23rd ABCM International Congress of Mechanical Engineering
December 6-11, 2015, Rio de Janeiro, RJ, Brazil

COMPARISON ON THE APPLICATION BETWEEN MULTI-OBJECTIVE OPTIMIZATION METHODS: GENETIC ALGORITHM, PARTICLE SWARM AND FIREFLY ALGORITHM TO AVAILABILITY AND COST OPTIMIZATION OF THE REDUNDANT SYSTEM

Cassio Pereira de Paula
Andrei Bavaresco Rezende
Hélio Fiori de Castro
UNICAMP-FEM, Department of Mechanical Design, P.O. Box 6051, Campinas, SP 13083-860, Brazil
cassio2014@fem.unicamp.br / abavaresco@fem.unicamp.br / heliofc@fem.unicamp.br

Abstract. *Efficient and reliable engineering systems are a necessity nowadays. The increase of system availability can be got though the increase of each component's availability or with the use of redundant system, however, this brings higher cost. This paper presents an availability optimization and an overall cost optimization of an engineering system assembled in a series configuration with redundancy of units. As more than one objective function is considered, the multi-objective optimization is applied and returns a Pareto optimal set of solutions. The Pareto ranking is applied to evaluate the fitness of each potential solution.*
The optimization methods used are genetic algorithm, firefly algorithm and particle swarm. In this context, the aim of this paper is compare all the results for the multi-objective optimization: maximization the availability and minimization of the overall cost.

Keywords: *Availability, Multi-objective optimization, Genetic algorithm, Particle swarm optimization, Firefly algorithm.*

1. INTRODUCTION

The efficiency of engineering systems is related to the optimization of many parameters that influence the performance. Nowadays it is necessary to develop systems with high availability. Thus, the system may work well for long periods of time keeping its design function. The availability increase of a system can be reached through two possibilities: increasing the availability of each component (through reliability or maintenance improvement) or using redundant system. However, it will increase the design and maintenance cost.

Availability of each component that composes an engineering system can be increased by improving reliability and maintainability. If the reliability is increased, the system can work for longer periods and if the maintenance program is improved, the system can be repaired quickly.

It's possible to note that almost all systems deteriorate due to age and its usability. This deterioration raises operating cost and produces less competitive products, see Chang (2014).

Defining reliability as the probability that a system or component will perform its design function successfully at an period of time, and maintainability as the ability to renew the system after a failure in a period of time, enabling it continue performing its design function. We can easily obtain the reliability and maintainability function respectively by failure time and repair time analysis. The use of redundant components also increase availability, but it's important to observe it can increase the overall cost and other engineering parameters as volume and weight.

The exponential statistical distribution will be used to model data in this paper and it's characterized by a constant failure rate (λ) in the time domain, which is suitable for components or systems with long life, such as electronic components. Others distributions as Weibull (suitable for mechanical system), Normal, Lognormal, Rayleigh and Gamma can be applied also in failure analysis and maintenance analysis. However, the failure rate of these functions is not constant in time.

It is possible to note which the increase of the availability can make the product more expensive, so the application of optimization techniques, in order to relate the availability increasing and cost reducing, is an interesting proposal to determine the ideal number of redundancies and maintenance resources in a project. This problem involves the simultaneous solution of multiple performance criteria. A multi-objective method is proposed to consider individual analysis of each objective function (availability and cost). As more than one objective function is considered, Pareto ranking technique is applied too.

A design process usually aims at high system availability with low associated costs. However, in general, the relation between availability and cost is direct, i.e., the higher the availability level, the higher the costs. Thus the objectives of designing a system with high availability and also with low associated costs are normally conflicting, according to Lins and Droguett (2008).

Inside engineering problems, like the availability optimization of redundant system, it's possible to involve simultaneous considerations of multiple performance criteria, in which objectives are non-commensurate and are frequently in conflict among themselves. In multi-objective problems, it is very important the search for a good dispersion of solutions, to offer to the decision maker a set of good possibilities, so that the choice of which solution is most appropriate to the project can be made more easily, said previously, Castro and Cavalca (2005).

