International Journal of Hybrid Information Technology
Vol.8, No. 5 (2015), pp. 113-118
http://dx.doi.org/10.14257/ijhit.2015.8.5.12

A DE Algorithm Combined with Lévy Flight for Reliability Redundancy Allocation Problems

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

In this paper, a differential evolution (DE) algorithm combined with Lévy flight is proposed to solve the reliability redundancy allocation problems. The Lévy flight is incorporated to enhance the ability of global search of differential evolution algorithm. DE is used for local search mainly. The method considers the trade-off of the diversification and the intensification simultaneously. Experimental results for three benchmark problems demonstrate that the proposed algorithm is more effective for solving the reliability redundancy allocation problems.

Keywords: *nonlinear programming, differential evolution, Lévy flight, reliability redundancy allocation*

1. Introduction

The reliability redundancy allocation problems are very important in industry and engineering fields. Usually two main ways have been used to enhance system reliability. They are by increasing the reliability of components and by using redundant components in the subsystems. The reliability redundancy allocation problems (RRAP) of maximizing the system reliability obey multiple nonlinear constraints [1].They belong to mixed integer programming problems. It can be formulated as following model uniformly:

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

Herein r_i is the reliability of *i*th subsystem, n_i is the count of components of *i*th subsystem. The f (.) is the objective function; the g_j (.) is the *j*th constraint function; bj is the *j*th upper limitation; m is the number of subsystems. The goal of RRAP is to get the number of redundant components and the components' reliability in each subsystem in order to maximize the overall system reliability.

RRAP has been proven to be NP-hard problem. It has been studied for decades. A lot of different optimization technologies have been utilized to resolve it. Some methods called heuristics and meta-heuristics have been presented and applied [2-6].Recently some hybrid meta-heuristic methods have been proposed to solve the reliability redundant allocation problems [7-8], [12].

In this paper, a DE combined with Lévy flight algorithm is proposed. This method considers the trade-off of the diversification and the intensification simultaneously. It is used to solve three problems on reliability redundancy allocation problems. The experimental results demonstrate that the proposed algorithm has higher precision more effectiveness for reliability redundancy allocations problems.

ISSN: 1738-9968 IJHIT

2. The Algorithm Based on DE and Lévy Flight

2.1. Lévy Flight

In random search strategy, Lévy flight is a kind of random walk model used widely. Its walking step follows a heavy-tailed (heavy tailed) distribution (named Lévy distribution). It is named for by French famous mathematician Pierre Lévy who suggests it.

In the nature, birds and insects find the food in a random way. In general, the search path of these animals is an effectively random flight because the next walk step is based on both the current position and the transition probability to the next position. The studies show that the flight action of many birds and insects proves the representative characteristic of Lévy flight. The flight distance from origin of these animals tends to a stable distribution after a lot of steps. Xin-she Yang simplifies the Lévy distribution and does Fourier transform, then gets the probability density function of Lévy distribution [13] as follows:

$$\text{Lévy} \sim u = t^{-\lambda},\ (1<\lambda\leq 3) \tag{2}$$

It has an infinite variance. When $\lambda=3$ it corresponds to Brownian motion, while $\lambda=1$ it has random tunneling function which can be more efficient to jump out local optima.

For simplifying to be easy to program, a formulas for simulating Lévy flight proposed by Mantegna [15] is adopted. That is

$$s = \frac{\mu}{|\nu|^{1/\beta}} \tag{3}$$

Where s is random step, it obeys Lévy distribution. The μ and ν follow the normal distribution respectively as follows:

$$\mu \sim N(0, \sigma_\mu^2),\ \nu \sim N(0, \sigma_\nu^2) \tag{4}$$

$$\sigma_\mu = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right)^{1/\beta},\ \sigma_\nu = 1 \tag{5}$$

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

2.2. Differential Evolution Algorithm (DE)

Differential evolution algorithm is an excellent evolutionary algorithm using real number code. Compared with the former genetic algorithm, differential evolution algorithm produces new population by mutation and crossover operations, and then uses the competition strategy of one to one to update the population. Now some variant of the DE algorithms have been appeared. But the DE/rand/1/bin has been widely used. This procedure is described as follows:

S1: Initialize parameters F, CR and M. Wherein F is scale factor, CR is crossover rate; M is the number of population.

