

Improved Bat Algorithm for Reliability-Redundancy Allocation Problems

Yubao Liu

College of Computer Science and Technology, Changchun University,
Changchun, China
Liuymb2015_mailbox@163.com

Abstract

The bat algorithm is a recently proposed meta-heuristic algorithm. Usually in solving the problem of optimization, the position of virtual bats is updated by flying speed, which decreases efficiency of the algorithm and accuracy of the solution. This paper has improved the location update strategy and individual selection strategy of bat algorithm, then puts forward an improved bat algorithm. The algorithm is used to solve three typical reliability-redundancy allocation problems, and the simulation experiment results show that the presented algorithm greatly enhances the computation efficiency, convergence speed and precision of the optimal solution in addressing the problem of reliability redundancy optimization. When compared with the best results reported in the previous literatures, the algorithm achieves a better or equally good optimal solution. It is demonstrated that the proposed algorithm is effective in solving reliability-redundancy allocation problems.

Keywords: reliability optimization, reliability-redundancy allocation, bat algorithm, nonlinear programming

1. Introduction

Reliability-redundancy optimization problem[1,2,7] is an important and complex mathematical programming problem in the field of engineering. It has been a hot topic of researchers and engineers. The goal tries to design an optimal redundancy allocation scheme to achieve the optimal value of the system reliability with the constraint conditions such as cost, volume and weight. So this problem is also called reliability-redundancy allocation problem(RRAP) [23,25]. Generally the problem can be described as:

$$\begin{aligned} \text{Max } R_s &= f(r,n) \\ \text{s.t.} \\ g_i(r,n) &\leq b_i, i = 1, \dots, p \end{aligned} \quad (1)$$

R_s is the reliability of the whole system, p is the number of constraints, g_i is the i th constraint, b_i is the i th upper limit of resource; $r=(r_1, r_2, \dots, r_m)$ is the reliability vector of selected components in the system. $n=(n_1, n_2, \dots, n_m)$ is the redundant amount vector of the selected components in the system, m is the number of types of available redundancy components in the system. $f(\cdot)$ is the objective function for the overall system reliability. The solution vector of reliability-redundancy optimization problem consists of two parts: the choice of component reliability values and their corresponding redundancy level (the number of redundancy).

Reliability-redundancy allocation problem is regarded as NP-hard problem. In the 1970s the traditional parameter optimization techniques appeared to be

inadequate in solving complex reliability-redundancy allocation problems. So finding effective optimization algorithm has always been a hot spot in the field. A wide variety of meta-heuristic algorithms emerge. At present, the main meta-heuristic algorithms for this problem are: genetic algorithm(GA) [2,3,4,5] immune algorithm(IA)[6,29], variable neighborhood search(VNS)[8], particle swarm optimization(PSO) [9,10,11,19], harmony search(HS)[12,13], ant colony optimization(ACO)[14], Tabu Search(TS)[15], imperialist competitive algorithm(ICA)[16], improved cuckoo search(ICS) algorithm[18] *et al.* Recently, some researchers have proposed hybrid meta-heuristic algorithms by combining with two kinds of meta-heuristic methods to solve the reliability-redundancy allocation problem. Safaei *et al.*[17] proposed particle swarm optimization algorithm based on simulated annealing (APSO). Wang & Li[31] proposed coevolution method based on differential evolution and harmony search(CDEHS). Chen *et al.* [20] presented method combined simulated annealing with artificial bee colony algorithm (artificial bee colony, ABC) method (SAABC). Kanagaraj *et al.*[21] put forward CS-GA method combined cuckoo search (CS) algorithm with genetic algorithm (GA).

Bat algorithm[32] is a new developed meta-heuristic optimization method inspired by echolocation behavior of bats. But it has never been applied to solve reliability-redundancy allocation problem. This paper presents an improved bat algorithm(IBA) for reliability redundancy optimization problem. The results show that the improved bat algorithm is not only simple, but also faster in convergence speed and effective in obtaining better optimal solution for solving reliability-redundancy allocation problem.

2. Improved Bat Algorithm (IBA)

Bat algorithm is a new meta-heuristic algorithm, which is proposed by Xin-she Yang recently[32]. Bat algorithm simulates echolocation behavior of some bats to solve optimization problems.

For simplicity, the idealized assumptions and rules are as follows[32]: All bats use echolocation to perceive distance, and they can also "know" the different preys and obstacles in some magical way.

