System Reliability-Redundancy Allocation by Evolutionary Computation

Mohamed Arezki Mellal
LMSS, Faculty of Engineering Sciences
M'Hamed Bougara University
Boumerdes, Algeria
e-mail: mellal.mohamed@gmail.com,
mellal.mohamed@univ-boumerdes.dz

Enrico Zio
Chair System Science and the Energy Challenge,
Fondation Electricité de France (EDF), CentraleSupélec,
Université Paris-Saclay, France
Politecnico di Milano, Milan, Italy

***Abstract*—This paper addresses the system reliability-redundancy allocation problem by resorting to three evolutionary computation methods, namely genetic algorithm, cuckoo optimization algorithm with penalty function, and penalty guided stochastic fractal search. The numerical results are compared in order to highlight the differences in the solution methods.**

Keywords-reliability-redundancy allocation; genetic algorithm; cuckoo optimization algorithm with penalty function; penalty guided stochastic fractal search

I. INTRODUCTION

Competitive industrial plants require highly reliable systems. The reliability-redundancy allocation is one way to increase the overall reliability of a system [1].

Mathematically, the reliability-redundancy allocation problem includes integer and real variables and is subject to design constraints, e.g. with respect to volume, weight, and cost. Several research works have been proposed for trying to efficiently solve this optimization problem. Recently, solution approaches have been proposed, based on evolutionary computation methods. The immune based algorithms (IA) have been used in [2], [3], whereas the artificial bee colony has been applied for solving different system reliability configurations in [4]. In [5], a particle swarm optimization based on Gaussian distribution and chaotic sequence has been proposed. Genetic algorithms have been applied several times [6]–[11]. A modified imperialist competitive algorithm based on attraction and repulsion concepts has been used in [12]. In [13], the particle swarm optimization has been modified for this purpose. Recently, a penalty guided stochastic fractal search has been implemented and its effectiveness has been proven. In all works, the number of subsystems of the system considered is limited to twenty.

The aim of the present paper is to address the reliability-redundancy allocation problem for systems consisting of ten and thirty subsystems connected in series. The problem involves sixty decision variables. Genetic algorithm (GA), cuckoo optimization algorithm with penalty function (PFCOA) [14] and penalty guided stochastic fractal search (PSFS) [15] are applied for solving the problem. The remainder of the paper is organized as follows: Section 2 presents the reliability-redundancy allocation problem. The results with a discussion are given in Section 3. Finally, the last section concludes this paper.

II. RELIABILITY-REDUNDANCY ALLOCATION PROBLEM

A. Case Study 1

The system studied here is a pharmaceutical plant containing ten subsystems (see Figure 1) [15], [16].

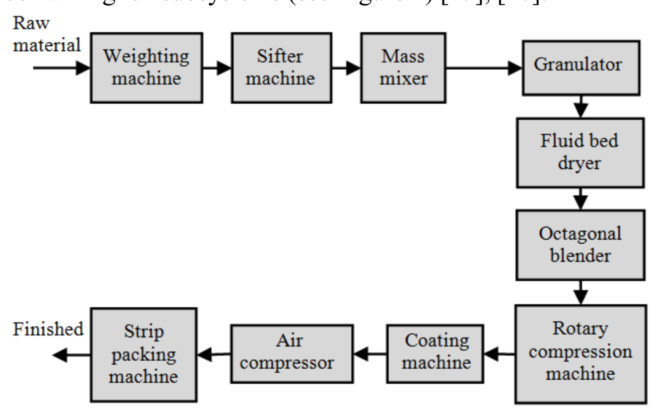

Figure 1. Pharmaceutical plant.

