Int. J. Fuzzy Syst.
https://doi.org/10.1007/s40815-021-01068-z

Multi-objective System Design Optimization via PPA and a Fuzzy Method

Mohamed Arezki Mellal[1,2] · Abdellah Salhi[3]

Received: 19 December 2019 / Revised: 1 January 2021 / Accepted: 9 February 2021

Abstract System design deals with various challenges of targets and resources, such as reliability, availability, maintainability, cost, weight, volume, and configuration. This paper deals with the multi-objective system availability and cost optimization of parallel–series systems by resorting to the multi-objective strawberry algorithm also known as the Plant Propagation Algorithm or PPA and a fuzzy method. It is the first implementation of this optimization algorithm in the literature for this kind of problem to generate the Pareto Front. The fuzzy method allows helping the decision maker to select the best compromise solution. A numerical case study involving 10 subsystems highlights the applicability of the proposed approach.

Keywords Multi-objective system design · Availability · Cost · Parallel–series system · Plant Propagation Algorithm · Fuzzy method

1 Introduction

Resource allocation is one of the most commonly used methods in system design and exploitation. It allows optimally using available resources and respecting product specifications. Industrial plants consist of many components and their system dependability (RAMS + C: reliability, availability, maintainability, safety, and cost) should be optimized [1–5]. In the last few decades, efforts have been devoted to the question of optimal allocation in system dependability. These efforts may be divided into two categories depending on the nature of the formulated problems: single objective and multi-objective optimization problems. Most solution approaches are based on bio-inspired optimization techniques, also called soft computing methods or artificial intelligence methods, as these methods proved their effectiveness in practice.

In [6], the reliability of parallel–series systems has been optimized by the redundancy allocation with component choices under the constraints of cost and weight. Several authors proposed solution approaches for the optimal system reliability–redundancy allocation subject to cost, volume, and weight [7–13]. Chen [7] and Hsieh and You [8] developed immune algorithms. Garg et al. [9] and Garg [10] used the artificial bee colony and cuckoo search, respectively. Mellal and Zio implemented a penalty guided stochastic fractal search for 10 case studies in [11], whereas a pharmaceutical plant consisting of 10 subsystems connected in series has been investigated in [12] using the above algorithm, the cuckoo optimization algorithm with penalty function, and the genetic algorithm. A large-scale reliability–redundancy allocation problem has been solved in [13] using the cuckoo optimization algorithm, particle swarm optimization, and the genetic algorithm. The optimization problem of the reliability–redundancy allocation has been simplified in [14] by resorting to the theory of survival signature. On the other hand, Juybari et al. [15] and Mellal and Zio [16] considered the cold-standby strategy for the redundant components. Chebouba et al. [17] considered the reliability and cost as objectives under

✉ Mohamed Arezki Mellal
mellal.mohamed@gmail.com; mellal.mohamed@univ-boumerdes.dz; mamellal@umd.edu

1 LMSS, Faculty of Technology, M'Hamed Bougara University, 35000 Boumerdes, Algeria

2 Center for Advanced Life Cycle Engineering (CALCE), University of Maryland, College Park, MD 20742, USA

3 Department of Mathematical Sciences, University of Essex, Colchester, UK

Published online: 09 April 2021

data uncertainties. In [18–20], the goal was to allocate the optimal number of redundant components, failure rates and repair rates in each subsystem for maximizing the system availability under the cost limit and design constraints. The author of [18, 19] used the Tabu-genetic algorithm, whereas in [20] the authors used five optimization techniques: cuckoo optimization algorithm, genetic algorithms, flower pollination algorithm, differential evolution, and particle swarm optimization. The study showed that the cuckoo optimization algorithm provided better results. In [21, 22], the authors considered both the system availability and the cost function described in [18–20] as objective functions using the weighted sum methods and the non-dominated sorting genetic algorithm II (NSGA-II), respectively. The bi-criteria problem has been tackled by converting it into a single objective optimization one using two weighted sum methods. The system cost under availability constraint with failure dependencies has been optimized in [4, 23] using an adaptive cuckoo optimization algorithm and the genetic algorithm, respectively. The optimum number of redundant components and repair teams have been allocated in each subsystem for this purpose.