Bats fly randomly with speed v_i at position x_i with a fixed frequency f_{\min} , changing wavelength λ and loudness A_0 to search for prey. They can automatically adjust to wavelength (or frequency) of pulse and the rate of pulse emission r ($r \in [0,1]$) according to the proximity of their object.

Though the loudness may be varied in many ways, Loudness is assumed to change from the large positive value A_0 down to a small constant value A_{\min} .

In order to further simplify the algorithm, it is assumed that the frequency $f \in [0, f_{\max}]$. The higher the frequency is, the shorter the wavelength is, and the smaller the propagation distance is. r is the rate of pulse. It can simply be in the range of $[0,1]$. If r is 0 means no pulses, and 1 indicates the max rate of pulse emission. The wavelength λ is not used. Because λf is a constant value, changing the frequency f can achieve the desired effect.

Based on the above idealized assumptions, bat algorithm can be described as follows[32]:

Objective function $f(x)$, $x = (x_1, \dots, x_d)$

Initialize the bat population x_i ($i = 1, 2, \dots, n$) and v_i
 Define pulse frequency f_i at x_i
 Initialize pulse rates r_i and the loudness A_i
while ($t < \text{Max number of iterations}$)
 Generate new solutions by adjusting frequency,
 and updating velocities and locations/ solutions [equations (2) to (4)]
 if ($\text{rand} > r_i$)
 Select a solution among the best solutions
 Generate a local solution around the selected best solution
 end if
 Generate a new solution by flying randomly
 if ($\text{rand} < A_i \ \& \ f(x_i) < f(x^*)$)
 Accept the new solutions
 Increase r_i and reduce A_i
 end if
 Rank the bats and find the current best x^*
end while
 postprocess results and visualization

In the D-dimensional search space, the position and velocity of the virtual bats are represented by D-dimensional vector x_i and v_i . For each virtual bat, the update of velocities and the generation of the new position are implemented by formula (2)-(4).

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (2)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i \quad (3)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (4)$$

Where, t is the time step. β is random vectors subjected to uniformly distributed in the range of $[0, 1]$. x^* is the current global optimum position. Typically $f_{\min} = 0$, $f_{\max} = 100$, the initial frequency of each virtual bat is uniformly distributed random number in the range of $[f_{\min}, f_{\max}]$.

In the local search, a random walk approach is applied for generating a new solution, as shown in formula (5).

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t \quad (5)$$

Wherein, ε is a random number, its value is $[-1, 1]$, A_t is average pulse loudness of all bats in time step t .

In the process of approximating its prey, the loudness of pulse will gradually decrease, and the rate of emission pulse will increase. Changes of loudness and rate are implemented by using formula (6) and (7).

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (7)$$

Among them, the initial value of A is set in accordance with the specific problems. For simplifying the problem, usually $A_0 = 1$, $A_{\min} = 0$. $A_{\min} = 0$ means that bats stop sounding when closing to the prey. r_0 is the initial rate of pulse emission, $r_0 \in [0, 1]$. T is the time step. α and γ are the adjustment

coefficients and can take the appropriate value according to the specific problem, usually $\alpha=\gamma=0.9$.

Some preliminary studies demonstrate that the bat algorithm is a promising meta-heuristic method. But there are some shortcomings such as the low precision, premature convergence, and the results are greatly influenced by the parameters. In order to improve the performance of bat algorithm to solve optimization problems, enhance search capabilities for the solution space so as to obtain better optimal solution, the bat algorithm has been improved and applied to the reliability -redundancy allocation problem. The improved algorithm is called improved bat algorithm(IBA).

For improving the global search ability of the algorithm, the position of each virtual bat is updated by using formula (8) :

$$x_{i,d}^{t+1} = x_{i1,d}^t + \lambda_1 (x_{i2,d}^t - x_{i3,d}^t) + \lambda_2 (x_{best}^t - x_{i,d}^t) \quad (8)$$

Where i_1, i_2, i_3 is random numbers, $i_1, i_2, i_3 \in [1, M]$, M is the population size, d represents the d th elements in solution vector. x^t represents the t th generation solution vector. x_{best} is current global optimal solution. λ_1, λ_2 are adjustment coefficients, $\lambda_1, \lambda_2 \in (0,1)$. Usually $\lambda_2 = (1-\lambda_1)$, can be also set the proper value depending on the specific problem.

