Availability Optimization for Series/ Parallel Systems using Evolutionary Algorithm

Alain Ratle
Daoud Ait-Kadi
Mohamed-Larbi Rebaiaia

December 2013

CIRRELT-2013-78

Availability Optimization for Series/Parallel Systems using Evolutionary Algorithm

Alain Ratle†, Daoud Ait-Kadi, Mohamed-Larbi Rebaiaia*

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mechanical Engineering, 1065, avenue de la Médecine, Université Laval, Québec, Canada G1V 0A6

Abstract. One of the best solutions provided to improve the availability of industrial and communication systems is to add redundant elements. Usually this task tries to achieve a performance level with minimal cost. The problem of total investment cost minimization, subject to availability constraints, is a pure redundancy optimization problem which is known to be intractable and for which no deterministic algorithm is expected to succeed. The difficulties come from the constraints and variable domains which are non-linear and of mixed types. To tackle the redundancy optimization problem for a system with different capacities, we propose two problem-specific evolutionary algorithms for the cost minimization and for the availability maximization. For the first case, the algorithm makes use of evolutionary operators that map feasible points into other feasible points. Specialized operators for boundary search allow an optimal utilization of available resources, since the search is restricted to solutions corresponding to a full utilization of these resources. In the second case, boundary search operators cannot be explicitly stated. The proposed alternative is a repair algorithm that maps infeasible solutions to feasible ones as close as possible to the boundary of feasibility, if not exactly on it.

Keywords. Reliability, availability, optimization, networks, genetic algorithms.

† This research was carried out while Alain Ratle was doing his researches at Department of Mechanical Engineering, Université Laval.

* Corresponding author: Mohamed-Larbi.Rebaiaia@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2013

1. Introduction

The resort to redundancy, active or passive, has largely been exploited in system design in order to improve their performance in terms of reliability, maintainability, flexibility and availability. For maintenance planning and resources management, redundancy provides more flexibility (Jin T. et al. (2013); Chen et al., 2013; Rebaiaia and Ait-Kadi, 2010 (2)) ; Coit and Konak, 2006). The optimal allocation problem has been deeply discussed in the literature (Aggarwal *et al.*,1975; Ait-Kadi et al.,1998; Coit *et al.*, 1996 (1); Coit *et al.*, 1996; Coit *et al.*, 1996 (2); Tillman *et al.*,1977. It is proved that it is NP-complete problem. The proposed algorithm provides a good result. It has been used by the authors to determine the spare parts requirement which guaranties some predetermined availability level. In this aim, the standby (passive) redundancy, rather than the parallel (active) was used. Only the availability expressions of the subsystems are affected. The proposal algorithm is efficient and simple to implement. The optimal design of parallel/series systems where redundancy of parallel components is allowed has been a major concern in reliability engineering for at least a half of century, following the work of Aggarwal (Aggarwal et al, 1975) and Fyffe (Fyffe, 1996). The original problem consists of finding the optimal number of parallel redundant elements on each of the *N* stages of a serial system, together with the particular type of elements to be used on each stage from a finite set of design alternatives. The concept of a parallel/series system is illustrated in Figure 1. Most of the early approaches to this problem consider only the case of such a purely integer formulation, since the classical programming methods can hardly deal with mixed variable types, although a first mixed variables approach have been proposed by Tillman (Tillman, 1977). In a mixed-variables formulation, the number of components is represented by integers, but the availability or reliability of each component is a real value rather than a finite set of values. This paper deals with the mixed-variables formulation of the problem.

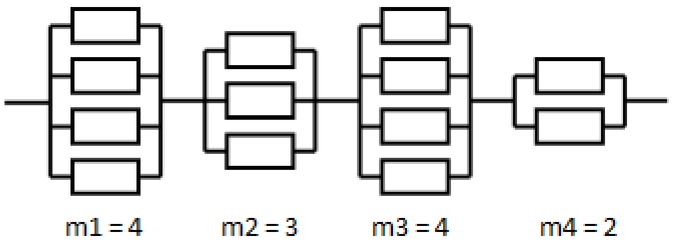

Figure 1: An example of a serial system with redundant components in parallel (m_i: $stage\ i$).

A first problem-specific optimization method was proposed by Aggarwal (1975). His procedure is initiated by providing a feasible point, and each iteration adds a redundant element to the stage where the payoff is greatest. Whenever a constraint is violated, the move is cancelled, and another one is tried out. The algorithm terminates if a constraint is exactly satisfied or no more moves can give a feasible solution. Clearly, the Aggarwal's algorithm is a local search method, since the information available at each step of the optimization algorithm is limited to the close neighbourhood of the current solution. For such a problem, a global search method is likely to gives better results, since the payoff function is probably far from a unimodal.