The aim of the present work is to consider the conflicting and nonlinear objectives of availability and cost of parallel–series systems when redundancy, failure rate, and repair rate allocations are considered as design variables. An implementation of a multi-objective optimization algorithm, called multi-objective strawberry algorithm, is presented in order to generate the Pareto front. A fuzzy method is applied to select the best compromise solution for the decision maker. The remainder of the paper is organized as follows: Sect. 2 describes the multi-objective optimization problem. Section 3 presents the principles of the multi-objective strawberry algorithm. Section 4 highlights the method applied for selecting the best compromise solution. A numerical case study with results and discussion are illustrated in Sects. 5 and 6, respectively. Finally, the last section concludes the paper.

2 Problem Description

The design of the parallel–series system shown in Fig. 1 when considering the redundancy, failure rate and repair rate allocations as design variables is given as follows [18–21]:

System cost

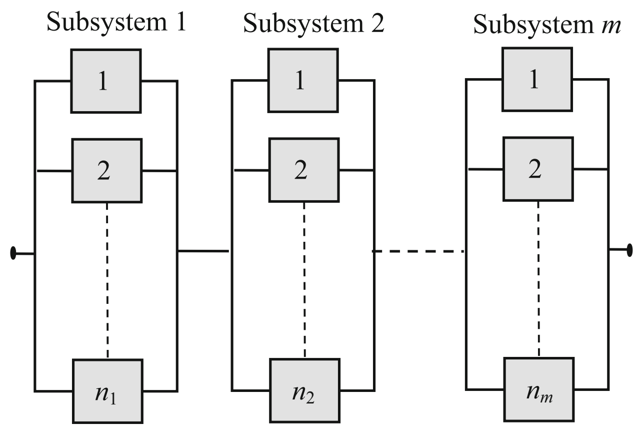

Fig. 1 Parallel–series system

$$C_s(n,\lambda,\mu) = \sum_{i=1}^{m}\left[\left(\alpha_i(\lambda_i)^{-\beta_i} + \mu_i mc_i\right)(n_i + \exp(n_i/4))\right] \tag{1}$$

Asymptotic system availability

$$A_s(n,\lambda,\mu) = \prod_{i=1}^{m}\left[1 - \left(1 - \frac{\mu_i}{\lambda_i + \mu_i}\right)^{n_i}\right] \tag{2}$$

System design configuration

$$\sum_{i=1}^{m} p_i(n_i)^2 \le D_1 \tag{3}$$

$$\sum_{i=1}^{m} w_i n_i \exp(n_i/4) \le D_2 \tag{4}$$

$$\begin{aligned} & n_i \ge 1 \ (n_i \in \mathbb{Z}^+) \\ & \lambda_i \in [\lambda_i^L, \lambda_i^U] \subset \Re^+, \mu_i \in [\mu_i^L, \mu_i^U] \subset \Re^+ \\ & A_s \ge A_s^* \\ & C_s \le C_s^* \end{aligned} \tag{5}$$

where $C_s(\bullet)$ is the total system cost, n_i is the number of identical redundant components to be used in the ith subsystem, λ_i is the failure rate of the components in the ith subsystem, μ_i is the repair rate of the components in the ith subsystem, m is the total number of subsystems in the system. β_i and α_i are parameters representing the physical features of each component in each subsystem i. p_i is the product of weight and volume per component in each subsystem i, and w_i is the weight of one component in each subsystem i. D_1, D_2, (A_s^* and C_s^* are the design limits. λ_i^L, μ_i^L, λ_i^U and μ_i^U are the lower and upper bounds for the failure and repair rates, respectively. It should be noted that the weight constraint given in Eq. (4) is increased by the interconnecting links modeled by $\exp(n_i/4)$.

In [18–20], the authors solved the problem by considering the system availability as a single objective problem, whereas in [21] the objectives of system availability and cost have been converted into a single objective by resorting to two weighted sum methods. In the present work, both objectives are considered in a Pareto front. Therefore, the above problem is formulated as follows:

$$\text{Minimize } C_s(n, \lambda, \mu) \tag{6}$$

$$\text{Maximize } A_s(n, \lambda, \mu) \tag{7}$$

subject to

Equations (3)-(5)

3 Multi-objective Strawberry Algorithm

First introduced by Salhi and Fraga [24], the strawberry algorithm also known as the Plant Propagation Algorithm or PPA is a Nature-Inspired heuristic that emulates the way plants and in particular the strawberry plant propagation. The strawberry plant can propagate through seeds, but as a hybrid, it relies more on runners. A runner is a long branch that grows over ground. When it touches the ground, it produces roots which then give rise to another strawberry plant. Strawberry plants use runners to explore the landscape where they happen to be to find good places to grow and propagate. Typically, a good place is one which is sunny, has enough nutrients and humidity. Note that we are not explicitly concerned with these growth factors. To improve its chances of survival in nature, a strawberry plant implements a very basic strategy which is:

1. In a good spot, send many short runners (exploitation);
2. In a poor spot, send few long runners (exploration).

This strategy which is not unique to the strawberry plant can be implemented for any type of optimization/search problem including those involving two or more objective functions. For min $f(x)$, where $f(x)$ is a vector function, PPA can be described as shown in Algorithm 1 [25].