In the process of local search, the following selection strategy is used to obtain the local optimal solution:

- (1) A new position s_d around the current global optimal solution is generated by using random walk, as shown in formula (9) :

$$s_d = x_{best,d}^t + \eta \text{rnd}_j \quad (9)$$

Where η is the adjustment coefficient, it's value is usually between 0 and 1. And rnd_j is a normal distribution random number, $j = 1,2, \dots, D$, D is the dimension of the solution vector. The item ηrnd_j generates a random step, it makes the current optimal solution move at the current position randomly. This can make the algorithm to avoid local optimal area so as to avoid the premature convergence. It can improve global search ability of the IBA algorithm.

- (2) The latest position of the i th virtual bat is determined by the strategy shown in Equation (10).

$$x_i^{t+1} = \begin{cases} s_d & \text{if } \text{Fitness}(s_d) < \text{Fitness}(x_i^{t+1}) \\ x_i^{t+1} & \text{otherwise} \end{cases} \quad (10)$$

- (3) Updating the global optimal solution by formula (11).

$$x_{best} = \begin{cases} x_i^{t+1} & \text{if } \text{Fitness}(x_i^{t+1}) < \text{Fitness}(x_{best}) \\ x_{best} & \text{otherwise} \end{cases} \quad (11)$$

Improved bat algorithm is described as follows:

Initialize the bat population x_i ($i = 1,2, \dots,n$)

Initialize pulse rates r and the loudness A

while ($t < \text{Max number of iterations}$)

$$x_{i,d}^{t+1} = x_{i1,d}^t + \lambda_1 (x_{i2,d}^t - x_{i3,d}^t) + \lambda_2 (x_{best}^t - x_{i,d}^t)$$

```

if (rand > r) // rand > r implies increasing ri
    sd = xbest,dt + η randj
end if
if Fitness(sd) < Fitness(xit+1)
    xit+1 = sd
else
    xit+1 = xit+1
end if
if (rand < A & Fitness (xit+1) < Fitness (xbest))
    // rand < A implies reducing Ai
    xbest = xit+1
else
    xbest = xbest
end if
end while
output xbest
    
```

3. Simulations and Comparisons

In the study of reliability-redundancy allocation problem, there are three benchmark problems: series systems, series parallel systems, complex (bridge) system. The optimization algorithms proposed by many researchers are based on these models.

3.1. Problem 1(P1): Series system

The structure of series system [26] is shown in Figure 1:

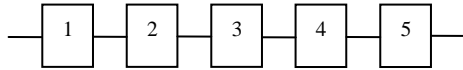


Figure 1. Series system

The reliability optimization model is:

$$\begin{aligned}
 \text{Max} \quad & f(r, n) = \prod_{i=1}^m R_i(n_i) \\
 \text{s.t.} \quad & g_1(r, n) = \sum_{i=1}^m w_i v_i^2 n_i^2 \leq V \\
 & g_2(r, n) = \sum_{i=1}^m \alpha_i (-1000 / \ln r_i)^\beta (n_i + \exp(n_i / 4)) \leq C \\
 & g_3(r, n) = \sum_{i=1}^m w_i n_i \exp(n_i / 4) \leq W \\
 & 0 \leq r_i \leq 1, n_i \in \mathbb{Z}^+, 1 \leq i \leq m
 \end{aligned} \tag{12}$$

The input parameters for this problem are shown in Table 1:

Table 1. The Parameters of Series System and Complex (bridge) System

Subsystem i	$10^5 \alpha_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.33	1.5	1	7	110	175	200
2	1.450	1.5	2	8			
3	0.541	1.5	3	8			
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

3.2. Problem 2(P2): Series Parallel System

The architecture of series parallel system[28] is shown in Figure 2:

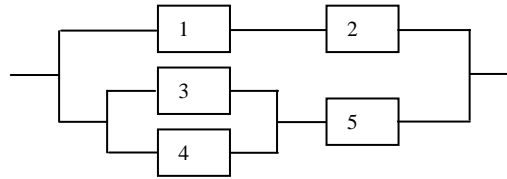


Figure 2. Series-Parallel System

The reliability optimization model is::

$$\text{Max } f(r, n) = 1 - (1 - R_1 R_2)(1 - (1 - (1 - R_3)(1 - R_4))R_5) \quad (13)$$

The constraints are the same as the problem 1. This problem is nonlinear mixed integer programming problem.

