



# An Improved Bacterial Foraging Optimization with Differential and Poisson Distribution Strategy and its Application to Nurse Scheduling Problem

Jingzhou Jiang<sup>1</sup>, Xiaojun Xiong<sup>2</sup>, Yikun Ou<sup>2</sup>, and Hong Wang<sup>2</sup>(✉)

<sup>1</sup> College of Mathematics and Statistics, Shenzhen University, Shenzhen, China

<sup>2</sup> College of Management, Shenzhen University, Shenzhen, China  
ms.hongwang@gmail.com

**Abstract.** Bacterial Foraging Optimization (BFO) has been predominantly applied to some real-world problems, but this method has poor convergence speed over complex optimization problems. In this paper, an improved Bacterial Foraging Optimization with Differential and Poisson Distribution strategies (PDBFO) is proposed to promote the insufficiency of BFO. In PDBFO, the step size of bacteria is segmented and adjusted in accordance with fitness value to accelerate convergence and enhance the search capability. Moreover, the differential operator and the Poisson Distribution strategy are incorporated to enrich individual diversity, which prevents algorithm from being trapped in the local optimum. Experimental simulations on eleven benchmark functions demonstrate that the proposed PDBFO has better convergence behavior in comparison to other six algorithms. Additionally, to verify the effectiveness of the method in solving the real-world complex problems, the PDBFO is also applied to the Nurse Scheduling Problem (NSP). Results indicate that the proposed PDBFO is more effective in obtaining the optimal solutions by comparing with other algorithms.

**Keywords:** Bacterial Foraging Optimization · Differential strategy · Poisson Distribution · Nurse Scheduling

## 1 Introduction

In a seminal paper published in 2002, Passino showed how bacterial individuals and groups find nutrients and how to model it as a distributed optimization process, which he named the Bacterial Foraging Optimization (BFO) [1]. The algorithm is designed to simulate foraging behavior of animals in nature, which has excellent capability to search optimal value of functions with steep function images [2]. Moreover, in the standard BFO, bacteria can avoid falling into the local optimum to some extent and also cooperating [3]. However, this approach often leads to the problem of slow convergence speed [1]. In recent decades, many

researchers have optimized BFO to enhance the performance. In the improvement of BFO scholars focus on two aspects: BFO's operational strategy and algorithm combination.

In terms of algorithm strategy improvement, Mishra et al. combined the fuzzy rule system of Takagi-Sugeno (TS) in the BFO and then put forward a Fuzzy Bacterial Foraging (FBF) algorithm [4]. Besides, compared with classical algorithms, adaptive chemotactic operators have better performance. Ben Niu proposed an improved BFO with adaptive chemotaxis step and a non-linearly decreasing exponential modulation model [5]. The BFO of automatic chemoattractant step size is brought up by Majhi [6].

In regards to algorithm combination, Tang suggested a multi-level threshold method using the modified Bacterial Foraging Optimization (MBFO) [7] so as to enhance the practicability of the optimal threshold technology. Researchers found that bacteria could learn from the best position in the population by integrating Particle Swarm Optimization (PSO) into each chemotactic step, which enhanced the global search capability of the algorithm [8]. D.H. Kim et al. combined the crossover and mutation operators of the Genetic Algorithm (GA) with the BFO [9]. Luh considered biological evolution and came up with Bacterial Evolutionary Algorithm (BEA) [10]. Combining the Bacterial Foraging and Particle Swarm Optimization, Biswas proposed the hybrid optimization algorithm and applied it to solve the optimization of multi-modal functions [11].

Despite the great efforts of the researchers, the problems of premature convergence and slow convergence speed remain tricky. In order to promote the development of BFO algorithm, this paper proposes an improved BFO with Differential and Poisson Distribution strategies (PDBFO) to shed some light on the problems. To begin with, differential operators are incorporated into standard BFO to increase the convergence accuracy. Meanwhile, the segmentation step size change strategy is adopted. Finally, the Poisson Distribution strategy is used to disperse the bacteria.

Optimization algorithms are widely used to solve practical problems with high complexity. Alireza Goli applied the Accelerated Cuckoo Algorithm to the vehicle routing problem [12]. Precup, R.E. suggests the use of Grey Wolf Optimizer algorithms to optimize the parameters of Takagi-Sugeno proportional-integral-fuzzy controllers [13]. To further demonstrate the effectiveness of PDBFO in solving the real-world problem, a real Nurse Scheduling Problem (NSP) which improves the working efficiency and quality of nurses is employed [14].

