Development of hybrid algorithm based on simulated annealing and genetic algorithm to reliability redundancy optimization

Bruno Dalanezi Mori, Hélio Fiori de Castro and Katia Lucchesi Cavalca
Department of Mechanical Design, School of Mechanical Engineering, State University of Campinas, Campinas, São Paulo, Brazil

Abstract

Purpose – The purpose of this paper is to present an application of the simulated annealing algorithm to the redundant system reliability optimization. Its main aim is to analyze and compare this optimization method performance with those of similar application.

Design/methodology/approach – The methods that were used to compare results are the genetic algorithm, the Lagrange Multipliers, and the evolution strategy. A hybrid algorithm composed by simulated annealing and genetic algorithm was developed in order to achieve the general applicability of the methods. The hybrid algorithm also tries to exploit the positive aspects of each method.

Findings – The results presented by the simulated annealing and the hybrid algorithm are significant, and validate the methods as a robust tool for parameter optimization in mechanical projects development.

Originality/value – The main objective is to propose a method for redundancy optimization in mechanical systems, which are not as large as electric and electronic systems, but involves high costs associated to redundancy and requires a high level of safety standards like: automotive and aerospace systems.

Keywords Reliability management, Optimization techniques

Paper type Research paper

Introduction

It is increasingly necessary to design reliable systems as there is a great demand for products that offer quality and safety. Another way of improving the reliability of a system is to use redundant components or redundant sub-systems. In this case, redundant components should take place of components that fail.

However, an increase in the number of components and sub-systems consequently results in project costs, mass and volume of the system, and other design parameters increasing. Hence, it is necessary to use optimization techniques in order to obtain an optimum system within the design constrains.

This paper deals with reliability optimization using redundant components connected in parallel (Yang *et al.*, 1999; Painton and Campbell, 1995). The availability of the redundant system was also optimized (Castro and Cavalca, 2003; Elegbede and Adjallah, 2003) focusing maintenance.

A hybrid algorithm is developed using simulated annealing and genetic algorithm sequentially. This procedure is applied to reliability optimization of redundant

The authors would like to thank Fapesp, CNPq and CAPES for supporting this work.

International Journal of Quality & Reliability Management
Vol. 24 No. 9, 2007
pp. 972-987

0265-671X
DOI 10.1108/02656710710826225

systems. Wang and Zheng (2001) have also developed a hybrid algorithm based in genetic algorithm and simulated annealing to solve job-shop scheduling problems, which is an optimization problem with a different point of view with respect to the problem considered here.

Notation

- R_s The reliability of the system.
- C_s The total cost of the system.
- W_s The weight of the system.
- R_i The reliability of the ith component.
- C_i The cost of the ith component.
- w_i The weight of the ith component.
- n The number of components in the ith sub-system.
- y_i The number of components in the ith sub-system or in the ith stage.

Formulation of the problems

In this study, the components redundancy model considers a system in series, with parallel and active redundant components.

Each item arranged in a series can be considered a subsystem, where the maximum number of subsystems is represented by the variable n. The redundancy number varies in each stage. And it is determined by optimization processes, considering parameters of reliability, cost, weight, and other project parameters of the system (Figure 1).

The reliability of this system can be obtained by using Equation 1, where R_i is the reliability of the components of system i, and y_i is the number of components in the subsystem (or stage) i.

$$R_S = \prod_{i=1}^{n} \left[1 - (1 - R_i)^{y_i}\right] \tag{1}$$

The total cost is given by Equation 2, where c_i the individual costs of the components and y_i are the number of components in each stage.

$$C_s = \sum_{i=1}^{n} c_i . y_i \tag{2}$$

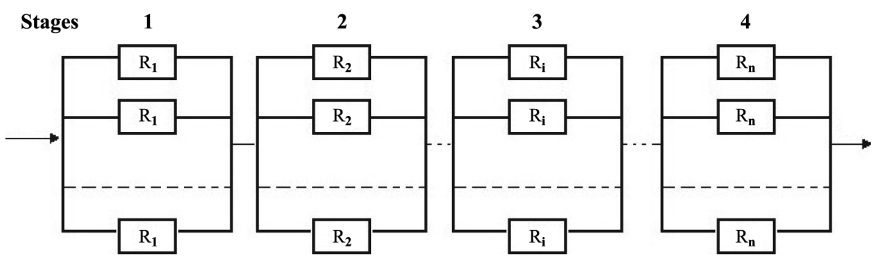


Figure 1. Redundant systems scheme

Similarly the weight of the system is given by Equation 3, where w_i are the individual weights of the components:

$$W_s = \sum_{i=1}^{n} w_i . y_i \tag{3}$$

In this work, three optimization problems for redundant systems are developed using the reliability as an objective function, or a constraint parameter. Some current problems are proposed (Table I) of which the main objective is to validate the optimization methods. The target is to compare the results obtained here with those obtained by traditional methods, as Lagrange Multipliers (Castro and Cavalca, 2002; Banerjee and Rajamani, 1973), that does not enable quite complex systems solution. The aim here is to obtain the number of components in each subsystem, i.e. the variable y_i.