Traditional methods, such as Lagrange Multiplier are inefficient when problems involving large number of parameters, because they demand complex mathematics which difficult the computational implementation.

Considering this context, this paper proposes a comparison between techniques of multi-objective optimization based on genetic algorithms, particle swarm and fireflies' algorithm for availability maximization and costs minimization. The multi-objective genetic algorithm is a search method that simulates the adaptation of biological systems, solutions are represented by individuals who form a population where the fittest can be combined and operated through rolling to provide new individuals or solutions. The multi-objective Particle Swarm algorithm (PSO) is based on a population of simple agents interacting locally with one another and with their environment, where each movement of optimized particles depends on three parameters: sociability factor, individuality factor (cognitive) and inertia weight, the algorithm combines these parameters with a randomly generated number to determine the next location of the particle.

Lastly, the firefly algorithm is based on bioluminescent behavior of fireflies, the method is organized considering a firefly as a possible solution of the problem and each solution might moves randomly in the sample space, attracting or being attracted to other fireflies. The evaluation of method is made through this intensity of the brightness generated.

2. REDUNDANT SYSTEM

It is necessary to ensure that an equipment is able to accomplish satisfactorily its design functions. Thus the aim to use a redundant system is to keep the capacity of a system to overcome the failure of its components through the use of redundant resources, in other words, a redundant system has a second device that is immediately available for use when happen the failure in the first device. Currently this concept is used in several departments to always ensure high availability.

Redundant systems can be defined as passive, when the redundant component just start its works after the main component failure, or can be active when the redundancy works together with others components. Other important definition is about the redundant system organization. The concept of series and parallel redundancy can be applied in approach of availability where a set of component are considered organized in series when the success of the operation depend of the proper operation of each component. However, the parallel system is observed when the operation success depends at least of the success of one of its components.

In this paper, we are using a redundant system, which can be represented by a series of parallel system with five stages as shown in Fig. 1.

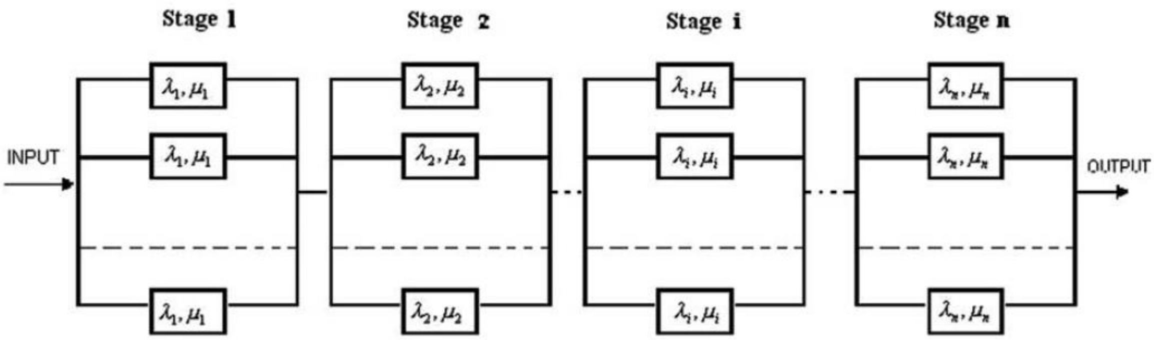

Figure 1. Redundant system (CASTRO and CAVALCA, 2005)

3. AVAILABILITY OPTIMIZATION PROBLEM

As previously mentioned, there are two ways to improve the availability and in both cases you also increase cost. The first method is to increase the availability of each component through increase of failure time or decrease of repair time. The second approach is to introduce redundant components in the global system.

Optimization methods are necessary to keep cost, volume and weight within acceptable limits, while achieving high availability levels, see Castro and Cavalca (2005).

