

Chapter 16

Solving Multi-objective Optimal Design and Maintenance for Systems Based on Calendar Times Using NSGA-II



Andrés Cacereño, Blas Galván, and David Greiner

Abstract Due to technical progress and business competition, design alternatives and maintenance strategies have to be contemplated to optimize the performance of physical assets when new facilities are projected and built. That combined optimization (Design & Maintenance) is required by all industrial installations to develop their activity in an increasingly competitive environment. The Design and Maintenance combined optimization process is a complex problem which requires research and development. The objectives to optimize are Unavailability (due to production losses) and Maintenance Cost (due to overcharge when it is not optimal). The Design and Maintenance strategy for a technical system are optimized jointly by modifying its Functionability Profile, which is closely related to the system's availability. The Functionability Profile is generated by applying Monte Carlo Simulation that allows characterizing the process' randomness until the failure and to modify that Functionability Profile by the optimal Maintenance strategy. An application case is presented, where several configurations of the elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) are used to optimize the multi-objective problem, successfully finding non-dominated solutions with optimum performance for the simultaneous Design and Maintenance strategy combination.

Keywords Reliability · Availability · Multi-objective optimization · Preventive maintenance · Design · NSGA-II · Evolutionary multi-objective algorithms

A. Cacereño (✉) · B. Galván · D. Greiner
Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI),
Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain
e-mail: acacereño@iusiani.ulpgc.es

B. Galván
e-mail: blas.galvan@ulpgc.es

D. Greiner
e-mail: david.greiner@ulpgc.es

16.1 Introduction

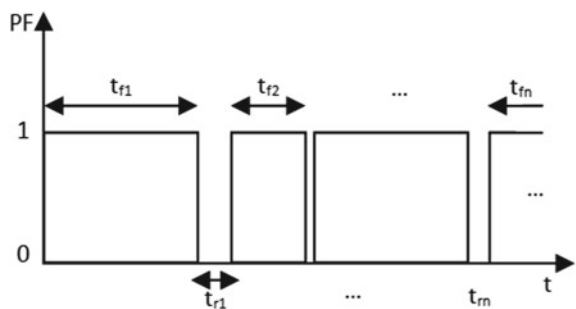
System Reliability is defined as the probability of being operating under particular conditions during a certain period [1]. The problem of systems design optimization based on their Reliability has been dealt by several authors, both in single-objective [2, 3] and multi-objective [4] cases, as an application of the well-known use of evolutionary algorithms/metaheuristics to solve complex problems in engineering design [5, 6]. However, it is still a live problem because of technical advances, the increase in the complexity of systems and the demand of consumers (among other aspects) [7].

The parameter which includes the process until the failure and recuperation for repairable systems is Availability. In repairable systems, information about the probability of being available at certain time to achieve their functions is given by their Availability.

System’s Availability can be deduced through its Functionability Profile. An example of Functionability Profile is shown in Fig. 16.1. The better the system Reliability is, the better its Availability will be. A priority objective in the industry is to obtain the maximum availability because if a system is “available”, resources will be being generated. However, when a system is not “available”, not only resources are not being generated, but also resources are being consumed until to recover the “available” state. When the system is not “available”, it is driven into unproductive phase [8].

The main reasons why a continuous operation system stops are a failure (after that, a recovery time is required) or a scheduled stop to perform a maintenance activity. The global improvement of system’s Reliability and Availability is possible through preventive maintenance [9]. If a preventive maintenance activity is performed, the unproductive phase will be more controlled than when reparations have to be performed because of a failure. Therefore, it is interesting to identify the optimum moment to make a stop to develop a preventive maintenance activity. In an ideal way, it has to be done before the occurrence of a failure but as near as possible to maximize the system’s “available” time. The Maintenance Optimization problem has been studied extensively [10].

Fig. 16.1 Functionability profile of a component (or device, or system)



From the foregoing, it can be deduced that both the system's optimum design and maintenance strategy improve its Reliability and Availability. Traditionally, the problem of improving Reliability by optimizing the system's design and maintenance strategy has been treated separately.

However, there are some works in which they have been jointly studied. In C.P. de Paula et al. [11] system's Availability and Cost are optimized through a decision process in which the number of redundant elements for a system (design) and the percentage of total resources allocated to maintain it are decided.

