Application of genetic algorithm for optimization the safety system of the nuclear reactor

Mohamed El-Sayed Wahed,
Wesam Z. Ibrahim,
Ahmed M. Effat

Abstract. 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. Multi-objective optimization is a powerful tool for resolving conflicting objectives in engineering design and numerous other fields. One approach to solve multi-objective optimization problems is the non-dominated sorting genetic algorithm (NSGA). Genetic algorithm (GA) was applied in regarding the choice of the time intervals for the periodic testing of the components of the chimney water injection system (CWIS) of the 22 MW open pool multipurpose reactor (MPR), ETRR-2, at the Egyptian Atomic Energy Authority, has been used as a case study.

Key words: genetic algorithm • non-dominated sorting • chimney water injection system (CWIS) • Egypt nuclear reactor

M. El-Sayed Wahed
Faculty of Science,
Mathematics Department,
Zagazic University,
Egypt

W. Z. Ibrahim✉, A. M. Effat
Atomic Energy Authority,
54-amr bin yasser-11351-Helioples, Cairo, Egypt,
Tel.: +20 114 641995, Fax: +20 282 7236,
E-mail: zakariawesam@yahoo.com

Received: 21 February 2008
Accepted: 20 October 2008

Introduction

Safety is one of the main concerns about the nuclear reactors design to be licensed. The reactor design should demonstrate that the technical objectives for the reactor safety are fulfilled. These technical objectives are to ensure the general prevention of accidents with high confidence margin; to ensure that, for all accidents taken into account in the design, even those of very low probability, the radiological consequences, if any, would be minor, and to ensure by prevention, protection, and mitigation measures that severe accidents with significant radiological consequences are extremely unlikely. However, 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 [8, 11].

In this paper we present the genetic algorithms approach to multi-objective optimization and apply it within the reliability/availability analysis framework to choice the time intervals for the periodic testing of the components of the chimney water injection system of Egypt Second Research Reactor (ETRR-2).

The power of genetic algorithms (GA's) comes from the fact that the technique is robust and can deal successfully with a wide range of difficult problems. GA's are not guaranteed to find the global optimum solution to a problem, but they are generally good at finding "acceptably good" solutions to problems "acceptably quickly". Where specialized techniques exist for solving particular problems, they are likely to outperform GA's in both speed and accuracy of the final result. Even where existing techniques work well, improvements have been made by hybridizing them with a GA. The basic mechanism of a GA is so robust that, within fairly wide margins, parameter settings are not critical.

A problem with GA's is that the genes from a few comparatively highly fit (but not optimal) individuals may rapidly come to dominate the population, causing it to converge on a local maximum. Once the population has converged, the ability of the GA to continue to search for better solutions is effectively eliminated: crossover of almost identical chromosomes produces little that is new. Only mutation remains to explore entirely new ground, and this simply performs a slow, random search.

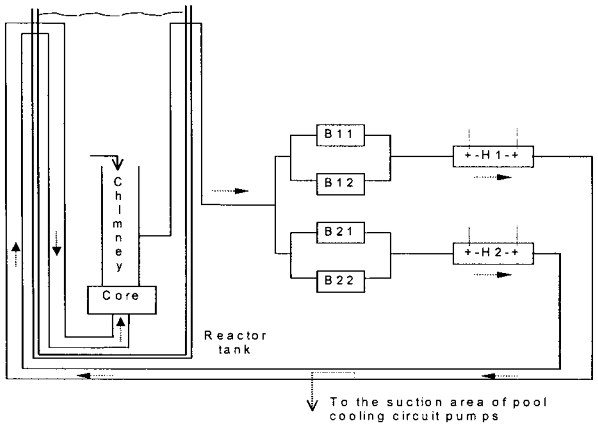

Fig. 1. Primary cooling system.

Chimney water injection safety system (CWIS)

The Egypt Second Research Reactor, ETTR-2, is an open pool type research reactor. The reactor nominal power is 22 MW with a maximum thermal neutron flux of 2.7×10^{14} n/cm$^{-2} \cdot$ s^{-1}. Several experimental devices are installed at the reactor so that it can be used for radioisotope production, basic and applied research in science and engineering, material testing, neutron radiography, neutron activation analysis, and for training [11]. The reactor coolant and moderator is light water and the reflector is beryllium. The reactor uses U_3O_8-Al plate type fuel with Al cladding and 19.75% enrichment [9].