3.1 Availability analysis

Availability is defined as the probability of a system's successful operation in determined period of time, it can be calculated by the ratio between life time and total time to failures of the equipment. Defining *MTTF* as the life time, which can be obtained from failure analysis, and *MTTR* is defined as the mean time to repair that can be calculated from maintenance analysis and represents the repair time. Therefore, the availability may be calculate by Eq. (1) below.

$$A = \frac{MTTF}{MTTF + MTTR} \tag{1}$$

In this paper, it will be used the exponential distribution to be representative for the reliability and maintainability, it's possible to assume that *MTTF* is the inverse of failure rate and *MTTR* is the inverse of repair rate. It can be presented on following Eq. (2) and Eq. (3).

$$MTTF = \frac{1}{\lambda} \tag{2}$$

$$MTTR = \frac{1}{\mu} \tag{3}$$

Where λ is failure rate and μ is repair rate. Giving sequence, the availability is expressed in the Eq. (4) below.

$$A = \frac{\mu}{\mu + \lambda} \tag{4}$$

3.2 Dependability analysis

The concept of dependability (d) was developed by Wohl (1966) and presents an important cost-benefit ratio between the maintenance and systems life time.

Dependability is an important design parameter because provides a single measurement of the performance conditions by combining failure and repair rates, associated to reliability and maintainability, according to Castro and Cavalca (2005). This concept is defined as the probability that component does not fail, or does fail and can be repaired in an acceptable period of time.

During the analysis of the dependability, the failure and repair rates are assumed constant, so we can express (d) by the Eq. (5).

$$d = \frac{MTTF}{MTTR} = \frac{\mu}{\lambda} \tag{5}$$

It is possible easily obtain a relationship between availability and dependability from the combination of equations (4) and (5).

$$A = \frac{\mu}{\mu + \lambda} = \frac{\frac{\mu}{\lambda}}{1 + \frac{\mu}{\lambda}} = \frac{d}{1 + d} \tag{6}$$

The first objective function represented by Fig. 1, where A_i is the availability of the components of subsystem (i) and Y_i is the number of redundant components inside this subsystem (i). The equation that represents the total availability is:

$$A_s = \prod_{i=1}^{n} [1 - \left(\frac{1}{1+d_i}\right)^{Y_i}] \tag{7}$$

4. COST OPTIMIZATION PROBLEM

The availability rate increase raises some cost as maintenance and development, but in some cases, the safety and reliability are essential. An example is the aeronautical industry where just one failure can cause many deaths. Some systems where high availability is needed and high performance, the system overall cost is raised because the maintenance operations should receive high investment. Therefore, optimization methods must be applied to reach the maximum of these parameters within certain limits cost.

The cost analysis which represent a system shown in Fig. 1, can be described as the sum of the design cost shown in Eq. (8) plus the sum of cost of corrective maintenance for a fixed time (T) shown in Eq. (9).

$$C = \sum_{i=1}^{n} c_i y_i \tag{8}$$

$$CM(T) = \sum_{i=1}^{n} Rec_i Crec_i + \sum_{i=1}^{n} q_i(T) y_i cm_i \tag{9}$$

Defining c_i as the design cost of each component, Y_i as the number of components in each stage, Rec_i as the percentage of corrective maintenance resource, $Crec_i$ as the cost of corrective maintenance resource and $q_i(T)$ as the failure probability of a component in subsystem (i). It is important to detach that by corrective maintenance resource, we would like to approach the complete support: maintenance team, equipment, financial resource and others.

In order to describe the influence of corrective maintenance resource on availability value, we can introduce the impact index I, defined as the amount by the *MTTR* decrease if 100% of corrective maintenance resource are applied to a specific component.

The impact index (I) expresses the dependability ratio sensitivity to the corrective maintenance resource, according Castro and Cavalca (2005).

The dependability d_i can be expressed in terms of I. Where $d1_i$ is the dependability ratio when no maintenance resource is applied.

$$d_i=[(I\text{-}1)Rec_i+1]d1_i \tag{10}$$

5. PROBLEM FORMULATION

In terms of a single objective optimization, just one unique solution is obtained and it's called optimal solution. However, when multiple conflicting objectives are considered, finding a single solution is very difficult to obtain or it might even not exist. Under this circumstance, it is desired to obtain a set of non-dominated solutions. This set is also known as optimal Pareto set, in regard to the Italian economist Vilfredo Pareto, who generalized the "optimal" concept in multi objective context, see Coelho (2002).

