if

5. Experiments.

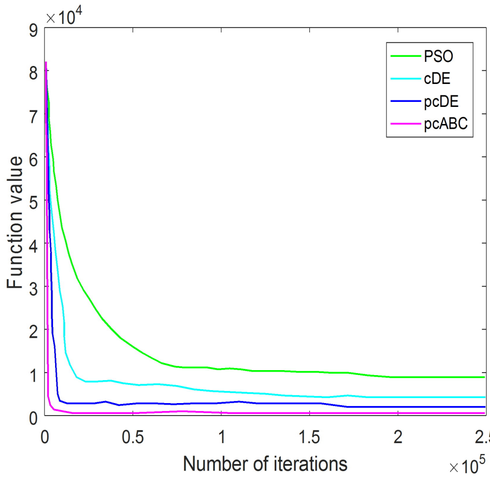
In this section, in order to demonstrate the performance of the proposed algorithm, the CEC2013 test functions are used and compared with other previous versions of the ABC algorithm [36]. The CEC2013 contains 28 functions, among which f1 f5 are unimodal functions for testing the calculation accuracy and efficiency of the algorithm. The functions of f6 f20 in the CEC2013 are multi-peak functions for testing the global search performance, and the functions of f21 f25 are synthetic functions for testing the combined performance of the algorithm [37]. All the tested algorithms are set with the same initial parameters.

In this experiment, there are two tasks being tested, one is to compare the performance of the PCABC with other general optimization algorithms such as the CDE [38,39],PCDE [26], CBA and PSO [40], and the other one is to test the PCABC and compare with several improved versions of the ABC algorithms [24], such as the ABC, MABC [7], PABC [41], EABC [42] and CABC [16]. The test platform is MATLAB, and each of the tasks is run for 30 times and the average value is calculated. The dimension of all test functions is set to 50, and the number of iterations is 3000. There are 5 parallel groups in each of the tests, and the value of NP is set to $2 \times n$. The parameters of other comparing algorithms are set according to the references. Table 1 shows the results of the first test, and the data obtained are the average value $\pm$ the corresponding standard deviation. Based on the method in [43] the confidence level is set to 0.95. In the table, ($\geq$) indicates that this algorithm is better than the proposed PCABC, and ($\leq$) means the opposite while ($\equiv$) represents that this algorithm shows the same performance with the PCABC.

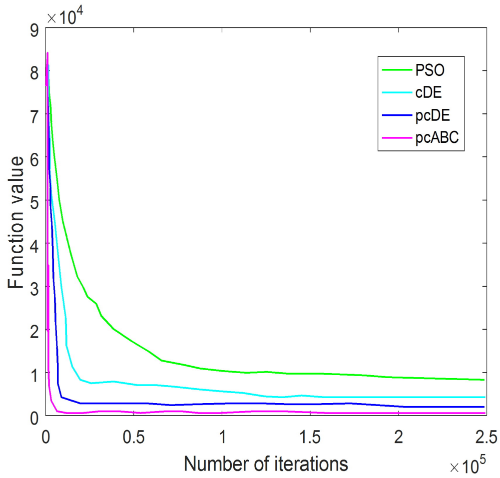
TABLE 1. Result of the first test.

