International Journal of Multimedia and Ubiquitous Engineering
Vol. 9, No. 12 (2014), pp. 375-382
http://dx.doi.org/10.14257/ijmue.2014.9.12.33

Reliability Redundancy Optimization Algorithm based on Eagle Strategy and PSO

Yubao Liu[1,2] and Guihe Qin[1,3]

[1]*College of Computer Science and Technology, Jilin University, Changchun, China*
[2]*College of Computer Science and Technology, Changchun University, Changchun, China*
[3]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China*
[1,2]*lyb_mailbox@126.com*, [1,3] *qingh@jlu.edu.cn*

Abstract

A new algorithm combined Eagle Strategy with PSO is proposed. The new algorithm performs by two phases: First Eagle Strategy is used to do global search; Second PSO algorithm is used to do fast local search around a promising solution. The balance of global search and local search is considered simultaneously. It is not only help to jump out of local optimum but also accelerate local convergence. Experimental results on three benchmark problems illustrate that the presented approach is effective, efficient and accurate for solving reliability redundancy optimization problems.

Keywords: *nonlinear programming, Eagle Strategy, PSO, reliability redundancy optimization*

1. Introduction

The reliability redundancy optimization problems are very important in modern industry. In general, System reliability can be increased by two major ways: raising the reliability of components and using redundant components. Usually the last way is by providing the components reliability selection and components redundancy numbers to get the highest system reliability. But the cost, weight, volume *etc.*, So it forms a difficult optimization problem with no-linear constraints on the cost, weight and volume etc. Such reliability redundancy optimization problems are called as RRAP (reliability redundancy allocation problem) [1]. RRAP has been proven to be NP-hard problem. So far many heuristics and meta-heuristics algorithms have been widely studied and used to solve reliability redundancy optimization problems [4]. They offer better feasible solution.

In general, reliability redundancy allocation problems are defined as maximizing the system reliability subject to multiple nonlinear constraints. They are belong to nonlinearly mixed-integer programming problems and can be described as following model uniformly:

$$\text{Max } R_s = f(r,n)$$
s.t.
$$g_j(r,n) \le b_j, j=1,\ldots,m; n_j \in \text{ positve integer}, 0 \le r_j \le 1 \qquad (1)$$

Where r_i and n_i are the reliability and the number of components of *i*th subsystem respectively. The f(r,n) is the objective function; the $g_j(r,n)$ is the jth constraint function; b_j is the *j*th upper limitation of the constraints; m is the number of subsystems.

ISSN: 1975-0080 IJMUE
Copyright ⓒ 2014 SERSC

For solving the system reliability optimization problems, many meta-heuristics methods have been proposed [6-9, 15, 22]. Recently some hybrid meta-heuristic methods have been proposed to solve the reliability redundant allocation problems [11].

In this paper, a reliability redundancy optimization algorithm based on Eagle Strategy and PSO is proposed. It is used to solve three problems on reliability redundancy optimization problems. And this method is demonstrated that it is effectiveness for reliability redundancy optimization problems.

2. Eagle Strategy and PSO

2.1. Eagle Strategy

Eagle strategy is a two-stage optimization strategy which presented by X. S. Yang and S. Deb in 2010[2]. The algorithm imitates feeding behavior of the eagle. When hunting, Eagle first searches target for a wide range in the sky. Once the prey is found, Eagle can accurately locate target and dive to capture prey at very fast speed. The search process of eagle in the sky is much like global search in the solution space. When feasible solution is found, a fast local search around the feasible solution is done to find the optimal solution. This process is similar to the process that eagle captures prey rapidly. Therefore, the eagle strategy is a two-stage search process: the first stage is global search in the solution space; the second phase is local search near the feasible solution. The trade-off between global search and local search is considered in two stages simultaneously. It is not only help to avoid the local optima but also accelerate the convergence speed.

2.2. The Particle Swarm Optimization (PSO)

Particle Swarm Optimization is inspired by feeding action of birds. It is a meta-heuristic algorithm to be used to solve the optimization problems [16], the solution is related to the position of the bird (called "Particle") in the solution space. In the process of flight, every particle has its own position and speed. They are used to determine the direction and distance. Each particle also has a fitness value calculated by objective function. This value is utilized to evaluate the current particle.

In PSO algorithm, the optimal solution is got by iterations. The position and velocity of particles are updated by formula (1) and (2) in every iteration. They are described as follows:

$$v_{id}^{t+1} = v_{id}^{t} + c_1 r_1 \times (pbest_{id}^{t} - x_{id}^{t}) + c_2 r_2 \times (gbest_{id}^{t} - x_{id}^{t}) \quad (2)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (3)$$

Where, the pbest is personal optimal value point of the particle i. The gbest is global optimal value of all particles. The parameters c_1 and c_2 are accelerating factor, usually $c_1 = c_2 = 2$. The parameters r_1 and r_2 are random number, and $r_1, r_2 \in [0,1]$.