The input parameters are shown in Table 2:

Table 2. The Parameters of Series-Parallel System

Subsystem i	$10^5 \alpha_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.500	1.5	2	3.5	180	175	100
2	1.450	1.5	4	4.0			
3	0.541	1.5	5	4.0			
4	0.541	1.5	8	3.5			
5	2.100	1.5	4	4.5			

3.3. P3:Complex (Bridge) System

The structure of complex (bridge) system [24] is shown in Figure 3:

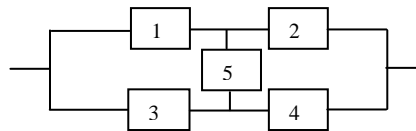


Figure 3. Complex (bridge) system

The reliability optimization model is:

$$\begin{aligned} \text{Max } f(r, n) = & R_1 R_2 + R_3 R_4 + R_1 R_4 R_5 + R_2 R_3 R_5 \\ & - R_1 R_2 R_3 R_4 - R_1 R_2 R_3 R_5 - R_1 R_2 R_4 R_5 \\ & - R_1 R_3 R_4 R_5 - R_2 R_3 R_4 R_5 + 2R_1 R_2 R_3 R_4 R_5 \end{aligned} \quad (14)$$

The constraints are the same as the problem 1. This problem is nonlinear mixed integer programming problem. The parameters value of this problem are shown in Table 1.

4. Simulation Results, Analysis and Discussion

In order to evaluate the performance of the algorithm, the proposed IBA is applied to solve three benchmark reliability-redundancy allocation problems. And the simulation results are compared and analyzed. The following parameters of simulation experiment are used in the IBA: The maximum number of iterations is 1500, population size $M = 40$, the loudness $A = 0.9$, rate of pulse emission $r = 0.1$, $\lambda_1 = 0.4$, $\lambda_2 = 0.6$, $\eta = 0.1$.

To measure the improvement of IBA than BA, BA is also applied to solve the three benchmark reliability-redundancy allocation problems. The parameters are

set in the BA: The maximum number of iterations is 15000(If the maximum number of iterations is set to 1500, BA can't get the satisfactory optimal solution.), population size $M = 40$, the loudness $A = 0.9$, rate of pulse emission $r = 0.1$, the frequency $f_{\min} = 0$, $f_{\max} = 10$.

In order to avoid violating the constraints, it is necessary that unfeasible solutions should be adapted to feasible solutions. In this paper, a penalty function is used for handling constraints. Thus the constrained optimization problem is transformed into the unconstrained one. Usually the maximization of $f(x)$ can be converted into the minimization of $-f(x)$. So the reliability- redundancy allocation problem formulated by formula (1) can be described as follows:

$$\min F(x) = -f(x) + \mu \sum_{j=1}^p \max\{0, g_j(x)\} \quad (15)$$

Where $F(x)$ is penalty function, $f(x)$ is objective function. $g_j(x)$ ($j = 1, 2, \dots, p$) is the j th constraint, and μ is penalty coefficient, and it is set to 10^{15} here.

In the reliability-redundancy allocation problem, n_i which represents the number of redundancy components is discrete variable. But in the proposed IBA, n_i is adjusted by a real number, the handling method transforms n_i into the nearest integer.

The IBA and BA are all coded in C++ and the simulations are on the Intel(R) Core(TM) i3-2130 CPU @3.40Ghz 3.40Ghz with 2GB memory.

The IBA and BA all run 50 times independently for each optimization problem. The simulation results and statistical features are listed in table 4-9.

The comparisons of statistical results for P1,P2,P3 and P4 are listed in Table 4-7, where Best represents the best results, Worst represents the worst result, Mean denotes the mean results and SD denotes standard deviation.