	CDE		PCDE		PSO		PCABC
f1	1.948e+04±2.08e+04	<	5.755e+00±4.88e-01	<	4.215e+02±3.58e+01	<	1.709e-12±1.64e-13
f2	8.581e+07±10.6e+06	<	4.689e+07±5.78e+06	<	2.787e+08±10.7e+07	<	6.917e+06±4.58e+05
f3	1.764e+10±1.07e+10	<	3.322e+09±1.11e+08	>	8.721e+10±1.82e+09	<	6.084e+08±3.91e+07
f4	8.574e+04±9.41e+03	<	4.664e+04±9.6e+03	=	8.803e+04±3.1e+03	<	4.305e+04±3.41e+03
f5	3.013e-01±5.96e-02	<	2.742e+00±3.65e+00	>	5.892e+02±5.19e+01	<	7.108e-04±2.88e-05
f6	7.743e+02±9.14e+01	<	2.116e+02±6.05e+01	>	3.615e+02±1.25e+00	<	4.564e+01±9.57e-01
f7	3.055e+02±8.65e+01	<	2.366e+02±9.59e+01	>	4.647e+02±6.37e+01	<	2.598e+02±7.06e+01
f8	2.95e+01±4.32e+00	<	2.938e+01±4.01e+00	<	2.836e+01±5.22e+00	>	2.927e+01±3.04e+00
f9	5.852e+01±1.69e+00	>	6.855e+01±8.8e+01	>	4.646e+01±8.57e+00	>	7.439e+01±2.07e+01
f10	1.418e+02±10.13e+02	<	2.566e+01±4.1e+00	<	6.74e+03±2.89e+02	<	9.756e-02±4.98e-03
f11	4.022e+02±6.13e+02	>	5.681e+00±6.38e-01	<	5.12e+02±7.28e+01	<	4.064e+02±9.13e+00
f12	5.455e+02±8.85e+02	>	5.332e+02±4.03e+01	>	1.487e+03±9.39e+02	<	8.718e+02±8.46e+01
f13	6.969e+02±4.45e+02	>	6.538e+02±4.6e+01	>	5.841e+03±9.97e+01	<	7.505e+02±9.43e+01
f14	3.484e+03±8.53e+02	>	4.946e+02±2.81e+01	>	6.477e+03±4.09e+01	<	4.792e+03±1.12e+02
f15	1.767e+04±9.64e+03	<	8.947e+03±8.31e+03	<	9.588e+03±3.41e+02	<	7.256e+03±9.37e+02
f16	2.073e+00±4.61e+00	<	2.172e+00±6.58e+00	<	1.974e-01±3.26e-02	<	1.427e+00±6.66e-01
f17	3.347e+02±2.22e+01	>	5.842e+01±1.75e+01	<	7.82e+02±2.21e+01	<	5.718e+02±3.71e+01
f18	9.403e+02±4.49e+02	<	5.679e+02±4.22e+01	>	9.653e+02±9.97e+01	<	1.888e+03±4.56e+02
f19	6.538e+01±9.74e+01	<	2.999e+00±2.95e+01	>	3.254e+03±5.42e+02	<	2.133e+01±5.85e+00
f20	2.276e+01±9.59e+00	<	2.176e+01±4.85e+00	<	2.274e+01±7.69e+00	<	1.822e+01±7.14e+00
f21	3.326e+03±2.89e+04	<	8.812e+03±5.31e+02	<	5.097e+03±11.4e+02	<	1.564e+03±5.32e+02
f22	6.035e+03±2.99e+02	<	7.741e+03±8.54e+00	<	9.326e+03±9.71e+02	<	3.463e+03±9.43e+02
f23	1.669e+04±4.86e+03	<	1.109e+04±7.76e+03	<	1.637e+04±4.32e+03	<	9.084e+03±3.58e+02
f24	4.216e+02±3.91e+02	<	4.038e+02±1.24e+01	<	7.173e+02±7.09e+01	<	3.837e+02±1.24e+01
f25	4.581e+02±2.49e+03	<	4.252e+02±5.06e+01	<	5.932e+02±7.19e+01	<	3.535e+02±9.53e+01
f26	5.863e+02±4.87e+01	<	1.997e+02±7.88e+01	>	2.261e+02±8.87e+01	<	2.034e+02±2.67e+01
f27	2.376e+03±1.97e+02	>	2.631e+03±7.27e+02	<	2.819e+03±3.63e+03	<	2.383e+03±9.19e+01
f28	8.338e+03±6.83e+03	<	5.345e+02±9.39e+01	>	4.597e+03±8.51e+02	<	3.237e+03±9.15e+02