Methods

Simulated annealing (SA)

This algorithm explores the analogy between the search for a minimum in an optimization process and the gradual cooling of a metal into a minimum energy crystalline structure. The search for a minimum requires the definition of boundary constraints of the problem. It also requires a cost evaluation method of a particular solution, which can be used in all optimization problems. The algorithm carries out an iterative search for a better solution in the neighborhood of the current solution. A new solution can become the starting point for successive steps trying to find better solutions, or become a path to avoid a local minimum. This procedure must be carried on until reaches the stop criteria. When there are no local solutions that improve the quality of the solution, the algorithm stops at the local solution. The local optimum trap makes the local search a heuristic restriction for many combinatory optimization problems. This is because it strongly depends on the initial point. A desirable property of any algorithm is its ability to obtain the global optimum independently of the starting point.

So, the paradigm of the simulated annealing offers us an escape from the local optimum by analyzing the boundary of the current solution, and by accepting solutions that worsen the current solution with a certain probability. This is aimed at finding a better way to obtain the global optimum.

Around 1953, Metropolis *et al.* (1953) presented a simple algorithm to simulate the evolution of a solid from its liquid state to its thermal equilibrium.

The physical process of annealing can be modeled successfully using simulation methods of condensed matter physics. The temperature (control parameter) is slowly cooled after a number of searches in the neighborhood of the current state. For this reason, some analogies are drawn between a particle physics system and a combinatory optimization problem. The solutions in an optimization problem are equivalent to states in a physical system. The cost of a solution is equivalent to the energy of a state. Choosing a solution in the neighborhood of an optimization problem is equivalent to the perturbation of a physical state. The global optimum of a combinatory problem is equivalent to the fundamental state of a system of particles. A local optimum of a combinatory problem is equivalent to a meta-stable structure in a system of particles. With an iterative implementation, it is possible to obtain an

Problem 1	Problem 2	Problem 3
Cost minimization for an acceptable minimum reliability This problem uses Equation 2 as an objective function, and Equation 1 as a restrictive function	Reliability maximization for an acceptable maximum cost This problem uses Equation 1 as an objective function, and Equation 2 as a restrictive function	Reliability maximization for an acceptable maximum cost and weight This problem uses Equation 1 as an objective function, and Equation 2 and 3 as a restrictive function
$\min U = \sum_{i=1}^{n} c_i.\mathrm{y}_t$ subject to, $\psi = \prod_{i=1}^{n} \left[1-(1-R_i)^{\mathrm{y_i}}\right] \geq R_{\min}$	$\max U = \prod_{i=1}^{n} \left[1-(1-R_i)^{\mathrm{y_i}}\right]$ subject to, $\psi = \sum_{i=1}^{n} c_i.\mathrm{y}_t \leq C_{\max}$	$\max U = \prod_{i=1}^{n} \left[1-(1-R_i)^{\mathrm{y_i}}\right]$ subject to, $\psi_1 = \sum_{i=1}^{n} c_i.\mathrm{y_i} \leq C_{\max}$ $\psi_2 = \sum_{i=1}^{n} w_i.\mathrm{y_i} \leq W_{\max}$

Table I.
Proposed problems

algorithm for combinatory optimization problems (Metropolis *et al.*, 1953; Laarhoven and Aarts, 1989).

Genetic algorithm (GA)

The genetic algorithm is a search strategy that employs random choice to guide a highly exploitative search, striking a balance between exploration of the feasible domain and exploitation of "good" solutions, see Holland (1975). This strategy is analogous to biological evolution. From a biological perspective, it is conjectured that the ability of an organism structure to survive in an environment ("fitness"), is determined by its DNA. An offspring, which is a combination of both parents DNA, inherits traits from both parents and other traits that the parents may not have, due to recombination. These traits may increase offspring fitness, yielding a higher probability of surviving more frequently and passing the traits on to the next generation. Over time, the average fitness of the population improves.

In GA terms, the DNA is currently represented by a binary string where each position in the string is a finite set of values. It makes possible to work with integer and real numbers together in the same optimization process, applying a decoding transform of this variable in binary numbers.

Members of a population are subjected to operators in order to create offspring. Commonly used operators include selection, reproduction, crossover, and mutation. The selection operator compares the individuals of the population. The individuals that are closest to the optimum point have a major probability to produce a new offspring by reproduction, crossover and mutation (Holland, 1975; Goldberg, 1989).