Genetic algorithm (GA) in evolutionary computation has been used intensively because of its potential of being a very effective design optimization technique for solving various NP hard/ill-structured problems (Holland, 1975; Goldberg, 1989). A GA-based algorithm involves the evaluation of a population of solutions that are revised over successive generations. They are search and optimization methods based evolution in nature. They were first developed by John Holland (Holland, 1975). The power of GA’s is, instead of working with particular point in the solution space, they proceed using more than one point and they are not necessary concerned with finding the optimal solution, but they produce a satisfactory one. The structure of a genetic algorithm is based on natural selection (see. Figure 1). First an initial population of feasible solutions is randomly generated. The initial population consists of *chromosomes*. The selection takes place between members of a population, and a child is formed from the combination of the parents chromosomes. Whether or not the child becomes a member of the population depends on its *fitness* value. Each new child is compared against the worst member and the better one is kept in the population. By producing new generations, the population improves and the best member of the final population is the solution returned by the algorithm. In GA algorithms two particular operators are used to introduce new prospective design solutions at each generation. They are named *crossover* and *mutation*. Crossover involves the selection of parent and their recombination to produce new prospective solutions. Parent selection is random, but biased by the ordinal objective function ranking within a current population (Taboada and al.,*; Taboada and Coit, 2012). Rebaiaia et al. (2002, 2006) proposed a toolset of algorithms based on genetic algorithms for recognizing fingerprint minutiae. It has been demonstrated that the implemented software program can retrieve all the forms that characterize a digital fingerprint with an accuracy approaching 99%. The algorithm has been performed using real fingerprints. Again, Rebaiaia et al. (2002) used another genetic algorithm as a technique to verify reactive systems. Recently, Chen et al. (2013) used a variant of standard GA algorithm for three types of preventive maintenance activities of a reusable rocket engine. The objective was to obtain an optimal scheduling plan for conducting these preventive maintenance actions by minimizing the total cost under the system reliability constraint. Aghaie et al. (2013) proposed an Advanced Progressive Real Coded Genetic Algorithm (APRCGA) to optimize the availability of standby systems with preventive maintenance scheduling. APRCGA code was used in two nuclear power plant emergency systems. The objective of this work was to apply preventive maintenance scheduling that keeps unavailability of systems within safe and reliable conditions. Volkanovski et al (2008) introduce a new method for optimisation of the maintenance scheduling of generating units in a power system. The proposed method uses genetic algorithm to obtain the best solution resulting in a minimal value of the annual loss of load expectation. Yang and Yang (2012) proposed a genetic algorithm for optimizing the cost of a scheduling maintenance plan of aircraft. The feasibility of the model and algorithm is verified by an example of an airline. All solutions proposed in such articles are similar.

In GA's, mutation operator for example, involves the addition or removal of components in accordance with a pre-selected mutation rate. This prevents premature convergence to local optima. The culling operator involves the selection of the solutions with the highest objective function from among the prior population and the newly formed solutions.

The algorithm continues for a pre-determined maximum number of generations (*G*) (see figure 2). More efficient genetic algorithms have been proposed by Coit and Smith, using a specific carling tailored to the structure of the problem and an adaptive penalty method for handling the constraints (Coit, 1996(1); Smith, 1996 and Coit, 1996(2)).

As discussed previously, a genetic algorithm should proceed using the following steps:

- Starts: Creation of a random initial population (*n chromosomes*).
- Creation of a sequence of new populations by evaluating the fitness of each individual member in the current generation to create the next population by performing the following actions:

    - Scores each member of the current population by computing its fitness value.
    - Scales the raw fitness scores to convert them into a more usable range of values.
    - Selects members, called parents, based on their fitness (*selection* operation).
    - Some of the individuals in the current population that have lower fitness are passed to the next population.
    - Produces children from the parents using the *mutation* operation or by combining the vector entries of a pair of parents (*crossover* operation).
    - With a mutation probability mutate new offspring at each position in the chromosome.
    - Replaces the current population with the children to form the next generation.

- If the end condition is satisfied, the algorithm stops iterating and return the best solution in current population.

Note that because the genetic algorithm uses random number generators, the algorithm returns slightly different results each time you run it.

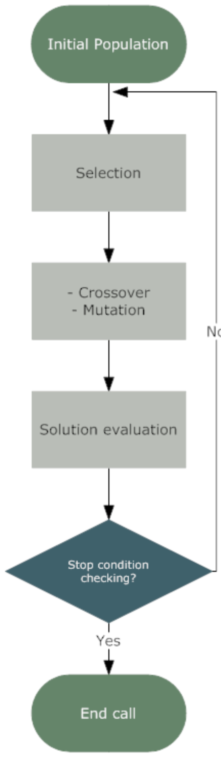

Figure 2 : Flowchart of a simple genetic algorithm

In this paper, the proposed approach is the use of a problem-specific evolutionary algorithm that integrates as much information as possible on the problem. The basic evolutionary operators are redefined with respect to the solution carling and to the constraints on the feasibility of solutions. A constrained optimization problem might be greatly simplified if one has the insight that the best feasible solution necessarily lies at the edge of the feasible domain. In this case, a specialized algorithm working exclusively around this particular region shall certainly salves the problem more efficiently than a general-purpose algorithm exploring the whole domain, as long as such a specialized algorithm can be designed with reasonable efforts. The basic idea of using specialized evolutionary operators to explore the edge of the feasible domain has been recently proposed by Michalewicz and Schoenauer (1996) for numerical optimization problems. Several forms of general design frameworks for specialized operators have been proposed for numerical optimization problems (Michalewicz, 1996; Schoenauer, 1997 (1) and Schoenauer, 1997 (2)), using mappings to general formulations of constraint equations. On the other hand, numerous researches have been conducted on the use of evolutionary algorithms to solve combinatorial problems. These problems require in most cases the use of specialized evolutionary operators since they cannot be easily stated in a form suitable for standard operators. A well-known case is the Traveling Salesman Problem, for which a feasible solution is a permutation of *N* variables (Fox, 1991 and Davis, 1985). These various works show the interest in problem-specific evolutionary operators.