This paper is organized as follows. Section 2 introduces the standard BFO algorithm. Then, the proposed algorithm is illustrated in details in Sect. 3. Section 4 presents the model of NSP. Section 5 shows the performance comparisons. Finally, conclusions and further work are presented in Sect. 6.

## 2 Bacterial Foraging Optimization

### 2.1 Chemotaxis

Chemotaxis is the most important process of BFO, in which bacteria gradually approach the optimal value through rotation and swimming. This process simulates the behaviour of swimming and tumbling through flagella of *E. coli*. Suppose  $\theta_i(j, k, l)$  represents the  $i$ th bacterium at  $j$ th chemotactic,  $k$ th reproductive, and  $l$ th elimination-dispersal process. The movement of  $i$ th bacterium is expressed as follows:

$$\theta_i(j+1, k, l) = \theta_i(j, k, l) + C(i) \varphi(i) \quad (1)$$

where  $C(i) > 0$  represents the step size of each step forward, and  $\varphi(j)$  represents a random forward direction vector selected after tumbling.

### 2.2 Reproduction

For reproduction, bacteria are ranked according to their health degree  $Jhealth$ . The smaller  $Jhealth$  is, the healthier the bacterium is. The larger half population will be replaced by the better half. In this way, the population of bacteria remains the same. The algorithm runs  $N_{re}$  times reproduction operations. The health degree of bacteria  $Jhealth$  is calculated as follows:

$$Jhealth(i) = \sum_{j=1}^{N_c} J(i, j, k, l) \quad (2)$$

where  $J$  represents fitness value computed by objective function,  $i$  represents each individual,  $j$  represents the number of chemotactic,  $k$  represents the number of reproduction and  $l$  represents the number of migration.

### 2.3 Elimination-Dispersal

In the evolutionary process, elimination and dispersal events may occur, which bacteria in one area are killed or a group of them are dispersed to a new environment. Dispersal may disrupt chemotaxis, whereas may also assist chemotaxis since it may direct the bacteria approaching better places with fruitful nutrition. This operation simulates the migration of bacteria to new environments by water currents or other organisms and the bacteria will be dispersed to random locations within the search area. The algorithm runs the dispel operation  $N_{ed}$  times. The specific operation is to give a fixed value of  $P_{ed}$  which is chosen within  $[0,1]$ . When the random number of bacterial individual is less than  $P_{ed}$ , the bacteria will die. Otherwise, a new individual will be generated randomly so as to achieve the purpose of migration.

### 3 Improved Bacterial Foraging Optimization Algorithm

In order to improve the search capability and convergence speed of the BFO, step size segmentation strategies are introduced by adjusting the step size according to the fitness values. Additionally, the differential operators and greedy selection strategies are employed at the end of swimming operations. During the migration process, Poisson Distribution strategy is used to choose the individuals to be dispelled. The Pseudocode of PDBFO has been provided in Algorithm 1.

#### 3.1 Segmentation Step Size Changes on Chemotaxis Operation

In [15], experiment results have indicated that with the change of the current fitness value, changing of Chemotaxis step size can lead to better convergence performance in comparison to the case when the step size is fixed. In this paper, Chemotaxis step size is exploited by adopting a segmentation strategy. The fitness values of the bacteria are sorted in ascending order. Individuals with the smaller fitness values indicate good localization and could be assigned with small step size, whereas individuals with larger fitness values indicate poor bacterial localization and are assigned with a large step size.

Thus, the Chemotaxis step size can be adjusted as follows:

$$C_i = \begin{cases} C_{max} + (N_c - j) \frac{C_{max} - C_{min}}{N_c}, & i \leq 0.2S \\ C_{min} - (N_c - j)(C_{max} - C_{min}), & i \geq 0.8S \\ 0.1, & i \in (0.2S, 0.8S) \end{cases} \quad (3)$$

where  $S$  represents the population number,  $N_c$  is the total number of swimming,  $j$  is the current swimming times, and  $C$  represents the step size.