Hybrid algorithm (HA)

The hybrid method (HA) has generated promising results in many applications, mainly in highly complex problems. While the SA uses the local movement to generate a new solution only by modifying the old one, the GA generates solutions by combining two different solutions. However, these facts do not necessarily make the algorithm better or worse than others.

It is important to note that the GA and the SA are also bounded by the assumption that good solutions are probably found "near" the best known solutions, rather than chosen from a whole set of solutions. Otherwise these procedures do not carry out with a better search than the random search.

The hybrid method is developed by arranging the algorithms in series. First the genetic algorithm is used, followed by the Simulated Annealing. The HA parameters are set up so that the stop criteria of the GA interrupts it before the complete convergence to a global optimal value, once the GA only generates the first input to the SA. In order to comprehend the parameters influence, a sensitivity analysis of GA/SA parameters is carried out simultaneously, indicating the ideal parameters and also defining the stop criteria of both algorithms. Consequently, the simulated annealing works with a pre-checked starting point in a wide universe of solutions.

The strategy is the GA makes the least part of the processing, and the SA works in the largest processing, limiting the GA number of generations considerably.

The flow chart of the hybrid algorithm is shown in Figure 2. This algorithm was used in the three optimization problems proposed.

It is important to highlight that genetic algorithm and simulated annealing parameters have influence in the convergence of the optimization problem. So, in the

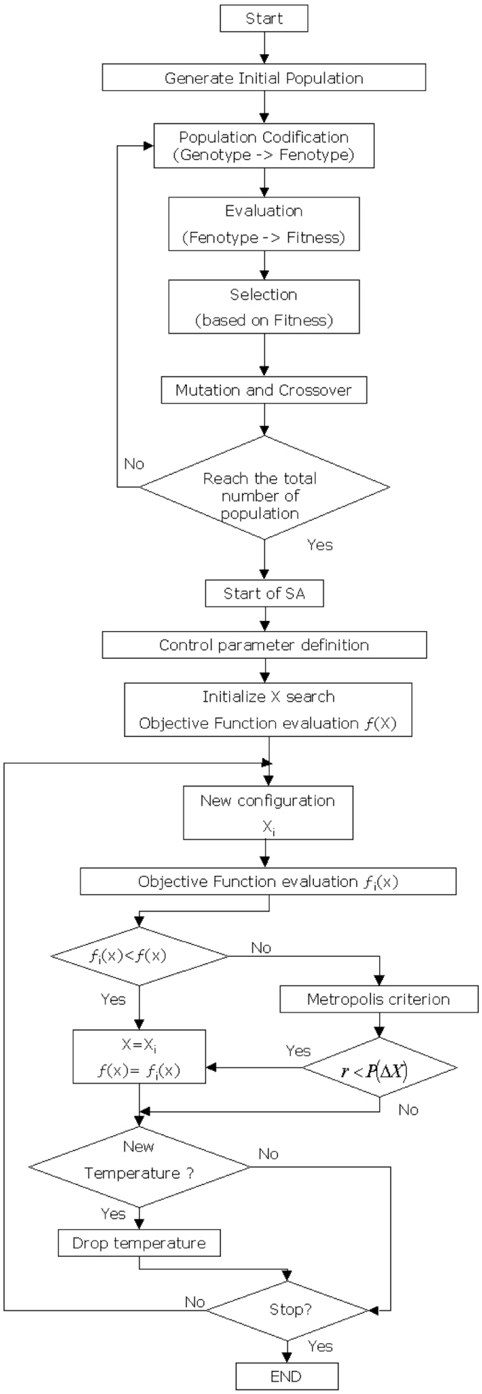


Figure 2.
Flow chart of the hybrid algorithm

next section, parameters influence and the more relevant parameters interaction are analyzed.