This paper is structured as follows:

Section 2 introduces the formal problem statement. In section 3, we present Problem 1 and which evolutionary operators proper to genetic algorithms are discussed. Similar to Section 3, Section 4 presents the Problem 2. GA implementation for the availability optimization, results and comparison with Aggarwal's algorithm are showed in Section 5, and Section 6 concludes this work.

2. Formal problem statement

Many different forms of the redundancy/availability allocation problem are found in the literature. This paper considers two cases. The first one, is called *labelled Problem 1*, and consists of minimizing the total cost of a system under a constraint of minimal availability (Zhou et a., 2012; Wang et Zhang, 2011; Rebaiaia and Ait-Kadi, 2010). The problem is stated as follow:

$$\text{Minimize } \left\{ Z = \sum_{i=1}^{N} C_i\left(A_i(t), m_i\right) \right\}$$

$$\text{Subject to } \prod_{i=1}^{N}\left[1-\left(1-A_i(t)\right)^{m_i}\right] \geq A^* \tag{1}$$

$$\text{and,} \qquad 0 \leq A_i \leq 1, \forall i$$

where $A_i(t)$ is the availability of a component on the i^{th} stage, or more specifically, the probability that the component remains available during an operating time of duration *t* (Rebaiaia M-L, Ait-Kadi D., 2010; Arturo et al., 2009). The redundancy m_i is the number of components in parallel on stage *i*, and the value A^* is the minimal acceptable availability (Rebaiaia, 2010). The A_i's are coded by real variables bounded between 0 and 1, while the redundancies take integer values only. The minimal number of components on any stage is 1, and there might or might not be an upper bound. The cost C_i of each stage is given by the following equation (Aït-Kadi and Chelbi

(1998), Amari and Pham, 2007, Moghaddam, 2010; Mettas and Zhao, 2005; Chou and Le, 2011, Dedopoulos and Smeers, 1998; Bartholomew et al, 2009):

$$C_i = \alpha_i \left(\frac{-t}{\ln(A_i)} \right)^{\beta_i} \left[m_i + \exp\left(\frac{m_i}{4} \right) \right] \tag{2}$$

The constants α_{i} and β_i are problem-dependent values given by the statistical distribution of component failures (Tillman, 1977). The second case, labelled Problem 2, consists of maximizing the system's availability under various constraints, representing upper limits on the cost, weight and volume allowed for the global system. The problem is expressed as follow, for an *N*-stages parallel/series system:

$$\begin{aligned} &\text{Maximize } \left\{ A_s = \prod_{i=1}^{N} \left[1 - (1 - A_i(t))^{m_j} \right] \right\} \\ &\text{Subject to } \sum_{i=1}^{N} g_{ij}\left(A_j(t), m_j\right) \le b_j \quad j = 1 \ldots K \\ &\text{and,} \quad 0 \le A_i \le 1, \forall i \end{aligned} \tag{3}$$

The constraints under the generic formulation $\sum_{i=1}^{N} g_{ij}\left(A_j(t), m_j\right)$ may represent any limit imposed on the resources allocated to the system. In the present study, three constraints are considered. The first one is an upper bound on the cost:

$$\sum_{i=1}^{N} C_i \le C_{\max} \tag{4}$$

The cost C_i of each stage is the same as given by Eq. 2. The two other constraints impose upper limits on the total weight W_{max} and volume V_{max} of the system:

$$\sum_{i=1}^{N} W_i m_i \exp\left(m_j / 4\right) \le W_{\max} \tag{5}$$

$$\sum_{i=1}^{N} W_i V_i (m_i)^2 \le V_{\max} \tag{6}$$

The W_i's and V_i's are constant values defining the weight and volume of each component. The term given by exp(m_j/4) is usually associated with the cost of the hardware required for the connections between the various components of the redundant system. The number of necessary connections grows exponentially with the number of components to be connected (Tillman, 1977).

3. Evolutionary Operators for Problem 1

The Problem 1 implies the minimization of the total cost of a system under a minimal availability constraint. It is clear from Eq. 2 that the payoff function is monotonically growing with respect to the *A_i's*. Consequently, the minimal cost value satisfying the constraint should correspond to a point of minimal availability A^*. This means that the search can be restricted to the region defined by the expression of the objective function of the system (3) without loss on the quality of the final solution. An evolutionary optimization algorithm that performs this task is developed in the following way. Let the global availabilities *Ai* of the *i*th stage be defined as:

$$A_i = 1 - (1 - A_i)^{m_i} \tag{7}$$

The proposed optimization algorithms works on the redundancies and the global availabilities *Ai* instead of the component availabilities A_i. The minimal availability constraint is transformed into:

$$\prod_{i=1}^{N} A_i \geq A^* \tag{8}$$

The problem is now to devise a search algorithm restricted to the subspace defined by

$$\prod_{i=1}^{N} A_i = A^*$$

Using these variables, fitness is evaluated by the inverse transform, that is,

$$A_i = 1 - (1 - A_i)^{1/m_i}$$

There are no constraints on the numbers of components *m_i's* other than lower and possibly upper bounds.