In this context of optimization approach, it is possible define the objective functions in Eq. (11) and Eq. (12), where the first function is unavailability (1-*As*) and the second function represent the overall cost. It is necessary to minimize both. In this paper the simulations were prepared to minimize the objective functions then it's significant to note that for to have the availability maximized it's possible to minimize the unavailability.

$$f(1) = 1 - As = 1 - \prod_{i=1}^{n} [1\text{-}(\frac{1}{1+[(I\text{-}1)Rec_i+1]d1_i})^{y_i} \tag{11}$$

$$f(2) = CT = \sum_{i=1}^{n} c_i y_i + \sum_{i=1}^{n} Rec_i Crec_i + \sum_{i=1}^{n} q_i(T) y_i cm_i \tag{12}$$

Our decision variables are the percentage of corrective maintenance resource applied in subsystem (i) Rec_i and the number of redundant components in each stage y_i.

6. SEARCH METHODS

6.1 Multi Objective Genetic Algorithm (NSGA-2)

The Genetic Algorithm (GA) is the most popular technique in the evolutionary program, it was developed by Holland (1975) that aimed to design artificial systems based in natural systems evolution. The main advantage of this method is that not sensitive to noise during the search. The Genetic algorithm transforms a population of individuals using reproductive principles and selection of the fittest, by applying operations such as recombination and mutation, Lobato (2008). According Cortes (2013), these algorithms have been used to efficiently find solutions in a variety of complex problems. Such as: optimization functions numerical, generally combinatorial optimization, machine learning, image processing, robotics.

In terms of GA a potential solution is called individual and a set of solution is known as population, where every member of this population has a phenotype and genotype. The selection is made by comparing the individual's fitness with other individual's fitness inside a population and an individual with best fitness has more probability to be selected (elitism). The fitness is calculated by the phenotype and represents as an individual is closer to the optimum point.

Individuals are changes through mutations where a gene or part of the genetic load is modified. The genetic background of individuals can recombine by crossover or recombination. In this process, two individuals produce children, which have genetic load of recombined parents. There are other operators in the evolutionary algorithms such as, for example, inversion, altering the order of the genetic chain.

An advantage of the usage of the (GA) is that, different to the classic optimization algorithms, it does not work with only one point in the search space, but with a group of points simultaneously. The number of points is previously determined by a parameter known as population size, according to Castro and Cavalca (2005).

This paper proposes the use of an algorithm called NSGA-II to simulate genetic algorithm, because of its simplicity and demonstrated superiority over other existing applications. This code can be found in a package available in Matlab software inside optimization toolbox. The Non-dominated Sorting Genetic Algorithm (NSGA2) was proposed by Srinivas and Deb (1994), and is based on several layers of classifications of the individuals. Before selection is performed, the population is ranked on the basis of domination (using Pareto ranking): all nondominated individuals are classified into one category, see Mukta et al. (2012).

6.2 Multi Objective Firefly Algorithm (MOFA)

Fireflies are characterized by their flashing light produced by its bioluminescence. Such flashing light may serve as the primary courtship signals for mating. The algorithm was developed by Yang for continuous optimization and it’s based on the bioluminescent behavior of fireflies colonies and it's organized considering that a firefly is a possible solution of the problem and moves randomly in the sample space, attracting or being attracted to other fireflies.

A firefly tends to be attracted towards other fireflies with higher flash intensity. This algorithm is thus different from Particle Swarm Optimization (PSO) and can have two advantages: local attractions and automatic regrouping. As light intensity decreases with distance, the attraction among fireflies can be local or global, depending on the absorbing coefficient, and thus all local modes as well as global modes will be visited, according Yang (2013).

This algorithm is based on a physical formula of light intensity (I) that decreases with the increase in the square of the distance (r^2), said previously Fister et al, (2013).

The firefly algorithm also has some rules:

1) The attractiveness of a firefly is proportional to its brightness and it can decrease with distance;

2) Fireflies are unisex, so one firefly can be attracted to other fireflies regardless of their sex;

3) The attractiveness β is relative, it depend of the eyes of the beholder or judged as the other fireflies. Thus, it will vary with the distance between firefly i and firefly j.