In the present paper, we face an unpublished problem, where the multi-objective optimization problem of minimization of the cost and maximization of the availability (or minimization of the unavailability) are handled: a set of optimal balanced solutions between Availability and Cost are provided, on the one hand, from the elements potentially included in the design, and on the other hand from the identification of the optimum moment in which the maintenance activity has to be performed. To obtain that, Functionability Profiles for system's devices have to be readjusted and, consequently, the system's Functionability Profile. Those Functionability Profiles, which are built and adjusted by using Discrete Events Simulation, are product of the Design and Maintenance Strategy.

The paper is organized as follows. Section 16.2 resumes the Methodology. Section 16.3 presents an application case. In Sect. 16.4 results are shown, and finally Sect. 16.5 introduces conclusions.

16.2 Methodology

16.2.1 Availability and Functionability Profile

Reliability is an intrinsic characteristic to a component (or device, or system, depending on disaggregation level, from now on device) which is related to the way in which the device has been designed and built. Maintainability can be intrinsic to devices when it is related to conditions of design (a piece that is difficult to access will be more complex to maintain) or extrinsic, for example, when it is related to availability of spares or to human team who has to perform the maintenance operation.

In Availability, those two parameters (Reliability and Maintainability) are related to define the way in which the device is able to fulfill the function for which it was designed during a period. In the present paper, the system's Availability is characterized by using its Functionability Profile. An example of Functionability Profile is shown in Fig. 16.1.

Functionability Profiles depend on times to failures ($t_{f1}, t_{f2}, \dots, t_{fn}$) and recovery times (t_{r1}, \dots, t_{rn}). In continuous operation devices, when Functionability Profiles are set to logical 1, it is considered that devices are operating. Conversely, when Functionability Profiles are set to logical 0, it is considered that devices are stopped (they are being maintaining or repairing after the failure). It is possible to deduce

from Fig. 16.1 that after an operation time (time to failure or time to perform a scheduled maintenance activity), a recovery time is necessary (time to repair after failure or time to perform a preventive maintenance activity).

As previously mentioned, Availability is tightly related to Functionability Profiles. Availability is characterized through the relation between device's operation times and the hoped operation total time for that device. That device will be able to fulfill its purpose during t_f times, so it is possible to characterize Availability $A(t)$ by using Eq. 16.1.

This approximation to characterize the Availability is called Operational Availability. Andrews and Moss [12] explain that Availability is an important measure of performance for repairable devices, which is represented in Eq. 16.2.

$$A(t) \cong \frac{t_{f1} + t_{f2} + \dots + t_{fn}}{t_{f1} + t_{f2} + \dots + t_{fn} + t_{r1} + t_{r2} + \dots + t_{rn}} \quad (16.1)$$

$$A(t) = \frac{MTTF}{MTTF + MTTR} \quad (16.2)$$

Mean Time To Failure ($MTTF$) and Mean Time To Repair ($MTTR$) are distinguished in Eq. 16.2. The approach shown in Eq. 16.2 is the base of the approximation that allows using Eq. 16.1. Availability ($A(t)$) is a variable with value between 0 and 1. The opposite of Availability is Unavailability ($Q(t)$), so $A(t) + Q(t) = 1$ and $Q(t) = 1 - A(t)$.

A priori, operation and recovery times are not known. They are random variables so they allow a statistical treatment. If a historic of both times is compiled and a statistical analysis is performed, these variables could be defined as probability density functions and probability distribution functions through their respective parameters. Functions can arise from a specific typology (exponential, Weibull, normal, for example). There are several Data Bases in the market (OREDA [13], CCPS [14]) which supply the characteristic parameters for the refereed functions, so operation and recovery times can be characterized for different failure modes of devices.

The economic Cost is a variable directly associated to recovery times. When systems are operating, economic income is generated. Conversely, when systems are recovering, economic cost is generated to return it to its operation state. If we want to avoid long recovery times, it is necessary to carry out a preventive maintenance activity ideally before the failure. Because of that stop is scheduled (for reasons such as human personnel are willing and trained, or spare parts are available) recovery times will be shorter. Therefore, it is possible to modify Functionability Profiles for system's devices by including preventive maintenance activities.