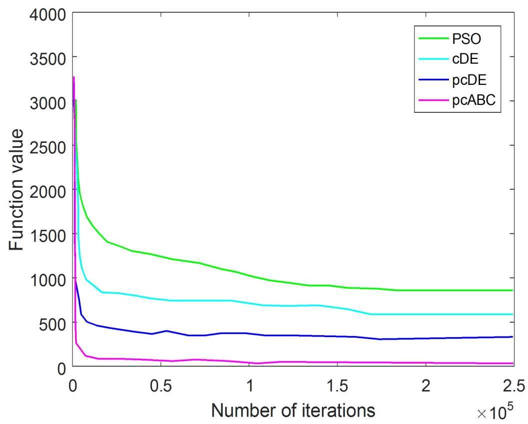
In order to test the convergence performance, the optimization of the functions f6, f10, f24 and f2 are specifically tested and the function values progressed with iterations are shown in (a)-(d). From (a)-(d) we can see that the PCABC performs better than other schemes in convergence speed and accuracy.



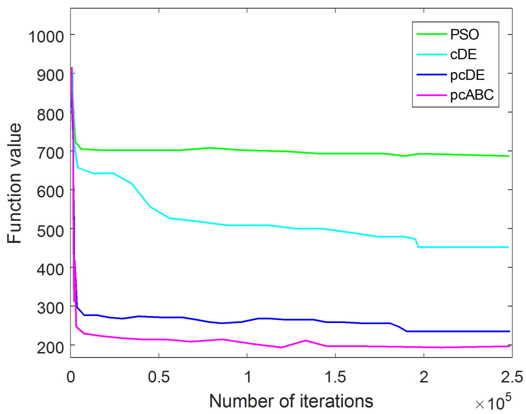
(a) Average performance on $f6$



(b) Average performance on $f10$



(c) Average performance on $f24$



(d) Average performance on $f25$

Table 2 shows the results from the second test, and the results show that the PCABC scheme outperforms other algorithms.

TABLE 2. Result of the second test.