Algorithm 1 Pseudo-code of the implemented MOPPA [25].

Algorithm 1. Pseudo-code of the implemented MOPPA [25].

```
Given: f(x); a vector function; ng: number of generations to perform; np: the
    propagation size; nr: maximum number of runners to propagate:
Output: z: vector approximation to Nondominated frontier:
    p ← initial random population of size np
    for ng generations do
        prune population p; removing similar solutions
        N ← fitness(p)                                  > Use rank based fitness
        p ← ϕ                                           > Empty set
        for i ← 0 … np do
            x ← select(p;N)                             >Tournament fitness based selection
            for each runner to generate do              > Nbre prop. to fitness rounded up
            x' ← new solution(x; 1 - N)                 > Distance inversely proportional to
            fitness
            p ← x' ∪ p                                  > Add to new population
            endfor
        endfor
        p ← p ∪ Nondominated(p)                         >New population with elitism
    endfor
    z ← Nondominated(p)
```

Only N, the population size has to be set arbitrarily and possibly the number of generations for the stopping criterion. So, PPA only requires two arbitrary parameters. Compare this with the seven parameters that the genetic algorithm implementation requires and the five parameters that the simulated annealing implementation requires [26].

PPA has been shown to be competitive in continuous global optimization [27]. It has also been shown to work well on discrete optimization problems such as the TSP [26]. A variant of PPA which emulates propagation by seeds has been implemented and shown to work also very well on continuous problems [28]. Another variant that combines propagation via runners and seeds has also been introduced in [29, 30].

From its inception, PPA [24] has been considered for multi-objective optimization. Indeed, in [25], a bi-criterion optimization problem arising in chemical engineering has been considered. The two objectives were combined in an additive fashion using a couple of parameters λ_1 and λ_2 such that $\lambda_1 + \lambda_2 = 1$. Here, we are concerned with the design of the parallel–series system problem (see Fig. 1). As described in Sect. 2, it involves two objectives namely the cost and the availability. The issue is to optimize with respect to these two conflicting objectives with the aim of providing the system designer with an optimum or near optimum decision. This decision is found in a Pareto Front which is a set of potential solutions. An approach to choosing the best solution from the PF is described below.

4 Best Compromise Solution

Solving multi-objective optimization problems is confronting its main disadvantage which is the Pareto Front (PF). PF is a set of optimal solutions and selecting a single solution is challenging for the decision maker. Several works have been devoted to the development of methods allowing the selection of the best compromise (called also preferred) solutions. An overview of these methods is listed in [31–36].

In this paper, the fuzzy set method is used to determine the best compromise solution from the obtained PF. When the multi-objective strawberry algorithm has generated the PF, the following algorithm is implemented [35–38]:

– Perform fuzzy-based mechanisms:

Table 1 Data of the system

Subsystem i	α_i (10^{-5})	β_i	mc_i	p_i	w_i
1	1.25	1.5	500	2	6
2	2.70	1.5	500	4	9
3	8.10	1.5	500	3	7
4	4.50	1.5	500	2	6
5	1.90	1.5	500	4	8
6	3.55	1.5	500	2	5
7	2.45	1.5	500	4	3
8	6.30	1.5	500	3	9
9	1.80	1.5	500	2	7
10	5.25	1.5	500	2	5

Table 2 Number of non-dominated solutions (Conditions 1)

No.	Number of non-dominated solutions
1	36
2	49
3	51
4	51
5	72
6	56
7	82
8	62
9	58
10	53

Table 3 Number of non-dominated solutions (Conditions 2)

No.	Number of non-dominated solutions
1	49
2	**91**
3	84
4	39
5	47
6	49
7	83
8	40
9	37
10	50