#### 3.2 Bacterial Foraging Algorithm Combined with Differential Strategy

From BFO, we can find that the chemotactic operator searches the field through random movement to ensure the local search capability of bacteria. However, the bacteria do not make full use of the information of other bacteria in the environment. Consequently, the convergence speed of the algorithm is slow, resulting in premature convergence of the algorithm and difficulty in obtaining the global optimal [16]. Therefore, differential operator and greedy selection mechanism are embedded in the proposed algorithm.

After the swimming operation, the bacteria enter the stage of differential operation. The differential vector is established first, and then the vector is synthesized with the individual to generate the new individual. Then the fitness values of each new individual and the corresponding original individual are compared, and the greedy strategy is used to retain the better performing individuals [17]. In addition, an adaptive scaling operator  $F$  is introduced in the process of generating intermediate individuals. The adaptive scaling factor changes according to the following equation:

$$F = 2F_0 \cdot e^{\frac{1-N_c}{N_c+1-j}} \quad (4)$$

where,  $F_0$  is the initial mutation operator,  $F_0 = 0.4$ .  $N_c$  and  $j$  represent the total number of chemotaxis and the current number of chemotaxis.

The generation equation of intermediate individuals is

$$G_i = P_i + F \cdot (R_1 - R_2) \quad (5)$$

where,  $G$  is the intermediate individual, and  $P$  is the corresponding original individual.  $R_1$  and  $R_2$  are two random individuals which are different from  $P$ .

The selection process is expressed as follows:

$$P_{i+1} = \begin{cases} G_i, & \text{if } f(G_i) < f(P_i) \\ P_i, & \text{otherwise} \end{cases} \quad (6)$$

where,  $P_{i+1}$  is the location of the selected new individual,  $G_i$  is the location of the intermediate individual, and  $P_i$  is the location of the original individual.

#### PSEUDOCODE FOR PDBFO ALGORITHM

```

1 Initialize:
  (a) Set parameters:  $N_{ed}$ ,  $N_{re}$ ,  $N_c$ ,  $N_s$ ,  $S$ ,  $C_{max}$ ,  $C_{min}$ ,  $F_0$ ,  $Dim$ 
  (b) Initialize bacterial population
2 Eliminating and dispersal loop: for  $l = 1, 2, \dots, N_{ed}$ 
3   Reproduction loop: for  $k = 1, 2, \dots, N_{re}$ 
4     Chemotaxis Loop: for  $j = 1, 2, \dots, N_c$ 
5       for  $i = 1, 2, \dots, S$ 
6         Compute Fitness Value  $J(i, j, k, l)$ , and set  $J_{last} = J(i, j, k, l)$ 
7         Tumble by Eq.(1), then compute and sort Fitness Value  $J(i, j, k, l)$ 
8         while  $m < N_s$ 
9           update  $P_i$  by Eq.(1) and Eq.(3)
10        end while
11        Compute  $F$  by Eq.(4) and  $G_i$  by Eq.(5)
12        Select  $G_i$  and  $P_i$  by Eq.(6)
13      end for
14    end for
15    Reproduction, compute the all bacteria'  $J_{health}$  by Eq.(2) and sort bacteria
16    for  $j = 1, 2, \dots, S/2$ 
17      Rerandomize the location of bacteria  $P_j$ 
18    end for
19  end for
20  Generate poisson distribution Numbers by Eq.(7) and compute  $J(i, j, k, l)$ 
21  for  $m = 1, 2, \dots, S$ 
22    Eliminate each bacterium as described in subsection 3.3
23  end for
24 end for

```

### 3.3 Poisson Distribution in Elimination-Dispersal Operations

Elimination-dispersal is an integral part of BFO. By means of elimination-dispersal, the bacteria trapped in local optimum can reposition themselves to avoid premature convergence. Elimination-dispersal makes the algorithm have a better random searching capability, and increases the diversity of the population. However, for high-dimensional optimization problems, elimination-dispersal will greatly slow down the convergence speed due to the increasing of dimensionality and complexity. Worse still, it is possible to drive away the best-performing bacteria, resulting in redundant computation [18].