16.2.2 Building Functionability Profiles

As we want to analyze the system's Availability, we are going to show how it is possible to build Functionability Profiles for devices by using Discrete Events Simulation. With this end, information about how to characterize operation times to failure (TF) and recovery times after failure (TR) is needed. Characteristic parameters about their probability distribution laws are needed. In this book chapter, all possible device's failures are grouped in a unique failure mode. From the characterization of probability density and probability distribution functions both for operation times (TF) and recovery times (TR), Functionability Profiles for system's devices will be built by generating random times (Discrete Events Simulation). To modify Functionability Profiles, attending to preventive maintenance activities, operation times to preventive maintenance (TR) and recovery times due to preventive maintenance (TRP) will be introduced by generating random times. The process is shown below:

1. System's Life Cycle has to be decided and then, the process continues for all devices.
2. The device's Functionability Profile has to be initialized.
3. A time to preventive maintenance (TP) is extracted from the individual of the population that is being evaluated and a recovery time for preventive maintenance (TRP) have to be randomly generated, between limits previously fixed.
4. Attending to the device's distribution probability law, an operation time to failure (TF) has to be randomly generated, between limits previously fixed.
5. If $TP < TF$, a preventive maintenance activity is performed before a failure occurs. In this case, as many logical "ones" as TP units followed by as many logical "zeros" as TRP units have to be added to the device's Functionability Profile.
6. If $TP > TF$, a failure occurs before a preventive maintenance activity would be done. In this case, attending to the device's distribution probability law, a recovery time after failure (TR) has to be randomly generated, between limits previously fixed. Then, as many logical "ones" as TF units followed by as many logical "zeros" as TR units have to be added to the device's Functionability Profile.
7. Steps 4 to 6 have to be repeated until the end of the device's Life Cycle.
8. Steps 2 to 7 have to be repeated until Functionability Profiles have been built for all devices.
9. After to build Functionability Profiles, attending to the logic due to the serial (AND) or parallel (OR) distribution for the system's devices, the system's Functionability Profile has to be built.
10. Finally, system's Availability will be established by using Eq. 16.1, while the system operation cost is computed by adding partial costs due to recovery times.

Economic costs due to recovery times after failure and for preventive maintenance activities have to be established. With this purpose, a cost will be associated to unavailable time units. That cost will be bigger for recovery times after failure due to lack of foresight. The cost has to be computed while device's Functionability Profiles are built.

16.2.3 Multi-objective Optimization

Optimization results useful in practically all areas of our life. Our activities have to be optimized when we want to get the best possible result. However, when we have to solve complex problems we become aware of the suitability of employing that methodology. Optimization is very useful specially when the number of potential solutions is high and getting the best solution is very difficult. However, it will be possible to obtain sufficiently good solutions [15].

Optimization problems can be minimized or maximized for one or more objectives. In most cases, real world problems present various objectives for optimising at the same time (frequently in conflict each other). These problems are so-called "multi-objective" and their solutions arise from a solution set which represent the best compromise between objectives (Pareto optimal set) [16, 17]. This kind of problems are described by Eq. 16.3 (considering a minimization problem in this case) [15].

$$\min_x f(x) = \min_x [f_1(x), f_2(x), \dots, f_k(x)] \quad (16.3)$$

In Optimization problems defined by this way, the k functions have to be optimized at the same time. Classical optimization methods suggest converting the multi-objective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at time. Due to their ability to find multiple Pareto-optimal solutions in one single simulation run, a number of multi-objective evolutionary algorithms (MOEAs) were suggested after. In this paper, a MOEA is used to optimize an application problem. This algorithm is the so-called Non-dominated Sorting Genetic Algorithm II [18] (NSGA-II). The selection method in this algorithm is based on the concept of non-dominance.

In this paper, the problem is to optimize the Design and Maintenance strategy for an industrial system based on two different objectives in conflict, Availability and Cost. We wish maximum Availability and minimum maintenance Cost. The more investment in maintenance, the greater system's Availability will be obtained. However, this policy implies a higher unwanted cost, being this the conflict between objectives. Not only maintenance strategy is considered but also the system's design is optimized too based on Availability and its influence in Costs due to Maintenance strategy. The process is discussed below.

16.3 Application Case

The proposed methodology has been applied to a fluid injection system from industry, based on⁴ as an example. That system is basically formed by cut valves (V_i) and impulsion pumps (P_i) as is shown in Fig. 16.2.