Bold indicates the Highest number of non-dominated solutions

For system cost (minimizing function),

$$\mu_j = \begin{cases} 1, \ F_j \leq F_j^{\min} \\ \dfrac{F_j^{\max} - F_j}{F_j^{\max} - F_j^{\min}}, \ F_j^{\min} < F_j < F_j^{\max} \\ 0, \ F_j \geq F_j^{\max} \end{cases} \tag{8}$$

For system availability (maximizing function),

$$\mu_j = \begin{cases} 0, \ F_j \leq F_j^{\min} \\ \dfrac{F_j - F_j^{\min}}{F_j^{\max} - F_j^{\min}}, \ F_j^{\min} < F_j < F_j^{\max} \\ 1, \ F_j \geq F_j^{\max} \end{cases} \tag{9}$$

where μ_j is the membership function, F_j is the jth objective function, and $(F_j^{\min}, F_j^{\max})$ are its minimum and maximum values, respectively.

– Calculate the normalized membership value:

Table 4 Best Pareto front

No.	C_s	A_s	No.	C_s	A_s
1	248.0301	0.9675	51	226.6152	0.9508
2	228.6186	0.9541	52	227.0615	0.9534
3	199.0064	0.9167	53	195.7784	0.9098
4	208.2397	0.9307	54	219.7293	0.9459
5	233.2392	0.9601	55	210.9245	0.9344
6	210.8452	0.9322	56	247.0194	0.9672
7	230.3806	0.9568	57	195.1605	0.9077
8	202.3160	0.9195	58	219.5095	0.9452
9	246.4674	0.9671	59	245.1320	0.9653
10	232.6731	0.9571	60	241.7951	0.9645
11	209.8697	0.9317	61	230.3806	0.9568
12	241.4323	0.9640	62	241.2529	0.9638
13	245.1320	0.9653	63	234.0463	0.9603
14	247.3743	0.9673	64	216.5461	0.9412
15	235.7759	0.9617	65	198.9011	0.9148
16	209.8697	0.9317	66	195.1605	0.9077
17	245.5489	0.9669	67	202.7490	0.9215
18	203.8581	0.9263	68	210.8452	0.9322
19	240.9682	0.9637	69	238.5180	0.9628
20	245.5489	0.9669	70	240.7870	0.9634
21	233.0069	0.9581	71	240.9682	0.9637
22	223.7531	0.9483	72	245.3488	0.9658
23	218.2685	0.9413	73	227.0615	0.9534
24	246.4674	0.9671	74	206.9551	0.9291
25	240.8994	0.9636	75	239.6384	0.9631
26	226.6152	0.9508	76	232.6731	0.9571
27	195.7784	0.9098	77	216.5461	0.9412
28	241.7951	0.9645	78	202.7490	0.9215
29	193.2516	0.9043	79	240.8994	0.9636
30	235.7759	0.9617	80	212.9021	0.9349
31	233.0069	0.9581	81	233.0069	0.9581
32	223.7531	0.9483	82	199.0064	0.9167
33	218.2685	0.9413	83	208.2397	0.9307
34	246.4674	0.9671	84	224.3092	0.9495
35	240.8994	0.9636	85	234.0463	0.9603
36	226.6152	0.9508	86	247.0194	0.9672
37	195.7784	0.9098	87	243.9883	0.9651
38	241.7951	0.9645	88	245.3488	0.9658
39	193.2516	0.9043	89	193.2516	0.9043
40	235.7759	0.9617	90	243.9883	0.9651
41	198.9011	0.9148	91	197.1770	0.9147
42	219.7293	0.9459			
43	202.3160	0.9195			
44	241.2529	0.9638			
45	203.8581	0.9263			
46	212.9021	0.9349			
47	228.6186	0.9541			
48	206.9551	0.9291			
49	241.4323	0.9640			
50	218.2685	0.9413			

$$\mu^k = \frac{\sum_{j=1}^{2} \mu_j^k}{\sum_{l=1}^{M} \sum_{j=1}^{2} \mu_j^l} \tag{10}$$

where M is the number of solutions in PF. The solution having the maximum value of μ^k represents the best compromise solution.

5 Case Study

The parallel–series system (see Fig. 1) studied here contains 10 subsystems. Table 1 reports its data. The limits of the system design are: $C_s^* = 250$; $D_1 = 200$; $D_2 = 300$, in arbitrary units, and $A_s = 0.9$ [20, 21]. The lower and upper bounds for the design variables are: $n_i \geq 1$ ($n_i \in \mathbb{Z}^+$), $\lambda_i \in [10^{-7}, 10^{-3}] \subset \mathbb{R}^+$, and $\mu_i \in [32 \times 10^{-7}, 32 \times 10^{-3}] \subset \mathbb{R}^+$.