Table 4. Statistical Results of the Series System using Two Algorithms

Algorithm	Best	Worst	Mean	SD	Average Convergence Numbers	Average CPU Time(s)
BA	0.9316497228	0.9099985189	0.9253828803	6.6684e-03	324.9	6.42046
IBA	0.9316823879	0.9297531345	0.9313351223	7.4872e-04	287.44	0.60656

Table 5. Statistical Results of the Series-Parallel System Using Two Algorithms

Algorithm	Best	Worst	Mean	SD	Average Convergence numbers	Average CPU Time(s)
BA	0.9999765928	0.9997004634	0.9999639443	3.9397e-05	11283.08	6.2939
IBA	0.9999766491	0.9999701482	0.9999759817	1.8083e-06	897.16	0.61112

Table 6. Statistical Results of the Complex (bridge) System Using Two Algorithms

Algorithm	Best	Worst	Mean	SD	Average Convergence Numbers	Average CPU Time(s)
BA	0.9998893813	0.9996345736	0.9998677602	6.2831e-05	11738.22	8.32238
IBA	0.9998896376	0.9998893505	0.9998894767	1.439e-07	676.98	0.85636

It can be seen from Table 4, for series system(P1), the best, the worst, and mean values of the results obtained by IBA are all higher than those of BA. It can be illustrated that the IBA has made the better optimal solution, and has a stronger

capability for searching solution space. The total iterations and the average convergence numbers of IBA are significantly lower than those of BA, which shows that the IBA has faster convergence speed. The Average CPU time of IBA and BA are 0.60656s and 6.42046s, respectively, it is shown that the computational efficiency of IBA is significantly higher than that of BA. The standard deviation(SD) of IBA and BA are 7.4872e-04 and 6.6684e-03 respectively, IBA has a smaller standard deviation, which indicates that IBA has better stability than BA.

As above, for P2 and P3, we can clearly see from Table 5 and Table 6, the IBA has obtained large improvements on all measurement indexes than BA. In short, the IBA is greatly superior to BA for solving reliability-redundancy allocation problems.

The comparisons between the best solution of the IBA and other result reported in previous literatures for three benchmark problems(P1,P2,P3) are listed in Table 7-9.

In order to measure the degree of the improvements made by the IBA to the previous best known results, the maximum possible improvement(MPI) index was adopted. And it was described as follows:

$$MPI(\%) = (f_{IBA} - f_{other}) / (1 - f_{other}) \quad (16)$$

Where f_{IBA} represents the best results obtained by IBA, f_{other} represents the best result reported in the previous literatures.

Table 7. Best Results Comparison on Series System

Pamerters	Kuo <i>et al.</i> (1978)	Xu <i>et al.</i> (1990)	Hikita <i>et al.</i> (1992)	Chen (2006)	Hsieh <i>et al.</i> (2011)	Valian <i>et al.</i> (2013)	Kanagaraj <i>et al.</i> (2013)	This paper
f(r,n)	0.9275	0.931677	0.931363	0.931678	0.931682340	0.931682387	0.931682388	0.931682388
n ₁ -n ₅	(3,3,2,3,2)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)
r ₁	0.77960	0.77939	0.777143	0.779266	0.779462304	0.779416938	0.779398871	0.779398875
r ₂	0.80065	0.87183	0.867541	0.872513	0.871883456	0.871833278	0.871837021	0.871837014
r ₃	0.90227	0.90288	0.896696	0.902634	0.902800879	0.902885082	0.902885355	0.902885361
r ₄	0.71044	0.71139	0.717739	0.710648	0.711350168	0.711393868	0.711402515	0.711402516
r ₅	0.85947	0.78779	0.793889	0.788406	0.787861587	0.787803712	0.787799488	0.787799488
MPI(%)	5.769	0.0079	0.4653	0.0064	0.00007	0.000001	0	-
Slack(g1)	27	27	27	27	27	27	27	27
Slack(g2)	0.000010	0.013773	0.000000	0.001559	0.0000005284	0.000000265	0.000000	0.000000
Slack(g3)	10.57248	7.518918	7.518918	7.518918	7.518918	7.518918241	7.51891824	7.518918241

Note: (1) The bold values denote the best values of those obtained by all the algorithms.
 (2)Slack is the unused resources.