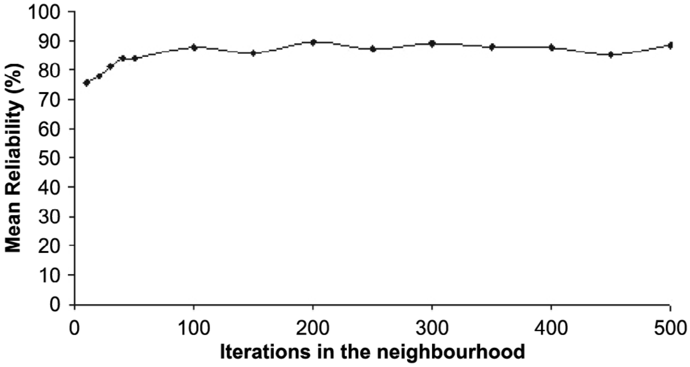
Parameter analysis

Simulated annealing

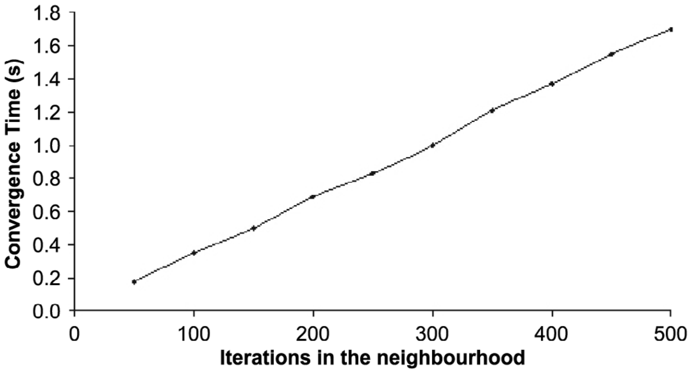
After implementing the simulated annealing, each parameter of the algorithm was evaluated: the cooling factor, the number of iterations in the neighborhood, and the number of temperature reduction steps.

These parameters strongly influence the final solution and the number of iterations to convergence. To evaluate such influence, tests are carried out changing each parameter separately. To obtain the mean processing time and the mean objective function value, ten samples of each method are tested, in a non-sequential order.

Number of iterations in the neighborhood. It is observed that a large number of iterations for the same temperature (Figure 3a) do not necessarily result in a better final solution. This is partly because many worse solutions are acceptable at high temperatures, and this could mislead the algorithm. As the number of iterations increases, the convergence time linearly increases too (Figure 3b).



(a) Mean reliability



(b) Convergence time and the total number of iterations in the neighbourhood

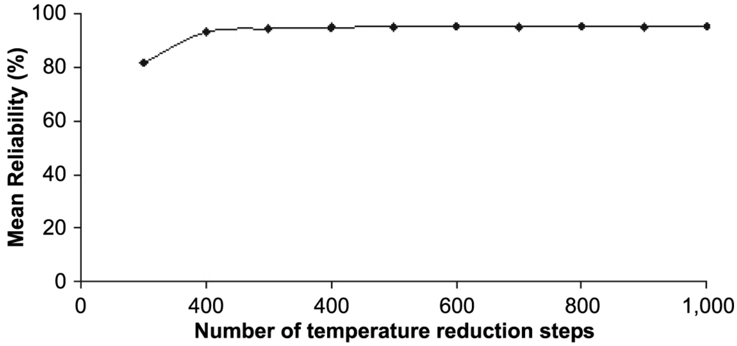
Figure 3.
The relationship between the (a) mean reliability or (b) convergence time and the total number of iterations in the neighborhood

The iteration in the neighborhood is related to a number of possible solutions y in a given temperature. As higher is this number, higher is the convergence time, but if this number is not enough high the convergence may not be reached.

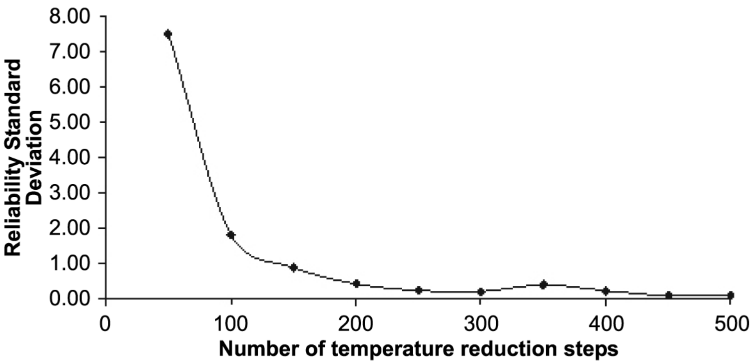
Temperature reduction steps. The mean reliability presents a rapid initial growth, and then it slowly converge to a fixed value as the number of steps increases (Figure 4a). But significant decreasing in the reliability variance confirms the convergence as the number of cooling steps increases (Figure 4b).

A higher temperature provides a higher worse-solution acceptance during the optimization process. As the temperature diminishes during the optimization, the probability of a worse solution to be accepted decreases.