The CWIS is responsible for the injection of water to the reactor chimney in order to maintain the core covered with water in case of an eventual drop of the reactor pool water below the chimney upper edge. This system is triggered by a signal of low water level in the pool. The system has been designed to maintain the chimney filled with water during at least 24 h, thus compensating losses due to residual decay heat (after the reactor shutdown). The system consists of four identical non-redundant tanks (each holds 25% of the required water) and their discharge lines. The lines from the four tanks are combined into a common line that passes through an orifice plate. The flow finally passes through two redundant solenoid operated valves that are used to trigger the system [4].

The CWIS together with the inherent core heat transfer characteristics are capable of keeping all core temperatures within specified safety limits during all shutdown conditions, including situations created by a breach on the reactor cooling system boundary.

The primary system, whose diagram is shown in Fig. 1, consists of a closed circuit through which the coolant is made to circulate driven by two centrifugal pumps (B), making it pass through the core in an upward flow and then through heat exchangers (H) where heat generated in the core is eventually transferred to the secondary system [7, 9].

The system mean unavailability is estimated to be 6.0×10^{-2}.

The genetic algorithm

The genetic algorithm (GA) is a method for solving optimization problems that is based on natural selection, the process that drives biological evolution. GA repeatedly modifies a population of individual solutions. At each step, GA selects individuals at random from the current population to be parents and uses them produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. GA uses three main types of rules at each step to create the next generation from the current population: selection rules select the individuals, called parents, that contribute to the population at the next generation. Crossover rules combine two parents to form children for the next generation, mutation rules apply random changes to individual parents to form children computation, selects the next population by computations that involve random choices [14].

The simple genetic algorithm follows the structure depicted in Fig. 2. Each of these operations will be described in the following subsections [3].

Selection plays a central role in genetic algorithms determining how individuals compete for gene survival. Selection weeds out the bad solutions and keeps the good ones. This can be done by "fitness propor-

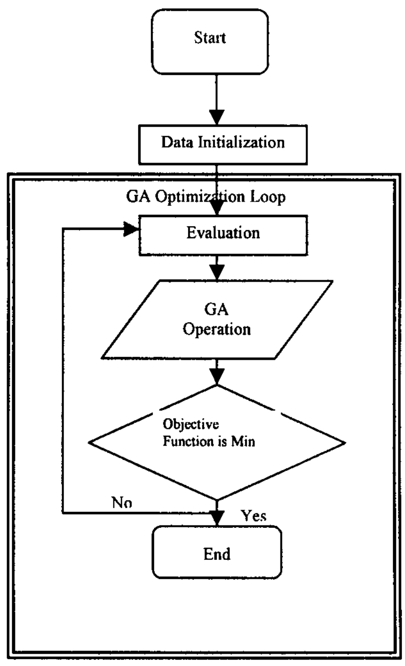

Fig. 2. Simple genetic algorithm.

tional selection" that assigns a selection probability in proportion to the fitness of the given individual. This, however, tends to be sub-optimal as the effective selection strength can be changed by adding an offset. More commonly used is "tournament selection", where a number of randomly picked individuals are compared to each other. The goal of GA is essentially to find a set of parameters that maximize or minimize the output of a function or fitness. The individual with the best fitness is then selected to be a part of the next generation. Selection in GA's is usually done on the whole original population and usually repeated for all individuals in the population [1].

The simple genetic algorithm can be mathematically described as:

– For each chromosome S_i, $i = 1,2,\dots$ *pop_size* compute the fitness value

$$eval(S_i) = f(S_i) \tag{1}$$

– Compute the total fitness of the population

$$F = \sum_{i=1}^{pop\_size} eval(S_i) \tag{2}$$

– Compute the probability of a selection p_i for each chromosome S_i

$$p_i = eval(S_i) / F \tag{3}$$

– Compute the cumulative probability of a selection q_i for each chromosome S_i

$$q_i = \sum_{i=1}^{i} p_j \tag{4}$$

– Generate a random number $r \in [0,1]$.