	ABC		EABC		PABC		MABC		CABC		PCABC
f1	2.966e-12±7.73e-13	<	7.506e-12±7.11e-12	<	2.049e-12±1.38e-13	>	2.291e-12±9.07e-13	<	1.579e-01±6.23e-01	<	2.276e-12±1.73e-13
f2	7.476e+09±6.02e+08	<	5.338e+07±5.32e+06	<	2.041e+07±9.86e+06	<	2.822e+07±6.18e+06	<	5.232e+08±3.95e+07	<	6.574e+06±3.25e+05
f3	2.082e+10±5.65e+09	<	2.512e+09±4.36e+09	<	1.409e+09±7.92e+08	<	1.018e+09±1.81e+09	<	2.659e+11±9.11e+10	<	6.247e+08±4.02e+07
f4	4.521e+06±1.33e+06	<	1.101e+05±1.86e+04	<	1.071e+05±4.45e+04	<	1.206e+05±2.07e+04	<	7.796e+04±9.64e+03	<	3.026e+04±2.38e+03
f5	1.723e+05±5.29e+04	<	6.146e-02±9.45e-03	<	2.001e-12±9.73e-13	>	2.228e-12±9.65e-12	>	1.143e+03±3.85e+02	<	6.814e-04±3.72e-05
f6	4.378e+02±6.32e+00	<	5.195e+02±5.96e+01	<	4.686e+02±4.69e+01	<	4.952e+01±7.71e+00	<	3.288e+03±7.50e+02	>	4.246e+01±9.09e-01
f7	2.568e+03±5.52e+02	<	1.842e+02±5.91e+01	<	2.644e+02±6.20e+01	<	1.506e+02±4.83e+01	>	2.087e+02±9.37e+02	<	1.864e+02±5.86e+01
f8	3.309e+01±4.43e+00	<	2.148e+01±2.97e+00	<	2.136e+01±9.27e+00	<	2.142e+01±3.89e+00	<	2.831e+01±6.39e+00	<	2.125e+01±4.57e+00
f9	9.363e+01±1.83e+00	<	7.466e+01±2.26e+00	>	8.412e+01±1.16e+00	<	6.555e+01±4.97e+00	>	8.529e+01±3.58e+01	<	7.540e+01±1.54e+01
f10	2.257e+03±4.64e+02	<	5.097e+00±7.36e-01	<	2.586e+00±9.01e-02	<	2.961e+00±5.65e-01	<	2.258e+12±4.13e+02	<	9.953e-02±5.19e-03
f11	3.753e+02±4.79e+01	=	9.847e-01±4.77e+00	>	3.171e-13±1.86e-14	>	5.603e-13±8.34e-14	>	9.153e+02±5.38e+01	<	3.753e+02±9.51e+00
f12	2.830e+03±3.91e+02	<	4.571e+02±5.42e+02	<	5.078e+02±9.67e+01	<	8.494e+02±1.90e+02	<	8.215e+02±6.93e+01	<	4.564e+02±7.51e+01
f13	8.413e+02±7.95e+01	<	4.138e+02±5.12e+02	<	6.443e+02±3.87e+01	<	6.620e+02±3.83e+01	<	8.936e+02±6.99e+01	<	4.113e+02±7.66e+01
f14	4.925e+03±9.13e+02	<	2.573e+02±9.76e+02	>	4.234e+01±5.10e+00	<	3.478e+00±4.84e-01	>	2.069e+04±2.12e+04	<	3.802e+03±2.00e+02
f15	8.379e+03±3.43e+02	<	9.409e+03±2.95e+02	<	9.823e+03±7.18e+02	<	7.446e+03±3.73e+02	=	2.873e+04±7.77e+03	>	7.447e+03±9.34e+02
f16	7.903e+00±3.58e-01	<	4.215e+00±2.34e+00	<	3.323e+00±6.41e+00	<	2.883e+00±8.94e-01	<	4.208e+00±5.53e+00	<	2.557e+00±7.04e-01
f17	4.252e+02±1.86e+01	>	6.986e+02±6.82e+00	<	6.108e+01±2.92e+00	>	6.062e+01±4.28e+01	>	1.888e+03±1.72e+03	<	6.946e+02±5.29e+01
f18	5.825e+02±9.42e+01	<	2.701e+02±4.86e+02	<	5.133e+02±4.59e+01	<	7.416e+02±2.97e+01	<	1.394e+03±9.01e+02	<	1.902e+02±4.93e+02
f19	2.991e+04±6.83e+03	<	6.236e-01±4.22e-02	<	3.673e+00±2.93e-01	>	9.703e+01±4.94e+02	<	4.669e+05±3.62e+01	<	3.905e+01±6.39e+00
f20	4.23e+01±1.91e+00	<	6.681e+01±9.75e+00	<	9.003e+01±9.18e+00	<	3.795e+01±2.88e+00	<	3.141e+01±4.54e+00	>	3.389e+01±7.61e+00
f21	2.814e+03±7.91e+02	<	2.924e+02±2.68e+01	>	2.889e+02±4.73e+01	>	2.904e+03±9.74e+02	<	4.949e+03±4.19e+03	<	2.009e+03±5.14e+02
f22	3.817e+03±1.61e+02	>	2.829e+01±3.42e+00	>	6.501e+01±1.93e+00	>	3.219e+03±6.44e+02	<	1.819e+04±8.15e+03	>	3.067e+03±9.06e+02
f23	2.147e+04±9.27e+03	<	2.313e+04±6.04e+03	<	2.209e+04±5.02e+03	<	9.652e+04±7.23e+03	<	1.897e+04±5.23e+03	<	9.777e+03±2.98e+02
f24	4.046e+02±7.06e+01	<	3.475e+02±4.73e+02	>	3.198e+02±1.76e+01	>	3.612e+02±3.02e+01	<	5.011e+02±5.56e+01	<	3.534e+02±2.06e+01
f25	6.628e+02±4.09e+01	<	4.142e+02±2.84e+01	<	3.634e+02±8.78e+01	>	4.124e+02±1.85e+01	<	4.939e+02±2.05e+02	<	3.733e+02±9.14e+00
f26	8.116e+02±6.87e+01	<	2.542e+02±4.48e+01	>	2.818e+02±4.28e+01	<	2.525e+02±4.28e+01	=	5.706e+02±3.91e+01	<	2.579e+02±2.69e+01
f27	4.844e+03±4.17e+03	<	3.527e+03±1.26e+02	<	3.337e+03±6.56e+02	<	3.598e+03±1.43e+03	<	3.504e+03±2.02e+02	<	3.283e+03±8.48e+01
f28	2.329e+03±1.67e+02	>	4.502e+02±1.98e+01	>	4.191e+02±4.22e+01	<	4.518e+03±3.45e+02	<	8.246e+03±4.09e+03	<	2.525e+03±8.43e+02