Cooling factor (α). Figure 5 shows the influence of the cooling factor in the convergence. The value of α affects the solution quality. The cooling factor α generates low mean reliabilities when its value is near 1, so the algorithm demands a large number of temperature reduction steps. But convergence is observed for the three values of the cooling factor. From previous observations, it is known that if the number of cooling steps increases, the mean reliability value rises. Since the cooling is geometric, smaller cooling factors result in a sharp decreasing in the temperature. Despite the fast temperature drop, a convergence is observed for the problems analyzed. It is important to point out that, for each problem, there is a relationship between the number of steps and the cooling factor. Therefore it is not advisable to use



(a) Mean value



(b) Standard deviation

Figure 4.
Relationship between the reliability and the number of temperature reduction steps

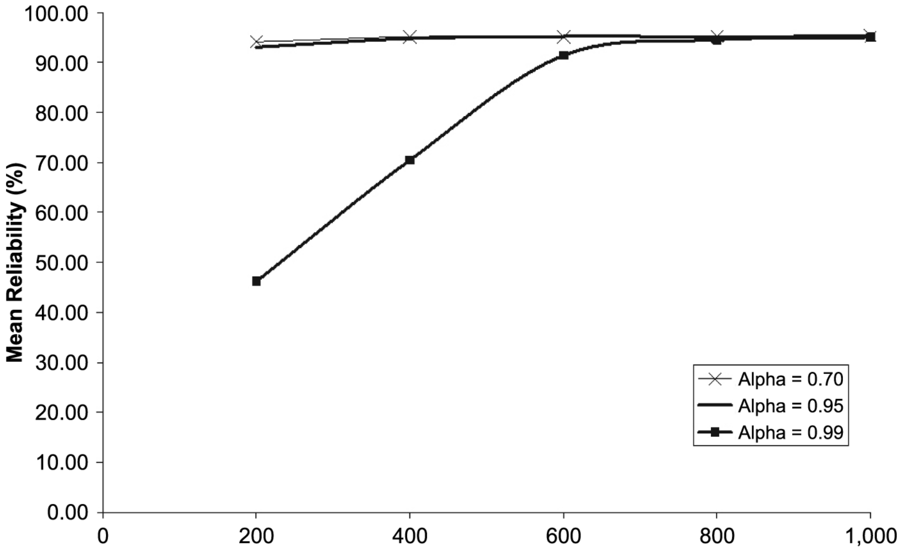


Figure 5.
Relationship between the mean reliability and the cooling factor

small cooling factors (the lower the cooling factor is the higher is the temperature drop), since the algorithm can move toward the local optimum in this situation.

Genetic algorithm

For the genetic algorithm, the main aspects of the parameters analyzes carried out by Castro and Cavalca (2003) are highlighted. The population size and the total number of generations should vary according to the complexity of the system to be optimized. The population growth exerts a strong influence on the convergence time. Thus, it is important to vary the total number of generations. It is advisable to have a high probability of mutation and a low probability of crossover for the examples analyzed. The genetic algorithm Parameter can be also related to the optimization problem parameter. The population size and the number of generation are related to the estimation of the different combinations the problem variable (number of redundancies y) can present and the probabilities of crossover and mutations have influence on the new solutions searching that the number of redundancies y is submitted.

Hybrid algorithm

It is essential to understand how the parameters of both methods influence the convergence and the processing time of the new algorithm. Hence, a variance analysis is carried out, based on a factorial planning to determine the parameter influence on the convergence of the hybrid algorithm. The factorial planning is capable of detecting the degree of influence of each parameter and their iterations. The following parameters are considered due to relevance:

- Size of the population – related to the possibilities of number of redundancies combinations.
- Total number of generations – related to the generation of new solutions.
- Cooling factor – define the convergence velocity to the fundamental state (mean reliability value).

- Temperature reduction steps – related to the partial state (reliability value) of the system.

Table II gives the maximum and minimum intervals of the parameters to be analyzed.

The parameters of crossover, mutation and inversion are fixed. For the tests, a 15-stage system with a very high degree of complexity was adopted. The combinations of the maximum and minimum values of the parameters were tested three times each. The results are presented in Table III.

An effect analysis and evaluation of the F0 Ratio shows that the cooling steps (C) presents the strongest influence on the convergence of the method. The number of generations (B) and the cooling Factor (D) are almost equivalent and exert a considerably weak influence. Then the population size (A) and the iterations AB and BC, take place with a weak but not negligible influence on the method. The other iterations influence are not so relevant to the results. After the analysis with the hybrid algorithm, the number of cooling steps is selected to be varied. An increase in the number of cooling steps increase the mean reliability value. During the implementation of the hybrid algorithm, it can be noticed that the GA carried out only few steps of the operation, leaving the most of processing to the SA.

