Multiobjective optimization by genetic algorithms: application to safety systems

P. Giuggioli Busacca, M. Marseguerra*, E. Zio

Department of Nuclear Engineering, CESNEF, Polytechnic of Milan, Via Porzio 34/3, 20133 Milan, Italy

Received 26 September 2000; accepted 4 December 2000

Abstract

When attempting to optimize the design of engineered systems, the analyst is frequently faced with the demand of achieving several targets (e.g. low costs, high revenues, high reliability, low accident risks), some of which may very well be in conflict. At the same time, several requirements (e.g. maximum allowable weight, volume etc.) should also be satisfied. This kind of problem is usually tackled by focusing the optimization on a single objective which may be a weighed combination of some of the targets of the design problem and imposing some constraints to satisfy the other targets and requirements. This approach, however, introduces a strong arbitrariness in the definition of the weights and constraints levels and a criticizable homogenization of physically different targets, usually all translated in monetary terms.

The purpose of this paper is to present an approach to optimization in which every target is considered as a separate objective to be optimized. For an efficient search through the solution space we use a multiobjective genetic algorithm which allows us to identify a set of Pareto optimal solutions providing the decision maker with the complete spectrum of optimal solutions with respect to the various targets. Based on this information, the decision maker can select the best compromise among these objectives, without a priori introducing arbitrary weights.

q 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Multiobjective optimization; Genetic algorithms; Pareto optimal solutions

1. Introduction

The optimal design of modern engineered systems entails the consideration of several targets and requirements to be satisfied along the system life such as high reliability, low costs, low workers’ health risk and low environmental pollution. These aspects are often in conflict with one another, hence a compromise solution has to be sought [1].

Two different approaches have been considered for handling such optimization problems:

(1) weighed aggregation of all the targets into a single objective function to be optimized [2–4];
(2) optimization of one of the targets by itself and imposition on the other targets of appropriate constraints to be satisfied [3,5].

Both these single objective approaches are inevitably open to criticism, the first one because of the homogenization of different quantities, such as reliability, costs, health consequences, to a common unit of measure and the second one because of the choice of which objective function optimize and of the setting of the constraints levels.

The complexity of industrial systems and the nonlinearity of their behavior is such that explicit functions modeling the system evolution are not readily available. These difficulties pose severe limitations to the application of classical analytical and semi-analytical optimization methods such as those based on an evaluation of the gradient of the system function with respect to the solution variables [6]. Thus, thanks to the ever increasing computing power available, new numerical search algorithms are becoming popular. In particular, here we focus attention on Genetic Algorithms (GAs). These are numerical search tools which function according to procedures that resemble the principles of natural selection and genetics [7,8]. Because of their flexibility, ease of operation, minimal requirements and global perspective, GAs have been successfully used in a wide variety of problems in several areas of engineering and life science [9–12]. In recent years an increasing number of GAs applications to single-objective optimizations have been observed in the field of reliability, maintainability and availability analysis [2,5,13–17]. In these applications, the...
performance of any candidate system design solution is measured through the value of the objective function, called fitness.

A more informative approach is one which considers all individual targets separately, aiming at identifying a set of solutions better than others with respect to all targets, but ‘comparatively good’ among themselves. Each member of this set is better or equal to the others of the set with respect to some, but not all, of the targets. The set thereby identified provides a spectrum of ‘good’ solutions which the decision maker can subjectively handle according to which targets he believes to be more or less important. For example, between two solutions a decision maker could prefer the one with highest reliability although obtained at higher costs or vice versa he might privilege low costs, thus giving up some reliability.

The application of genetic and other evolutionary algorithms to multiobjective optimization is subject to great attention in the technical community, as demonstrated by the recent flourishing of publications in the field [18–22]. In few words, the extension of GAs for the treatment of multiobjective optimization problems entails the introduction of a multivalued approach for comparing two solutions with respect to the multiple objectives considered [23,24]. In the case of a single-objective the comparison is trivial since a vector solution \( X \) is better than \( Y \) if the corresponding objective function (fitness) value \( f(X) \) is greater than \( f(Y) \). When the situation is extended to \( N \) objectives we are dealing with \( N \) objective functions \( f_i(\cdot), i = 1, 2, \ldots, N \) and, as we shall see, two solutions \( X \) and \( Y \) must be compared in terms of dominance of one solution over the other with respect to all \( N \) objectives. As a result of the multiobjective search process, convergence is achieved on a Pareto-optimal region of nondominated solutions which can be subjectively managed by the decider to identify his or her preferred solution.

In this paper we present the GAs’ approach to multiobjective optimization and apply it within the reliability/availability analysis framework. In the next section we present the basic principles behind the GA here adopted, formulate the multiobjective optimization problem within the frame of Pareto optimality and provide the details of the extension of the adopted GA within a dominance scheme for multiobjective optimization. A general Fortran computer code called MOGA (MultiObjective Genetic Algorithm) has been developed by the authors and two applications have been worked out with this code: these are presented in Sections 3 and 4, the first one regarding the allocation of redundancy in a simple system and the second one, based on a literature case study [5,25], refers to the choice of the time intervals for the periodic testing of the components of the High Pressure Injection System (HPIS) of a Pressurized Water Reactor (PWR). The paper ends with a section devoted to some conclusions and discussions.

It should be noted that, independent of the optimization approach employed, whether single-or multi-objective, reliability/availability design and inspection/maintenance strategies are typically subject to physical and normative constraints which come into play imposing restrictions that the candidate solutions have to satisfy. These can certainly be handled in a straightforward manner within both the single- and the multi-objective GA optimization procedures but may introduce obstacles in the convergence property of the algorithm. With respect to this issue, for simplicity, in our case studies we do not impose any “a priori” constraint to be satisfied by the candidate solutions. This choice leads perhaps to a somewhat degraded speed of convergence due to the larger search space but allows us to verify the consistency of the experimental constraints with the set of input data governing the problem.