This paper proposes a bacterial selection mechanism based on Poisson Distribution (PD) strategy to solve this problem. When the algorithm enters the dissipation process, the bacteria are firstly sorted according to the fitness value from small to large. Generate  $S$  random numbers that conform to the PD, and then compare the serial number of the bacteria with the corresponding random number to determine whether the bacterium should be dispersed. Through this mechanism, it is possible to ensure that most of the excellent bacteria will not be dispersed, and the bad bacteria will have the opportunity to be retained for further search. The probability function for the PD is

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, 2 \dots \tag{7}$$

where  $k$  is a random number,  $\lambda$  is the mean and variance of the PD. After experiments, when  $\lambda$  takes 25, the algorithm performs better.

## 4 Application of Improved Algorithm in Nurse Scheduling Problem

### 4.1 Problem Description

NSP refers to scheduling a specific group of nurses within a given scheduling period. The scheduling of nurses should meet some constraints (such as hard and soft constraints) and minimize the total salary of nurses [19].

The worksheet of nurse can be probably divided into three kinds: the early shift (0a.m.–8a.m.), the day shift (8a.m.–4p.m.) and the night shift (4p.m.–0p.m.). The wages are allocated by the level of nurses (Junior, Middle, Senior). The objective function of the Nurse Scheduling Problem is provided as follows [20]:

$$\min f(x) = \sum_{i=1}^{nn} \sum_{j=1}^{sk} \sum_{k=1}^{ss} \sum_{d=1}^{sd} x_{ijkd} \cdot w_{jk} + c \cdot \sum_{i=1}^{nn} \sum_{j=1}^{sk} \sum_{k=1}^{ss} \sum_{d=1}^{sd} m_{jkd} \tag{8}$$

where,  $nn$  is the total number of nurses,  $sk$  is the grade,  $ss$  is the type of shifts and  $sd$  is the period of scheduling.  $x_{ijkd}$  represents the  $i$  nurse belonging to  $j$  level has  $k$  shift in the  $d$  day,  $w_{jk}$  is the wage of nurse on the  $k$  shift at the

$j$  level and  $m_{jkd}$  means the amount of unsatisfied shifts. And  $c$  is the penalty coefficient, whose value is 1000.

Equations (9–11) are hard constraints of NSP and Eq. 12 is the soft constraint:

$$\sum_{k=1} X_{ijkd} \leq 1 \tag{9}$$

$$\text{if } X_{i,j,3,d} = 1, \text{ then } X_{i,j,3,d+1} = 0 \tag{10}$$

$$lp \leq \sum_{d=1}^{sd} \sum_{k=1}^{ss} X_{ijkd} \leq up \tag{11}$$

$$m_{ijkd} > 0 \tag{12}$$

Equation (9) means that each nurse can have no more than one shift per day. Equation (10) represents each nurse cannot be on consecutive shifts within two days. Equation (11) presents that the working hour of nurses cannot exceed the lower or upper limit in a scheduling period. Equation (12) shows that the number of nurses on one shift is no less than the actual demand.

## 5 Experiment Results and Discussions

This section illustrates the PDBFO’s performance and comparisons among the proposed PDBFO and the other BFOs, the GA [20], the HCO [21] and the PSO [22]. In this experiment, for above algorithms, the running times is 30, the swarm size is 50, 10000 is the maximum iterations and the dimension of the search space is 30. Specifically, as for GA, the crossover probability is 0.8 and the mutation probability is 0.1. While for BFO methods, the number of

**Table 1.** Function characteristics

Function	Name	Dimensions	Domain of definition
$f_1$	Ackley	30	$[-32.768, 32.768]$
$f_2$	Levy	30	$[-10, 10]$
$f_3$	Powell	30	$[-4, 5]$
$f_4$	Sphere	30	$[-5.12, 5.12]$
$f_5$	Sum of powers	30	$[-1, 1]$
$f_6$	Zakharov	30	$[-5, 10]$
$f_7$	Dixon-price	30	$[-500, 500]$
$f_8$	Griewank	30	$[-600, 600]$
$f_9$	Rotated hyper	30	$[-65.536, 65.536]$
$f_{10}$	Rastrigin	30	$[-5.12, 5.12]$
$f_{11}$	Sum squares	30	$[-10, 10]$

swimming, chemotaxis, reproduction and elimination-dispersal are respectively  $N_s = 4$ ,  $N_c = 1000$ ,  $N_{re} = 5$  and  $N_{ed} = 2$ . In addition, in PDBFO, the  $\lambda$  in Eq. (7) is 25 and the  $F_0$  in Eq. (4) is 0.4. While for PSO, the parameter setting is  $c_1 = c_2 = 1.5$ ,  $w = 0.8$ . More parameters settings of HCO is: the maximum flow times is 3, the evaporation and rainfall probability is 0.2. We have selected eleven well-known benchmark functions [23]. Table 1 summarizes the search scope of all functions.