$$\text{Maximize} \quad R_s(r,n) = \prod_{i=1}^{10}\left[1-(1-r_i)^{n_i}\right] \qquad (1)$$

subject to the following constraints

$$g_1(r,n) = \sum_{i=1}^{10} v_i n_i^2 \le V \qquad (2)$$

$$g_2(r,n) = \sum_{i=1}^{10} \alpha_i(-T/\ln r_i)^{\beta_i}\left[n_i + \exp(n_i/4)\right] \le C \qquad (3)$$

$$g_3(r,n) = \sum_{i=1}^{10} w_i n_i \exp(n_i/4) \le W \qquad (4)$$

$$0.5 \le r_i \le 1$$

$$r_i \in [0,1] \subset \mathbb{R}^+$$

$$n_i \in \mathbb{Z}^+$$

$$i = 1,2,\ldots,10$$

978-1-5386-3322-9/17/$31.00 ©2017 IEEE

where r_i is the reliability of each component in subsystem i, n_i is the number of components in subsystem i, v_i is the volume of each component in subsystem i, w_i is the weight of each component in subsystem i, α_i and β_i are parameters representing physical features (shaping and scaling factors, respectively) of each component at subsystem i. V, C and W are the upper limit on the volume, cost and weight of the system, respectively. T is the operating time during which the component must not fail (mission time).

The bounds considered in this paper are as follows: $1\leq n_1\leq 6$, $1\leq n_2$, $n_6\leq 4$, $1\leq n_3\leq 7$, $1\leq n_4$, n_7, $n_{10}\leq 3$, $1\leq n_5$, n_8, $n_9\leq 5$, V=270, C=480, and W=519. The data are reported in Table I.

TABLE I. DATA USED IN CASE STUDY 1.

Subsystem i	$10^5\alpha_i$	β_i	v_i	w_i
1	0.611360	1.5	4	9
2	4.032464	1.5	5	7
3	3.578225	1.5	3	5
4	3.654303	1.5	2	9
5	1.163718	1.5	3	9
6	2.966955	1.5	4	10
7	2.045865	1.5	1	6
8	2.649522	1.5	1	5
9	1.982908	1.5	4	8
10	3.516724	1.5	4	6

B. *Case Study 2*

This case study considers a system with thirty subsystems:

$$\text{Maximize} \quad R_s(r,n) = \prod_{i=1}^{30}\left[1-(1-r_i)^{n_i}\right] \tag{5}$$

subject to the following constraints

$$g_1(r,n) = \sum_{i=1}^{30} v_i n_i^2 \leq V \tag{6}$$

$$g_2(r,n) = \sum_{i=1}^{30} \alpha_i(-T/\ln r_i)^{\beta_i}\left[n_i + \exp(n_i/4)\right] \leq C \tag{7}$$

$$g_3(r,n) = \sum_{i=1}^{30} w_i n_i \exp(n_i/4) \leq W \tag{8}$$

$$0.5 \leq r_i \leq 1$$

$$r_i \in [0,1] \subset \mathbb{R}^+$$

$$1 \leq n_i \leq 12,\ n_i \in \mathbb{Z}^+$$

$$i = 1,2,\ldots,30$$

The numerical data of the system are given in Table I.

TABLE II. DATA USED IN CASE STUDY 2.

Subsystem i	$10^5\alpha_i$	β_i	v_i	w_i
1	0.8	1.5	3	11
2	0.5	1.5	4	7
3	1.5	1.5	1	8
4	0.6	1.5	7	6
5	3.1	1.5	6	6
6	1.2	1.5	5	10
7	2.1	1.5	3	4
8	1.4	1.5	2	9
9	2.7	1.5	3	10
10	2.4	1.5	4	4
11	1.8	1.5	4	11
12	1.9	1.5	3	9
13	1.6	1.5	6	2
14	1.8	1.5	5	4
15	2.9	1.5	1	10
16	1.5	1.5	8	9
17	2.8	1.5	4	4
18	1.1	1.5	3	4
19	1.9	1.5	1	6
20	2.4	1.5	2	10
21	1.7	1.5	5	8
22	0.5	1.5	3	11
23	3.7	1.5	6	7
24	2.5	1.5	5	9
25	2.8	1.5	1	3
26	1.3	1.5	4	5
27	2.0	1.5	6	10
28	1.7	1.5	3	8
29	0.9	1.5	2	5
30	3.5	1.5	1	7

The mission time considered in both case studies is 1000h, whereas the upper limits constraining the system are V=1200, C=1500, and W=1100.