2. Genetic algorithms

In the following sections we provide some background on the GA optimization approach with reference to the traditional single-objective GA [8] and then present the extension of the approach to multiobjective problems [23,24].

2.1. Generalities

GAs, first formalized as an optimization method by Holland [7], are search tools modeled after the genetic evolution of natural species. The GAs differ from most optimization techniques because of their global searching from one population of solutions rather than from one single solution. Every proposal of solution is represented by a vector \( X \) of the independent variables, which is coded in a chromosome, constituted by as many genes as the number of independent variables of the problem; a binary coding is widely used.

In its general form, GA works through the following steps:

1. creation of a random initial population of \( N_p \) potential solutions to the problem and evaluation of these individuals in terms of their fitnesses, i.e. of their corresponding objective function values;
2. selection of a pair of individuals as parents;
3. crossover of the parents, with generation of two children;
4. replacement in the population, so as to maintain the population number \( N_p \) constant;
5. genetic mutation.

Every time a new solution \( X \) is proposed by the GA the objective function is evaluated and a ranking of the individuals in the current population is dynamically updated, based on their fitness values. This ranking is used in the selection procedure which is performed in such a way that in the long run the best individuals will have a greater probability to be selected as parents, in resemblance to the
natural principles of the “survival of the fittest”. Similarly, the ranking is used in the replacement procedures to decide who, among the parents and the daughters, should survive in the next population. An algorithm based on these procedures is often referred to as a steady-state GA [5].

When using GAs, sufficient genetic diversity among solutions in the population should be guaranteed. Lack of such diversity would lead to a reduction of the search space spanned by the GA and consequently to a degradation of its optimization performance with selection of mediocre individuals resulting in premature convergence to a local minimum. On the other hand, an excess of genetic diversity, especially at later generations, may lead to a degradation of the optimization performance, resulting in very late, or even no, convergence.

2.2. Definitions

The GAs owe their name to the fact that their functioning is inspired by the rules of the natural selection: correspondingly, the adopted language contains many terms borrowed from biology, which need to be suitably redefined to fit the algorithmic context. Thus, when we say that the GA operate on a set of (artificial) chromosomes, these must be understood as strings of numbers, generally sequences of binary digits 0 and 1. If the objective function has many arguments, each string is partitioned in as many substrings of assigned lengths, one for each argument and, correspondingly, we say that each chromosome is analogously partitioned in (artificial) genes. The genes constitute the so called genotype of the chromosome and the substrings, when decoded in real numbers called control factors, constitute its phenotype. When the objective function is evaluated in correspondence of the values of the control factors of a chromosome, its value is called the fitness of that chromosome. Thus each chromosome gives rise to a trial solution to the problem.

To code/decode the ith gene in a control factor, that is in an argument of the objective function, the user:

- Defines the range \((a_i, b_i)\) of the corresponding argument in the objective function.
- Assigns the resolution of that independent variable by dividing the range \((a_i, b_i)\) in \(2^n\) intervals. A number \(n_i\) of bits is then assigned to the substring representative of the gene and the relation between a real value \(x \in (a_i, b_i)\) and its binary counterpart \(\beta\) is

\[
x = a_i + \beta \frac{b_i - a_i}{2^n}
\]

The values \(a_i, b_i, n_i\) are called the phenotyping parameters of the gene.

Fig. 1 shows the constituents of a chromosome made up of three genes and the relation between the genotype and the external environment, i.e. the phenotype, constituted by three control factors, \(x_1, x_2, x_3\), one for each gene. The passage from the genotype to the phenotype and vice versa is ruled by the phenotyping parameters of all genes, which perform the coding/decoding actions. Each individual chromosome is characterized by a fitness, defined as the value of the objective function calculated in correspondence of the control factors pertaining to that individual. Thus, a population is a collection of points in the solution space, i.e. in the space of \(f\).

An important feature of a population is its genetic diversity: if the population is too small, the scarcity of genetic diversity may result in a population dominated by almost equal chromosomes and then, after decoding the genes and evaluating the objective function, in the quick convergence of the latter towards an optimum which may well be a local one. At the other extreme, in too large populations, the overabundance of genetic diversity can lead to clustering of individuals around different local optima: then the mating of individuals belonging to different clusters can produce children (newborn strings) lacking the good genetic part of either of the parents. In addition, the manipulation of large populations may be excessively expensive in terms of computer time. The management of the genetic patrimony stored in the population is obviously influenced by the procedures of population evolution and its governing parameters, as detailed below.

In most computer codes the population size is kept fixed at a value set by the user so as to suit the requirements of the model at hand. The individuals are left unordered, but an index is sorted according to their fitnesses. During the search, the fitnesses of the newborn individuals are computed and the fitness index is continuously updated.

As stated above, the GA search is performed by constructing a sequence of populations of chromosomes, the individuals of each population being the children of those of the previous population and the parents of those of the successive population. The initial population is generated by randomly sampling the bits of all the strings. At each step, the new population is then obtained by manipulating the strings of the old population in order to arrive at a new population hopefully characterized by an increased mean fitness. This sequence continues until a termination criterion.
is reached. As for the natural selection, the string manipulation consists in selecting and mating pairs of chromosomes in order to groom chromosomes of the next population. This is done by repeatedly performing on the strings the four fundamental operations of reproduction, crossover, replacement and mutation, all based on random sampling. These operations will be detailed below with reference to the standard, single-objective GA.