Defining the attractiveness β of a firefly by Eq. (13).

$$\beta = \beta_0 e^{-\gamma r^2} \tag{13}$$

Where β_0 (max attractiveness when r = 0) and γ (coefficient of light absorption) are parameters before determined. For any given two fireflies x_i and x_j, the movement of a firefly (i) when it is attracted to another more attractive (brighter) firefly (j), is determined by Eq. (14).

$$xi^{t+1} = x^t + \beta o e^{-\gamma rij^2} (xj^t - xi^t) + \alpha t \epsilon i^t \tag{14}$$

In Eq. (14) the second term is due to the attraction and the third term is related the randomization with αt being the randomization parameter, lastly the $\in i^t$ is a vector of random numbers drawn from a Gaussian distribution or uniform distribution. In a multi objective approach, it is possible to solve an optimization problem following two ways: the first ways is combining all objectives into a single objective or another way is to extend the firefly algorithm to produce Pareto optimal front directly, by using the basic ideas of FA.

6.3 Multi Objective Particle swarm algorithm (MOPSO)

The particle swarm optimization algorithm is search method that belongs to the category of swarm intelligence techniques, where each particle establishes its trajectory combining their past experiences with the experiences of their neighbors. The changes of the position of the particles within the search space are based on the social psychological tendency of individuals to emulate the success of other individuals, Reddy and Kumar (2007).

The particle swarm optimization is an artificial intelligence technique that relies on social behavior pattern or social intelligence, which is observed in several bird species, fish schooling or ants’ colonies, according Castro (2013).

Inside a PSO algorithm, each solution to the optimization problem is a particle in the search space, which adjusts its position according to its own flying experience and the flying experience of other particle.

Accoding Suresh, et al., (2008), the PSO algorithm has high speed of convergence more than the other evolutionary search algorithm. In PSO, the particles remember the best position in the past and the best position ever attained by the particles. This property helps the particles to search the multi-dimensional space faster.

Due to its natural ability to converge faster, PSO is also use to solve multi objective optimization problems, in this case the algorithm is called MOPSO. This algorithm use the concept of Pareto dominance to determine the flight direction of a particle (convergence direction) and it maintains nondominated particles, previously found, in a global place that is later used by others particles to guide their flight. The parameters of these population’s members are adjusted with respect to their previous experience, and is derived from the particle as an individual and as a member of the entire of the population.

To perform the algorithm, each individual in the swarm is represented by four characteristics, where: X_i is the particle’s current position, V_i is the particle’s current velocity. XP_i is the particle’s personal best position and XG is the global best position. In the MOPSO algorithm the particle's position X_i have updated their positions according to their speed V_i in every interaction. After calculated the new position of the particle (i), it will evaluate the quality of X_i compared to the best position that the particle has visited or that some neighboring particle has visited P_i. If X_i is better than P_i, the best position is updated to the current position. The movement's velocity equation is shown in Eq. (15) where, $W(t)$ is the inertia weight, c_1 and c_2 are the acceleration constants, r_1 and r_2 are generated from the uniform distribution U(0,1).

$$V_i(t+1) = W(t)V_i(t) + c_1r_1[XP_i(t)-X_i(t)]+c_2r_2[XG(t)-X_i(t)] \tag{15}$$

The parameters that control the influence of convergence on the search process are: c_1 as the cognitive factor and c_2 as the social factor.

7. NUMERICAL SIMULATION

In order to analyze the results of the proposed problem, a system with five subsystem was chosen and two objective functions were prepared, where the Eq. (11) represent the unavailability (minimize) of system and Eq. (12) is the overall cost (minimize). The MTBF, MTTR, the design, maintenance and maintenance resource costs are shown in Tab. 1. The constraint considered in this problem is that 100% of the corrective maintenance resources should be used distributed through all subsystem.

In order to get the corrective maintenance cost, the value of the cumulative failure distribution in the time domain is necessary. Time T is set to 100 time units (the same unit as MTBF and MTTR). The impact variable I is set to 3.5.