3. The Algorithm based on Eagle Strategy and PSO

Eagle strategy uses Lévy walk to generate new solution. Lévy walk produces a stochastic step length following Lévy distribution. Its probability density function is [12]:

$$\text{Lévy} \sim u = t^{-\lambda}, (1<\lambda\leq 3) \quad (4)$$

When a new solution is produce, the following Lévy flight is applied [13]:

$$x_i^{t+1} = x_i^{t} + \alpha \oplus \text{Lévy}(\lambda) \quad (5)$$

where $\alpha > 0$ is scale factor.

Here we assume that Lévy(λ) = s, so the formula can also be described as follows:

$$x_i^{t} = x_i^{t+1} + \alpha \oplus s \quad (6)$$

 Copyright ⓒ 2014 SERSC

Where s is random step, it can be calculated by the formula (7) [14]:

$$s = \frac{\mu}{|\nu|^{1/\beta}} \tag{7}$$

The μ and ν obey the normal distribution respectively as follows:

$$\mu \sim N(0, \sigma_\mu^2), \nu \sim N(0, \sigma_\nu^2) \tag{8}$$

$$\sigma_\mu = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right)^{1/\beta}, \sigma_\nu = 1 \tag{9}$$

Γ is the Gamma function. $\beta = \lambda - 1$ and $\beta \in (0, 2]$.

PSO is an excellent algorithm for optimization problems. We proposed a new approach combined PSO with Eagle Strategy. The optimization process is divided into two phases: First Eagle Strategy is used to do global search; Second, PSO algorithm is used to do fast local search around a promising solution. The proposed algorithm considers the balance of the global search and local search at the same time. It is not only help to jump out of local optimum but also speed up the local convergence. Its main procedure of proposed algorithm is shown as follows:

Begin
  Initialize a random population x
  While(stop criterion)
    For i = 1 to M
      $x_i^{t+1} = x_i^t + \alpha \oplus s$
    EndFor
    Find the promising solution gbest
    For i = 1 to M
      $V_{id}^{t+1} = \omega V_{id}^t + c_1 r_1(pbest_{id} - x_{id}^t) + c_2 r_2(gbest_d - x_{id}^t)$
      $X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1}$
    EndFor
    If $F(x_i^{t+1}) < F(pbest_i)$
      Updating $pbest_i$
    EndIf
    If (a better solution is found)
      Updating gbest
    EndIf
  EndFor
  t = t + 1
End
  Output the gbest
End

4. Simulations and Comparisons

In this section, the proposed algorithm is applied to solve three benchmark reliability redundancy optimization problems. The results are compared with some other typical methods from the literatures.

A penalty function is used to handle constrains, it can be described as follows:

$$\min \quad F(x) = -f(x) + \lambda \sum_{j=1}^{p} \max\{0, g_j(x)\} \tag{10}$$

Where F(x) is penalty function, f(x) represents objective function. gj(x), (j = 1, 2, …, p) represents the jth constraint, and λ is a large positive constant which imposes penalty on unfeasible solutions, and it is named as penalty coefficient. It is set to 10^{15} here.

4.1. P1: Series System

The problem [17] is shown as Figure 1:

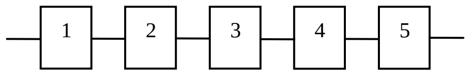

Figure 1. Series System

It is described by formula (11):

$$
\begin{aligned}
&Max \quad f(r,n) = \prod_{i=1}^{m} R_i(n_i) \\
&s.t. \quad g_1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 \le V \\
&\qquad g_2(r,n) = \sum_{i=1}^{m} \alpha_i(-1000 / \ln r_i)^{\beta_i} (n_i + \exp(n_i/4)) \le C \\
&\qquad g_3(r,n) = \sum_{i=1}^{m} w_i n_i \exp(n_i/4) \le W \\
&\qquad 0 \le ri \le 1, n_i \in Z^+, 1 \le i \le m
\end{aligned}
\tag{11}
$$

Wherein m is the number of subsystems. n_i is the number of components of subsystem i. R_i (n_i) is the reliability of subsystem i. f(r,n) is the reliability of the system; w_i ,v_i and r_i is the weight of each component, the volume and the reliability of each component in subsystem i respectively; $\alpha_i(-1000/\ln r_i)\beta_i$ is the cost of every component in *i*th subsystem, α_i and β_i is the constant value(usually given),1000 is the work time of the components(commonly denoted by Tm); V ,C and W is the upper limit of total volume ,total cost and total weight of the system respectively. The parameters for this case are given 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			