2.3. Creation of the initial population

The initial population is generated by random sampling the bits of all the strings. This procedure corresponds to uniformly sampling each control factor within its range. The chromosome creation, while quite simple in principle, presents some subtleties worth to mention. In particular, it may happen that the admissible hypervolume of the control factors is only a small portion of that resulting from the cartesian product of the ranges of the single variables, so that one must try to reduce the search space by resorting to some additional condition. In particular, in some cases the physics of the problem may be such that the value sampled for a gene drastically reduces the admissible range for the successive gene of the chromosome. Thus, by conditionally sampling the values for the successive genes, the search hypervolume can be drastically reduced.

2.4. The traditional breeding algorithm

The breeding algorithm is the way in which the \((n + 1)\)th population is generated from the \(n\)th previous one.

The first step of the breeding procedure is the generation of a temporary new population. Assume that the user has chosen a population of size \(N_p\) (generally an even number). The population reproduction is performed by resorting to the Standard Roulette Selection rule: to find the new population, the cumulative sum of the fitnesses of the individuals in the old population is computed and normalized to sum to unity. The new population is generated by random sampling individuals, one at a time with replacement, from this cumulative sum which then plays the role of a cumulative distribution function (cdf) of a discrete random variable (the position of an individual in the population). By so doing, on the average, the individuals in the new population are present in proportion to their relative fitness in the old population. Since individuals with relatively larger fitness have more chance to be sampled, most probably the mean fitness of the new population is larger.

The second step of the breeding procedure, i.e. the crossover, is performed as indicated in Fig. 2: after having generated the new (temporary) population as above said, \(N_p/2\) pairs of individuals, the parents, are sampled at random without replacement and irrespective of their fitness, which has already been taken into account in the first step. In each pair, the corresponding genes are divided into two portions by inserting at random a separator in the same position in both genes (one-site crossover): finally, the first portions of the genes are exchanged. The two chromosomes so produced, the children, are thus a combination of the genetic features of their parents. A variation of this procedure consists in performing the crossover with an assigned probability \(p_c\) (generally rather high, say \(p_c \geq 0.6\)) : a random number \(R\) is uniformly sampled in \((0,1]\) and the crossover is performed only if \(R < p_c\). Vice versa, if \(R \geq p_c\), the two children are copies of the parents.

The third step of the breeding procedure, performed after each generation of a pair of children, concerns the replacement in the new population of two among the four involved individuals. The simplest recipe, again inspired by natural selection, just consists in the children replacing the parents: children live, parents die. In this case each individual breeds only once.

The fourth and last step of the breeding procedure eventually gives rise to the final \((n + 1)\)th population by applying the mutation procedure to the (up to this time temporary) population obtained in the course of the preceding steps. The procedure concerns the mutation of some bits in the population, i.e. the change of some bits from their actual values to the opposite one (0 → 1 and vice versa). The mutation is performed on the basis of an assigned mutation probability for a single bit (generally quite small, say \(10^{-3}\) or lower). The product of this probability by the total number of bits in the population gives the mean number \(\mu\) of mutations. If \(\mu < 1\) a single bit is mutated with probability \(\mu\). Those bits to be actually mutated are then located by randomly sampling their positions within the entire bit population.

End of the search The sequence of successive population generations is usually stopped according to one of the following criteria:

(i) When the mean fitness of the individuals in the population increases above an assigned convergence value.
(ii) When the median fitness of the individuals in the population increases above an assigned convergence value.
(iii) When the fitness of the best individual in the population increases above an assigned convergence value. This
2.5. More sophisticated breeding procedures

Reproduction. Alternative procedures are:

(i) Hybrid Roulette Selection: the main disadvantage of the Standard Roulette Selection procedure follows from the fact that the new individuals are actually sampled from a multinomial distribution, so that their fitnesses are fairly dispersed around the mean and the convergence towards the best solution can be delayed or even lost. The Hybrid Roulette Selection rule starts by normalizing the fitnesses to their sum and by sampling one of them as in the Standard Roulette Selection case. This normalized fitness is then multiplied by the population size (number of individuals in the population) and the integer part of the product yields the number of individuals, identical to that having fitness \( f \) in the old population, which are deterministically assigned to the new population (of course, this number may be zero). The remainder of the above product is then treated as the probability of adding a further identical individual to the new population. By so doing the permanence of good individuals, i.e. those with relatively higher fitness, is favored along the population sequence. However, the genetic diversity may decline.

(ii) Random Selection and Mating: the two parents are randomly selected with replacement over the entire population, regardless of the fitnesses of the individuals. With respect to both Roulette Methods, on the average, this procedure is more disruptive of the genetic codes: in other words, the chromosomes of the two parents, suitably decoded, can give rise to points very far from each other in the control factor space and, correspondingly, the fitnesses of the newborn children can be largely far apart in the solution space.

(iii) Fit–Fit Selection and Mating: the population is scanned by stepping through the fitness index and pairing each individual with the next fittest individual. On the average, this procedure is highly conservative of the genetic information and a (generally local) maximum of the objective function is soon attained since weak individuals are rapidly eliminated.

(iv) Fit–Weak Selection and Mating: as in the preceding case, the population is scanned by stepping through the fitness index, but this time each individual is paired with that individual located in the symmetric position of the fitness index, with respect to the mid of the fitness list. On the average, this procedure is highly disruptive of the genetic codes, but it helps in improving the genetic diversity. It is seldom adopted.

In all the described procedures, after having selected the two parents and before proceeding to the selection of another couple, the two parents are crossed and the two individuals resulting from the adopted replacement procedure are immediately replaced in the population. Most important, before selecting the successive pair of parents, the fitness index is immediately updated: by so doing the sampling is performed on a dynamically varying population.

Crossover. An obvious generalization of the simple one-site crossing described above, is the multi-site crossing, consisting in the interposition of more than one separator in the substrings representative of the homologous genes of the parents, followed by the exchange of pieces of the involved substrings. The simplest case is the two-site crossing: two separators are randomly positioned in the homologous substrings and the bits between the two points are interchanged. However it should be said that the multi-site crossing is rarely adopted and that the simple, one-site, crossover remains the most popular technique.