The Multi objective genetic algorithm (NSGA II) parameters for this simulation are:

- Total number of generations: 1000;
- Population size: 50; probability of crossover: 60%; probability of mutation: 10%

The Multi objective firefly algorithm (MOFA) parameters for this simulation are:

- Total number of interactions: 1000; $\alpha = 0.4$; $\beta = 1$; $\gamma = 2$; Population size: 50.

The multi objective particle swarm algorithm (MOPSO) parameters for this simulation are:

- Total number of interactions: 1000; population size: 50; $c_1 = 1$ and $c_2 = 2$;

Observing the Pareto front in Fig. 2 for three methods used, where in green color is shown the NSGA II results, in blue the MOPSO and in red color is MOFA, it is possible to verify that the solutions are very similar. Analyzing the time of convergence of each method, it was found that NSGA II converged fast for a state like the final state with only 285 interactions, followed by MOPSO with 328 interactions and finally the most slow method is the MOFA, where only with 689 interactions was possible to note convergence to the final state. The convergence time analysis for the three methods is shown in Fig. 3.

The Tables 2 and 3 show some highlighted results from the simulations in terms of decision variables Rec_i and Y_i, and also in terms the respective optimizations results to the functions. It is important to note that the highlighted points in Fig. 2 corresponds to the points in the Tab. 2 and Tab. 3.

Table 1. Experimental data for optimization problem

SUBSYSTEM	MTBF $\frac{1}{\lambda}$	MTTR $\frac{1}{\mu}$	DESIGN COST (ci)	CORRECTIVE MAINTENANCE COST (cmi)	CORRECTIVE MAINTENANCE RESOURCE COST (creci)
1	500	50	50	60	7
2	550	35	55	40	3
3	600	40	55	45	4
4	750	30	40	30	2
5	500	30	60	50	5

Finally, after the simulations it was possible to verify the efficiency of these algorithms to solve multi objective optimizations problems, where several solutions were found with an acceptable cost and availability satisfactory. To use these methods, it is not necessary to apply a complex mathematics, resulting in an easy computational implementation. The result for percentage of the corrective maintenance resource (Rec_i) and for the number of redundant component in each subsystem (Y_i), are shown in Tab. 2, where it is possible to note that constraint of use 100% of the Rec_i , distributed through all subsystem was attended.