3.1 Initialization Operator

An initialization operator for the redundancy /availability allocation problem has to generate a string of *N* integers and *N* real values, arranged in some predetermined way. The string should represent a solution lying on the edge of the feasible domain. The proposed operator makes use of the following relation:

$$A_i = (A^*)^{r_i} \text{ and } \prod_{i=1}^{N} A_i = A^* \Leftrightarrow \sum_{i=1}^{N} r_i = 1 \tag{9}$$

The operation consists of generating a set of N uniformly distributed random numbers r_i, with their sum being equal to 1. Moreover, since no global availability is allowed to be greater than 1, all of the r_i's should be positive, and consequently an upper limit of 1 is to be imposed on each random r_i. The numbers r_i are calculated as follow:

- Draw N independent random numbers uniformly distributed between 0 and 1.
- Classify these numbers in increasing order and label them k_i, with i = 1... N.
- Take $r_1 = k_1$,
- $r_i = k_i - k_{i-1}$ for $1 \langle i \langle N$
- and $r_N = 1 - k_{N-1}$.

For the redundancies, random values between 1 and the upper bound can be drawn independently for each of the stages.

3.2 Crossover Operator

In the same way as for the initialization operator, the crossover should be handled differently for the redundancies and the global availabilities. The formers can be crossed-over using a simple uniform Boolean operator: starting from two parents having on their i^{th} stage the redundancies m_i^0 and m_i^1, the value m_i inherited by the offspring is:

$$m_i = \chi m_i^0 + (1-\chi)m_i^1 \;\forall i \in \{1...N\},\; \chi \in \{0,1\} \tag{10}$$

The Boolean random variable X is drawn independently for each i. For the global availabilities, the constraint is clearly respected using the following crossover operator:

$$A_i = \left(A_i^0\right)^{\alpha}\left(A_i^1\right)^{1-\alpha} \quad \forall i \in \{1...N\},\; 0 \le \alpha \le 1 \tag{11}$$

The uniform random variable α is drawn only once for each offspring. Crossover under the form given by Eq. 11 has been first proposed by Michalewicz and Schoenauer (1996) for a different problem, subject to a constraint of the same shape.

3.3 Mutation Operator

The mutation operator designed for the problem at hand is intended to be applied stochastically to a fraction of the offsprings. Whenever a mutation is applied to one particular individual, it is applied either to redundancies or to availabilities with a probability of 50% in each case. Mutation of a redundancy is clone by giving a discrete random value to an arbitrarily chosen *mi*. For the availabilities, the following approach is proposed: select randomly two variables, A_k and A_l, give a random value to the first one and modify the second one in order to balance the constraint. The value of the second variable is given by:

$$A_l' = \frac{A_k A_l}{A_k'} \tag{12}$$

4. Evolutionary Operators for Problem 2

Boundary operators are highly desirable whenever the best solution to an optimization problem is expected to lie on the edge of the feasible domain. These operators can not however be easily developed for any problem, at least with reasonable efforts. In these cases, another potentially interesting approach is the use of a repair operator which maps any non-feasible solution into a feasible one. The proposed approach for the problem 2 is a compromise between boundary search algorithms and repair-based algorithms: a repair operator is utilized for mapping a non-feasible solution to one that lies as close as possible to the interior boundary of the feasible do- main. For this problem, although a feasible solution can not be easily built from scratch, any non-feasible one can be mapped to the boundary of feasibility with little computational efforts. In the same fashion, mutation and crossover operators restricted to the edge of feasibility cannot be easily devised. The proposed alternative consists of using constraint-independent operators together a problem-specific repair operator. Information on the constraints is included only into the repair operator while the other operators act as blind search operators with respect to the constraints. These operators are still problem-specific in some way, since the information structure of the genetic code is inspired from the physics of the problem.

4.1 Initialization Operator

In the same way as for the problem 1, the initialization operator has to generate a string of N integers and N real values. The values should only respect upper and lower bounds on redundancies (between 1 and some maximal value) and availabilities (between 0 and 1), since the other constraints are dealt with using a repair algorithm.

4.2 Crossover Operator

The crossover operator also remains simple as long as there are no constraints to be implicitly handled. Uniform crossover of discrete variables is simply clone by giving to the offspring the value from one of its two parents, with a probability of one half for each one. The availability A_i of the offspring is obtained from the availabilities A_i^0 and A_i^1 from the two parents by the following relation:

$$A_i = \alpha_i A_i^0 + (1-\alpha_i)A_i^1 \quad \forall i \in \{1 \ldots N\}$$
$$\text{with } 0 \le \alpha_i \le 1 \tag{13}$$

where α_i is a uniformly distributed random variable between 0 and 1 that is drawn independently for each of the N stages.