Replacement. Alternative procedures are:

(i) Fittest individuals: out of the four individuals involved in the crossover procedure, the fittest two, parent or child, replace the parents. This procedure should not be used when weak individuals are discarded in the parent selection step, otherwise the weak individuals have a large chance to remain forever in the population.

(ii) Weakest individuals: the children replace the two weakest individuals in the entire population, parents included, provided the children fitness is larger. This technique shortens the permanence of weak individuals in the successive populations and it is particularly efficient in large populations.

(iii) Random replacement: the children replace two individuals randomly chosen in the entire population, parents included. By this technique, weak individuals have the same chance as the fit ones of being included in the new population: therefore the method is particularly efficient in small populations since it can give rise to a deep search in the space of the control factors.

2.6. Multiobjective genetic algorithms

In order to treat simultaneously several objective functions, it is necessary to substitute the single-fitness based procedure employed in the single-objective GA for comparing two proposals of solution. The comparison of two
chromosome-coded solutions with respect to several objectives may be achieved through the introduction of the concepts of Pareto optimality and dominance [23,24] which enable solutions to be compared and ranked without imposing any a priori measure as to the relative importance of individual objectives, neither in the form of subjective weights nor arbitrary constraints.

In practice, often, constraints exist, based on experience, which impose restrictions that the candidate solutions have to satisfy. Such constraints may be handled, just as in the case of single-objective GAs, by testing for the fulfillment of the criteria by the candidate solutions during the population creation and replacement procedures. The introduction of external constraints speeds up the convergence of the algorithm because it reduces the search space. In the multiobjective applications of this paper we chose not to impose any constraint a priori on the proposed candidate solutions. This was done so as to ascertain that the experience-based constraints are consistent with the set of input data governing the problem. Results quite different from those suggested by experience would imply a deeper investigation to find whether a suboptimal solution is achieved or a more careful analysis on the input data and model consistency.

Let us consider \( N \) different objective functions \( f_i(X) \), \( i = 1,..,N \) where \( X \) represents the vector of independent variables identifying a generic proposal of solution. We say that solution \( X \) dominates solution \( Y \) if \( X \) is better on all objectives [3], i.e. if

\[
f_i(X) > f_i(Y) \quad \text{for} \quad i = 1,..,N
\]

If a solution is not dominated by any other in the population, it is said to be a nondominated solution. Using this definition, a ranking of the population can be readily performed (Fig. 3). All nondominated individuals in the current population are identified. These solutions are considered the best solutions, and assigned the rank 1. Then, these solutions are virtually removed from the population and the next set of nondominated individuals are identified and assigned rank 2. This process continues until every solution in the population has been ranked.

The selection and replacement procedures of the multiobjective GAs are based on this ranking: every solution belonging to the same rank class has to be considered equivalent to any other of the class, i.e. it has the same probability of the others to be selected as a parent and survive the replacement.

During the optimization search, an archive of a given number of nondominated solutions representing the dynamic Pareto optimality surface is recorded and updated. At the end of each generation, nondominated solutions in the current population are compared with those already stored in the archive and the following archival rules are implemented:

1. If the new solution dominates existing members of the archive, those are removed and the new solution is added.
2. If the new solution is dominated by any member of the archive, it is not stored.
3. If the new solution neither dominates nor is dominated by any member of the archive then:
   - If the archive is not full, the new solution is stored.
   - If the archive is full, the new solution replaces the most similar one in the archive. (An appropriate concept of distance being introduced to measure the similarity between two solutions: in this paper we shall adopt a euclidean distance based on the values of the fitnesses of the chromosomes.)

The setup of an archive of nondominated solutions can also be exploited by introducing an elitist parents’ selection procedure which should in principle be more efficient. Every solution in the archive (or a pre-established fraction of the population size \( N_p \), typically \( N_p/4 \), if the archive’s size is too large) is chosen once as a parent in each generation. This should guarantee a better propagation of the genetic code of nondominated solutions, and thus a more efficient evolution of the population towards Pareto optimality.

At the end of the search procedure, the result of the optimization is constituted by the archive itself which gives the Pareto optimality region.

3. Multiobjective optimization of system design: a simple application

When designing a system such as an industrial plant, one must give proper account to the constraints coming from the safety and reliability requirements as well as from the limitations on budget and resources. In particular, the problem here considered regards the choice among different potentially valid system redundancy configurations made up of components which can differ for their failure and repair characteristics. The choice of a higher redundancy or of a
The repair rate of component are given in relation to its failure rate. Hence, the failure rate as follows:

\[ \lambda_i = \alpha \sqrt{\mu_i^k} \]  

where \( \alpha \) is a pre-defined constant parameter, expressed in \( 1/\sqrt{y} \).

The conjecture behind this relationship is that a more reliable component is technologically more complicated and will therefore require longer restoration times (and, as we shall see below, greater costs). We also assume, for the sake of simplicity, that the number of repairmen is equal to the number of components constituting the system.

Two separate objectives of the optimization problem are considered: the net profit drawn from system operation during the mission time (\( T_M \)) and the reliability at mission time.

The profit is made up of the following contributions [2]:

- profit from plant operation \( P \);
- purchase and installation costs \( C_A \);
- repair costs \( C_R \);
- penalties during downtime, due to missed delivery of agreed service \( C_{NS} \).