S2: Randomly generating initial population.

S3: Evaluate the population.

S4: Mutation.

S5: Crossover.

Step 6: Selection.

Step 7: stopping criterion.

If the stopping criterion is satisfied, the procedure is end. Otherwise, go back to S4.

3. The DE Algorithm Combined with Lévy Flight

The proposed algorithm used a random walk method called Lévy flight for enhanced global random search. Then differential evolution algorithm was adopted to realize the local quick search. When generating new *k*th solutions, the Lévy flight is used to change position location of the global optimal solution. It is advantage to avoiding the local optima. The formula [14] is as follows:

$$vbest_i^k = xbest_i^{k-1} + \alpha \oplus Lévy(\lambda) \quad (6)$$

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

$$vbest_i^k = xbest_i^{k-1} + \alpha \oplus s \quad (7)$$

Where α is step size, it should be set according to the scale of the optimization problem.

In DE algorithm, the local evolutionary strategy is adopted to generate the new candidate solution. This can increase the speed of getting the global optimal solution. These consider the diversification and the intensification of algorithms simultaneously. The local evolutionary strategy is shown as:

$$V_i^k = xbest_i^{k-1} + F\times (x_{i1}^{k-1} - x_{i2}^{k-1}) \quad (8)$$

v_i^k is the trial vector. The x_{i1}^k, x_{i2}^k are two different individuals randomly selected from *(k-1)*th generation population, i_1,i_2 is random number ranged from 1 to M, and mutation factor F is a scale factor.

The main procedure of the algorithm is shown as follows:

Begin
Objective function f(x), $x = (x_1, x2,\ldots,x_d)^T$
Generating initial population
Get the current optimal solution xbest
While (t<MaxGeneration) or (stop criterion)
$vbest_i^k = xbest_i^{k-1} + \alpha \oplus s$
Calculating the fitness value of $vbest_i^k$
If (fitness ($vbest_i^k$) is better than fitness ($xbest_i^{k-1}$))
    Xbestik-1 = vbestik
End If
For i = 1 to M
    Randomly generate three integers i_1, i_2 in [1, M], and $i_1 \neq i_2 \neq i$.
    $v_i^k = xbest_i^{k-1} + F\times (x_{i1}^{k-1} - x_{i2}^{k-1})$
    Randomly generate an integer r_d in the range [1, N]
    For j = 1 to N
        If rand < CR or j = r_d
            $u_{i,j}^k = v_{i,j}^k$
        Elseif
            $u_{i,j}^k = x_{i,j}^{k-1}$
        End If
    EndFor
    If $f(u_i^k) < f(x_i^{k-1})$
        $x_i^k = u_i^k$
    Elseif
        $x_i^k = x_i^{k-1}$
    End If
EndFor
Get the current optimal solution $xbest^k$ from this generation population
If (a better solution is found)
Update the current optimal solution
End If
t = t + 1
End While

Post press the result and visualization
End

4. Case Studies and Comparisons

In this part, the simulations based on three benchmark problems to test the performances of the proposed method for reliability redundancy allocation problems are implemented. And we compared with some other typical algorithms from the former literatures.

To resolve the problem of violation of constraints, a penalty function approach is used to handle constrains. That is

$$\min \quad F(x) = -f(x) + \lambda \sum_{j=1}^{p} \max \quad \{0, g_j(x)\}^2 \tag{9}$$

Where F (x) represents penalty function, f (x) represents objective function. $g_j(x)$, (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. This penalty function is used to convert the constrained optimization to unconstrained optimization.