As it was exposed above, optimization objectives are, on the one hand, to maximize the system's Availability and, on the other hand, to minimize Costs due to system's unproductive phases (both because the system is being recovered and because the system is being maintained). To do that:

- For all system's devices, the optimum moment to perform a preventive maintenance activity has to be established.
- Including redundant devices as P2 and/or V4 has to be decided by evaluating Design alternatives. Including redundant devices will improve the system's Availability but it will worsen its Maintenance Cost.

Population individuals for the Optimization process will be characterized by its chromosome. Chromosomes will be shaped by real number strings with 0 as minimum value and 1 as maximum value (decision variables). They will be codified as $[B_1 B_2 T_1 T_2 T_3 T_4 T_5 T_6 T_7]$, where the presence of redundant devices, P2 and V4, is decided by B_1 and B_2 , respectively, and optimum times to perform a preventive maintenance activity to devices are represented by T_1 to T_7 . Data set for system's devices used to the optimization process are shown in Table 16.1.

The Software Platform PlatEMO [19] (programmed in MATLAB) was used to optimize the problem. The open source platform PlatEMO includes more than 50 multi-objective evolutionary algorithms, more than 100 multi-objective test problems, along with several widely used performance indicators. In this case, the reliability and maintenance analysis software has been developed and implemented to solve the problem described above in the platform.

The parameters set used to configure the simulation process is shown in Table 16.2. The evolutionary multi-objective algorithm used in this paper is the so-called Non-dominated Sorting Genetic Algorithm II (NSGA-II), a method based on the concept of non-dominance. The method was configured with several parameters. All cases were running five times with a stopping criterion of 5,000,000 evaluations, with Simulated Binary Crossover (SBX), and crossover distribution and mutation distribution indexes of 20. Two population sizes were analysed with 50 and 100 individuals. Mutation probabilities were changed between 0.5, 1 and 1.5 genes per chromosome

Fig. 16.2 Application case: fluid injection system

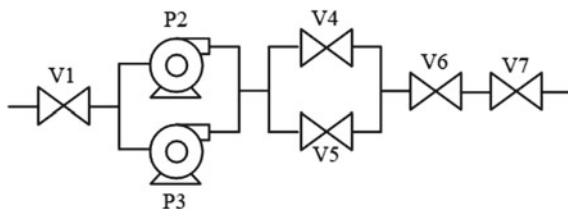


Table 16.1 Data set for the system’s devices

Parameter	Quantification	Source
Life Cycle	700,800 h	–
Recovery Cost	0.5 units	Expert judgement
Maintenance Cost	Recovery Cost/4	Expert judgement
Pump TF min	1 h	Expert judgement
Pump TF max	70,080 h	Expert judgement
Pump λ Exponential Law	$159.57 \cdot 10^{-6}$ h	OREDA 2009
Pump TR min	1 h	Expert judgement
Pump TR max	24.33 h	$\mu + 4\sigma$
Pump TR μ Normal Law	11 h	OREDA 2009
Pump TR σ Normal Law	3.33 h	$(\mu - TR_{min})/3$
Pump TP min	2,920 h	Expert judgement
Pump TP max	8,760 h	Expert judgement
Pump TRP min	4 h	Expert judgement
Pump TRP max	8 h	Expert judgement
Valve TF min	1 h	Expert judgement
Valve TF max	70,080 h	Expert judgement
Valve λ Exponential Law	$44.61 \cdot 10^{-6}$ h	OREDA 2009
Valve TR min	1 h	Expert judgement
Valve TR max	20.83 h	$\mu + 4\sigma$
Valve TR μ Normal Law	9.5 h	OREDA 2009
Valve TR σ Normal Law	2.83 h	$(\mu - TR_{min})/3$
Valve TP min	8,760 h	Expert judgement
Valve TP max	35,040 h	Expert judgement
Valve TRP min	1 h	Expert judgement
Valve TRP max	3 h	Expert judgement

Table 16.2 Simulation configuration parameters

Parameter	Configuration
Method	NSGA-II
Evaluations	5,000,000
Population	50–100
Crossover probability	1
Crossover distribution index	20
Mutation probability	0.055–0.111–0.166
Mutation distribution index	20
Executions	5

(0.055, 0.111 and 0.166 respectively). Six cases (combination of two population sizes and 3 mutation rates) were finally evaluated.

16.4 Results

The different configurations for the optimization method were executed five times each. The Hypervolume [20] (HV) average value evolution (among five executions and for each configuration) is shown in Fig. 16.3. The higher the number of evaluations, the higher the improvement of the Hypervolume is observed.