4.3 Mutation Operator

The mutation operator for this problem is also applied stochastically to a fraction of the individuals. When- ever it is applied, the mutation takes place either on the availabilities or on the redundancies, with equal probabilities. Mutation of a redundancy is clone by giving a discrete random value to one of the m_i's. Mutation of the availabilities is performed by the addition of a random noise vector to the N availabilities of a solution. A truncated normal distribution is employed for that purpose. An implicit satisfaction of the upper and lower bounds on the availabilities cannot be ensured with a normally distributed mutation, since the normal distribution is theoretically unbounded. For this reason, a slight modification is clone by truncating the distribution to a point where the density of probability is almost zero. A fraction of 99.73% of the probability density of a normal distribution $N(0, \sigma^2)$ is comprised between -3σ and 3σ. The distribution can therefore be truncated between these values with no significant bias. The proposed mutation model is the following:

$$A_i' = \begin{cases} A_i + \Delta(t, 1 - A_i) & \textit{- if a random bit is 0} \\ A_i - \Delta(t, A_i - A^*) & \textit{- else} \end{cases} \tag{14}$$

With

$$\Delta(t,y) = \begin{cases} \frac{y}{3} \cdot N(0,1) \cdot \left(1 - \frac{t}{t_{max}}\right)^b & \text{if } N(0,1) < 3 \\ y \end{cases} \tag{15}$$

The time t represents the index of the current generation while t_{max} is the maximum number of generations, b is a control parameter, and $N(0, 1)$ is a normally distributed random variable of zero mean and unit variance.

4.4 Repair operator

So far, problem-specific operators have been devised but no cares have been taken about the feasibility of the solutions, this aspect being handled by the repair operator. It is observed from Eq. 2 and 4 that the cost constraint makes use of the availabilities and the redundancies, while the weight and volume constraints depend only on the redundancies. It is by the way possible to deal first with the weight and volume constraints only, working with the redundancies, and then to fine tune the availabilities in such a manner that the solution lies precisely on the edge of the cost constraint. A simple way to correct infeasibilities with respect to the integer-valued constraints (weight and volume) is to choose randomly a stage *i*, decrease its redundancy by one, and repeat until the solution becomes feasible with respect to weight and volume. Care should be taken of not decreasing the number of components on a given stage when there is nothing but one component.

Once the *m_i's* have been corrected in such a way that the weight and volume are feasible, the total cost *C* is corrected if it still lies above the maximal value of C_{max}. This correction is clone by a manipulation on the *Ai's* alone. Since the total cost is the sum of the costs associated with each stage, a solution to this problem is to reduce each cost C_i by a factor C_{max}/C. The new availabilities A_i' are calculated from the old (illegal) ones A_i using the following relation:

$$A_i' = A_i^{\left(\frac{1}{(C_{\max}/C)^{1/\beta_i}}\right)} \qquad (16)$$

5. Some preliminary results

The specialized evolutionary algorithms proposed in this paper have been tested on two redundancy/availability allocation problems. The first one is a simple 5 stages problem proposed by Aït-Kadi and Chelbi (1998). The parameters defining this problem are given on Table I. The optimization was performed assuming an operating duration of 1000 units of time, a minimal availability of 0.9, and maximal cost, weight, and volume of 350, 400 and 220 units respectively.

Table I : Definition of the 5-stages test problem

i	α_i	β_i	W_i	V_i
1	$2.33.10^{-5}$	1.5	7	0.14
2	$1.45.10^{-5}$	1.5	8	0.25
3	$5.41.10^{-5}$	1.5	8	0.38
4	$8.05.10^{-5}$	1.5	6	0.67
5	$1.95.10^{-5}$	1.5	9	0.22

The second problem to be carried out is a 25-stages system that consists of a serial combination of 5 of the systems defined by the first problem. The same parameters given on Table I are used. To keep all things equals, the minimal availability is corrected to 0.95, and the other constraints are scaled up in a similar way. Solutions for the two test problems have been found using a basic evolutionary optimization algorithm with the parameters given on Table II.

Table II: Optimization parameters for the basic evolutionary algorithm.

Parameter	Value	
	5-stages	25-stages
Population size	40	100
Number of generations	100	1000
Selection (tournament size)	3	4
Survivors per generation	4	10

5.1 Mutation or Crossover actions

The relative contributions of the mutation and crossover operators can be isolated by comparing the solutions obtained with various mutation rates, all other things being kept equals. Experiments have been clone with one mutation every two individuals, one every ten and no mutation at all. Results are presented on Figure 3 for the 5 stages version of the problem 1, and on Figure 4 for the 25 stages version. The fitness value for these two cases is the total cost *Z* given by Eq. 1. In both cases, no significant difference is observed either on the convergence rate or on the quality of the final solution. This suggests that all the improvements in solutions quality are due solely to the crossover operator together with natural selection. Similar results have been obtained for the problem 2 on Figure 5 for the 5-stages problem and on Figure 6 for the 25 stages problem. This time, the fitness is equal to the availability given by Eq. 3.