– If $r < q_1$ then select the first chromosome S_1, otherwise select the i-th chromosome S_i, $i = 1,2,\dots$ *pop_size* such that $q_{i-1} < r < q_1$.

Recombination of individuals is done to investigate the performance of new individuals that resemble the exiting ones. This is done on the genotype level of the individuals and leads to the construction of new intermediate solutions. The notion of generations arises as parent individuals recombine their genes to create offspring. Recombination is often done by crossover [13]. In the crossover phase, all of the chromosomes (except for the elite chromosome) are paired up, and with a crossover probability, p_c, they are crossed over. The crossover is accomplished by randomly choosing a site along the length of the chromosome, and exchanging the genes of the two chromosomes for each gene past this crossover site [14]. The crossover operation proceeds in the following manner:

– For each chromosome S_i in the population.

– Generate a random number $r \in [0,1]$.

– If $r < p_c$ then select the given chromosome for crossover.

– Compute select chromosomes randomly.

– For each pair of coupled chromosomes, generate a random integer number $pos \in [1,\dots m-1]$, m is a number of bits in each chromosome, the number *pos* indicates the position of crossing point the following two chromosomes are crossed over as follows:

$$\begin{matrix} (b_1 b_2 \dots b_{pos+1} b_m) \quad \text{and} \\ (c_1 c_2 \dots c_{pos+1} c_m) \end{matrix} \tag{5}$$

Are replaced by a pair of their offspring

$$\begin{matrix} (b_1 b_2 \dots b_{pos} c_{pos+1} c_m) \quad \text{and} \\ (c_1 c_2 \dots c_{pos} b_{pos+1} b_m) \end{matrix} \tag{6}$$

Mutation. After the crossover, for each of the genes of the chromosomes (except for the elite chromosome), the gene will be mutated to any one of the codes with a mutation probability, p_m. With the crossover and mutations completed, the chromosomes are once again evaluated for another round of selection and reproduction. Even if most of the search is being performed by recombination, mutation can be vital to provide the diversity which recombination needs. The mutation operation proceeds in the following manner:

For each chromosome in the population apply:

– Generate a random number $r \in [0,1]$.

– If $r < p_m$ mute this bit by change its value from 0 to 1 or *vice versa*.

Genetic algorithm for a multi-objective optimization problem

Multi-objective genetic algorithms usually try to find all the non-dominated solutions of an optimization problem with multiple-objectives. Let us consider the following multi-objective optimization problem with n objectives:

$$\text{Maximize} \quad f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots f_n(\boldsymbol{x}) \tag{7}$$

where x is a vector to be determined, and $f_1(x), f_2(x), \dots, f_n(x)$ are functions to be maximized. If a feasible solution is not dominated by any other feasible n objective solutions of the multi-objective optimization problem, that solution is said to be a non-dominated solution. When the following inequalities hold between two solutions y, it x and is said that the solution x is dominated by the solution [3]:

$$\forall i: f_i(x) \le f_i(y) \ \text{and} \ \exists j: f_j(x) < f_j(y) \tag{8}$$

The aim of our multi-objective algorithms is not to determine a single final solution but to find all the non-dominated solutions of the multi-objective optimization problem in Eq. (3). Since it is difficult to choose a single solution for a multi-objective optimization problem without iterative interaction with the decision maker, one general approach is to show the set of non-dominated solutions to the decision maker. Then, one of the non-dominated solutions can be chosen depending on the preference of the decision maker. Since Deb's work [3], extensions of GA's to multi-objective optimization problems have been proposed in several manners. Fonseca and Fleming [5], have published an excellent survey on GA's for multi-objective optimization. Almost all approaches which have already been proposed can be categorized into one of two classes: a "population-based non Pareto approach" or a "Pareto-based approach" by their selection schemes [12].

In this Section we present the extension of the genetic algorithm approach to multi-objective problems [1, 12]. 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 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.

During the optimization search, an archive of a given number of non-dominated solutions representing the dynamic Pareto optimality surface is recorded and updated. At the end of each generation, non-dominated solutions in the current population are compared with those already stored in the archive and the following archival rules are implemented [3]:

1. If the new solution dominates over the existing members of the archive, they 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.