The net profit objective function \( G \) (gain) can then be written as follows:

\[ G = P - (C_A + C_R + C_{NS}) \] (2)

where

\[ P = P_f \int_0^{T_M} A(t) \, dt \] (3)

is the plant profit in which \( P_f \) represents the amount of money per unit time paid by the customer for the plant service, and \( A(t) \) is the instantaneous plant availability

\[ C_A = \sum_{k=1}^{N_n} \sum_{i=1}^{n_k} C_{Ai} \] (4)

is the acquisition and installation cost of the \( N_n \) nodes, the \( k \)th of them constituted of \( n_k \) components; \( C_{Ai} = (\gamma^2/\sqrt{\lambda_i^k}) \) is the contribution due to component \( i \) in node \( k \) and \( \gamma^k (($/\sqrt{y}) \) is a proportionality constant equal for all components of the \( k \)th node

\[ C_R = \sum_{k=1}^{N_n} \sum_{i=1}^{n_k} C_{R,i} \] (5)

is the mean repair cost of all components of the system, with \( C_{R,i} \) being the mean value of repair cost for component \( i \) in node \( k \). Such mean repair cost is assumed to obey the following model:

\[ C_{R,i} = N_{R,i} \left( \frac{1}{\mu_i^k} \right) C_{R,i} \] (6)

where

\[ C_{R,i} = \frac{\beta^k}{\mu_i^k} \] (yearly repair cost for component \( i \) in node \( k \);

\[ \beta^k \] is a proportionality constant equal for all components of the \( k \)th node, expressed in ($$/y^2$$)

\[ N_{R,i} = \frac{T_M}{\lambda_i^k + \frac{1}{\mu_i^k}} \] (mean number of failures during the mission time)

and finally

\[ C_{NS} = C_U \int_0^{T_M} [1 - A(t)] \, dt \] (7)

in which \( C_U \) is the economic penalty per unit time, i.e. the amount of money to be paid to the customer because of missed delivery of the agreed service when the plant is unavailable.

Note that in this single model we did not introduce the interest rates.

The second objective function considered is the reliability at mission time \( R(T_M) \). The evaluation of this objective is performed simply through the resolution of the Markovian process governing the stochastic evolution of the system considered.
Concerning the GA, one gene encodes the system configuration and one gene is used for the failure rate of each component, which are seven at most, so that the chromosome is made up by eight genes which give a complete description of a potential system solution to the optimization problem.

3.1. Results and discussion

Table 1 contains the parameters related to the system technical and economical specifications, whereas Table 2 contains the rules and the parameters for the GA implemented in order to solve the two-objectives optimization problem with the developed Fortran code MOGA. The values of the parameters were chosen based on experience and trial-and-error tuning, so as to achieve proper convergence.

The complete search space resulting from the failure rate ranges and the other system parameters is shown in Fig. 4, in which the values of the two objective functions for all possible solutions are represented.

The sharp changes in the shape of the search space are due to the configuration changes of the system nodes occurring when \( n \) in the 1-out-of-\( nG \) configuration varies.

The results of the GA optimization process are shown in Fig. 5. It can be seen how the GA efficiently identifies the Pareto optimal solutions, i.e. the nondominated solutions. The discontinuities in the Pareto surface are due to changes in the system configuration which
introduce jumps in the achievable values of the system reliability. For example, there are no nondominated solutions with reliability values in the range between 0.45 and 0.85.

Based on this information the decision maker can either impose a minimal reliability level for the plant as an a posteriori constraint, or he can decide to sacrifice part of the system revenues in favor of an increased system reliability. Actually, looking at the high reliability solutions, a natural option in our case would seem that of obtaining higher revenues with a little decrease in reliability. In any case, it is shown that adopting a multiobjective approach provides a wider information to the decision maker without introducing any ‘a priori’ arbitrariness which, instead, comes into play ‘a posteriori’, at the decision level.

4. Multiobjective optimization of a safety system inspection policy

Let us consider a standby safety system of a nuclear power plant (NPP) [5,25]. The system under consideration is the HPIS of a PWR. Fig. 6 shows a simplified schematics of a specific HPIS design. The system consists of three pumps and seven valves.

During normal reactor operation, one of the three charging pumps draws water from the volume control tank (VCT) in order to maintain the normal level of water in the primary reactor cooling system (RCS) and to provide a small high-pressure flow to the seals of the RCS pumps. Following a small loss of coolant accident (LOCA), the HPIS is required to supply a high pressure flow to the RCS. Moreover, the HPIS can be used to remove heat.

Fig. 6. The simplified HPIS system RWST = Radioactive Waste Storage Tank.
from the reactor core if the steam generators were completely unavailable. Under normal conditions the HPIS function is performed by injection through the valves \( V_3 \) and \( V_5 \) but, for redundancy, crossover valves \( V_4, V_6 \) and \( V_7 \) provide alternative flow paths if some failure were to occur in one of the nominal paths.

This stand-by safety system has to be inspected periodically to test its availability. The test interval (TI) specified by the technical specifications (TS) both for the pumps, and the valves is 2190 h. However, there are several restrictions on the maintenance procedures described in the TS, depending on reactor operations. For this study the following assumptions are made:

1. At least one of the flow paths must be open at all times.
2. If the component is found failed during surveillance and testing, it is returned to an as-good-as-new condition through corrective maintenance or replacement.
3. If the component is found to be operable during surveillance and testing it is returned to an as-good-as-new condition through restorative maintenance.
4. The process of inspection and testing requires a finite time; while the corrective maintenance (or replacement) requires an additional finite time, the restorative maintenance is supposed to be instantaneous.

Moreover, in this study the system components have been divided in three groups characterized by different test strategies. All the components belonging to a same group undergo testing with the same periodicity. The groups identified through the test period \( T_i \), \( i = 1, 2, 3 \), are:

\[ T^1 \rightarrow V_1, V_2 \]
\[ T^2 \rightarrow P_A, P_B, P_C, V_3, V_5 \]
\[ T^3 \rightarrow V_4, V_6, V_7 \]

Assuming a mission time of one year, the range of variability of the three TIs is [1,8760] h. Therefore, any solution to the optimization problem can be encoded using the following array of decision variables:

\[ x = \{ T^1, T^2, T^3 \}. \]

### 4.1. Problem formulation and objective functions

The goal is to optimize the effectiveness of the TIs of the HPIS with respect to three different criteria: (i) mean availability; (ii) cost; and (iii) workers’ time of exposure to radiation.