III. RESULTS AND DISCUSSION

The reliability-redundancy allocation problems described in Eqs. (1)–(8) with the data reported in Tables I and II are solved using GA [17], PFCOA [14] and PSFS [15]. The algorithms have been encoded in MATLAB 2015 and run on a PC with the following characteristics: Intel Pentium Processor G620 of 2.60 GHz with 4 GB of RAM and 3Mo Cache. In this paper, the penalty factors are dynamic values. It decreases when the violation of the constraints decreases.

Tables II, III, and IV summarize the parameters of the implemented evolutionary computation methods. The parameters values have been set by a systematic procedure of trial and error.

The optimal results provided by the GA, PFCOA and PSFS are reported in Tables VI–XI.

TABLE III. GENETIC ALGORITHM PARAMETERS AND RULES.

Population size	100
Selection technique	Standard roulette
Mutation probability	10^{-3}
Crossover probability	1

TABLE IV. CUCKOO OPTIMIZATION ALGORITHM WITH PENALTY FUNCTION PARAMETERS

Number of cuckoos	100
Minimum number of eggs	2
Maximum number of eggs	4

TABLE V. PENALTY GUIDED STOCHASTIC FRACTAL SEARCH PARAMETERS

Number of fractals	100
Penalty value	10^5

TABLE VI. OPTIMAL SOLUTIONS PROVIDED BY GA FOR CASE STUDY 1.

r	0.8626, 0.8599, 0.8628, 0.7886, 0.9039, 0.7073, 0.8855, 0.8017, 0.8287, 0.7893
n	3, 2, 2, 3, 2, 4, 2, 3, 3, 3
$\boldsymbol{R_s}$	0.9021
NFE	36000
CPU	183s

TABLE VII. OPTIMAL SOLUTIONS PROVIDED BY PFCOA FOR CASE STUDY 1.

r	0.8670, 0.7985, 0.8074, 0.8032, 0.8500, 0.8093, 0.8298, 0.7613, 0.8273, 0.8779
n	3, 3, 3, 3, 3, 3, 3, 4, 3, 2
$\boldsymbol{R_s}$	0.9379
NFE	8120
CPU	122s

TABLE VIII. OPTIMAL SOLUTIONS PROVIDED BY PSFS FOR CASE STUDY 1.

r	0.9276, 0.8067, 0.8072, 0.8089, 0.8596 0.8131, 0.8349, 0.7513, 0.8316, 0.8071
n	2, 3, 3, 3, 3, 3, 3, 4, 3, 3
$\boldsymbol{R_s}$	0.9452
NFE	5000
CPU	35s

TABLE IX. OPTIMAL SOLUTIONS PROVIDED BY GA FOR CASE STUDY 2.

r	0.8531, 0.8154, 0.9118, 0.9291, 0.7669, 0.9041, 0.8497, 0.9122, 0.9012, 0.7726, 0.8980, 0.9080, 0.8089, 0.8574, 0.9018, 0.6880, 0.8569, 0.8728, 0.8192, 0.8966, 0.8962, 0.8981, 0.8751, 0.8749, 0.6650, 0.8089, 0.8938, 0.9562, 0.8262, 0.7409
n	3, 3, 2, 2, 4, 2, 3, 2, 2, 3, 2, 2, 3, 3, 2, 4, 2, 3, 3, 2, 2, 2, 2, 2, 5, 3, 2, 1, 4, 4
$\boldsymbol{R_s}$	0.7556
NFE	54000
CPU	257s

TABLE X. OPTIMAL SOLUTIONS PROVIDED BY PFCOA FOR CASE STUDY 2.

r	0.9217, 0.9156, 0.8447, 0.8715, 0.8461, 0.8478, 0.8366, 0.9159, 0.8888, 0.8473, 0.8884, 0.8435, 0.8499, 0.8467, 0.8689, 0.9049, 0.8231, 0.8534, 0.8283, 0.8961, 0.8462, 0.8889, 0.9024, 0.8900, 0.7954, 0.7928, 0.8870, 0.8660, 0.8656, 0.8293
n	2, 3, 3, 3, 3, 3, 3, 2, 2, 3, 2, 3, 3, 3, 2, 2, 3, 3, 3, 2, 3, 3, 2, 2, 4, 3, 2, 3, 3, 3
$\boldsymbol{R_s}$	0.8338
NFE	10650
CPU	180s