4.1. Case Study 1: Series-Parallel System

This case study [9] is shown as Figure 1:

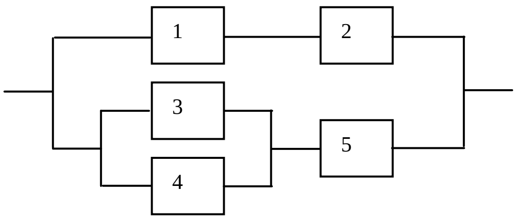

Figure 1. Series-Parallel System

It is formulated as follows:

$$\begin{aligned} &Max \quad f(r,n) = 1-(1-R_1R_2)(1-(1-(1-R_3)(1-R_4))R_5) \\ &s.t. \quad g_1(r,n) = \sum_{i=1}^{m} w_i v_i^2 n_i^2 \le V \\ &\quad g_2(r,n) = \sum_{i=1}^{m} \alpha_i(-1000 \ / \ln r_i)^{\beta_i}(n_i + \exp(n_i/4)) \le C \\ &\quad g_3(r,n) = \sum_{i=1}^{m} w_i n_i \exp(n_i/4) \le W \end{aligned} \tag{10}$$

Wherein m is the number of subsystems, n_i is the number of components of ith subsystem, R_i (n_i) is the reliability of ith subsystem , f (.) is the reliability of the system; w_i is the weight of each component in ith subsystem, v_i is the volume of each component in ith subsystem; r_i is the reliability of each component in ith subsystem; The item $\alpha_i(-1000/\ln r_i)^{\beta_i}$ is the cost of each component in ith subsystem , the parameters α_i and β_i is the constant value(usually assume that have been given),1000 is the task time of the components(it is commonly expressed in T_m); V is the upper limit of total volume of the system, C is the upper limit of total cost of the system, W is the upper limit of total weight of the system. The values of parameters are set in Table 1:

Table 1. The Parameters of Series-Parallel System

Subsy stem i	1 $0^5\alpha_i$	i	w $_i v_i^2$	i		(	
1	2 .500	.5	2	.5	80	75	00
2	1 .450	.5	4	.0			
3	0 .541	.5	5	.0			
4	0 .541	.5	8	.5			
5	2 .100	.5	4	.5			

4.2. Case Study 2: Complex (bridge) System

This Case study [10] is shown as Figure 2:

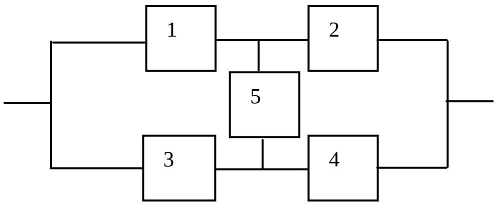

Figure 2. Complex (bridge) System

It is formulated as follows:

$$\begin{aligned} Max \quad f(r,n) &= R_1R_2 + R_3R_4 + R_1R_4R_5 + R_2R_3R_5 \\ &- R_1R_2R_3R_4 - R_1R_2R_3R_5 - R_1R_2R_4R_5 \\ &- R_1R_3R_4R_5 - R_2R_3R_4R_5 + 2R_1R_2R_3R_4R_5 \end{aligned} \tag{11}$$

The constraints are the same as case study 1. The values of parameters are listed in Table 2:

Table 2. The Parameters of Complex (bridge) System

Subsys tem i	1 $0^5\alpha_i$	i	[v $_i v_i^2$	i		C	V
1	2 .33	.5	1		10	1 75	2 00
2	1 .450	.5	2				
3	0 .541	.5	3				
4	8 .050	.5	4				
5	1 .950	.5	2				

4.3. Case study 3: Over Speed Protection System

The Case study 3 is used to over speed protection of a gas turbine. Once the over speed occurs, the system will be stop. This problem [11] is shown as Figure 3:

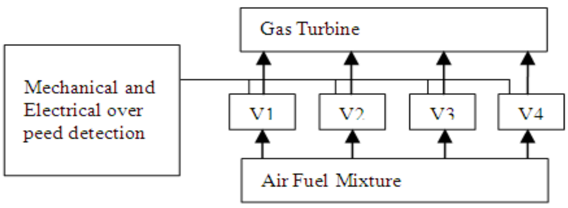

Figure 3. The Over Speed Protection System of a Gas Turbine

This system can be viewed as an N-stage (N=4) mixed series-parallel systems. It is formulated as follows:

$$\begin{aligned}
Max \quad & f(r,n) = \prod_{i=1}^{m}[1-(1-r_i)^{n_i}] \\
s.t. \quad & h_1(r,n) = \sum_{i=1}^{m} v_i n_i^2 \le V \\
& h_2(r,n) = \sum_{i=1}^{m} C(r_i)\cdot[n_i + \exp(n_i/4)] \le C \\
& h_3(r,n) = \sum_{i=1}^{m} w_i n_i \exp(n_i/4) \le W \\
& 1.0 \le n_i \le 10,\ n_i \in Z^+ \\
& 0.5 \le r_i \le 1-10^{-6},\ r_i \in R^+
\end{aligned} \tag{12}$$

Wherein $C(r_i) = \alpha_i(-T/\ln r_i)^{\beta_i}$, T is the mission time of the components, α_i and β_i are the same as series-parallel systems.