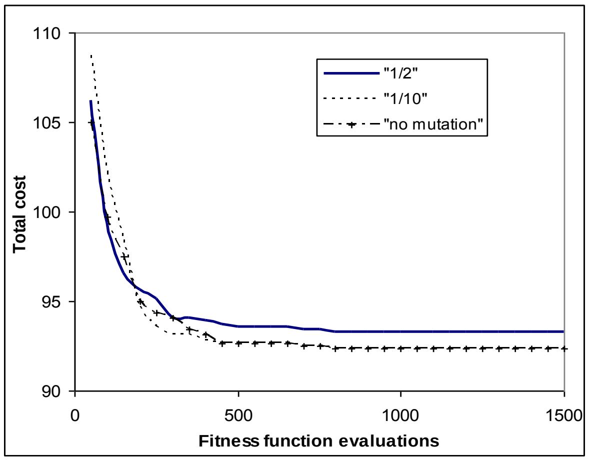

Figure 3: Minimization of the total cost for the 5-stages problem.

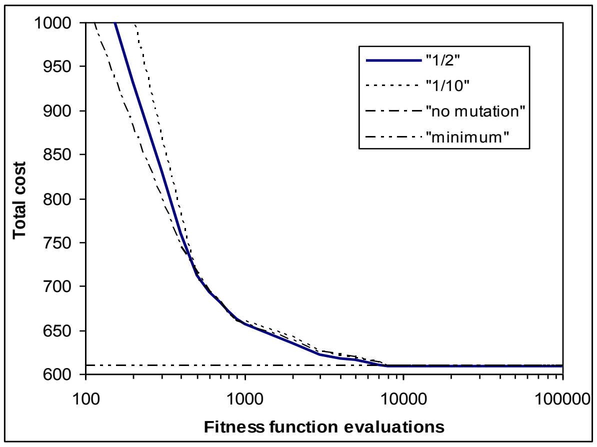

Figure 4: Minimization of the total cost for the 25-stages problem.

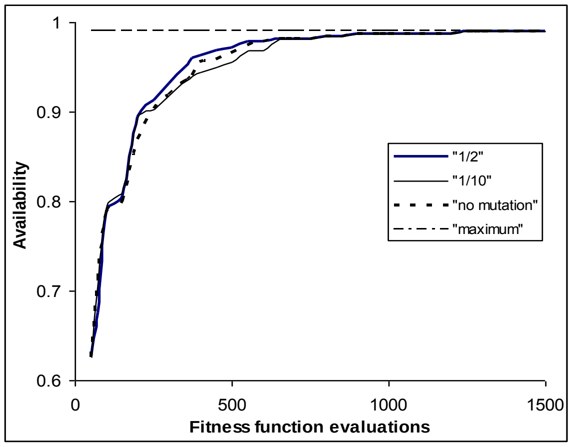

Figure 5: Maximization of the availability for the 5-stages problem.

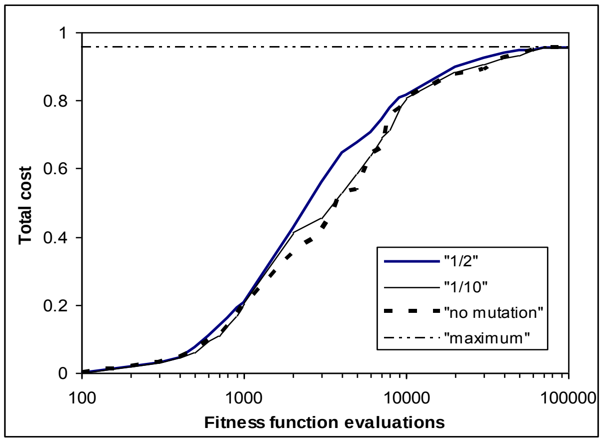

Figure 6: Maximization of the availability for the 25-stages problem.

5.2 Comparison with Aggarwal's Algorithm

The results obtained with one of the problem-specific evolutionary algorithm have been compared with those found by Aït-Kadi and Chelbi (1998) using the Aggarwal's algorithm

(Aggarwal, 1975)), for the maximal availability problem. Table III presents the best solution ever found in bath cases, that is, the maximal value of the total availability A_s for the 5-stages problem. Both solutions are feasible with respect to cost, weight, and volume constraints.

Table III: Comparison of the best solutions found using the problem-specific evolutionary algorithm and the Aggarwal's algorithm, for the 5-stages problem.

	Evolutionary Algorithm		Aggarwal's Algorithm	
i	A_i	m_i	A_i	m_i
1	0.7974	4	0.77455	3
2	0.8808	3	0.83419	3
3	0.7712	4	87300	2
4	0.7627	4	0.72007	3
5	0.8134	4	0.84830	2
A_s	0.98954		0.92291	
Cost	350		287	
Weight	377		189	
Volume	190		83	

The best solution found by the evolutionary algorithm has an availability of 0.98954, while the Aggarwal's algorithm has in the best case terminated with availability 0.92291. This large discrepancy is easily understood when the costs of the two solutions are compared. The evolutionary algorithm gave a solution that make use of the whole budget of 350 money units, while the Aggarwal's algorithm terminated with a cost of only 287 units. The repair operator employed by the evolutionary algorithm explicitly maps the infeasible solutions to the limit of the cost constraint, while the Aggarwal's algorithm terminate as soon as a constraint is satisfied, with no regard to the value of the other constraints. This point shows that the definition of optimality one gives for a problem is often questionable, since the amount of cost, weight and volume resources dedicated to the problem might better be utilized to their maximum extent.