The detail of the last evaluations is shown in Fig. 16.4. It is possible to check that the parameters configuration with population of 100 individuals and mutation probability of 0.055 (0.5 gen per chromosome) finally presents the higher Hypervolume average value.

The values of the main measures obtained for the final evaluations are shown in Table 16.3. These are the Average, Median, Minimum Value, Maximum Value and Standard Deviation of the Hypervolume metric. Firstly, the parameters configuration with population of 50 individuals and mutation probability of 0.055 (0.5 gen per chromosome) presents the higher median of the Hypervolume value. Secondly, the parameters configuration with population of 100 individuals and mutation probability of 0.055 (0.5 gen per chromosome) presents the higher average and minimum of the Hypervolume value. Thirdly, the parameters configuration with population of 100 individuals and mutation probability of 0.111 (1 gen per chromosome) presents the higher maximum of the Hypervolume value. Finally, the parameters configuration

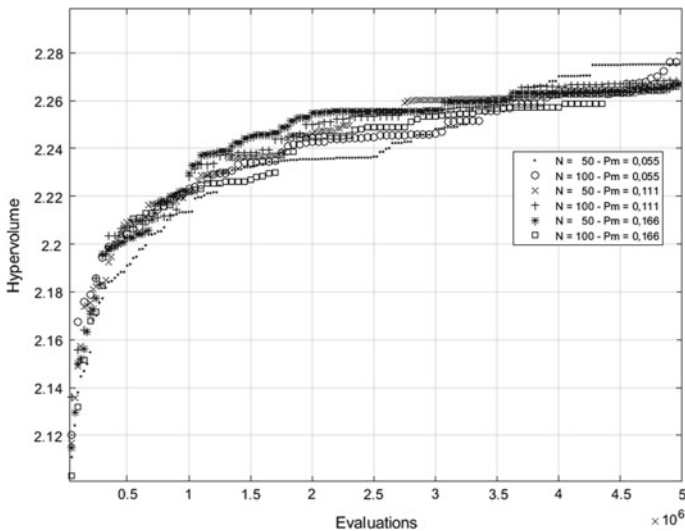


Fig. 16.3 Hypervolume average value evolution

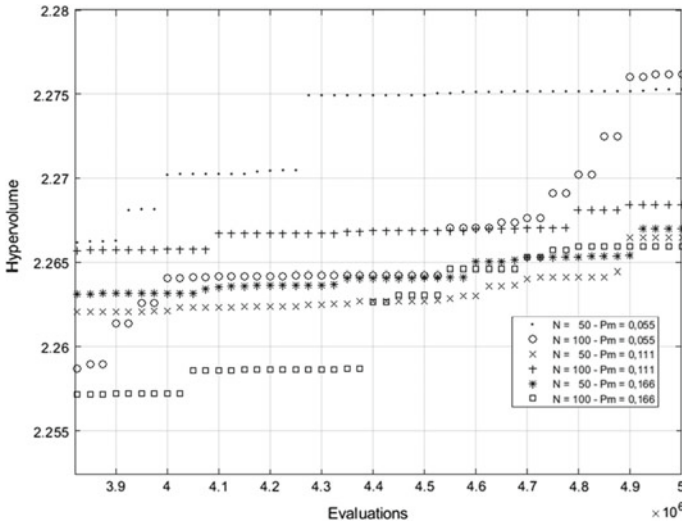


Fig. 16.4 Hypervolume average value evolution (detail)

with population of 50 individuals and mutation probability of 0.166 (1.5 gen per chromosome) presents the lowest standard deviation of the Hypervolume value.

Box plots of the final Hypervolume value distribution for the last evaluation are shown in Fig. 16.5. It is possible to observe some details described above, related to average, median, minimum, maximum and the standard deviation of the final Hypervolume values. The parameters configuration with population of 50 individuals and mutation probability of 0.055 presents the highest median of the Hypervolume value. The parameters configuration presents the higher minimum of the final Hypervolume value. The parameters configuration with population of 100 individuals and mutation probability of 0.111 presents the highest maximum of the final Hypervolume value. The parameters configuration with population of 50 individuals and mutation probability of 0.166 presents the lowest standard deviation of the final Hypervolume value.