6 Results and Discussion

The multi-objective strawberry algorithm and fuzzy set method have been encoded using MATLAB 2017 and run on a PC (Intel Core I5-7300U vPro 7th Gen, 2.7 GHz, 8 GB of RAM). The optimization algorithm has been run under two values of the number of runners (3 and 4) with the same population size (100) and maximum number of iterations (200). These values are called Conditions 1 and Conditions 2, respectively. The above parameters have been fixed by trial-and-error and based on experience. Ten independent runs have been performed with each value of number of runners in order to select the run with the highest number of non-dominated solutions. From Tables 2 and 3, it can be observed that the maximum number of non-dominated solutions is 91 and has been obtained with four runners. The consumed CPU time was 6.9165 s. Figure 2 shows this Pareto front, whereas the values of the system cost and availability of the 91 solutions are reported in Table 4. The points are relatively extensive on front.

The normalized membership value has been calculated for each value of Table 4 and is reported in Table 5. From this table, it can be observed that the maximum value is

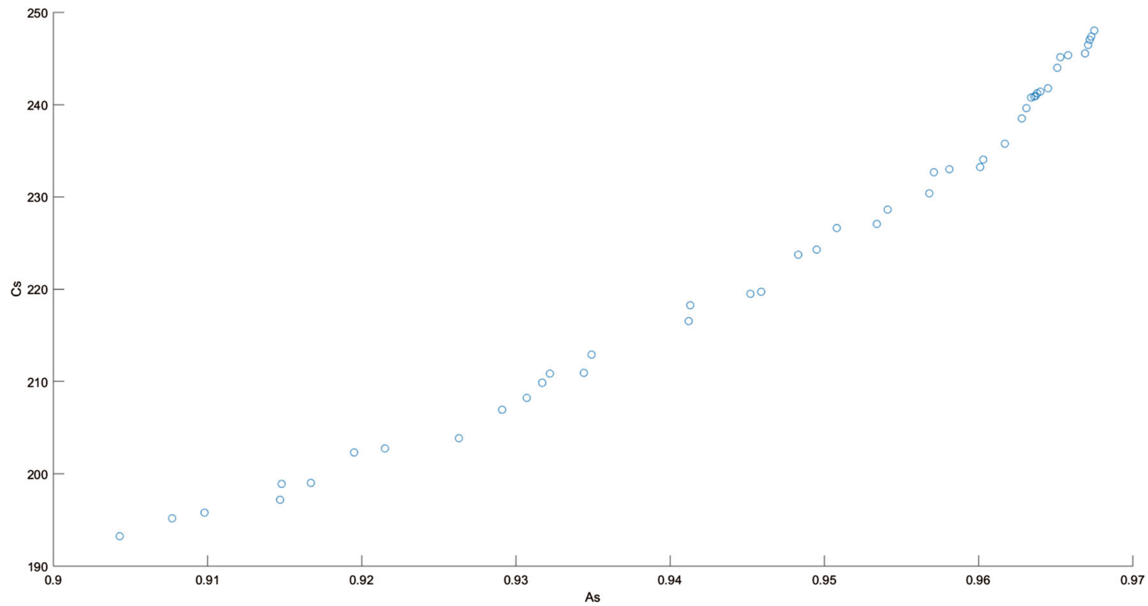

Fig. 2 Best Pareto front

those of the solutions 42 and 52 ($\mu^k = 0.0117856$). The related solutions represent the best compromise solutions of the obtained PF. The values of the decision variables of these solutions are given in Table 6. Therefore, this is the best compromise solution for the decision maker, where the system cost and availability are 219.7293 and 0.9459, respectively.

7 Conclusions

In this work, the multi-objective system availability and cost have been investigated. The problem has been solved by using the multi-objective strawberry algorithm which requires a few parameters. It was the first implementation of this efficient algorithm to solve this kind of problem. A fuzzy method has been used to determine the best compromise solution from the Pareto front for helping the decision maker. A numerical case study consisting of a system with 10 subsystems has been solved in order to highlight the applicability of the proposed solution approach.

In the future, this proposed approach will be hybridized with other methods to provide better performance metrics.