### 5.1 Experiment Results on Benchmark Functions

Table 2 provides the average results and the variances on benchmark functions over 30 runs. For the sake of observation, the results are treated with logarithms. Noted that the mean of minimum values and the variance of each group have been bolded to highlight the best performing algorithm.

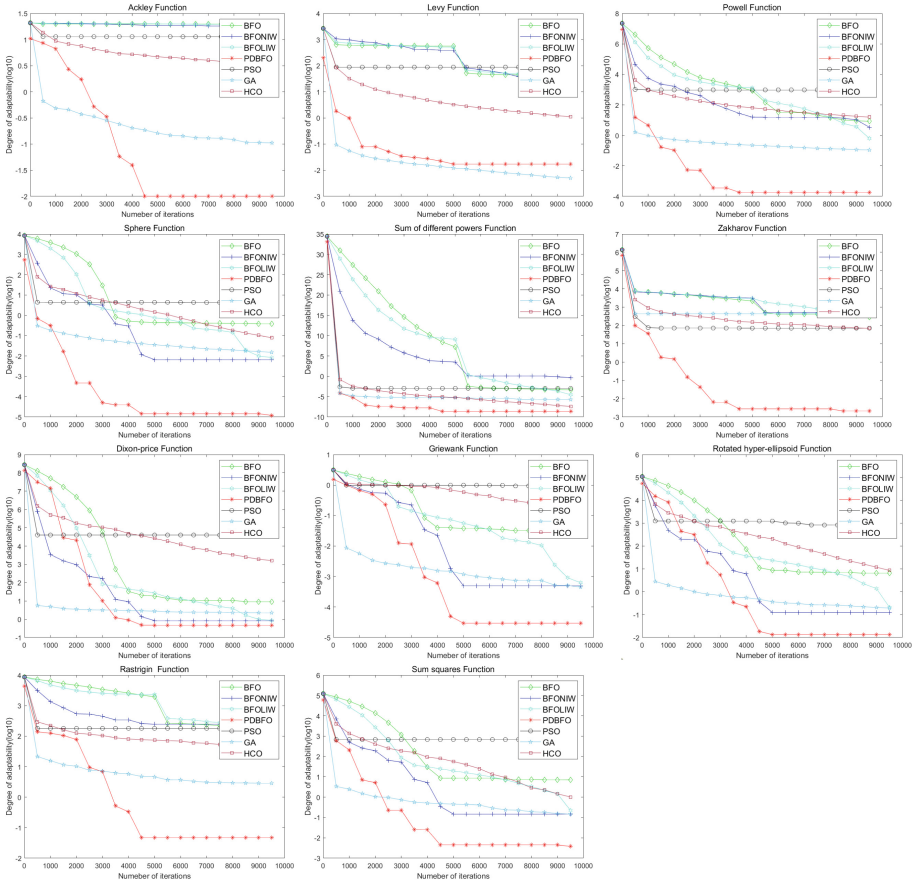
From Table 2 and convergence Fig. 1, it can be observed that the PDBFO outperforms other algorithms. For one thing, PDBFO obtains high quality mean of results in comparison to those of other algorithms in most cases. Although PDBFO performs worse than GA in  $f_2$ , it still has the edge over its counterparts. When others BFOs, GA, PSO and HCO find a solution that is close to the optimal value, they get stuck in poor local optimal and have trouble getting rid of it, while PDBFO can improved its solutions steadily (such as in  $f_3$ ,  $f_4$  and  $f_6$ ). Because differential operators and Poisson distribution strategies keep PDBFO from falling into poor local optima. Consequently, the proposed PDBFO