The values of parameters for this problem are set in Table 3:

Table 3. The Parameters of Over Speed Protection System

Subsystem i	$10^5\alpha_i$	β_i	i	i	V	C	W	T
1	1	1.5			250	400	500	1000
2	2.3	1.5						
3	0.3	1.5						
4	2.3	1.5						

To analyze the performance of the proposed algorithm, it is developed for three bench mark problems for reliability redundancy allocations problems. For this algorithm, the maximum number of iterations is set to 800, set F=0.7, CR=0.8, population size M=40. The parameters of α and β in Lévy flight are 0.01 and 2 respectively. The algorithm runs 50 times independently for every problem. The best results are listed in Table 4, Table 5, and Table 6.

MPI (maximum possible improvement) index is used to measure the relative improvement. And it is described as:

$$MPI\ (\%) = (f - f_{other})/(1 - f_{other}) \tag{13}$$

Where f represents the best value obtained by the proposed method, and f_{other} represents the best value obtained by one of the other approach in literatures. It should be emphasized that even very small improvements in reliability are very important in high reliability application systems.

Table 4. Best Results Comparison on Series Parallel System

Parameter	Hikita *et al.* [19]	Hsieh *et al.* [6]	Chen [7]	This paper
n_1-n_5	(3,3,1,2,3)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)
r_1	0.838193	0.785452	0.812485	0.81965932
r_2	0.855065	0.842998	0.843155	0.84498074
r_3	0.878859	0.885333	0.897385	0.89550642
r_4	0.911402	0.917958	0.894516	0.89550643
r_5	0.850355	0.870318	0.870590	0.86844775
f(r,n)	0.99996875	0.99997418	0.99997658	**0.99997665**
MPI (%)	25.28	9.57	0.30	-
Slack(g1)	53	40	40	40
Slack(g2)	0.000000	1.194440	0.002627	0.000000
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)Slack is the unused resources.

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

Parameter	Hikita. *et al.* [9]	Hsieh *et al.* [2]	Chen [3]	Coelho[16]	This paper
n_1-n_5	(3,3,2,3,2)	(3,3,3,3,1)	(3,3,3,3,1)	(3,3,2,4,1)	(3,3,2,4,1)
r_1	0.814483	0.814090	0.812485	0.826678	0.82808641
r_2	0.821383	0.864614	0.867661	0.857172	0.85780478
r_3	0.896151	0.890291	0.861221	0.914629	0.91424067
r_4	0.713091	0.701190	0.713852	0.648918	0.64814622
r_5	0.814091	0.734731	0.756699	0.715290	0.70416210
f(r,n)	0.9997894	0.99987916	0.99988921	0.99988957	**0.99988964**
MPI (%)	47.597	8.673	0.388	0.063	-
Slack(g 1)	18	18	18	5	5
Slack(g 2)	1.854075	0.376347	0.001494	0.000339	0.000000
Slack(g 3)	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) Slack is the unused resources.

Table 6. Best Results Comparison on Over Speed Protection System

Parameter	Yokota *et al.* [10]	Dhingra[11]	Chen[3]	Coelho [16]	This paper
n_1-n_4	(3,6,3,5)	(6,6,3,5)	(5,5,5,5)	(5,6,4,5)	(5,6,4,5)
r_1	0.965993	0.81604	0.903800	0.902231	0.90161482
r_2	0.760592	0.80309	0.874992	0.856325	0.84992114
r_3	0.972646	0.98364	0.919898	0.9481450	0.94814139
r_4	0.804660	0.80373	0.890609	0.883156	0.88822284
f(r,n)	0.999468	0.99961	0.999942	0.999953	**0.99995467**
MPI (%)	91.48	88.38	21.84	3.55	-
Slack(g 1)	92	65	50	55	55
Slack(g 2)	70.733576	0.064	0.002152	0.975465	0.000000
Slack(g 3)	127.583189	4.348	28.803701	24.801882	24.801882

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

Table 4 Table 5 and Table 6 compare the best results of three reliability optimization problems with those reported in the literatures. It is clear that the proposed algorithm can attain a better result than any other approach proposed in literatures.