A	Population size	Min.	10
		Max.	50
B	Number of generations	Min.	10
		Max.	100
C	Temp. reduction steps	Min.	100
		Max.	1,000
D	Cooling factor	Min.	0.7
		Max.	0.99

Table II.
Intervals of the parameters analyzed

						R			
				n1		n2		n3	
A	B	C	D	Time	Reliability	Time	Reliability	Time	Reliability
10	10	100	0.7	0.33	93.72	0.33	92.24	0.33	92.80
50	10	100	0.7	0.44	93.28	0.39	93.53	0.44	92.50
10	100	100	0.7	0.39	93.35	0.43	93.72	0.38	93.36
50	100	100	0.7	1.10	93.19	1.10	93.12	1.15	93.96
10	10	1,000	0.7	3.24	94.47	3.24	94.33	3.24	94.20
50	10	1,000	0.7	3.35	94.40	3.29	94.49	3.36	94.47
10	100	1,000	0.7	3.29	94.47	3.30	94.33	3.30	94.08
50	100	1,000	0.7	4.07	94.47	4.07	94.47	4.01	94.52
10	10	100	0.99	0.33	92.66	0.33	91.89	0.33	93.09
50	10	100	0.99	0.38	93.21	0.55	93.35	0.44	93.27
10	100	100	0.99	0.38	93.00	0.39	93.94	0.39	93.29
50	100	100	0.99	1.15	93.51	1.15	92.75	1.15	92.35
10	10	1,000	0.99	3.35	94.08	3.19	94.49	3.19	94.47
50	10	1,000	0.99	3.29	94.47	3.30	94.38	3.29	94.37
10	100	1,000	0.99	3.24	94.29	3.24	94.13	3.24	94.47
50	100	1,000	0.99	4.01	94.40	4.01	94.33	4.01	94.01

Table III.
Combinations of the parameters

The interaction between the number of cooling steps and the total number of generations (BC iteration) shows the importance of correctly relating these two parameters (the main stop method criteria). Hence, an increase in the total number of generations will slightly decrease the number of cooling steps towards the convergence point, and vice-versa. Only a very substantial increase in the total number of generations results in a slight decrease in the number of cooling steps. Since the GA carries out only a fraction of the operations, the total number of generations has a relatively weak influence on the convergence. And so it is not necessary to perform a large number of operations. The size of the population should also be limited because it has a weak influence on the convergence. But it presents a substantial increase in the computational time (number of iterations).

The cooling factor is also an important parameter of the hybrid algorithm convergence. For the hybrid algorithm (compared to the Simulated Annealing), it is possible to use smaller cooling factors because the hybrid method takes advantage of the GA initial solution.

The parameter analysis of the three methods revealed the behavior of each parameter and their interactions, what allows the comparison of the methods.

The results shown in Table IV were obtained.

Comparison of the methods

Numerical analysis are carried out to compare the efficiency of genetic algorithm, simulated annealing and hybrid algorithm with other optimization methods: Lagrange Multipliers and evolution strategy. The genetic algorithm and the last two methods were tested by Castro and Cavalca (2002). The Lagrange Multipliers (Banerjee and Rajamani, 1973) was applied to the reliability equation of the redundant system, depending on each component reliability, with maximum costs and weight constraints (problems 2 and 3). The algorithm was also applied to costs model with minimum reliability constraints (problem 1).

A different system was chosen for each one of the problems proposed, with distinct levels of complexity. The comparison of the methods was performed considering the

Table IV. Results of the factorial planning

Combination	Effect	F0
A	0.07	0.49
B	0.14	1.75
C	1.23	1.29
D	− 0.14	1.67
AB	− 0.19	0.10
AC	0.00	0.00
AD	− 0.02	0.00
BC	− 0.18	0.09
BC	− 0.08	0.01
CD	0.06	0.01
ABC	0.17	0.08
ABD	− 0.14	0.05
ACB	− 0.04	0.00
BCD	0.02	0.00
ABCD	0.11	0.03

processing time, number of objective function evaluations and the quality of the results. To obtain the mean convergence time, the example of each method is tested ten times each. And none of the tests is performed sequentially. System 1, shown in the Table V, is used to solve Problem 1 (cost minimization for any given minimum reliability), for a minimum reliability value of 99.9 per cent.

System 2, used to solve problem 2 (reliability maximization for a maximum cost allowed) is given in Table VI.

For problem 3 (reliability maximization for the maximum cost and weight allowed), the system is highly complex, with more stages and two constrains, allowing a better comparison of the methods. The system is presented in Table VII.