TABLE XI. OPTIMAL SOLUTIONS PROVIDED BY PSFS FOR CASE STUDY 2.

r	0.9265, 0.9355, 0.8530, 0.8871, 0.8288, 0.9143, 0.8400, 0.8553, 0.8204, 0.8266, 0.9067, 0.8491, 0.7952, 0.8487, 0.8949, 0.9132, 0.8280, 0.8647, 0.8459, 0.8977, 0.9077, 0.9351, 0.8247, 0.8375, 0.8218, 0.8511, 0.9027, 0.8503, 0.8741, 0.8195
n	2, 2, 3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 4, 3, 2, 2, 3, 3, 3, 2, 2, 2, 3, 3, 3, 3, 2, 3, 3, 3
$\boldsymbol{R_s}$	0.8566
NFE	5200
CPU	76s

From Tables VI, VII and VIII it can be observed that the overall system reliability provided by the genetic algorithm, cuckoo optimization algorithm with penalty function and penalty guided stochastic fractal search for the first case study are 0.9021, 0.9379 and 0.9452, respectively. Moreover, the results for the second case study are 0.7556, 0.8338 and 0.8566, respectively. Figures 2 and 3 clearly show that the best solution is provided by PSFS.

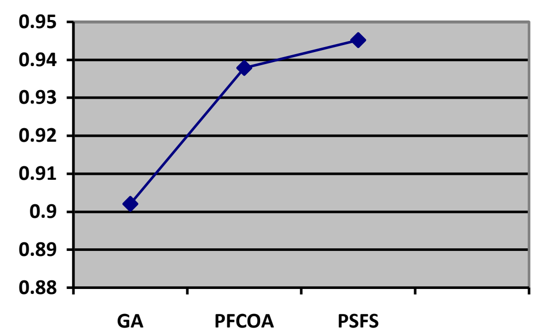

Figure 2. Overall system reliability provised by each evolutionary computation method (Case study 1).

Furthermore, Figures 4 and 5 show that the PSFS requires the smallest number of function evaluations.

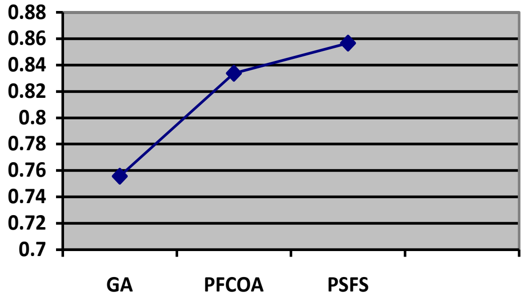

Figure 3. Overall system reliability provised by each evolutionary computation method (Case study 2).

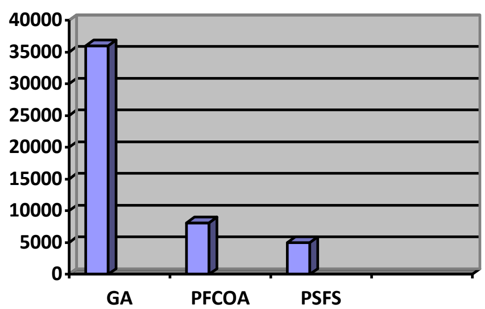

Figure 4. NFE required by each evolutionary computation method (Case study 1).

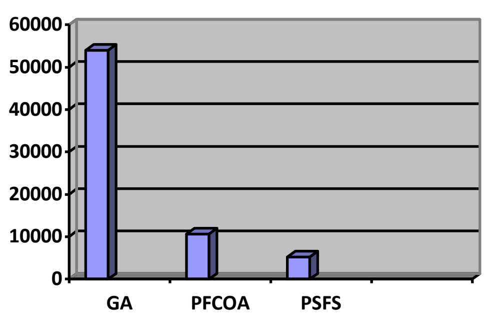

Figure 5. NFE required by each evolutionary computation method (Case study 2).