Table 5 Values of the normalized membership

No.	μ^k	No.	μ^k
1	0.0100301	51	0.0112983
2	0.0114616	52	0.0116248
3	0.0109487	53	0.0104333
4	0.0114702	**54**	**0.0117856**
5	0.0115622	55	0.0115671
6	0.0112388	56	0.0101616
7	0.0115565	57	0.0102174
8	0.0107814	58	0.0117108
9	0.0102555	59	0.0102107
10	0.0111854	60	0.0106976
11	0.0113291	61	0.0115565
12	0.0106830	62	0.0106878
13	0.0102107	63	0.0114469
14	0.0101119	64	0.0116146
15	0.0113554	65	0.0106559
16	0.0113291	66	0.0102174
17	0.0103917	67	0.0110212
18	0.0115771	68	0.0112388
19	0.0107117	69	0.0110275
20	0.0103917	70	0.0107073
21	0.0112829	71	0.0107117
22	0.0114236	72	0.0102476
23	0.0113192	73	0.0116248
24	0.0102555	74	0.0114560
25	0.0107202	75	0.0108589
26	0.0112983	76	0.0111854
27	0.0104333	77	0.0116146
28	0.0106976	78	0.0110212
29	0.0100301	79	0.0107202
30	0.0113554	80	0.0112912
31	0.0110275	81	0.0112829
32	0.0117108	82	0.0109487
33	0.0109577	83	0.0114702
34	0.0100301	84	0.0115134
35	0.0115134	85	0.0114469
36	0.0107073	86	0.0101616
37	0.0114236	87	0.0103821
38	0.0115622	88	0.0102476
39	0.0115671	89	0.0100301
40	0.0109577	90	0.0103821
41	0.0106559	91	0.0109632
42	**0.0117856**		
43	0.0107814		
44	0.0106878		
45	0.0115771		
46	0.0112912		
47	0.0114616		
48	0.0114560		
49	0.0106830		
50	0.0113192		

Bold indicates the Highest normalized membership value

Table 6 Best compromise solution

n	$\lambda(10^{-3})$	$\mu(10^{-2})$	C_s	A_s
(3, 2, 2, 3, 2, 3, 3, 2, 2, 2)	(0.5178, 0.5046, 0.6814, 0.9439, 0.6741, 0.9603, 0.7212, 0.7156, 0.8516, 0.9156)	(0.27249, 0.76399, 0.84547, 0.61919, 0.69907, 0.43691, 0.28401, 0.71324, 0.98624, 1.56609)	219.7293	0.9459

Bold indicates the Highest normalized membership value

Acknowledgements We are grateful to ESRC, Grant ES/L011859/1, for partially funding this research.

References

1. Marseguerra, M., Zio, E., Martorell, S.: Basics of genetic algorithms optimization for RAMS applications. Reliab. Eng. Syst. Saf. **91**, 977–991 (2006). https://doi.org/10.1016/j.ress.2005.11.046
2. Torres-Echeverría, A.C., Martorell, S., Thompson, H.A.: Design optimization of a safety-instrumented system based on RAMS+C addressing IEC 61508 requirements and diverse redundancy. Reliab. Eng. Syst. Saf. **94**, 162–179 (2009). https://doi.org/10.1016/j.ress.2008.02.010
3. Mellal, M.A., Zio, E.: An adaptive particle swarm optimization method for multi-objective system reliability optimization. J. Risk Reliab. **233**, 990–1001 (2019). https://doi.org/10.1177/1748006X19852814