Table VIII shows the processing time and the number of objective function evaluations for each problem. For problem 3, the evolution strategy reached the reliability of 91.77 per cent in 76 seconds (or 1,000,000 evaluations), which does not mean good results. The Lagrange Multiplier method simply does not converge, since

Stage	Cost	Reliability (%)	Optimum system
1	3.00	95	3
2	2.50	92	3
3	2.00	90	4
4	1.50	85	5
Constrain		99.9	
Optimal solution	32.00	99.92	

Table V. System for Problem 1

Stage	Cost	Reliability (%)	Optimum system
1	4.00	90	2
2	3.00	85	3
3	2.00	80	4
4	1.0	70	5
Constrain	30.00		
Optimal solution	30.00	98.27	

Table VI. System for Problem 2

Stage	Cost	Weight	Reliability (%)	Optimum system
1	5	8	90	2
2	4	9	70	4
3	9	6	65	5
4	7	7	80	3
5	7	8	85	3
6	8	9	75	3
7	9	7	90	2
8	8	8	85	3
9	6	9	80	3
Constrain	215	220		
Optimal solution	197	219	93.05	

Table VII. System for Problem 3

Table VIII.
Convergence time of the methods for each problem

		Lagrange multiplier	Genetic algorithm	Evolution strategy	Simulated annealing	Hybrid algorithm
Problem 1	Convergence time	0	3.02 s	1.04 s	0.88 s	0.50 s
	Number of evaluations	X	2,500	13,684	50,000	4,000
Problem 2	Convergence time	0	6.32 s	1.39 s	0.88 s	0.55 s
	Number of evaluations	X	5,000	18,289	50,000	4,000
Problem 3	Convergence time	ERROR	23.89 s	> 76 s	10.44 s	3.20 s
	Number of evaluations	ERROR	105,000	> 1,000,000	250,000	6,500

this case presents an internal numerical incompatibility for the non-linear system. The GA, SA and HA are able to reach the optimum solution.

For the methods analyzed, the hybrid algorithm presents the lowest convergence time and number of evaluations, except for the Lagrange Multipliers. Since the Lagrange Multiplier method uses differential calculus, it presents an almost real-time solution whenever possible. In these cases, the main aim is to compare the heuristic and random methods: Simulated Annealing, genetic algorithm, evolution strategy, and hybrid algorithm.

The evolution strategy presents good results for the case-studies of low complexity. But for complex problems, the processing time increases too much, and the method does not converge.

Convergence can be reached for the global optimum of the system for all examples tested using the genetic algorithm. However, the convergence time is relatively high.

The simulated annealing presented the second lowest convergence time among the methods. However, in more simple problems with fewer sub-systems (especially in the first two), the convergence time is not significantly different from the hybrid algorithm , except by the third problem that presents a considerable difference in the convergence time. These results motivate the analysis of a new complex problem with 15 stages.

The good performance of the hybrid algorithm in previous problems, also encourages its comparison with genetic algorithm and Simulated Annealing, in a more complex problem with 15 stages. The evolution strategy and Lagrange Multiplier method do not take part due to their lowest performance in the previous analysis.

The 15-stage system is represented in Table IX. Ravi *et al.* (1997) used this system to implement a new version of the simulated annealing which works with non-equilibrium stages.

It can be noticed that the genetic algorithm and the hybrid algorithm reaches an optimum solution, while the simulated annealing does not converge. Increasing the number of parameters of the simulated annealing cannot improve the solution of the method. In problems where the complexity does not increase linearly with the number of stages, the simulated annealing faces difficulties as in the 15-stage system. In the

Stage	Cost	Weight	Reliability (%)	Optimum system
1	5	8	90	3
2	4	9	75	4
3	9	6	65	6
4	7	7	80	4
5	7	8	85	3
6	5	8	93	2
7	6	9	78	4
8	9	6	66	5
9	4	7	78	4
10	5	8	91	2
11	6	9	79	3
12	7	12	77	4
13	9	6	67	5
14	8	5	79	4
15	6	7	67	5
Constrain	400	414		
Optimal solution	392	414	94.56	

Table IX.
System with 15 stages

work of Ravi *et al.* (1997), the conventional simulated annealing does not reach the optimum solution either. The efficiency of the genetic algorithm can be considered good even though it requires considerable computational time, because it has always presented consistent results regardless of the degree of complexity.

The hybrid algorithm presents the best results once again, obtaining an optimum solution in all problems analyzed, and demanding considerably lower computational time (or number of objective function evaluations) (Table X).

The main difference between the hybrid method proposed here and that proposed by Wang and Zheng (2001) is the application of the Simulated Annealing. Wang and Zheng combines genetic algorithm with simulated annealing to develop a hybrid framework, in which genetic algorithm was introduced to present parallel search architecture, and simulated annealing was introduced to increase escaping probability from local optima at high temperatures and perform a "fine" neighbor search at low temperatures. In the present paper, the method uses the simulated annealing to improve the partial solution generated by of genetic algorithm. So, the genetic algorithm is used to provide a start condition to the Simulated Annealing, allowing the genetic algorithm to use a lower number of generations and population size, as well as, reducing the number of iterations in the neighborhood and temperature reduction steps, decreasing the processing time. Wang and Zheng hybrid algorithm, when compared to genetic algorithm and Simulated Annealing, has shown a better convergence, but the time processing was higher. The hybrid method presented here has also shown a better convergence, and additionally a shorter processing time than genetic algorithm and Simulated Annealing.