The CPU time required by each method for providing the optimal solutions is 183s for the GA, 122s for the PFCOA, and 35s for the PSFS in the first case study, whereas 257s, 180s and 76s in the second case study. Therefore, the comparison between the optimal solutions and the execution performances reveals that PSFS has outperformed GA and PFCOA in solving the considered reliability-redundancy allocation problems. The dynamic values of the penalty factors have handled the NFE required by each algorithm. The PSFS is better since its optimization procedures are simple and powerful over GA and PFCOA.

IV. Conclusions

In this paper, the reliability-redundancy allocation problem has been addressed using three evolutionary computation methods: genetic algorithm (GA), cuckoo optimization algorithm with penalty function (PFCOA), and the penalty guided stochastic fractal search (PSFS). The systems considered in the case studies contain ten and thirty subsystems connected in series, respectively. It is shown that on the case studies considered, the PSFS has outperformed the two other methods in terms of better system reliability solution and execution performance. It should be noted that the dynamic penalty factors are an advantage in solving the reliability-redundancy allocation problem.

References

[1] E. Zio, “Reliability engineering: Old problems and new challenges,” *Reliab. Eng. Syst. Saf.*, vol. 94, pp. 125–141, 2009.

[2] Y. C. Hsieh and P. S. You, “An effective immune based two-phase approach for the optimal reliability-redundancy allocation problem,” *Appl. Math. Comput.*, vol. 218, no. 4, pp. 1297–1307, 2011.

[3] T. C. Chen, “IAs based approach for reliability redundancy allocation problems,” *Appl. Math. Comput.*, vol. 182, no. 2, pp. 1556–1567, 2006.

[4] H. Garg, M. Rani, and S. P. Sharma, “An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique,” *Comput. Oper. Res.*, vol. 40, no. 12, pp. 2961–2969, 2013.

[5] L. dos S. Coelho, “An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications,” *Reliab. Eng. Syst. Saf.*, vol. 94, no. 4, pp. 830–837, 2009.

[6] Y.-C. Hsieh, T.-C. Chen, and D. L. Bricker, “Genetic algorithms for reliability design problems,” *Microelectron. Reliab.*, vol. 38, no. 10, pp. 1599–1605, 1998.

[7] D. W. Coit and A. E. Smith, “Reliability optimization of series-parallel systems using a genetic algorithm,” *IEEE Trans. Reliab.*, vol. 45, no. 2, pp. 254–260, 266, 1996.

[8] D. W. Coit and A. E. Smith, “Penalty guided genetic search for reliability design optimization,” *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 895–904, 1996.

[9] P. Giuggioli Busacca, M. Marseguerra, and E. Zio, “Multiobjective optimization by genetic algorithms: application to safety systems,” *Reliab. Eng. Syst. Saf.*, vol. 72, pp. 59–74, 2001.

[10] M. Marseguerra, E. Zio, and L. Podofillini, “Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation,” *Reliab. Eng. Syst. Saf.*, vol. 77, no. 2, pp. 151–165, 2002.

[11] M. Marseguerra and E. Zio, “Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation,” *Reliab. Eng. Syst. Saf.*, vol. 68, no. 1, pp. 69–83, 2000.

[12] 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 Syst. Appl.*, vol. 40, no. 9, pp. 3794–3802, 2013.

[13] P. Wu, L. Gao, D. Zou, and S. Li, “An improved particle swarm optimization algorithm for reliability problems,” *ISA Trans.*, vol. 50, no. 1, pp. 71–81, 2011.

[14] M. A. Mellal and E. J. Williams, “Cuckoo optimization algorithm with penalty function for combined heat and power economic dispatch problem,” *Energy*, vol. 93, pp. 1711–1718, 2015.

[15] M. A. Mellal and E. Zio, “A penalty guided stochastic fractal search approach for system reliability optimization,” *Reliab. Eng. Syst. Saf.*, vol. 152, pp. 213–227, 2016.

[16] H. Garg and S. P. Sharma, “Reliability-redundancy allocation problem of pharmaceutical plant,” *J. Eng. Sci. Technol.*, vol. 8, no. 2, pp. 190–198, 2013.

[17] E. G. Talbi, “Population-based metaheuristics,” in *Metaheuristics: From design to implementation*, John Wiley., New Jersey, 2009, pp. 190–307.