4. Mellal, M.A., Zio, E.: An adaptive cuckoo optimization algorithm for system design optimization under failure dependencies. J. Risk Reliab. **233**, 1099–1105 (2019)
5. Chebouba, B.N., Mellal, M.A., Adjerid, S.: Multi-objective system reliability optimization in a power plant. In: The 3rd International Conference on Electrical Sciences and Technologies in Maghreb. Algiers, Algeria (2018)
6. Coit, D.W., Smith, A.E.: Reliability optimization of series-parallel systems using a genetic algorithm. IEEE Trans. Reliab. 45, 254–260, 266 (1996). https://doi.org/10.1109/24.510811
7. Chen, T.C.: IAs based approach for reliability redundancy allocation problems. Appl. Math. Comput. **182**, 1556–1567 (2006). https://doi.org/10.1016/j.amc.2006.05.044
8. Hsieh, Y.C., You, P.S.: An effective immune based two-phase approach for the optimal reliability–redundancy allocation problem. Appl. Math. Comput. **218**, 1297–1307 (2011)
9. Garg, H., Rani, M., Sharma, S.P.: An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique. Comput. Oper. Res. **40**, 2961–2969 (2013). https://doi.org/10.1016/j.cor.2013.07.014
10. Garg, H.: An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm. Beni-Suef Univ. J. Basic Appl. Sci. **4**, 14–25 (2015). https://doi.org/10.1016/j.bjbas.2015.02.003
11. Mellal, M.A., Zio, E.: A penalty guided stochastic fractal search approach for system reliability optimization. Reliab. Eng. Syst. Saf. (2016). https://doi.org/10.1016/j.ress.2016.03.019
12. Mellal, M.A., Zio, E.: System reliability-redundancy allocation by evolutionary computation. In: 2017 2nd International Conference on System Reliability and Safety. pp. 15–19. IEEE, Milan, Italy (2017)
13. Mellal, M.A., Williams, E.J.: Large scale reliability-redundancy allocation optimization problem using three soft computing methods. In: Modeling and Simulation based Analysis in Reliability Engineering, pp. 199–214. CRC Press, Francis & Taylor, Boca Raton (2018)
14. Huang, X., Coolen, F.P.A., Coolen-Maturi, T.: A heuristic survival signature based approach for reliability-redundancy allocation. Reliab. Eng. Syst. Saf. (2019). https://doi.org/10.1016/j.ress.2019.02.010
15. Juybari, M.N., Abouei-Ardakan, M., Davari-Ardakani, H.: A penalty-guided fractal search algorithm for reliability–redundancy allocation problems with cold-standby strategy. Proc. Inst. Mech. Eng. Part O J. Risk Reliab. (2019). https://doi.org/10.1177/1748006X19825707
16. Mellal, M.A., Zio, E.: System reliability-redundancy optimization with cold-standby strategy by an enhanced nest cuckoo optimization algorithm. Reliab. Eng. Syst. Saf. **201**, 106973 (2020). https://doi.org/10.1016/j.ress.2020.106973
17. Chebouba, B.N., Mellal, M.A., Adjerid, S.: Fuzzy multiobjective system reliability optimization by genetic algorithms and clustering analysis. Qual. Reliab. Eng. Int. (2020). https://doi.org/10.1002/qre.2809
18. Liu, G.S.: Availability optimization for repairable parallel-series system by applying Tabu-GA combination method. In: 10th IEEE International Conference on Industrial Informatics. pp. 803–808. Beijing, China (2012)
19. Liu, G.S.: Availability optimization for repairable n-stage standby system by applying Tabu-GA combination method. Int. J. Model. Optim. **3**, 245–250 (2013)
20. Mellal, M.A., Zio, E.: Availability optimization of parallel-series system by evolutionary computation. In: 3rd International Conference on System Reliability and Safety, pp. 198–202. Barcelona, Spain (2018)
21. Mellal, M.A., Salhi, A.: Parallel-series system optimization by weighting sum methods and nature-inspired computing. In: Applied Nature-Inspired Computing: Algorithms and Case Studies. Springer, Singapore (2020)
22. Mellal, M.A., Chebouba, B.N.: Cost and availability optimization of overspeed protection system in a power plant. In: 2019 Int. Conf. Adv. Electr. Eng., pp. 1–4 (2019). https://doi.org/10.1109/ICAEE47123.2019.9015115
23. Hu, L., Yue, D., Li, J.: Availability analysis and design optimization for a repairable series-parallel system with failure dependencies. Int. J. Innov. Comput. Inf. Control. **8**, 6693–6705 (2012). https://doi.org/10.4156/aiss.vol3
24. Salhi, A., Fraga, E.S.: Nature-inspired optimisation approaches and the new plant propagation algorithm. In: The International Conference on Numerical Analysis and Optimization (2011)
25. Rodman, A.D., Fraga, E.S., Gerogiorgis, D.: On the application of a nature-inspired stochastic evolutionary algorithm to constrained multi-objective beer fermentation optimisation. Comput. Chem. Eng. **108**, 448–459 (2018). https://doi.org/10.1016/j.compchemeng.2017.10.019
26. Selamoğlu, B.İ, Salhi, A.: The plant propagation algorithm for discrete optimisation: The case of the travelling salesman problem. In: Yang, X.S. (ed.) Nature-Inspired Computation in Engineering, pp. 43–61. Springer, Cham (2016)
27. Sulaiman, M., Salhi, A., Selamoglu, B.I., Kirikchi, O.B.: A plant propagation algorithm for constrained engineering optimisation problems. Math. Probl. Eng. (2014). https://doi.org/10.1155/2014/627416
28. Sulaiman, M., Salhi, A.: A seed-based plant propagation algorithm: The feeding station model. Sci. World J. (2015). https://doi.org/10.1155/2015/904364
29. Sulaiman, M., Salhi, A.: A hybridisation of runner-based and seed-based plant propagation algorithms. In: Yang, X.S. (ed.) Nature-Inspired Computation in Engineering, pp. 195–215. Springer, Cham (2016)
30. Sulaiman, M., Salhi, A., Kha, A., Shakoor, M., Khan, W.: On the theoretical analysis of the plant propagation algorithms. Math. Probl. Eng. (2018). https://doi.org/10.1155/2018/6357935
31. de la Fuente, D., Vega-Rodríguez, M.A., Pérez, C.J.: Automatic selection of a single solution from the Pareto front to identify key players in social networks. Knowl. Based Syst. (2018). https://doi.org/10.1016/j.knosys.2018.07.018
32. Zio, E., Bazzo, R.: A comparison of methods for selecting preferred solutions in multiobjective decision making. Comput. Intell. Syst. Ind. Eng. (2012). https://doi.org/10.2991/978-94-91216-77-0_2
33. Kong, X., Yong, C., Wang, C., Li, P., Yu, L., Chen, Y.: Multi-objective power supply capacity evaluation method for active distribution network in power market environment. Int. J. Electr. Power Energy Syst. **115**, 105467 (2020)
34. Zio, E., Bazzo, R.: A clustering procedure for reducing the number of representative solutions in the Pareto Front of multi-objective optimization problems. Eur. J. Oper. Res. (2011). https://doi.org/10.1016/j.ejor.2010.10.021
35. Qu, B.Y., Zhu, Y.S., Jiao, Y.S., Wu, M.Y., Suganthan, P.N., Liang, J.J.: A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. Swarm Evol. Comput. **38**, 1–11 (2018)
36. Abido, M.A.: Multiobjective evolutionary algorithms for electric power dispatch problem. IEEE Trans. Evol. Comput. **10**, 315–329 (2006)
37. Sakawa, M., Yano, H., Yumine, T.: An interactive fuzzy satisficing method for multiobjective linear programming problems and its application. IEEE Trans. Syst. Man. Cybern. **17**, 654–661 (1987)
38. Mellal, M.A., Pecht, M.: A multi-objective design optimization framework for wind turbines under altitude consideration. Energy