6. Application of the proposed algorithm on operation optimization of cascade hydropower plants.

In this section, the proposed PCABC algorithm is applied to the optimization of cascade hydropower plant and compare with other commonly used optimization schemes. The water conservancy characteristic parameters of the plant reservoirs are shown in Table 3. Hydropower plant A is an upstream reservoir of hydropower plant B, and the operation period is from November to the next May. The setting of other parameters is based on [44-46]. Table 4 shows the optimization results obtained from the test.

TABLE 3. The water conservancy characteristic parameters of the plant reservoirs

Plant	Lowest pool level (m)	Normal pool level (m)	Installed capacity (MW)	Firm capacity (MW)	Max turbine flow (m3.s-1)	Combined efficiency coefficient
A	145	175	2250	499	2900	8.5
B	62	66.5	321	104	2360	8.5

TABLE 4. Optimization results obtained from the test

Algorithm	Output of Plant A	Output of Plant B	Total output
ABC	404.3521	86.4927	490.8448
GABC	404.4217	86.3215	490.7432
PABC	405.2103	86.7324	491.9427
MABC	404.8219	86.6813	491.5032
CABC	404.5794	86.5021	491.0815
PCABC	405.3231	86.5973	491.9204

It can be seen from Table 4 that the proposed PCABC algorithm has helped to achieve the maximum total power generation, which is 11.772×107kw·h and 10.756kw·h more than the GABC and ABC algorithms respectively. It can be seen from Table 5 that the ABC algorithm is better than the GABC in this application, which explains that different problems require a suitable scheme for optimization.

TABLE 5. Optimization data of various ABC algorithms for hydropower generation

Algorithm	Output of plan A	Output of plan B	Total output
ABC	404.3521	86.4927	490.8448
GABC	404.4217	86.3215	490.7432
PABC	405.2103	86.7324	491.9427
MABC	404.8219	86.6813	491.5032
CABC	404.5794	86.5021	491.0815
PCABC	405.4231	86.6973	492.1204

FIGURE 1.shows the convergence speed of each of the algorithms. It can be seen that the PCABC algorithm achieves the optimal value at 5×104 iterations, while others need at least 16×104 iterations. Therefore, the convergence performance of the PCABC is also better than other algorithms.