**Table 2.** The numerical results on benchmark functions  $f_1$  through  $f_{11}$

Function	Result	PDBFO	BFO	BFONIW	BFOLIW	PSO	GA	HCO
$f_1$	Variance	<b>1.01E-04</b>	1.82E-03	1.84E+00	4.18E-04	1.83E-01	1.52E-04	2.57E+00
	Mean	<b>1.00E-02</b>	1.96E+01	1.70E+01	1.90E+01	1.13E+01	9.82E-02	1.60E+00
$f_2$	Variance	2.08E-02	2.57E+00	9.70E-01	2.19E+00	6.59E+01	<b>9.78E-07</b>	2.13E-01
	Mean	1.44E-01	4.82E+01	2.69E+01	3.96E+01	8.01E+01	<b>4.94E-03</b>	4.61E-01
$f_3$	Variance	<b>1.09E-05</b>	2.12E+00	1.42E-02	1.44E-02	4.86E+04	1.05E-03	4.10E+01
	Mean	<b>3.30E-03</b>	6.87E+00	1.72E+00	4.03E-01	4.12E+02	1.22E-01	6.41E+00
$f_4$	Variance	<b>3.16E-08</b>	1.45E-03	7.90E-08	6.78E-13	1.83E+00	3.37E-05	7.76E-04
	Mean	<b>1.78E-04</b>	3.56E-01	6.71E-03	4.13E-03	6.72E+00	1.85E-02	2.79E-02
$f_5$	Variance	<b>1.11E-16</b>	6.56E-08	4.82E-02	1.27E-12	8.34E-11	5.60E-14	4.20E-16
	Mean	<b>2.41E-09</b>	5.21E-04	2.20E-01	6.68E-06	4.78E-05	2.65E-07	4.71E-09
$f_6$	Variance	<b>4.41E-06</b>	9.81E+00	1.85E+04	6.75E+02	7.55E+02	1.01E+05	2.50E+03
	Mean	<b>2.10E-03</b>	2.07E+02	5.27E+02	3.25E+02	7.19E+01	4.42E+02	5.00E+01
$f_7$	Variance	2.28E-01	7.99E-03	<b>1.09E-05</b>	2.42E-05	6.91E+08	1.02E+00	1.01E+06
	Mean	<b>4.78E-01</b>	9.13E+00	8.41E-01	7.58E-01	3.95E+04	2.27E+00	1.00E+03
$f_8$	Variance	<b>8.77E-11</b>	1.36E-05	4.22E-09	1.71E-10	4.06E-03	1.41E-10	2.29E-03
	Mean	<b>2.96E-05</b>	2.77E-02	4.97E-04	3.70E-04	9.33E-01	4.18E-04	4.79E-02
$f_9$	Variance	<b>1.83E-05</b>	3.08E-01	8.25E-04	2.80E-05	1.33E+04	1.94E-04	3.97E+00
	Mean	<b>1.35E-02</b>	6.42E+00	1.25E-01	9.22E-02	8.27E+02	1.83E-01	1.99E+00
$f_{10}$	Variance	<b>2.19E-03</b>	3.08E+00	1.53E+01	6.14E+00	2.28E+02	6.69E-02	3.92E+02
	Mean	<b>4.68E-02</b>	1.86E+02	1.83E+02	1.31E+02	1.81E+02	2.31E+00	1.98E+01
$f_{11}$	Variance	<b>1.46E-05</b>	5.33E-02	3.94E-04	3.53E-05	3.72E+04	2.96E-03	2.71E-01
	Mean	<b>3.83E-03</b>	7.11E+00	1.38E-01	1.15E-01	6.88E+02	1.45E-01	5.21E-01



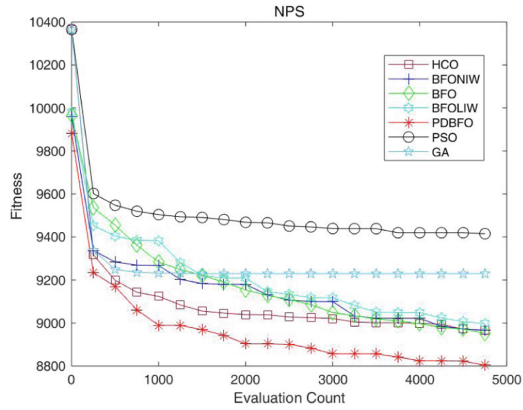
has desirable global search capability and strong robustness. For another thing, although the convergence speed of PDBFO is slower than GA (such as in  $f_2$  and  $f_7$ ) or PSO (such as in  $f_9$ ) in the early stage of some cases, PDBFO has a satisfactory convergence speed in comparison to other BFOs.