Table 8. Best Results Comparison on Series-Parallel System

Pamerters	Hikita <i>et al.</i> (1992)	Hsieh <i>et al.</i> (1998)	Chen (2006)	Wang <i>et al.</i> (2012)	Hsieh <i>et al.</i> (2011)	Valian <i>et al.</i> (2013)	Kanagaraj <i>et al.</i> (2013)	This paper
f(r,n)	0.99996875	0.99997418	0.99997658	0.99997665	0.999976649	0.999976649	0.99997665	0.99997665
n ₁ -n ₅	(3,3,1,2,3)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)
r ₁	0.838193	0.785452	0.812485	0.819596	0.819591561	0.819927087	0.819660256	0.819659319
r ₂	0.855065	0.842998	0.843155	0.845000	0.844951068	0.845267657	0.844981615	0.844980736
r ₃	0.878859	0.885333	0.897385	0.895514	0.895428548	0.895491554	0.895519305	0.895506418
r ₄	0.911402	0.917958	0.894516	0.895519	0.895522339	0.895440692	0.895492245	0.895506432
r ₅	0.850355	0.870318	0.870590	0.868456	0.868490229	0.868318775	0.868447587	0.868447751
MPI (%)	25.28	9.5662	0.2989	0	0.0043	0.0043	0	-
Slack(g1)	53	40	40	40	40	40	40	40
Slack(g2)	0.000000	1.194440	0.002627	0.000007	0.000000	0.0000161	0.000000017	0.000000000
Slack(g3)	7.110849	1.609289	1.609829	1.609289	1.609829	1.6092890	1.60928897	1.6092889667

Note: (1) The bold values denote the best values of those obtained by all the algorithms.
 (2)Slack is the unused resources.

Table 9. Best Results Comparison on Complex (Bridge) System

Pamerters	Hikita. <i>et al.</i> (1992)	Hsieh <i>et al.</i> (1998)	Chen (2006)	Coelho (2009)	Hsieh <i>et al.</i> (2011)	Valian <i>et al.</i> (2013)	Kanagaraj <i>et al.</i> (2013)	This paper
f(r,n)	0.9997894	0.99987916	0.99988921	0.99988957	0.9998893505	0.99988964	0.99988964	0.99988964
n ₁ -n ₅	(3,3,2,3,2)	(3,3,3,3,1)	(3,3,3,3,1)	(3,3,2,4,1)	(3,3,3,3,1)	(3,3,2,4,1)	(3,3,2,4,1)	(3,3,2,4,1)
r ₁	0.814483	0.814090	0.812485	0.826678	0.816624176	0.828094038	0.82808567	0.828086469
r ₂	0.821383	0.864614	0.867661	0.857172	0.868767396	0.858004485	0.85780605	0.857804770
r ₃	0.896151	0.890291	0.861221	0.914629	0.858748781	0.914162924	0.91424006	0.914240669
r ₄	0.713091	0.701190	0.713852	0.648918	0.710279379	0.647907792	0.64814375	0.648146216
r ₅	0.814091	0.734731	0.756699	0.715290	0.753429200	0.704565982	0.70418228	0.704162105
MPI (%)	47.5973	8.6726	0.3881	0.0634	0.2616	0	0	-
Slack(g1)	18	18	18	5	18	5	5	5
Slack(g2)	1.854075	0.376347	0.001494	0.000339	0.000000	0.00007929	0.00000002	0.000000000
Slack(g3)	4.264770	4.264770	4.264770	1.560466	4.264770	1.560466288	1.56046629	1.5604662880

Note: (1) The bold values denote the best values of those obtained by all the algorithms.
 (2)Slack is the unused resources.

It can be seen from Table 7, for series system(P1), the best results reported by Kuo *et al.*[26], Xu *et al.*[27], Hikita *et al.*[28], Chen[6], Hsieh[29] and Valian[18] are 0.9275, 0.931677, 0.931363, 0.931678,0.931682340 and 0.931682387 respectively. The best result of the proposed IBA is better than the best results obtained by the above six algorithms. And the corresponding improvements(measured by MPI) are 5.769%, 0.0079%, 0.4653%, 0.0064%, 0.0007% and 0.000001%. The best result obtained by Kanagaraj[21] is 0.931682388, it is the same as the best results obtained by IBA in this paper. It is worth mentioning that, even if the reliability value is obtained in a very little improvement, it is very important for high reliability application system.

Table 8 shows that, for series parallel system(P2), the best solutions obtained by Hikita *et al.*[28], Hsieh *et al.*[5], Chen[6], Hsieh[29] and Valian[18] are 0.99996875, 0.99997418, 0.99997658,0.999976649 and 0.999976649 respectively. The best solution of the proposed IBA is better than the best results of above five algorithms, the corresponding improvements were 25.28%, 9.5662%, 0.2989%, 0.0043% and 0.0043% respectively. The best result obtained by Wang[31] and Kanagaraj[21] is 0.99997665, it is as good as best results obtained by IBA.