In order to establish if any of the six parameter configurations works better than others, a statistical significance hypothesis test was conducted. Particularly, the procedure starts detecting significant differences among the results obtained by applying the Friedman’s test. It responds the question: “Are there results with different median?” When there are two or more result sets, the null hypothesis (H_0) claims that median are equals (no differences among methods). If H_0 is rejected, differences among methods exist, and a post hoc test is run in order to find the concrete pairwise comparisons which produce differences. In our case, the average rank computed through the Friedman’s test is shown in Table 16.4.

The parameters configuration with population of 100 individuals and mutation rate of 0.055 presents the lowest average rank computed through the Friedman’s test (the best in this case, as a maximization problem is analyzed -maximum Hypervolume

Table 16.3 Hypervolume statistics of the optimization results

ID	Parameter	N = 50—Pm = 0.055
1	Average	2.2753
	Median	2.2775
	Minimum Value	2.2572
	Maximum Value	2.2879
	Standard Deviation	0.0124
ID	Parameter	N = 100—Pm = 0.055
2	Average	2.2762
	Median	2.2732
	Minimum Value	2.2676
	Maximum Value	2.2934
	Standard Deviation	0.0101
ID	Parameter	N = 50—Pm = 0.111
3	Average	2.2665
	Median	2.2700
	Minimum Value	2.2442
	Maximum Value	2.2909
	Standard Deviation	0.0199
ID	Parameter	N = 100—Pm = 0.111
4	Average	2.2726
	Median	2.2680
	Minimum Value	2.2542
	Maximum Value	2.2939
	Standard Deviation	0.0167
ID	Parameter	N = 50—Pm = 0.166
5	Average	2.2671
	Median	2.2667
	Minimum Value	2.2558
	Maximum Value	2.2746
	Standard Deviation	0.0071
ID	Parameter	N = 100—Pm = 0.166
6	Average	2.2659
	Median	2.2643
	Minimum Value	2.2413
	Maximum Value	2.2896
	Standard Deviation	0.0173

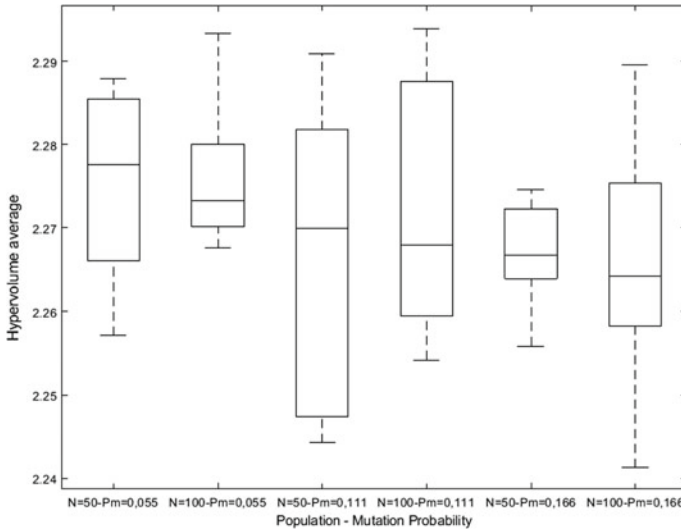


Fig. 16.5 Box plots of the final hyper volume value distribution

Table 16.4 Average rank computed through the Friedman’s test (best in bold type)

ID	NSGA-II Configuration	Average Rank
1	N = 50—Pm = 0,055	3.00
2	N = 100—Pm = 0,055	2.60
3	N = 50—Pm = 0,111	4.40
4	N = 100—Pm = 0,111	3.20
5	N = 50—Pm = 0,166	3.60
6	N = 100—Pm = 0,166	4.19

is desired-). However, the p-value computed by Friedman’s test is 0.6212. This p-value is higher than the level of significance α (0.05) so the null hypothesis “median are equals” can’t be rejected. This implies it is not possible to establish that any parameter configuration performs better than any other. In the conditions in which the experiment was developed, there aren’t significant differences between performances from different configurations. A procedure for conducting multiple comparisons involving all possible pairwise comparisons, as, e.g. described by Garcia S. and Herrera F. in [21], is therefore here not necessary.

The possible solutions to the problem provided through the last generation of the evolutionary process of the five accumulated executions for all configurations are shown in Fig. 16.6. Some optimum solutions belonging to the obtained non-dominated front are shown in Table 16.5 (these solutions are rounded and numbered in Fig. 16.6). Unavailability is shown in fraction, Cost is shown in economic units and the rest of variables represent, for the respective devices, optimum times to perform a preventive maintenance activity in hours.