Convers. Manag. **222**, 113212 (2020). https://doi.org/10.1016/j.enconman.2020.113212

Mohamed Arezki Mellal is an Associate Professor at the Department of Mechanical Engineering, Faculty of Technology, M'Hamed Bougara University, Algeria and was a Visiting Scholar at the Center for Advanced Life Cycle Engineering, Department of Mechanical Engineering, University of Maryland, College Park, MD, USA. Likewise, he was a Visiting Scholar at various universities. He has published in several journals and conference proceedings. He has edited five books and authored seven book chapters. He is a member of the Algerian National Laboratory for Maintenance Education in conjunction with the European Union (Erasmus+). He has also been a committee member for over seventy international conferences. He serves as a regular reviewer for eighteen SCI-indexed journals and an editorial board member in seven peer-reviewed international journals.

Abdellah Salhi is a Professor of Operational Research at the University of Essex, UK. He studied for a Ph.D. at Aston University, in Birmingham, UK and thereafter, worked as a Research Fellow at the Universities of Leeds, Edinburgh, and Southampton, UK, before settling in Essex. His research work covers a variety of topics in OR and Data Science. He designed the Plant Propagation Algorithm and applied it to a number of challenging problems in design and optimization. His most recent work includes the design and implementation of a new approach to protecting the security of data. He has lead many research projects concerned with operations at container ports, green distribution in online shopping, smart town centres, optimum pollination with honeybees and others. His work appears in a number of prominent journals such as the ACM Transactions in Evolutionary Computation, the Annals of Operations Research, the Journal of the Operational Research Society, Computers and Operations Research, to name a few. He co-organizes the yearly EGL Workshop at one of the universities of Essex, Greenwich and UCL. He has been the Head of the Department of Mathematical Sciences of Essex University from 2010 to 2016.