The presented algorithm runs 50 times independently, the results are as follows:

Table 2. Best Results Comparison on Series System

Parameter	Hikita et al. [34]	Kuo et al.[40]	Chen[16]	Xu et al. [12]	This paper
f(r,n)	0.931363	0.9275	0.931678	0.931677	**0.9316824**
n1	3	3	3	3	3
n2	2	3	2	2	2
n3	2	2	2	2	2
n4	3	3	3	3	3
n5	3	2	3	3	3
r1	0.777143	0.77960	0.779266	0.77939	0.7793997
r2	0.867541	0.80065	0.872513	0.87183	0.8718379
r3	0.896696	0.90227	0.902634	0.90288	0.9028848
r4	0.717739	0.71044	0.710648	0.71139	0.7114028
r5	0.793889	0.85947	0.788406	0.78779	0.7877971
MPI(%)	0.465	5.767	0.006	0.008	-

Copyright ⓒ 2014 SERSC

Slack(g1)	27	27	27	27	27
Slack(g2)	0.000000	0.000010	0.001559	0.013773	0.000000007
Slack(g3)	7.518918	10.57248	7.518918	7.518918	7.518918

Note: (1) the bold values denote the best values of those obtained by all the algorithms.
(2) MPI (%) = $(f - f_{other})/(1 - f_{other})$.
(3)Slack is the unused resources.

It can be seen from Table 2, that the best results reported by Hikita, *et al.*, Hsieh, *et al.*, Chen and Xu et al. were 0.931363, 0.9275, 0.931678 and 0.931677 for the series system respectively. The result obtained by the proposed algorithm is better than the above four best solution, and the corresponding improvements made by the presented method are 0.465%, 5.767% , 0.006% and 0.008% respectively.

4.2. P2: Series-parallel System

The problem [18] is shown as Figure 2:

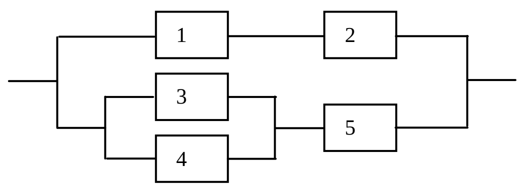

Figure 2. Series-parallel System

It is described by formula (12):

$$Max \quad f(r,n) = 1 - (1 - R_1 R_2)(1 - (1 - (1 - R_3)(1 - R_4)) R_5) \qquad (12)$$

Its constraints are the same as series system. The parameters for this case are set in Table 3:

Table 3. 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			

The presented algorithm runs 50 times independently, the results are as follows:

Table 4. Best Results Comparison on Series Parallel System

Parameter	Hikitaet al.[29]	Hsieh et al. [11]	Chen[12]	This paper
f(r,n)	0.99996875	0.99997418	0.99997658	**0.99997665**
n_1	3	2	2	2
n_2	3	2	2	2
n_3	1	2	2	2
n_4	2	2	2	2
n_5	3	4	4	4
r_1	0.838193	0.785452	0.812485	0.819655
r_2	0.855065	0.842998	0.843155	0.844975
r_3	0.878859	0.885333	0.897385	0.895509
r_4	0.911402	0.917958	0.894516	0.895509
r_5	0.850355	0.870318	0.870590	0.868449
MPI (%)	25.28	9.56	0.30	-
Slack(g1)	53	40	40	40

Copyright ⓒ 2014 SERSC

Slack(g2)	0.000000	1.194440	0.002627	0.000000008
Slack(g3)	7.110849	1.609289	1.609829	1.609289

Note: (1) the bold values denote the best values of those obtained by all the algorithms.
(2) MPI (%) = $(f - f_{other})/(1 - f_{other})$.
(3)Slack is the unused resources.

It can be seen from Table 4, that the best results reported by Hikita, *et al.*, Hsieh, *et al.*, and Chen were 0.99996875, 0.99997418 and 0.99997658 for the series–parallel system respectively. The result obtained by the proposed algorithm is better than the above three best solution, and the corresponding improvements made by the presented method are 25.28%, 9.56% and 0.30% respectively.