The TIs then represent the decision variables of the optimization problem and different choices of their values will lead to different performances with respect to the three above-mentioned objectives.

### 4.2. Mean unavailability

To compute the system unavailability we have developed the fault tree for the top event “no flow out of both injection paths A and B” (here not reported for brevity’s sake). The boolean reduction of the corresponding structure function has allowed us to determine the \( N \) system minimal cut sets (MCS). Then, the mean system unavailability \( \bar{U} \) can be expressed as [5]:

\[ \bar{U} = \sum_{j=1}^{N} \prod_{i=1}^{n_j} \bar{a}_i \]  

where \( N \) is the number of MCS, \( n_j \) is the number of basic events relevant to the \( j \)th minimal cut set and \( \bar{a}_i \) represents the mean unavailability associated with the \( i \)th component belonging to the \( j \)th MCS.

As for the mean unavailability \( \bar{u}_i \) of a generic individual component \( i \), several models have been proposed in the literature to account for the different contributions coming from failure on demand, human errors, maintenance etc. In this study the following model is assumed [5,21]:

\[ \bar{u}_i = \rho_i + \frac{1}{2} \lambda_i \tau + (\rho_i + \lambda_i \tau) \frac{d_i}{\tau} + \frac{t_i}{\tau} + \gamma_0 \]  

where \( \rho_i \) is the probability of failure on demand, \( \lambda_i \) the failure rate for the \( i \)th component, \( \tau \) the test interval for the \( i \)th component, \( t_i \) the mean downtime due to testing, \( d_i \) the mean downtime due to corrective maintenance and \( \gamma_0 \) the probability of human error. Eq. (9) is valid for \( \rho < 0.1 \) and \( \lambda \tau < 0.1 \) which are reasonable assumptions when considering safety systems.

Obviously, the adopted model of Eqs. (8) and (9) is a practical but simplified model: a more realistic approach would require the use of Monte Carlo simulation for the evaluation of the system unavailability [2,17].

### 4.3. Cost function

We assume that the cost objective \( C \) is made up of two major contributions:

1. \( C_{S&M} \) costs associated with surveillance and maintenance (S&M);
2. \( C_{accident} \) costs associated with consequences related to accidents possibly occurring at the plant, therefore

\[ C = C_{S&M} + C_{accident} \]  

For a given component \( i \) the S&M costs are computed on the basis of given yearly inspection (\( C_{ins} \)) and corrective maintenance (\( C_{m} \)) costs.

For a given mission time, \( T_M \), the number of inspections performed on component \( i \) are \( (T_M/\tau_i) \); of these, on average, a fraction equal to \( (\rho_i + \lambda_i \tau_i) \) demands also a corrective
maintenance action. Thus the surveillance and maintenance costs amount to:

$$C_{\text{S&M}} = \sum_{i=1}^{N_c} \left[ C_{\text{h},i} \left( \frac{T_i}{\tau_i} \right) t_i + C_{\text{hc},i} \left( \frac{T_i}{\tau_i} \right) d_i (\rho_i + \lambda_i \tau_i) \right]$$

(11)

As for what concerns the accident costs contribution, $C_{\text{accident}}$, this is intended to measure the costs associated to damages of accidents which are not mitigated due to the HPIS failing to intervene. A proper analysis of such costs implies that we account for the probability of the corresponding accident sequences. To this aim we have referred to a small LOCA event tree found in the literature [13] and here reported in Fig. 7. Actually, the HPIS plays an important role in many other accident sequences generating from other initiators such as intermediate LOCA, station blackout and turbine trip. In our example, for simplicity we consider only the contribution due to small LOCAs, recognizing that by so doing we significantly underestimate the accident cost contribution related to the HPIS. Table 3, also taken from [13], reports the characteristics of the plant damages states (PDS) resulting from the various small LOCA accident sequences, and the economic damages of the associated consequences. The accident sequences considered for the quantification of the accident costs are those which involve the failure of the HPIS (solid lines in Fig. 7), so that the possible PDS are PDS1 and PDS3. Thus:

$$\begin{align*}
C_{\text{accident}} &= C_1 + C_3 \\
C_1 &= P(EI) (1 - U_{RT}) \bar{u} \{ U_{\text{LPIS}} + (1 - U_{\text{LPIS}}) U_{\text{SDC}} U_{\text{MSHR}} \} C_{\text{PDS1}} \\
C_3 &= P(EI) (1 - U_{RT}) \bar{u} (1 - U_{\text{LPIS}}) \{ U_{\text{SDC}} (1 - U_{\text{MSHR}}) + (1 - U_{\text{SDC}}) \} C_{\text{PDS3}}
\end{align*}$$

(12)
where \( C_1 \) and \( C_3 \) are the total costs associated with accident sequences leading to damaging states 1 and 3, respectively. These costs depend on the initiating event frequency and on the unavailability values of the safety systems which ought to intervene along the various sequences: these values are taken from the literature [13,27] for all systems except for the SDC and MSHR, which were not available and were arbitrarily assumed of the same order of magnitude of the other safety systems, and for the HPIS for which the unavailability is calculated from Eqs. (8) and (9) and it depends on the TIs of the components. Finally, for the values of \( C_{\text{PDS}_1} \) and \( C_{\text{PDS}_3} \), the accident costs for PDS1 and PDS3, respectively, we adopted the mean value of the uniform distributions given in Table 3. Table 4 summarizes the input data.