Table 4 shows that the best results reported by Hikita *et al.*[9], Hsieh, *et al.*[2] and Chen[3] were 0.99996875, 0.99997418 and 0.99997658 for the series–parallel system respectively. The result obtained by TSDE is better than the above three best solution, and the corresponding improvements made by the presented method are 25.28%, 9.57% and 0.30% respectively.

Table 5 shows that the best results reported by Hikita *et al.*[9], Hsieh *et al.*[2], Chen[3] and Coelho[16] were 0.9997894, 0.99987916, 0.99988921 and 0.99988957 for the complex (bridge) system respectively. The result obtained by TSDE is better than the above four best solution, and the corresponding improvements made by the presented method are 47.597%, 8.673%, 0.388% and 0.063% respectively.

Table 6 shows that the best results reported by Yokota *et al.*[10], Dhingra[11], Chen [3] and Coelho[16] were 0.999468, 0.99961, 0.999942 and0.999953 for the overspeed protection system respectively. The result is better than the above four best solution, and the corresponding improvements made by the presented method are 91.48%, 88.38%, 21.84% and 3.55% respectively.

In short, the proposed DE algorithm combined with Lévy flight is an effective algorithm, and it has got better solution than the other methods for reliability redundancy allocation problems.

5. Conclusion

In this paper, we proposed a DE algorithm combined with Lévy flight to solve the reliability redundancy allocation problems. The Lévy flight is used to enhance the ability of global search of differential evolution algorithm. DE is used for local search mainly. The proposed algorithm considers the trade-off of the diversification and the intensification simultaneously. Experiments results based on three benchmark problems are obtained and be compared with some methods in the literatures. It is showed that the presented algorithm was effective and outperformed the other methods in the literatures. In the future work, it will be used to solve other more complex optimization problems.

Acknowledgment

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 : 20140204048GX.

References

[1] H. Garg, and S. P. Sharma, "Multi-objective reliability-redundancy allocation problem using particle swarm optimization" , Computers & Industrial Engineering, vol. 64, no. 1, **(2013)**.
[2] Y. C. Hsieh, T. C. Chen and D. L. Bricker, "Genetic algorithms for reliability design problems", Microelectronics Reliability, vol. 38, no. 10, **(1998)**.
[3] T. C. Chen, "IAs based approach for reliability redundancy allocation problems", Applied Mathematics and Computation, vol. 182, no. 2, **(2006)**.
[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] M. A. Ardakan and A. Z Hamadani, "Reliability–redundancy allocation problem with cold-standby redundancy strategy", Simulation Modeling Practice and Theory, vol. 42, **(2014)**.
[6] L. D. Afonso, V. C. Mariani and L. Coelho, "Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization", Expert Systems with Applications, vol. 40, no. 9, **(2013)**.
[7] N. Safaei, R. T. 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)**.
[8] L. Wang, and L. Li, "A co-evolutionary differential evolution with harmony search for reliability-redundancy optimization", Expert Systems with Applications, vol. 39, no. 5, **(2012)**.
[9] M. Hikita, Y. Nakagawa and H. Harihisa, "Reliability optimization of systems by a surrogate constraints algorithm", IEEE Trans. Reliab, vol. 41, no. 3, **(1992)**.
[10] 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)**.
[11] A. K. Dhingra, "Optimal Apportionment of Reliability & Redundancy in Series Systems Under Multiple Objectives", IEEE Transactions on Reliability, vol. 41, no. 4, **(1992)**.
[12] S. Chen, A. Sarosh and Y. Dong, "Simulated annealing based artificial bee colony algorithm for global numerical optimization", Applied Mathematics and Computation, vol. 219, no. 8, **(2012)**.
[13] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms", 2nd Ed. Luniver Press, **(2010)**.
[14] 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.
[15] 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)**.
[16] L. 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)**.