4.3. P3: Complex (Bridge) System

This problem [19] is shown as Figure 3:

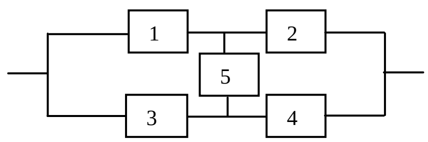

Figure 3. Complex (Bridge) System

It is described by formula (13):

$$\begin{aligned} \text{Max} \quad f(r,n) = {} & R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 \\ & - R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_2R_4R_5 \\ & - R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \end{aligned} \quad (13)$$

Its constraints are the same as series system. The parameters for this case are listed in Table 1:

The presented algorithm runs 50 times independently, the results are as follows:

Table 5. Best Results Comparison on Complex (Bridge) System

Parameter	Hikita. et al.[18]	Hsieh et al. [6]	Chen [7]	Coelho [10]	This paper
f(r,n)	0.9997894	0.99987916	0.99988921	0.99988957	**0.99988964**
n_1	3	3	3	3	3
n_2	3	3	3	3	3
n_3	2	3	3	2	2
n_4	3	3	3	4	4
n_5	2	1	1	1	1
r_1	0.814483	0.814090	0.812485	0.826678	0.828082
r_2	0.821383	0.864614	0.867661	0.857172	0.857812
r_3	0.896151	0.890291	0.861221	0.914629	0.914241
r_4	0.713091	0.701190	0.713852	0.648918	0.648155
r_5	0.814091	0.734731	0.756699	0.715290	0.704066
MPI (%)	47.596	8.671	0.386	0.061	-
Slack(g1)	18	18	18	5	5
Slack(g2)	1.854075	0.376347	0.001494	0.000339	0.000000008
Slack(g3)	4.264770	4.264770	4.264770	1.560466	1.560466

Note: (1) the bold values denote the best values of those obtained by all the algorithms.
(2) MPI (%) = $(f - f_{other})/(1 - f_{other})$.
(3)Slack is the unused resources.

It can be seen from Table 5, that the best results reported by Hikita, *et al.*, Hsieh, *et al.*, Chen and Coelho were 0.9997894, 0.99987916, 0.99988921 and 0.99988957 for the complex

 Copyright ⓒ 2014 SERSC

(bridge) system respectively. The result obtained by the proposed algorithm is better than the above four best solution, and the corresponding improvements made by the presented method are 47.596%, 8.671%, 0.386% and 0.061% respectively.

The statistical results comparison of three benchmark problems are listed in Table 6, Table 7 and Table 8 including the best results(Best), the worst results(Worst), the mean results (Mean)and standard deviation(SD).

Table 6. Statistical Results Comparison on Series System

Algorithm	Best	Worst	Mean	SD
ABC[21]	0.931682	NA	0.930580	8.14E-04
IA[8]	0.931682340	NA	0.931682222	1.3E-14
This paper	0.9316824	0.931536	0.9316622	3.79E-06

Table 7. Statistical Results Comparison on Series Parallel System

Algorithm	Best	Worst	Mean	SD
ABC[21]	0.99997731	NA	0.99997517	2.89E-06
CDEHS[15]	0.99997665	0.99996475	0.99997365	4.3E-06
This paper	0.99997665	0.99997652	0.99997662	3.91E-07

Table 8. Statistical Results Comparison on Complex (Bridge) System

Algorithm	Best	Worst	Mean	SD
ABC[21]	0.99988962	NA	0.99988362	1.03E-05
PSO [10]	0.99988957	0.99987750	0.99988594	6.9E-07
EGHS[9]	0.99988960	0.99982887	0.99988263	1.6E-05
CDEHS[15]	0.99988964	0.99988931	0.99988940	1.9E-07
This paper	0.99988964	0.99988811	0.99988914	4.32E-07

It can be clearly seen from Table 6 that the proposed algorithms in this paper have best value in terms of the best results and better value in terms of the mean results.

From Table 7, it can be seen that the proposed method can get best value about the best results and the worst results, and get better value about the average results.

Through the comparison in Table 8, we can see that the proposed method can find better value than ABC, PSO and EGHS in terms of performance indexes, and get the same good value as CDEHS on the best results.

5. Conclusions

In this paper, a reliability redundancy optimization algorithm based on eagle strategy combined with PSO is proposed. It is applied to solve the reliability redundancy optimization problems. This method works by two phases: one is to do global search by Eagle Strategy, the other is to do fast local search around a promising solution by PSO algorithm. The proposed algorithm considers the trade-off between the global search and local search at the same time. It is not only help to avoid local optimum but also enhance speed of the local convergence. Simulation experiments based on three benchmark problems and compared with some methods in literatures. The results showed that the proposed algorithm was effective, efficient and accurate for reliability redundancy optimization problems.