### 4.4. Exposure time

During testing operations, the technicians may be subjected to radiation exposure. With reference to the ICRP recommendation no. 60 [28], based on the well-known ALARA (As Low As Reasonably Achievable) and limit-dose-principles, the dose received by workers should be minimized. Assuming a constant exposure rate, the minimization of the dose is equivalent to that of the exposure time, so that the third objective function of our optimization problem can be assumed to be

\[
T_{\text{exp}} = \sum_{i=1}^{N_C} \left( \frac{T_M}{\tau_i} \right) t_i + \left( \frac{T_M}{\tau_i} \right) d_i \left( \rho_i + \lambda_i \tau_i \right)
\]  

(13)

Table 4
Frequencies associated to failures of safety systems in the ET and costs associated to PDSs of interest for sequences involving the HPIS failure

<table>
<thead>
<tr>
<th>PDS</th>
<th>Plant damage state</th>
<th>Health risk</th>
<th>Investment risk</th>
<th>Total risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Severe core damage or core melt; significant radioisotope release to containment</td>
<td>( 5 \times 10^4 ) person rem per event (( \sim U(5,50) )) M$/ event</td>
<td>( \sim U(1162, 3136) ) M$/ event median:</td>
<td>( \sim U(1167, 3186) ) M$/ event median:</td>
</tr>
<tr>
<td>2</td>
<td>Small LOCA leading to containment cleanup, valve and vessel repair to containment</td>
<td>( 3.8 \times 10^7 ) person rem per event (( \sim U(3.8,38) )) M$/ event</td>
<td>( \sim U(329, 924) ) M$/ event median:</td>
<td>( \sim U(332.8, 962) ) M$/ event median:</td>
</tr>
<tr>
<td>3</td>
<td>Possible damage to steam generator; minor containment cleanup and equipment checkout</td>
<td>( \sim U(32, 243) ) M$/ event median:</td>
<td>( 626.5 ) M$/ event median:</td>
<td>( 647.4 ) M$/ event median:</td>
</tr>
<tr>
<td>4</td>
<td>Possible primary system water loss; little or no spill into containment; no core or equipment damage</td>
<td>( \sim U(1, 6) ) M$/ event median:</td>
<td>( 137.5 ) M$/ event median:</td>
<td>( 137.5 ) M$/ event median:</td>
</tr>
</tbody>
</table>

with the same meaning of the symbols explained in the previous sections.

Note that, for simplicity, in computing the radiation exposure time with Eq. (13) we neglect the fact, often verified in practice, that work management procedures in nuclear power plants are such that exposure times associated with performing surveillance tests or corrective maintenance on a component are larger than its respective mean downtimes.

Expression (13) is similar to that of Eq. (11) for the surveillance and maintenance costs, \( C_{\text{S&M}} \). However, the presence of the accident contribution in the cost objective function is such that exposure time and cost are generally two distinct objectives to be optimized separately.

An analysis of the three objective functions hereby defined shows that they all share some common contributions but present some conflicting ones as well. For example, the cost function has a contribution relating to the unavailability of the HPIS due to economic damages of occurring accidents and a contribution associated to the time of surveillance and maintenance (and thus of exposition) due to the costs of such operations. On the other hand, the surveillance and maintenance time influences also the mean system unavailability, through the downtimes of the inspected components.

### 4.5. Genetic coding

The goal of the work is that of utilizing the multiobjective GA optimization procedure to determine the optimal values of inspection intervals, \( t^*_i \), \( i = 1, 2, 3 \) for the three groups of

Table 5
Genetic algorithm parameters and rules

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of chromosomes (population size, ( N_p ))</td>
<td>100</td>
</tr>
<tr>
<td>Number of generations (termination criterion)</td>
<td>500</td>
</tr>
<tr>
<td>Selection</td>
<td>Standard roulette</td>
</tr>
<tr>
<td>Replacement</td>
<td>Weakest</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.005</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>1</td>
</tr>
<tr>
<td>Number of nondominated chromosomes in the archive</td>
<td>400</td>
</tr>
</tbody>
</table>
components identified in the HPIS, which maximize separately the three objective functions: average availability ($A = 1 - U$), reciprocal of the cost ($1/C$) and reciprocal of the exposure time ($1/T_{exp}$). The decision variables of the optimization are then the three TIs $T_i$, $i = 1, 2, 3$. Such TIs are assumed to vary in the range $[1,8760]$ h so that at least one inspection on each component is carried out in one year. Each of the variables is coded by one 10-bit gene in the chromosome. The data relevant for the multiobjective GA procedure contained in Table 5 have been selected after appropriate tuning and constitute the input to the MOGA code.

### 4.6. Single objective optimization for model validation

A validation of the model has first been sought by comparison of the results of a constrained single objective optimization with results of literature [5]. The optimization considers two different cases: (i) optimization of the system mean unavailability with a constraint on cost; (ii) vice versa. No consideration is given to the exposure time, to the contribution of human error to component unavailability and to the accident costs. Table 6 contains the numerical values of the relevant parameters taken from Ref. [5] except for the human error frequency $\gamma_0$ which is taken from Ref. [26]. The constraints on costs and mean unavailability for the two cases have been computed taking as values for the TIs those provided by the TSs. In other words, acceptable TIs are only those which produce values of the mean unavailability and cost not larger than those obtained with the TI values of the TSs.

The results obtained are compared with those of literature in Table 7. The small differences in the optimal values are due to a slight difference in the encoding procedure adopted by us with respect to that adopted in the literature work. Finally, as a general result we note the convenience of changing the values of the TIs with respect to those provided by the TS: a convenience resulting in both an increased system availability and a reduced cost. In particular, an important result in agreement with that of Ref. [5] is that the TS prescribe TIs for the components of group 1 which are significantly longer than those resulting from the GA optimization. This relevant difference might be due to the fact that the inspection time of a component depends not only on the component itself but also on the system configuration in which it operates.