Through the comparison in Table 9, about complex(bridge) system(P3), it is obvious that the best values got by Hikita *et al.*[28], Hsieh *et al.*[5], Chen[6], Coelho[7] and Hsieh[29] are 0.9997894, 0.99987916, 0.99988921, 0.99988957 and 0.9998893505 respectively. The best value of the proposed IBA is better than those obtained by the above five algorithms, the corresponding improvements are improved 47.5973%, 8.6726%, 0.3881%, 0.0634% and 0.2616% respectively. The best result obtained by Valian[18] and Kanagaraj[21] is 0.99988964, it is equal to the best value of the proposed IBA.

Considering the overspeed protection system, Table 11 indicates that the best solutions obtained by Dhingra[30], Yokota *et al.*[24], Chen[6], Coelho[7], Hsieh[29] and kanagaraj[21] are 0.99961, 0.999468, 0.999942, 0.999953 ,0.999954675 and 0.999954675 respectively. The best solution of the proposed has got the better solution than the above six algorithms. The corresponding improvements are 88.3795%, 91.4812%, 21.8621%, 3.5745%, 0.011% and 0.011% respectively. The best results obtained by Valian[18] is 0.99995468, it is as good as the best result obtained by proposed IBA.

From the above, the proposed method has led to better or the same result. It is demonstrated that the presented IBA is an excellent algorithm for solving reliability-redundancy allocation problems.

5. Conclusion

In order to improve the precision and the convergence speed of the bat algorithm, an improved bat algorithm (IBA) was proposed in this paper. In the IBA, the update strategy of position and the strategy of generating local optimal solution are improved. The proposed approach is used to solve three benchmark reliability-redundancy allocation problems. The simulation results show that computational efficiency, convergence rate, precision of the proposed algorithm are all greatly enhanced and it has better stability. Compared with server typical approaches in the previous literatures, the proposed algorithm also meets or exceeds those in the best solution. It is demonstrated that the presented algorithm is effective for reliability-redundancy allocation problems. And it is a promising optimization algorithm.

Acknowledgment

This work supported by the Key Program of National Natural Science Foundation of Heilongjiang No.ZD201309, and Project supported by the Maor International Joint Research Program of China (Grant No. 2014DFB70120).

References

- [1] Chern M.S., "On the computational complexity of reliability redundancy allocation in a series system", *Operation Research Letter*, vol. 11, no. 5, (1992).
- [2] Ardakan M. A. and Hamadani A. Z., "Reliability–redundancy allocation problem with cold-standby redundancy strategy", *Simulation Modelling Practice and Theory*, vol. 42, no. 3, (2014).
- [3] Yu G.B , Ding G, Lin L, Zhao X.F, Zhao Y, Aircraft Engine Fuel Flow Prediction Using Process Neural Network, *International Journal of Control and Automation*, vol. 7, no. 3, (2013), pp. 53-61.
- [4] P.S. You and T.C.Chen, "An efficient heuristic for series–parallel redundant reliability problems," *Computers and Operations Research*, vol. 32, no. 8, (2005).
- [5] Yi-Chih Hsieh, Ta-Cheng Chen and Dennis L. Bricker, "Genetic algorithms for reliability design problems," *Microelectronics Reliability*, vol. 38, no. 10, (1998).
- [6] Ta-Cheng Chen, "IAs based approach for reliability redundancy allocation problems," *Applied Mathematics and Computation*, vol. 182, no. 2, (2006).
- [7] Coelho L.S., "Reliability–redundancy optimization by means of a chaotic differential evolution approach", *Chaos, Solitons and Fractals*, vol. 41, no. 2, (2009).
- [8] Y.C. Liang and Y.C. Chen, "Redundancy allocation of series–parallel systems using a variable neighborhood search algorithm," *Reliability Engineering and System Safety*, vol. 92, no. 8, (2007).
- [9] Garg H.,Rani M.,Sharma S.P. and Vishwakarma Y., "Bi-objective optimization of the reliability-redundancy allocation problem for series-parallel system", *Journal of Manufacturing Systems*, vol. 33, no. 3, (2014).
- [10] Wang Y. and Li L., "A PSO algorithm for constrained redundancy allocation in multi-state systems with bridge topology", *Computers & Industrial Engineering*, vol. 68, no. 2, (2014).
- [11] Garg H. and Sharma S.P., "Multi-objective reliability-redundancy allocation problem using particle swarm optimization", *Computers & Industrial Engineering*, vol. 64,no. 1, (2013).
- [12] Zou D. *et al.*, "A novel global harmony search algorithm for reliability problems", *Computers & Industrial Engineering*, vol. 58,no. 2, (2010).
- [13] Zou D. *et al.*, "An effective global harmony search algorithm for reliability problems", *Expert Systems with Applications*, vol. 38, no. 4,(2011).
- [14] Agarwal M. and Sharma V.K., "Ant colony approach to constrained redundancy optimization in binary systems", *Applied Mathematical Modelling*, vol. 34, no. 1, (2010).
- [15] Ouzineb M.,Nourelfath M. and Gendreau M., "Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems", *Reliability Engineering and System Safety*, vol. 93,no. 8, (2008).
- [16] Afonso L.D.,Mariani V.C. and Coelho L.S., "Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization", *Expert Systems with Applications*, vol. 40, no. 9, (2013).