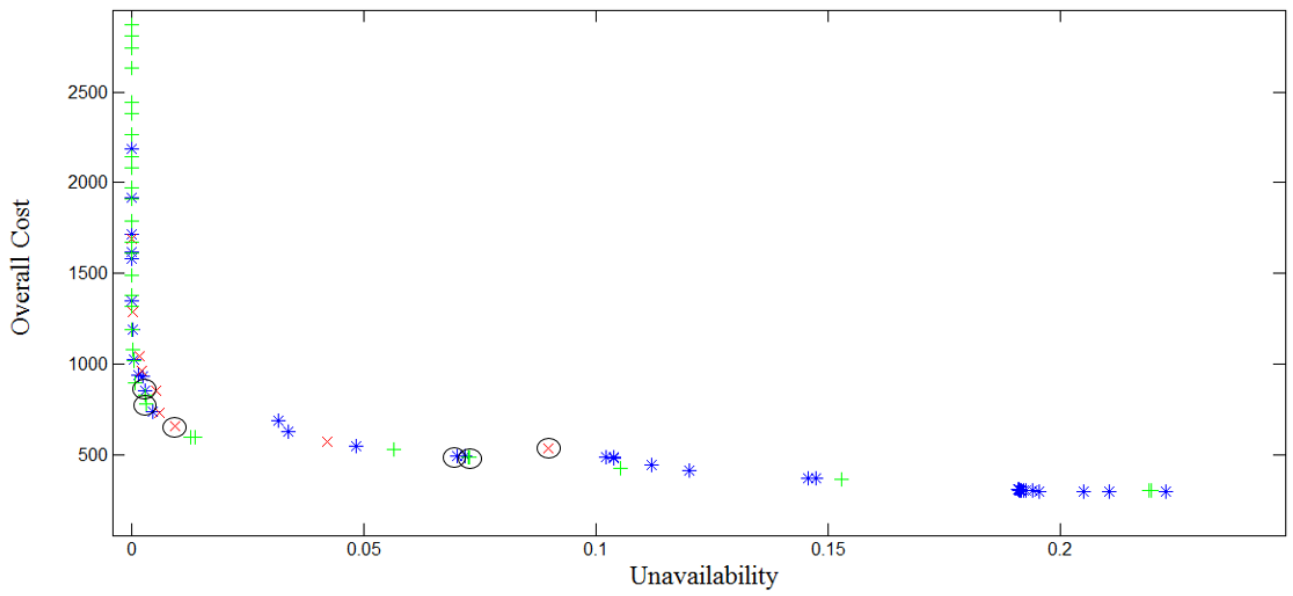

Figure 2. Pareto Front of methods

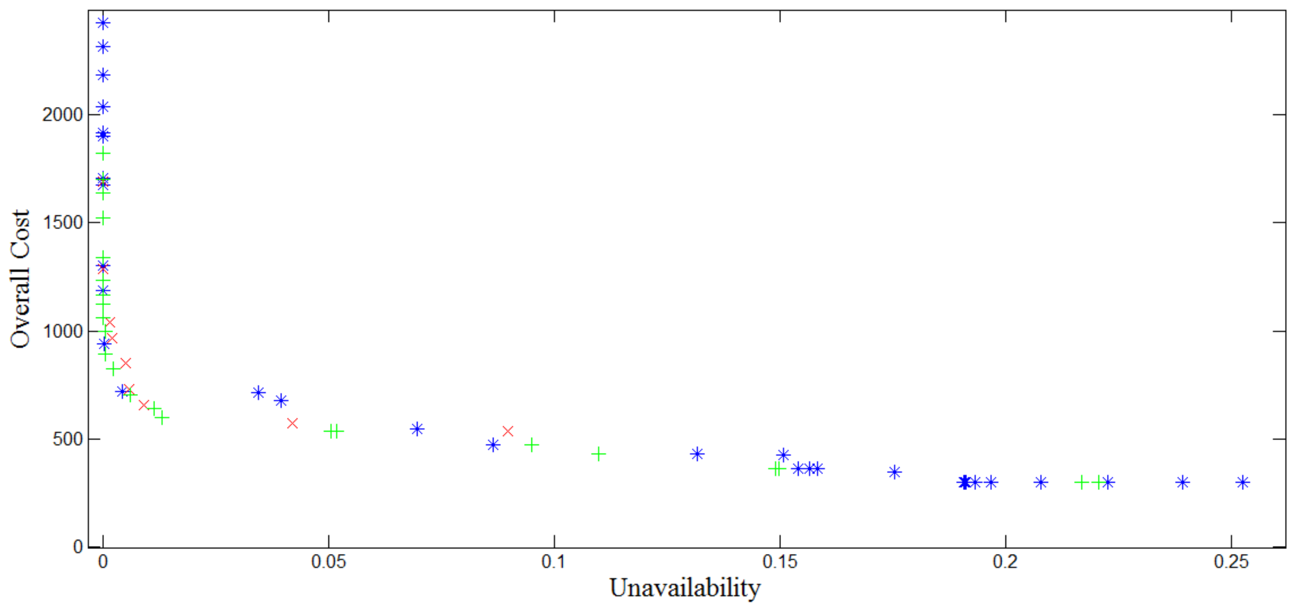

Figure 3. Pareto front for convergence time analysis of the methods

Table 2. Variable's results of optimization problem by NSGA II, MOFA, MOPSO

METHOD	REC1	Y1	REC2	Y2	REC3	Y3	REC4	Y4	REC5	Y5
GA 1	3,02%	2	2,86%	2	30,25%	2	53,66%	1	10,21%	1
GA 2	7,29%	3	4,5%	3	30,83%	3	47,01%	2	10,37%	2
MOFA 1	14,11%	2	2,13%	3	24,21%	2	53,34%	2	6,2%	2
MOFA 2	3%	1	21,25%	2	22,44%	2	30,11%	2	23,2%	2
PSO 1	21,4%	3	24,31%	4	20,68%	3	17,27%	2	16,35%	2
PSO 2	20,67%	2	22,05%	2	20,32%	2	18,7%	1	18,26%	1