7. Conclusion.

In this paper a Parallel Compact ABC (PCABC) method was proposed and applied to the optimization of hydropower plant. For the previous compact ABC method, only one solution is initialized and optimized, leading to uncertain result. Based on the compact ABC, the PCABC scheme adopts the parallel communication strategy to produce several solutions to improve the efficiency and accuracy. The Probability Distribution Function

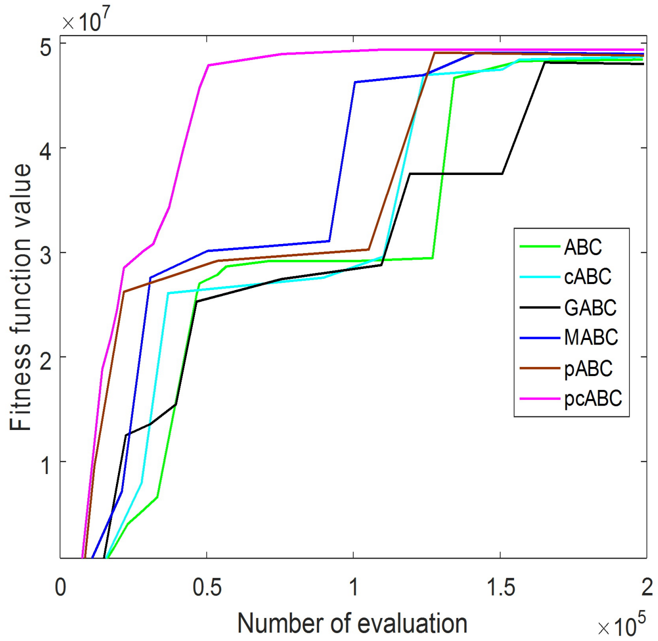


FIGURE 1. Hydropower plant output at different iterations.

(PDF) is used to estimate the virtual population in the CABC algorithm. In this way, the stability and the convergence speed of the algorithm can be greatly improved. The proposed algorithm was tested in several test functions, and the performance is shown to be better than the previous algorithms. The proposed PCABC scheme is also applied to the reservoir operation optimization of a cascade hydropower plant. Comparing with other algorithms, the proposed scheme produces more effective solution with less calculation.

Acknowledgment.
This work is partially supported by The high-level personnel startup program of Beibu Gulf University (Grant NO. 2018KYQD39);The Research Basic Ability Improvement Project of Young and Middle-aged Teachers in Guangxi Universities(2019KY0449);

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers,which have improved the presentation.

REFERENCES

[1] S. S. Jadon, R. Tiwari, H. Sharma, J. C. Bansal, Hybrid artificial bee colony algorithm with differential evolution, *Applied Soft Computing*, vol. 58, pp. 11-24, 2017.
[2] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, *Journal of Intelligent Manufacturing*, vol. 23, pp. 1001-1014, 2012.
[3] M. Sonmez, Artificial Bee Colony algorithm for optimization of truss structures, *Applied Soft Computing*, vol. 11, no. 2, pp. 2406-2418, 2011.
[4] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, vol. 8, no. 1, pp. 687-697, 2008.
[5] N. Taspinar, D. Karaboğă, M. Yildirim, B. Akay, Partial transmit sequences based on artificial bee colony algorithm for peak-to-average power ratio reduction in multicarrier code division multiple access systems, *IET Communications*, vol. 5, no. 8, pp. 1155-1162, 2011.
[6] Z. Meng, J.-S. Pan, L. Kong, Parameters with Adaptive Learning Mechanism (PALM) for the enhancement of Differential Evolution, *Knowledge-Based System*, vol. 141, pp. 92-112, 2018.
[7] X. Zhang, X. Zhang, S. L. Ho, W. N. Fu, A modification of artificial bee colony algorithm applied to loudspeaker design problem, *IEEE Transactions on Magnetics*, vol. 50, no. 2, 7018204, 2014.
[8] S.-S. Chu, H.-C. Huang, J. F. Roddick, J.-S. Pan, Overview of algorithms for swarm intelligence, *Third International Conference on Computational Collective Intelligence*, pp. 28-41, 2011.