**Fig. 1.** The convergence results on benchmark function

Furthermore, PDBFO also has excellent performance in terms of stability. As can be seen from Table 2, in most cases, the variance given by PDBFO is much smaller than the corresponding variance given by other algorithms, which reflects that the PDBFO comes with very small volatility. Except that the variance of PDBFO is greater than GA in  $f_2$ , PDBFO shows sufficiently competitive stability in the other ten benchmark function experiments.

Overall, the PDBFO maintains splendid global searching capability and satisfactory convergence speed in most benchmark function experiments and hence it is more competitive than other algorithms. In addition, the low variance of



**Fig. 2.** The convergence results on the nurse scheduling problem

**Table 3.** The preferred schedule selected by PDBFO

Nurse number	Sun.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
1	A	N	R	P	N	P	R
2	P	P	A	N	R	P	R
3	A	N	N	R	P	R	R
4	N	N	R	P	R	N	P
5	N	R	A	P	P	P	R
6	A	A	A	N	R	N	R
7	R	P	R	P	A	A	N
8	R	A	N	N	R	P	R
9	A	P	P	A	A	R	A
10	P	A	P	A	N	R	R
11	P	R	P	R	P	A	N
12	P	P	R	P	N	N	R
13	A	N	N	R	A	P	N
14	N	P	P	R	A	N	R
15	N	P	P	R	A	R	N
16	R	P	N	N	R	R	P
17	N	N	R	P	P	P	A
18	A	P	P	R	A	N	R
19	A	A	A	N	P	A	A
20	N	P	N	P	P	A	A

A, P, N and R represent morning work, noon work, evening work and rest respectively.

experimental results computed by PDBFO indicates that it is capable of delivering stable results.

## 5.2 Experiments on Nurses Scheduling Problem

PDBFO is compared with other BFOs, PSO, GA and HCO on solving the Nurse Scheduling Problem in this experiment. The schedule of 20 nurses for a week will be presented after calculation. Equation (8) is the objective function. The Function dimension is 30 dimension and the number of iterations is 5000.

As it is shown in Fig. 2, the proposed PDBFO has an edge over other algorithms on the Nurse Scheduling Problem. PDBFO has a better searching capability and convergence speed. An efficient shifts roster is shown in Table 3 to meet the needs of health care as well as improve the job satisfaction of nurses. The combination of scientific and reasonable nurse scheduling model and PDBFO can replace the low efficiency and low quality of manual scheduling scheme, contributing to enhancing the availability and optimization of the scheduling shifts. Due to the excellent search capability of PDBFO, an optimal scheduling scheme is more likely to be realized.

## 6 Conclusion and Future Work

PDBFO algorithm is proposed to improve the performance of BFO. The convergence speed is accelerated by introducing the change of segmentation step size. The Differential and Poisson Distribution strategies are used to reduce the problem of being trapped into local optimal. The benchmark function experiment results prove that the PDBFO significantly can deliver solutions with good quality and stability. Such an excellent performance demonstrates that PDBFO can balance different nursing requirements efficiently, making it an outstanding choice to solve the Nurse Scheduling Problem.

Due to the fact that PDBFO has shortcomings in the benchmark functions experiments, PDBFO can be further improved. More efficient bacterial swimming methods will be sought to improve the rate of convergence. Meanwhile, the improvement direction of the replication process will also be taken into consideration to improve the global optimization capability of the algorithm.

**Acknowledgment.** This work is partially supported by the Natural Science Foundation of China under Grant 71901152, Natural Science Foundation of Guangdong Province (2018A030310575), Natural Science Foundation of Shenzhen University (85303/00000155), Research Cultivation Project from Shenzhen Institute of Information Technology (ZY201717).