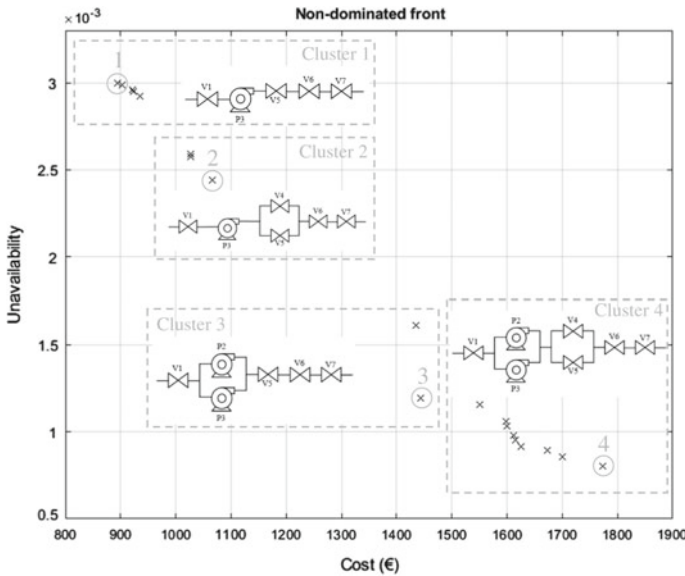


Fig. 16.6 Non dominated solutions (black crosses), and their configuration designs, clustered. Chosen representative solutions (Table 16.5) are additionally circled and numbered

Table 16.5 Sample of some optimum solutions

ID	Unavailability	Cost [un]	V1 [h]	P2 [h]	P3 [h]	V4 [h]	V5 [h]	V6 [h]	V7 [h]
1	0.0029979	894.20	30,887	∅	8,344	∅	29,391	24,051	33,860
2	0.0024687	1,066.92	21,430	∅	8,718	10,299	28,043	31,234	31,442
3	0.0011928	1,443.00	29,592	8,179	8,597	∅	20,005	33,923	29,773
4	0.0008019	1,772.59	34,766	8,386	8,467	29,272	34,531	32,968	31,000

The solution with the lowest Cost (ID1) (894.20 economic units) presents the biggest Unavailability (0.0029979). These values are followed by periodic optimum times (hours) measured from the moment in which the Life Cycle starts (time for performing the preventive maintenance activity (TR) is not included). In that case, it is possible to observe that periodic optimum times to preventive maintenance for devices P2 and V4 are not supplied. It is caused because the design alternative did not consider including such devices. The opposite case shows the biggest Cost (ID4) (1,722.59 economic units) and the lowest Unavailability (0.0008019). In this case, periodic optimum times to perform preventive maintenance activities are supplied for all devices. It is caused because the design alternative considered including devices P2 and V4. Other optimum solutions were found between those two solutions (ID2 and ID3). Decision makers, attending to their requirements, will have to decide which design is the preferable to choose.

Moreover, solutions have been clustered in Fig. 16.6 attending to their final design. Solutions contained by Cluster 1 are the solutions in which non redundant devices have been included in the design. Solutions contained by Cluster 2 are the solutions in which a redundant valve has been included in the design. Solutions contained by Cluster 3 are the solutions in which a redundant pump has been included in the design. Finally, solutions contained by Cluster 4 are the solutions in which both a redundant valve and a redundant pump have been included in the design. Final designs for each Cluster are shown in Fig. 16.6.

16.5 Conclusions

A successful methodology has been presented and demonstrated by a practical test case where proper non-dominated solutions for minimum unavailability and cost objectives have been generated. It has been possible by generating functionality profiles for several designs of the analyzed technical system, using Discrete Events Simulation, and varying those functionality profiles with the inclusion of maintenance activities before the failure. The evolutionary multi-objective algorithm NSGA-II was used to perform the optimization process. This method allowed obtaining optimum solutions attending to the design and maintenance strategy for the technical system. The goal for devices included in the design, was to obtain the sets of optimum times between maintenance activities with the best unavailability-cost relations. A system test case with 7 possible devices was used, including pumps and valves.