- [17] Safaei N.,Tavakkoli-Moghaddam R. and Kiassat C., “Annealing-based particle swarm optimization to solve the redundant reliability problem with multiple component choices”, *Applied Soft Computing*, vol. 12,no. 11, **(2012)**.
- [18] Valian E., “Improved cuckoo search for reliability optimization problems”, *Computers & Industrial Engineering*, vol. 64, no. 1, **(2013)**.
- [19] Khalili-Damghani K., Abtahi A.R. and Tavana M., “A new multi-objective particle swarm optimization method for solving reliability redundancy allocation problems”, *Reliability Engineering and System Safety*, vol. 111, no. 3, **(2013)**.
- [20] Chen S.M.,Sarosh,A. and Dong Y.F., “Simulated annealing based artificial bee colony algorithm for global numerical optimization”, *Applied Mathematics and Computation*, vol. 219, no. 8, **(2012)**.
- [21] Kanagaraj G.,Ponnambalam S.G. and Jawahar N., “A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems”, *Computers & Industrial Engineering*, vol. 66, no. 4, **(2013)**.
- [22] Gopal K.,Aggarwal K.and Gupta J.S., “An improved algorithm for reliability optimization”. *IEEE Transactions on Reliability*, vol. 27, no. 5, **(1978)**.
- [23] Coit D.W. and Smith A.E., “Redundancy allocation to maximize a lower percentile of the system time-to-failure distribution”, *IEEE Transactions on Reliability*, vol. 47, no. 1, **(1998)**.
- [24] T. Yokota, M. Gen and Y.X. Li, “Genetic algorithm for non-linear mixed integer programming problems and its applications,” *Comput. Ind. Eng.*, vol. 30, no. 4, **(1996)**.
- [25] W. Kuo and V.R. Prasad, “An annotated overview of system-reliability optimization,” *IEEE Transaction on Reliability*, vol. 49, no. 2, **(2000)**.
- [26] Kuo, W., Hwang, C.L., and Tillman, F.A., “A note on heuristic methods in optimal system reliability”. *IEEE Transactions on Reliability*, vol. R-27, no. 5, **(1978)**.
- [27] Xu Z., Kuo W. and Lin H.H.,“Optimization limits in improving system reliability”,*IEEE Transactions on Reliability*,vol. 39,no. 1, **(1990)**.
- [28] M. Hikita, Y. Nakagawa and H. Harihisa, “Reliability optimization of systems by a surrogate constraints algorithm,” *IEEE Trans. Reliab.*, vol. 41, no. 3, **(1992)**.
- [29] Hsieh Y.C. and You P.S., “An effective immune based two-phase approach for the optimal reliability–redundancy allocation problem”. *Applied Mathematics and Computation*, vol. 218, no. 4, **(2011)**.
- [30] Anoop K. Dhingra, “Optimal Apportionment of Reliability & Redundancy in Series Systems Under Multiple Objectives,” *IEEE Transactions on Reliability*, vol. 41,no. 4, **(1992)**.
- [31] Wang L. and Li L., “A coevolutionary differential evolution with harmony search for reliability-redundancy optimization”, *Expert Systems with Applications*, vol. 39, no. 5, **(2012)**.
- [32] Yang X.S. and Gandomi A.H., “Bat algorithm: a novel approach for global engineering optimization”. *Engineering Computations*, vol. 29, no. 5, **(2012)**.