[9] G. Zhou, H. Moayedi, M. Bahiraei,Z. Lyu, Employing artificial bee colony and particle swarm techniques for optimizing a neural network in prediction of heating and cooling loads of residential buildings, *Journal of Cleaner Production*, vol. 254, 120082, 2020.

[10] M. Bijandi, M. Karimi, B. F. Bansouleh, W. Knaap, Agricultural land partitioning model based on irrigation efficiency using a multi-objective artificial bee colony algorithm, *Transactions in GIS*, vol. 25, pp. 551-574, 2021.

[11] A.-Q. Tian, S.-S. Chu, J.-S. Pan, H. Cui, W.-M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, vol. 12, no. 3, 767, 2020.

[12] Z. Gu, Y. Zhu, Y. Wang, X. Du, M. Guizani, Z. Tian, Applying artificial bee colony algorithm to the multidepot vehicle routing problem, *Software: Practice and Experience*, 2020, https://doi.org/10.1002/spe.2838.

[13] J.-S. Pan, P. Hu, S.-S. Chu, Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power, *Processes*, vol. 7, no. 11, 845, 2019.

[14] G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287-297, 1999.

[15] L. Yu, Q. Zheng, S. Zhewen, Compact particle swarm optimization algorithm, *Journal-Xian Jiaotong University*, vol. 40, no. 8, 883, 2006.

[16] T.-K. Dao, S.-C. Chu, T.-T. Nguyen, C.-S. Shieh, M.-F. Horng, Compact artificial bee colony, *27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems*, pp. 96-105, 2014.

[17] P. Lu, J. Zhou, C. Wang, Q. Qiao, L. Mo, Short-term hydro generation scheduling of Xiluodu and Xiangjiaba cascade hydropower plants using improved binary-real coded bee colony optimization algorithm, *Energy Conversion and Management*, vol. 91, pp. 19-31, 2015.

[18] X. Wen, J. Zhou, Z. He, C. Wang, Long-term scheduling of large-scale cascade hydropower plants using improved differential evolution algorithm, *Water*, vol. 10, no. 4, 383, 2018.

[19] G. Xu, G. Ma, Y. Tu, Optimal operation of cascade hydropower plants based on ant colony algorithm, *Huoli Xuebao*, vol. 36, no. 8, pp. 978-982, 2005.

[20] T.-T. Nguyen, H.-J. Wang, T.-K. Dao, J.-S. Pan, J.-H. Liu, S. Wend, An improved slime mold algorithm and its application for optimal operation of cascade hydropower plants, *IEEE Access*, vol. 8, pp. 226754-226772, 2020.

[21] T. D. Seeley, The Wisdom of the Hive: The social physiology of honey bee colonies, *Harvard University Press*, 1996.

[22] V. Tereshko, Reaction-diffusion model of a honey bee colon's foraging behavior, *6th International Conference on Parallel Problem Solving from Nature*, pp. 807-816, 2020.

[23] V. Tereshko, T. Lee, How information mapping patterns determine foraging behavior of a honey bee colony, *Open Systems and Information Dynamics*, vol. 9, pp. 181-193, 2002.

[24] D. Karaboga, An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06*, 2005.

[25] B. Basturk, D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, *IEEE Swarm Intelligence Symposium*, 2006.

[26] X. Sui, S.-C. Chu, J.-S. Pan, H. Luo, Parallel compact differential evolution for optimization applied to image segmentation, *Applied Sciences*, vol. 10, no. 6, 2195, 2020.

[27] M. Younis, M. Youssef, K. Arisha, Energy-aware routing in cluster-based sensor networks, *10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pp.129-136, 2002.

[28] W. Gautschi, Error function and Fresnel integrals, *Handbook of Mathematical Functions*, pp. 297–308, 1972.

[29] R. Pemantle, A survey of random processes with reinforcement, *Probability Surveys*, vol. 4, pp. 1–79, 2007.

[30] E. B. Hreinsson, Optimal short term operation of a purely hydroelectric system, *IEEE Transactions on Power Systems*, vol. 3, no. 3, pp. 1072-1077, 1988.