### 4.7. Multiobjective optimization of the HPIS

After validating the model, we can now perform an actual multiojective optimization without resorting to any constraint but rather considering the goodness of the solutions with respect to the objective functions taken separately. Fig. 8 shows the results obtained through the GA procedure for maximizing the three objective functions of mean unavailability, reciprocal of costs and reciprocal of exposure time, simultaneously. In the figure, we report the values of the objective functions in correspondence of all the nondominated solutions (triplets of TIs) contained in the archive at convergence. These results certainly constitute a more informative set which the designer can handle for a more informed decision, free of a priori constraints or arbitrary weights.

It is clear that there exists a linear relationship between cost and exposure time. This is due to the fact that the safety systems failure frequencies and accidental costs are such that the contribution to cost due to accidents is negligible compared to that of surveillance and maintenance, which, in turn, is proportional to the surveillance and maintenance time and, thus, to exposure time. As a consequence of this linear relation, the results obtained would not change, as it was verified, if we were to perform a two-objective optimization in which the two objectives were the mean availability and one between the reciprocals of cost and exposure time. Along these lines, an optimization considering the reciprocal of cost and exposure time as the only objective

<table>
<thead>
<tr>
<th>Table 6</th>
<th>System data from [5] except for the human error frequency $\gamma_0$ which is taken from [26]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>$\lambda$ ($\text{h}^{-1}$)</td>
</tr>
<tr>
<td>Pump (P)</td>
<td>3.89</td>
</tr>
<tr>
<td>Valve (V)</td>
<td>5.83</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7</th>
<th>Results of the constrained optimizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
<td>Initial values (TS)</td>
</tr>
<tr>
<td>$T_1^*(\text{h})$</td>
<td>2184</td>
</tr>
<tr>
<td>$T_2^*(\text{h})$</td>
<td>2184</td>
</tr>
<tr>
<td>$T_3^*(\text{h})$</td>
<td>2184</td>
</tr>
<tr>
<td>$U$</td>
<td>$7.462 \times 10^{-5}$ (constraint)</td>
</tr>
<tr>
<td>$C$ ($\text{s}$)</td>
<td>1438.82 $\text{h}$ (constraint)</td>
</tr>
</tbody>
</table>
functions would result in a single-point solution, corresponding to the lowest cost and exposure time.

Finally, the TIs in the GAs archive (here not reported for brevity) give an indication that the HPIS can indeed be made more available, on average, by increasing the frequency of the inspections but, as reasonable, this leads to large inspectors’ exposure times and also renders the system more expensive. A thorough analysis of the results in the archive also shows that $T^1$ is somewhat dominant, as expected since it governs the inspections on the two valves $V_1$ and $V_2$ which constitute the most critical MCS of the system.

As mentioned before, if we were to account for the accident cost contributions due to all initiating events which demand the intervention of the HPIS we would find much higher values which could greatly influence the results of the multiobjective problem. An investigation is then in order to see what happens when the contribution to cost due to accidents becomes significant. To this aim, for simplicity, we unrealistically modified the initiating event frequencies of SDC and MSHR increasing them significantly as in Table 8. Fig. 9 shows the results of the multiobjective optimization by GAs.

In this case we loose the linearity relationship between cost and exposure time, and we obtain two distinct behaviors in the solution space. We have a region up front, characterized by lower values of the mean availability and higher values of the reciprocals of the exposure time and of the costs, in which the accident contribution to costs is dominating, and another region, characterized by larger values of mean availability and lower values of the reciprocals of the exposure time and costs, in which surveillance and maintenance costs take over. These two behaviors give rise to the curvature of the projection on the reciprocal-of-cost $1/C$ vs. reciprocal-of-exposure time $1/T_{exp}$ plane. Increasing values of the reciprocal of exposure time, i.e. decreasing values of the exposure time, are due to fewer inspections which allow some savings in surveillance expenditures; on the other hand, we also have a decrease in the mean availability of the system which is less properly maintained and undergoes more frequent stochastic failures, with a consequent increase in the accident costs and thus a decrease in the reciprocal of cost. Such conflicting situation between increasing accident costs and decreasing inspection costs lead to a maximum in the reciprocal of the costs objective function when inspection costs are at their lowest and accident costs are about to dominate.

### 5. Conclusions

Optimal plant design and logistic maintenance must achieve several targets and satisfy several requirements, which are often in conflict. The usual approach of tackling these problems by focusing on a single objective which is a weighed combination of some of the relevant objectives,
keeping into account the remaining objectives by imposing constraints on them, can be greatly improved by a more informative approach in which every objective is considered separately without any a priori imposition as to the relative importance of each objective and without any arbitrary constraints.

In this paper we proposed to perform such multiobjective optimization by means of GAs. The GA adopted considers a population of chromosomes, each one encoding a different solution to the optimization problem. For a given solution, there are more than one objective to be evaluated so that the performance of any given candidate solution is evaluated introducing the concepts of Pareto optimality and dominance.

The proposed multiobjective GA approach has been applied first to a simple design problem aiming at identifying the optimal system configuration and components with respect to a reliability and a monetary cost objectives.

The procedure was then applied to a realistically complicated system, taken from literature, for determining the optimal TIs of the components of a safety system in a nuclear power plant. The optimization performed with respect to availability, economic and workers’ safety objectives has shown the potentials of the approach and the benefits which can derive from a more informative multiobjective framework.

As a final remark we underline the fact that although more informative, Pareto optimality does not solve the decision problem. The decision maker is provided the whole spectrum of nondominated alternatives, and their performances with respect to the objectives, and he or she must ultimately select the preferred one according to his or her preference values.

Thus, the closure of the problem must still rely on techniques of decision making such as utility theory, multi-attribute value theory or fuzzy decision making, to name a few.

Acknowledgements

The authors are indebted to the two anonymous reviewers for their deep comments and suggestions which have allowed for a significant improvement of the paper.

References