## References

1. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **22**(3), 52–67 (2002)

2. Amjady, N., Fatemi, H., Zareipour, H.: Solution of optimal power flow subject to security constraints by a new improved bacterial foraging method. *IEEE Trans. Power Syst.* **27**(3), 1311–1323 (2012)
3. Liu, Q., Xu, J.: Traffic signal timing optimization for isolated intersections based on differential evolution bacteria foraging algorithm. *Procedia-Soc. Behav. Sci.* **43**(4), 210–215 (2012)
4. Mishra, S.: A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. *IEEE Trans. Evol. Comput.* **9**(1), 61–73 (2005)
5. Niu, B., Wang, H., Tan, L., Li, L.: Improved BFO with adaptive chemotaxis step for global optimization. In: 2011 Seventh International Conference on Computational Intelligence and Security, pp. 76–80. IEEE (2011)
6. Majhi, R., Panda, G., Majhi, B., Sahoo, G.: Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. *Expert Syst. Appl.* **36**(6), 10097–10104 (2009)
7. Tang, K., Xiao, X., Wu, J., Yang, J., Luo, L.: An improved multilevel thresholding approach based modified bacterial foraging optimization. *Appl. Intell.* **46**(1), 214–226 (2016). <https://doi.org/10.1007/s10489-016-0832-9>
8. Raju, M., Gupta, M.K., Bhanot, N., Sharma, V.S.: A hybrid PSO-BFO evolutionary algorithm for optimization of fused deposition modelling process parameters. *J. Intell. Manuf.* **30**(7), 2743–2758 (2019)
9. Kim, D.H., Cho, J.H.: A biologically inspired intelligent PID controller tuning for AVR systems. *Int. J. Control Autom. Syst.* **4**(5), 624–636 (2006)
10. Zobelas, G., Tarantilis, C.D., Ioannou, G.: A hybrid evolutionary algorithm for the job shop scheduling problem. *J. Oper. Res. Soc.* **60**(2), 221–235 (2009)
11. Biswas, A., Dasgupta, S., Das, S., Abraham, A.: Synergy of PSO and bacterial foraging optimization—a comparative study on numerical benchmarks. In: Corchado, E., Corchado, J.M., Abraham, A. (eds.) *Innovations in Hybrid Intelligent Systems*. AINSC, vol. 44, pp. 255–263. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74972-1\\_34](https://doi.org/10.1007/978-3-540-74972-1_34)
12. Goli, A., Aazami, A., Jabbarzadeh, A.: Accelerated cuckoo optimization algorithm for capacitated vehicle routing problem in competitive conditions. *Int. J. Artif. Intell.* **16**(1), 88–112 (2018)
13. Precup, R.E., David, R.C., Petriu, E.M., Szedlak-Stinean, A.I., Bojan-Dragos, C.A.: Grey wolf optimizer-based approach to the tuning of pi-fuzzy controllers with a reduced process parametric sensitivity. *IFAC-PapersOnline* **49**(5), 55–60 (2016)
14. Aickelin, U., Dowsland, K.A.: An indirect genetic algorithm for a nurse-scheduling problem. *Comput. Oper. Res.* **31**(5), 761–778 (2004)
15. Tong, Y.L.: Bacteria foraging optimization algorithm based on self-adaptive method. *Value Eng.* (2015)
16. Chatzis, S.P., Koukas, S.: Numerical optimization using synergetic swarms of foraging bacterial populations. *Expert Syst. Appl.* **38**(12), 15332–15343 (2011)
17. Dasgupta, S.: Analysis of a greedy active learning strategy. In: *Advances in Neural Information Processing Systems*, pp. 337–344 (2005)
18. Sahib, M.A., Abdulnabi, A.R., Mohammed, M.A.: Improving bacterial foraging algorithm using non-uniform elimination-dispersal probability distribution. *Alexandria Eng. J.* **57**(4), 3341–3349 (2018)
19. Chao, W., Dong, X.: Variable neighborhood search algorithm for nurse rostering problem. *J. Comput. Appl.* **33**(2), 338–341 (2013)
20. Booker, L.B., Goldberg, D.E., Holland, J.H.: Classifier systems and genetic algorithms. *Artif. Intell.* **40**(1–3), 235–282 (1989)

21. Yan, X., Niu, B.: Hydrologic cycle optimization part i: background and theory. In: Tan, Y., Shi, Y., Tang, Q. (eds.) ICSI 2018. LNCS, vol. 10941, pp. 341–349. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-93815-8\\_33](https://doi.org/10.1007/978-3-319-93815-8_33)
22. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN 1995-International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE (1995)
23. Surjanovic, S., Bingham, D.: Virtual library of simulation experiments: test functions and datasets. <http://www.sfu.ca/~ssurjano>. Accessed 20 Feb 2020