Table 3. Functions results of optimization problem by NSGA II, MOFA, MOPSO

METHOD	AVAILABILITY – (f 1)	OVERALL COST – (f 2)
GA 1	99,7%	$782
GA 2	92,8%	$485
MOFA 1	99,1%	$659
MOFA 2	91,0%	$537
PSO 1	99,7%	$850
PSO 2	92,8%	$494

8. CONCLUSIONS

The aim of this paper is to compare the application of three algorithms to solve availability and overall cost optimization problem. In these terms all algorithms showed efficiently to solve complex problems like this. In our example several solutions were found with a final availability above 99% with the cost ranging among $650 and $850 as showed in Tab. 3, which is considered satisfactory, given the complexity of the data. It's possible to see the variable's result in Tab. 2 in each subsystem, where 100% of corrective maintenance resources are applied.

The three methods were able to find the expected parameters. Highlighting that NSGAII converges faster than MOPSO and MOFA, but it is important also remember that NSGAII is a package ready in Matlab software, and the two others algorithms were adjusted. Therefore, an improvement of MOFA and MOPSO can be carried out, by a better definition of each method parameters. It's hard to indicate the best method because it also depends of the optimization problem, but in this simulation the NSGAII had the best results, followed for MOPSO and MOFA.

9. REFERENCES

Castro, H.F. and Cavalca, K.L., 2005. "Maintenance resource optimization applied to a manufaturing system". In Reliability engineering & system safety, v. 20, p. 01-08.

Castro, H.F., 2003. Reliability and availability optimization applied to redundant systems. Master Thesis, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas SP.

Chang, C. C., 2014. "Optimum preventive maintenance policies for systems subject to random working times, replacement, and minimal repair" In Computers & Industrial Engineering, v. 67, p. 185-194.

Coello, C.A.C., Veldhuizen, D.A.V. & Lamont, G.B., 2002. Evolutionary algorithms for solving multiobjective problems, Kluwer Academic, New York.

Cortes, J. M. R., 2003. A genetic algorithm for solving the problem of location of economic activities, Universidade Estadual Norte Fluminense – UENF, Campos dos Goytacazes – RJ.

Fister, I.,Fister Jr, I. and Yang, X.S., 2013. "A comprehensive review of firefly algorithms". In Swarm and Evolutionary Computation, v.13, p. 34-46.

Lins, I. D., Droguett, E. L., 2009. "Multiobjective optimization of availability and cost in repairable systems design via genetic algorithms and discrete event simulation". In Pesquisa operacional, v. 29, p.43-66.

Lobato, F.S., 2008. Multiobjective optimization to engineering system design, Ph.D thesis, Faculdade de engenharia mecânica, Universidade Federal de Uberlândia, Uberlândia.

Kalyanmoy D., Pratap, A., Agarwal, S. and Meyarivan, T., 2002. "A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II". In IEEE Transactions on Evolutionary Computation, v. 6(2), p.182 – 197.

Michalewicz, Z., 1996. "Genetic algorithms + data structures = evolution programs". In 3rd. Ed. New York: Springer, p. 387.

Mukta, M.S.H., Islam, T.M.R. and Hasnayen, S.M., 2012. "Multiobjective Optimization using Genetic Algorithm". In International Journal of emergin trends e technology in computer science, v.1 , p. 255-260.

Reddy, M. J., Kumar, D. N., 2007. "Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation". In Hydrological Processes in Wiley InterScience, Published online, 10 January.

Surech, S., Sujit, P.B., Rao, A.K., 2007. "Particle swarm optimization approach for multi-objective composite box-beam design". In Composite structures, v. 81, p. 598-605.

Yang, XS., 2013. "Multiobjective firefly algorithm for continuos optmization". In Reliability Engineering & System Safety, v.29, Ed. 2, p.175-184.

10. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.