Conclusion

This paper elaborates on the implementation and application of the simulated annealing in design parameters optimization of redundant systems.

The comparative tests of the methods revealed a better performance of the hybrid algorithm for all systems analyzed. The method is always capable of obtaining the optimum solution, at a lower computational time than the other methods. As the systems became more complex, most methods are not useful to reach the global optimum solution. Some aspects are highlighted: the simulated annealing always had the second best efficiency, but does not converge in the most complex problem studied (15 steps). Simultaneously, the genetic algorithm presents a consistent efficiency regardless of the degree of complexity, always obtaining good solutions even though it requires considerable computational time. Both algorithms can be seen as algorithms that do not bear the disadvantages of the local search algorithms (the possibility to converge to a local optimum). Unlike the local search algorithms, SA and GA have a mechanism to escape from the local optimum problem. The simulated annealing reveals its quality as a "quick starter". This mean that it is able to obtain good solutions in a short time, but it is not able to improve this solution significantly during

Table X. Reliability, processing time and number of objective function evaluations for the 15-stage system

Method	Genetic algorithm	Simulated annealing	Hybrid algorithm
Reliability reached (%)	94.56	94.33	94.56
Evaluations	700,000	2,000,000	303,000
Time	304.3 s	330.1 s	20.7 s

the operation. On the other hand, the genetic algorithm characterizes a "slow starter". However, it is capable of improving the solutions significantly, but it spends more time.

The hybrid algorithm is able to deal with the general applicability of the methods and to exploit the positive aspects of both. The proposed hybrid strategy is validated, allowing the achievement of better results than GA and SA methods.

References

Banerjee, S. and Rajamani, K. (1973), "Optimization of system reliability using a parametric approach", *IEEE Transactions on Reliability*, Vol. R-22, pp. 35-9.

Castro, H.F. and Cavalca, K.L. (2002), "Reliability optimization of redundant system", *SAE Technical Papers*, Vol. 1, pp. 1-9.

Castro, H.F. and Cavalca, K.L. (2003), "Availability optimization with genetic algorithm", *International Journal of Quality and Reliability Management*, Vol. 20 No. 7, pp. 847-63.

Elegbede, C. and Adjallah, K. (2003), "Availability allocation to repairable systems with genetic algorithms: a multi-objective formulation", *Reliability Engineering & System Safety*, Vol. 82, pp. 319-30.

Goldberg, D.E. (1989), *Genetic Algorithm in Search Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Holland, J.H. (1975), *Adaptation in Natural and Artificial System: An Introduction with Application to Biology, Control and Artificial Intelligence*, University of Michigan Press, Ann Arbor, MI.

Laarhoven, P.M. and Aarts, E. (1989), *Simulated Annealing: Theory and Applications*, Kluwer, Dordrecht.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E. (1953), "Equation of state calculations by fast computing machines", *J. of Chem. Physics*, Vol. 21.

Painton, L. and Campbell, J. (1995), "Genetic algorithm in optimization of system reliability", *IEEE Transactions on Reliability*, Vol. 44 No. 2, pp. 172-8.

Ravi, V., Murty, B.S.N. and Reddy, P.J. (1997), "Nonequilibrium simulated annealing-algorithm applied to reliability optimization of complex systems", *IEEE Transaction on Reliability*, Vol. 46 No. 2, pp. 233-9.

Wang, L. and Zheng, D.Z. (2001), "An effective hybrid optimization strategy for job-shop scheduling problems", *Computers & Operations Research*, Vol. 28 No. 6, pp. 585-96.

Yang, J., Hwang, M., Sung, T. and Jin, Y. (1999), "Application of genetic algorithm for reliability allocation in nuclear power plants", *Reliability Engineering & System Safety*, Vol. 65, pp. 229-38.

Further reading

Coit, D.W. and Smith, A.E. (1996), "Reliability optimization of series-parallel system using a genetic algorithm", *IEEE Transactions on Reliability*, Vol. 45 No. 2, pp. 254-66.

About the authors

Bruno Dalanezi Mori is a Mechanical Engineer, Hélio Fiori de Castro is PhD., MSc and Mechanical Engineering, and Katia Lucchesi Cavalca is Associated Professor, all at State University of Campinas, Campinas, Sao Paulo, Brazil. Katia Lucchesi Cavalca is the corresponding author and can be contacted at: katia@fem.unicamp.br