[31] M. T. L. Barros, J. E. G. Lopes, S.-L. Yang, W. W.-G. Yeh, Large-scale hydropower system optimization, *IAHS-AISH Publication*, 2001.

[32] G. K. Young Jr., A. M. ASCE, Finding reservoir operating rules, *Journal of the Hydraulics Division*, vol. 93, no. 6, pp. 297-322, 1967.

[33] T.-Y. Wu, L. Yang, Z. Lee, C.-M. Chen, J.-S. Pan, S. H. Lslam, Improved ECC-based three-factor multiserver authentication scheme, *Security and Communication Networks*, vol. 2021, 6627956, 2021.

[34] Y. Song, D. Wu, W. Deng, X.-Z. Gao, T. Li, B. Zhang, Y. Li, MPPCEDE: Multi-population parallel co-evolutionary differential evolution for parameter optimization, *Energy Conversion and Management*, vol. 228, 113661, 2021.

[35] D. Abramson, J. Abela, A parallel genetic algorithm for solving the school timetabling problem, Division of Information Technology, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.56.8965.

[36] J. Tvrdík, R. Poláková, Competitive differential evolution applied to CEC 2013 problems, *2013 IEEE Congress on Evolutionary Computation*, pp. 1651–1657, 2013.

[37] T.-K. Dao, T.-S. Pan, J.-S. Pan, A multi-objective optimal mobile robot path planning based on whale optimization algorithm, *2016 IEEE 13th International Conference on Signal Processing*, pp. 337-342, 2016.

[38] E. Mininno, F. Neri, F. Cupertino, D. Naso, Compact differential evolution. *IEEE Transaction on Evolution Computing*, vol. 15, no. 1, pp. 32–54, 2011.

[39] Z. Meng, J.-S. Pan, QUasi-affine TRansformation Evolutionary (QUATRE) algorithm: A parameter-reduced differential evolution algorithm for optimization problems, *2016 IEEE Congress on Evolutionary Computation*, pp. 4082-4089, 2016.

[40] J. L. Foo, J. Knutzon, V. Kalivarapu, J. Oliver, E. Winer, Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization, *Journal of Aerospace Computing, Information, and Communication*, vol. 6, pp. 271-290, 2009.

[41] W.-F. Gao, S.-Y. Liu, L.-L. Huang, A novel artificial bee colony algorithm with powell's method, *Applied Soft Computing*, vol. 13, no. 9, pp. 3763-3775, 2013.

[42] W.-F. Gao, S.-Y. Liu, L.-L. Huang, Enhancing artificial bee colony algorithm using more information-based search equations, *Information Science*, vol. 270, pp. 112-133, 2014.

[43] A. Banitalebi, M. I. A. Aziz, A. Bahar, Z. A. Aziz, Enhanced compact artificial bee colony, *Information Sciences*, vol. 298, pp. 491-511, 2015.

[44] A. T. Hammid, O. I. Awad, M. H. Sulaiman, S. S. Gunasekaran, S. A. Mostafa, N. M. Kumar, B. A. Khalaf, Y. A. Al-Jawhar, R. A. Abdulhasan, A review of optimization algorithms in solving hydro generation scheduling problems, *Energies*, vol. 13, no. 11, 2787, 2020.

[45] W. Wang, R. Tang, C. Li, P. Liu, L. Luo, A BP neural network model optimized by Mind Evolutionary Algorithm for predicting the ocean wave heights, *Ocean Engineering*, vol. 162, pp. 98-107, 2018.

[46] J.-S. Pan, Z. Meng, S.-C. Chu, H.-R. Xu, Monkey King Evolution: an enhanced ebb-tide-fish algorithm for global optimization and its application in vehicle navigation under wireless sensor network environment, *Telecommunication Systems*, vol. 65, pp. 351-364, 2017.