Acknowledgments

The authors wish to thank Department of education of Jilin Province, China. This work was supported by the science research project of Department of education of Jilin Province, China (No. 2013461). This work was also partially supported by Jilin Provincial Science & Technology Department, project number is 20140204048GX.

Copyright ⓒ 2014 SERSC

References

[1] M. A. Ardakan and A. Z. Hamadani, "Reliability–redundancy allocation problem with cold-standby redundancy strategy", Simulation Modelling Practice and Theory, vol. 42, **(2014)**.

[2] A. H. Gandomi, X. S. Yang, S. Talatahari and S. Deb, "Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization", Computers and Mathematics with Application, vol. 63, no. 1, **(2012)**.

[3] V. R. Prasad and W. Kuo, "Reliability optimization of coherent system", IEEE Trans. Reliab, vol. 49, no. 3, **(2000)**.

[4] H. Garg, M. Rani, S. P. Sharma and Y. Vishwakarma, "Bi-objective optimization of the reliability-redundancy allocation problem for series-parallel system", Journal of Manufacturing Systems, vol. 33, no. 3, **(2014)**.

[5] W. Kuo, V. R. Prasad, F. A. Tillman and C. L. Hwang, "Optimal Reliability Design: Fundamentals and Applications", Cambridge University Press, Cambridge, **(2001)**.

[6] Y.-C. Hsieh, T.-C. Chen and D. L. Bricker, "Genetic algorithms for reliability design problems", Microelectronics Reliability, vol. 38, no. 10, **(1998)**.

[7] T.-C. Chen, "IAs based approach for reliability redundancy allocation problems", Applied Mathematics and Computation, vol. 182, no. 2, **(2006)**.

[8] Y.-C. Hsieh and P.-S. You, "An effective immune based two-phase approach for the optimal reliability–redundancy allocation problem", Applied Mathematics and Computation, vol. 218, no. 4, **(2011)**.

[9] D. Zou, L. Gao, S. Li and J. Wu, "An effective global harmony search algorithm for reliability problems", Expert Systems with Applications, vol. 38, no. 4, **(2011)**.

[10] L. dos Santos Coelho, "An efficient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications", Reliability Engineering and System Safety, vol. 94, no. 4, **(2009)**.

[11] N. Safaei, R. Tavakkoli-Moghaddamb and C. Kiassat, "Annealing-based particle swarm optimization to solve the redundant reliability problem with multiple component choices", Applied Soft Computing, vol. 12, no. 11, **(2012)**.

[12] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms", 2nd Ed, Luniver Press, **(2010)**.

[13] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights", In Proceedings of World congress on nature & biologically inspired computing, **(2009)** December 9-11, Coimbatore, India.

[14] R. N. Mantegna and H. E. Stanley, "Stochastic Process with Ultraslow Convergence to a Gaussian: The Truncated Lévy Flight", Physical Review Letters, vol. 73, no. 22, **(1994)**.

[15] L. D. Afonso, V. C. Mariani and L. dos Santos Coelho, "Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization", Expert Systems with Applications, vol. 40, no. 9, **(2013)**.

[16] Z-H. Zhan, J. Zhang, Y. Li and Y-H Shi, "Orthogonal Learning Particle Swarm Optimization," IEEE Transactions on Evolutionary Computation, vol. 15, no. 6, **(2011)**.

[17] K. Gopal, K. Aggarwal and J. S. Gupta, "An improved algorithm for reliability optimization", IEEE Trans. Reliab, vol. 27, no. 5, **(1978)**.

[18] M. Hikita, Y. Nakagawa and H. Harihisa, "Reliability optimization of systems by a surrogate constraints algorithm", IEEE Trans. Reliab, vol. 41, no. 3, **(1992)**.

[19] 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)**.

[20] A. K. Dhingra, "Optimal Apportionment of Reliability & Redundancy in Series Systems Under Multiple Objectives", IEEE Transactions on Reliability, vol. 41, no. 4, **(1992)**.

[21] W. C. Yeh and T. J. Hsieh, "Solving reliability redundancy allocation problems using an artificial bee colony algorithm", Comput. Oper. Res, vol. 38, no. 11, **(2011)**.

[22] H. Garg and S. P. Sharma, "Multi-objective reliability-redundancy allocation problem using particle swarm optimization", Computers & Industrial Engineering, vol. 64, no. 1, **(2013)**.

[23] W. Kuo, C. L. Hwang and F. A. Tillman, "A note on heuristic methods in optimal system reliability", IEEE Transactions on Reliability, vol. R-27, no. 5, **(1978)**.

 Copyright ⓒ 2014 SERSC