6. Conclusion

This paper presented a problem-dependent approach for the evolutionary optimization of redundancy j availability allocation in parallel/series systems. Evolutionary algorithms can be considered as an assembly of building- blocks where some blocks are general and some others are problem-specific. The design of appropriate problem-specific blocks allows the search to be restricted to the boundary of the feasible domain. This approach has two important advantages over the use of general purpose algorithms. First, the computational cost of the optimization problem can be significantly reduced since the search is *a priori* biased toward potentially interesting regions by some physical knowledge over the problem. Secondly, the quality of the final solutions obtained is guaranteed to be equivalent or better to those found with other methods, since only the boundary of feasible domain is searched. In the redundancy/availability allocation problem, as well as in many other constrained problems, it is easily observed that the best solutions necessarily lies at the edge of the feasible domain. This ensures an optimal utilization of the available resources for a given problem.

Bibliography

Aggarwal, K.K., Gupta, J.S. and Misra, K.B. (1975), "A new heuristic criterion for solving a redundancy optimization problem", *IEEE Transactions on Reliability,* R-24(1), pp. 86-87.

Aghaie M., A. Norouzi,, A. Zolfaghari, A. Minuchehr, Z. Mohamadi Fard, R. Tumari, 2013, Advanced progressive real coded genetic algorithm for nuclear system availability optimization through preventive maintenance scheduling, Annals of Nuclear Energy 60 , pp. 64–72.

Aït-Kadi, D. and Chelbi, A. (1998), "Optimal design of systems using active redundancy with repairable components", International Journal of Industrial Engineering and Mechanical Production, 1(1).

Amari SV, Pham H., 2007, A novel approach for optimal cost-effective design of complex repairable systems, IEEE Transactions on Systems, Man, and Cybernetics, 37, pp. 406–15.

Merlano A., Ait-Kadi D., Rebaiaia M-L, 2009, Optimisation de la disponibilité d'un système assujettis à de la maintenance imparfaite sous des contraintes de budget. Publié dans les actes du congrès PErformances et Nouvelles TechnolOgies en Maintenance (PENTOM'09) Du 7 au 8 décembre 2009 à Grenoble (Autrans), France.

Bartholomew-Biggs M, Zuo MJ, Li X., 2009, Modelling and optimizing sequential imperfect preventive maintenance. Reliability Engineering and System Safety, 97, pp. 53–62.

Chen T., Li J., Jin P., Cai. G., (2013), Reusable rocket engine preventive maintenance scheduling using genetic algorithm, Reliability Engineering and System Safety 114 (2013) 52–60.

Coit, D.W. and Smith, A.E. (1994), "Use of a genetic algorithm to optimize a combinatorial reliability design problem", In *Froc. Third IIE Research Conf.,* pp. 467-472.

Coit, D.W. and Smith, A.E. (1996) (1), "Reliability optimization of series-parallel systems using a genetic algorithms", *IEEE Transactions on Reliability,* 45(2), pp. 254-266.

Coit, D.W. and Smith, A.E. (1996) (2), “Penalty guided genetic search for reliability design optimization” *Computers Ind. Engng.*, 30(4), pp. 895-904.

Coit A. W., Jin T. and Wattanapongsakorn, (2004), System optimization considering component reliability estimation uncertainty: a multi-criteria approach, *IEEE Transactions on Reliability*, 53(3), 369-380.

Coit A. W and Konak A., (2006), Multiple weighted objectives heuristic for the redundancy allocation problem, *IEEE Transactions on Reliability*, 55(3), 551-558.

Chou JS, Le TS., 2011, Reliability-based performance simulation for optimized pavement maintenance, Reliability Engineering and System Safety, 96.

Davis, L. (1985), “Applying adaptive algorithms to epistatic domains”, In *International Joint Conference on Artificial Intelligence,* pp. 162-164.

Dedopoulos IT, Smeers Y, 1998, Age reduction approach for finite horizon optimization of preventive maintenance for single units subject to random failures, Computers and Industrial Engineering, 34, pp. 643–54.

Fox, B.R. and McMahon, M.B. (1991). “Genetic operators for sequencing problems”, ln Gregory J. E. Rawlins, editor, *Foundations of Genetic Algorithms,* pp. 284-300.

Fyffe, D.E., Hynes, W.W. and Lee N.K. (1968), “System reliability allocation and a computational algorithm”, *IEEE Transactions on Reliability,* R-17(2), pp. 64-69.

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading, MA.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems,* MIT Press.

Jin T, Tian Y, Zhang C.W and Coit D, Multi-Criteria Planning for Distributed Wind Generation under Strategic Maintenance, IEEE Transactions on Power Delivery, vol. 28, no. 1.