A set of different evolutionary multi-objective algorithm parameters configuration has been tested with the purpose of determining its effect in the optimization process. The best non-dominated solutions were archived. A test hypothesis was built with the objective of determining what parameter configuration presents the best performance. It is possible to conclude that significant differences were not found so, in the conditions defined for the experiment, no parameter configuration worked better than any other.

As future work, a comparison among several state of the art evolutionary multi-objective optimizers (EMO) will be performed, including, as stated in e.g. [22], a representative of each of the different three main paradigms of evolutionary multi-objective optimizers attending to their selection method.

Acknowledgments A. Cacereño is recipient of a contract from the Program of training for Predoctoral research staff of University of Las Palmas de Gran Canaria. The authors are grateful for this support.

References

1. Misra KB (2008) Reliability engineering: a perspective. handbook of performability engineering, vol 2008. Springer, pp 253–259
2. Kuo W, Prasad VR (2000) An annotated overview of system-reliability optimization. *IEEE Trans Reliab* 49(2):176–87
3. Kuo W, Wan R (2007) Recent advances in optimal reliability allocation. *Computational intelligence in reliability engineering*, vol 2007. Springer, pp 1–36
4. Greiner D, Galván B, Winter G (2003) Safety systems optimum design by multicriteria evolutionary algorithms. *Evolutionary multi-criterion optimization. Lecture Notes in Computer Science*, vol 2632. Springer, pp 722–736
5. Greiner D, Periaux P, Quagliarella D, Magalhaes-Mendes J, Galván B (2018) Evolutionary algorithms and metaheuristics: applications in engineering design and optimization. *Math Probl Eng* 2018:1–4
6. Greiner D, Galván B, Périaux P, Gauger N, Giannakoglou K, Winter G (2015) Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences. *Computational Methods in Applied Sciences*, vol 36. Springer
7. Coit DW, Zio E, The evolution of system reliability optimization. *Reliab Eng Syst Saf*. <https://doi.org/10.1016/j.res.2018.09.008>
8. Boliang L, Jianping W, Ruixi L, Jiayi W, Hui W, Xuhui Z (2019) Optimization of high-level preventive maintenance scheduling for highspeed trains. *Reliab Eng Syst Saf* 183:261–275
9. Gao Y, Feng Y, Zhang Z et al (2015) An optimal dynamic interval preventive maintenance scheduling for series systems. *Reliab Eng Syst Saf* 142:19–30
10. Faddoul R, Raphael W, Chateaneuf A (2018) Maintenance optimization of series systems subject to reliability constraints. *Reliab Eng Syst Saf* 180:179–188
11. De Paula CP, Visnadi LB, De Castro HF (2019) Multi-objective optimization in redundant system considering load sharing. *Reliab Eng Syst Saf* 181:17–27
12. Andrews J D, Moss T R. Reliability and risk assessment 2nd Edition. Professional Engineering Publishing Limited, London and Bury St Edmunds, UK. ISBN 1 86058 290 7
13. OREDA participants. OREDA – Offshore reliability data handbook. 5th Edition. Published by: OREDA participants. Prepared by: SINTEF, Distributed by: Det Norske Veritas (DNV). ISBN 978-82-14-04830-8
14. Center for Chemical Process Safety. Guidelines for process equipment reliability data with data tables. Center for Chemical Process Safety of the American Institute of Chemical Engineers. New York: ISBN 0-8169-0422-7
15. Simon D (2013) Evolutionary optimization algorithms. John Wiley & Sons, Hoboken, New Jersey
16. Coello CA (2015) Multi-objective evolutionary algorithms in real-world applications: some recent results and current challenges. In: Greiner D et al (eds) *Advances in evolutionary and deterministic methods for design, optimization and control in engineering and sciences*, *Computational Methods in Applied Sciences*, vol 36, Springer, pp 3–18
17. Emmerich M, Deutz A (2018) A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Nat Comput* 17(3):585–609
18. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
19. Tian Y, Cheng R, Zhang X, Jin Y (2017) PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Comput Intell Mag* 12(4):73–87
20. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans Evol Comput* 7(2):117–132
21. García S, Herrera F (2008) An extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for all pairwise comparisons. *J Mac Learn Res* 9:2677–2694
22. Greiner D, Periaux P, Emperador J, Galván B, Winter G (2017) Game theory based evolutionary algorithms: A review with nash applications in structural engineering optimization problems. *Arch Comput Meth Eng* 24:703–750