Michalewicz, Z. and Schoenauer, M. (1996), “Evolutionary algorithms for constrained parameter optimization problems”, *Evolutionary Computation,* 4(1), pp. 1-32.

Painton, L. and Campbell, J. (1995), “Genetic algorithms in optimization of system reliability”, *IEEE Transactions on Reliability,* R-44(2), pp. 172-178.

Rebaiaia M-L, Ait-kadi D., Merlano A, 2009, Une méthodologie pour la modélisation de la fiabilité et de la disponibilité d’un réseau de radiocommunication, Journées d’optimisation, Montréal.

Rebaiaia Mohamed-Larbi and Ait-Kadi, Daoud, Model based binary decision diagrams for complex networks reliability optimization, IFAC-PapersOnLine, Editor: Milik, Adam, Hrynkiewicz, Edward, Programmable Devices and Embedded Systems, Volume # 10, Part# 1, IFAC-PapersOnLine.net, Elsevier, ISSN: 1474-6670, 2010.

Rebaiaia M-L, Ait-Kadi D., 2010, A Contribution for Modeling and Optimizing the Availability of Telecommunication Systems under Budget Constraints, Conference Internationale de Recherche Opérationnelle, CIRO'10, Marrakech, Maroc.

Rebaiaia M-L, Benmohamed M., 2002, Verification Algorithms for Integer Programming and Genetic Algorithms, Proc. of International Arab Conference of Information Technology, Doha, Qatar.

Jaam J. M., Rebaiaia M-L, Hasnah A., 2006, A Fingerprint Minutiae Recognition System Based on Genetic Algorithms, Int. Arab J. Inf. Technol. 3(3), pp. 242-248.

Mettas A, Zhao W, 2005, Modeling and analysis of repairable systems with general repair. Proceedings of the annual reliability and maintainability Symposium, Virginia, USA.

Moghaddam KS, 2010, Preventive maintenance and replacement scheduling models and algorithms, University of Louisville, USA.

Rebaiaia Mohamed-Larbi., Benmohamed M, Jihad Mohamad Jaam and Hasnah Ahmad, Verification Algorithms for Integer Programming and Genetic Algorithms, In the International Journal of Applied Sciences & Computations, Vol. 10, N°3, pp.191-204, December, ISSN : 1089-0025, 2003.

Schoenauer, M. and Michalewicz, Z. (1996), "Evolutionary computation at the edge of feasibility", In Werner Ebeling, editor, *Parallel Problem Solving from Nature IV*.

Schoenauer, M. and Michalewicz, Z. (1997) (1), "Boundary operators for constrained parameter optimization problems", In Thomas Bäck, editor, *7th International Conference on Genetic Algorithms*, pp. 322-329.

Schoenauer, M. and Michalewicz, Z. (1997) (2), "Sphere operators and their applicability for constrained parameter optimization problems", *7th Annual Conference on Evolutionary Programming*.

Smith, A.E., Coit, D.W. and Tate, D.M. (1996), "Adaptive penalty methods for genetic optimization of con- strained combinatorial problems", *INFORMS journal on Computing,* 8(2), pp. 173-182.

Taboada H.T., Beheranwala F. Coit D., Wattanapongsakorn N., Practical Solution of Multi-objective System Reliability design Problems using Genetic Algorithms, Research paper, paper: 05-011, Rutgers University.

Taboada H. and Coit D., (2012), "A New Multiple Objective Evolutionary Algorithm for Reliability Optimization of Series-Parallel Systems,", International Journal of Applied Evolutionary Computation, vol. 4, no. 2.

Tekiner-Mogulkoc H. and Coit D., 2011, "System Reliability Optimization Considering Uncertainty: Minimization of the Coefficient of Variation for Series-Parallel Systems," *IEEE Transactions on Reliability*, vol. 60, no. 3.

Tillman, F.A., Hwang C.L. and Kuo, W. (1977), "Determining component reliability and redundancy for optimum system reliability", *IEEE Transactions on Reliability,* R-26(3), pp. 162-165.

Volkanovskia A., Mavkoa B., Bosevskib T., Causevskib A., (2008), Genetic algorithm optimisation of the maintenance scheduling of generating units in a power system, Reliability Engineering and System Safety 93 757–767.

Wang GJ, Zhang YL., 2011, A bivariate optimal replacement policy for a cold standby repairable system with preventive repair. Applied Mathematics and Computation, 218(31), pp. 58–65.

Xiang Y., Coit D. and Feng Q., (2013) "n-Subpopulations Experiencing Stochastic Degradation: Reliability Modeling, Burn-in and Preventive Replacement Optimization," IIE Transactions, vol. 45, no. 4.

Yanga Z., Yang G., 2012, Optimization of Aircraft Maintenance plan based on Genetic Algorithm, Physics Procedia, 33, pp. 580 – 586.

Zia L. and Coit D, 2010, "Reliability Allocation for Series-Parallel Systems Using a Column Generation Approach," it, IEEE Transactions on Reliability, vol. 59, no. 4.

Zhou X, LuZ , Xi L., 2012, Preventive maintenance optimization for a multicomponent system under changing jobshop schedule. Reliability Engineering and System Safety; 101, pp. 14–20.