The setup of an archive of non-dominated solutions can also be exploited by introducing an elitist parents' selection procedure which should, in principle, be more efficient. Every solution in the archive is chosen once as a parent in each generation. This should guarantee a better propagation of the genetic code of non-dominated 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 [13].

The basic idea behind non-dominated sorting genetic algorithm (NSGA) is the ranking process executed before the selection operation, this approach is proposed by Srinivas and Deb [14]. This process identifies non-dominated solutions in the population, at each generation, to form non-dominated fronts [8], based on the concept of non-dominance criterion. After this, the selection, crossover, and mutation usual operators are performed. In the ranking procedure, the non-dominated individuals in the current population are first identified. Then, these non-dominated individuals are shared by dividing the dummy fitness value of an individual by a quantity called niche count, which is proportional to the number of individuals around it. In order to maintain diversity in the population, a sharing method is then applied. Afterwards, the individuals of the first front are ignored temporarily and the rest of population is processed in the same way to identify individuals for the second non-dominated front [6]. A dummy fitness value that is kept smaller than the minimum shared dummy fitness of the previous front is assigned to all individuals belonging to the new front. This process repeated until the whole population is classified into non-dominated fronts. Since the non-dominated fronts are defined, the population is then reproduced according to the dummy fitness values [2]. The sharing procedure used in this method can be summarized as follows [14]:

Given a set of n_k solutions in the k-th non-dominated each having a dummy fitness value f_k. The sharing procedure is preformed for each solution $i = 1,2,\ldots k$ as follows:

- Step 1. Compute the Euclidean distance measure with another solution j in the k non-dominated as:

(9) $$d_y = \sqrt{\sum_{p=1}^{p}\left(\frac{x_p^i - x_p^j}{x_p^u - x_p^i}\right)^2}$$

where p is the number of variables in the problem is; x_p^i, x_p^u is the lower and upper bound of the variables x_p.

- Step 2. The distance d_{ij} is computed with a prespecified parameter ∂_{share}

(10) $$sh(d_y) = \begin{cases} 1-\left(\dfrac{d_{ij}}{\partial_{\text{share}}}\right), & \text{if } d_y \le \partial_{\text{share}} \\ 0 & \text{otherwise} \end{cases}$$

The approximate value of ∂_{share} is:

(11) $$\partial_{\text{share}} \approx \frac{0.5}{\sqrt[p]{q}}$$

where q is the desired number of distinct Pareto optimal solution.

- Step 3. Increment j if $j \le n_k$ go to Step 1, if $j \succ n_k$ calculate nich count

(12) $$m_i = \sum_{j=1}^{n_k} sh(d_y)$$

- Step 4. Shared the dummy fitness f_k of the i^{ih} in the k^{ih} non-dominated as follows:

(13) Shared fitness $$f_i = \frac{f_k}{m_i}$$

The main advantage is that it can deal with any number of objectives, the sharing procedure is preformed in the parameter value space with ensues a good distribution of the individuals.

Optimization of the safety system

Let us consider the chimney water injection system (CWIS) of the Egypt Second Research Reactor (ESRR-2) [9, 11]. Figure 3 shows the simplified scheme of a specific CWIS design [3]. 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

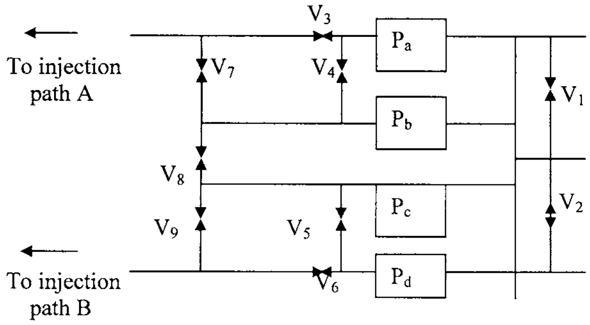

Fig. 3. Simplified CWIS system.

of coolant accident (LOCA), the CWIS is required to supply a high pressure flow to the RCS. Moreover, the CWIS can be used to remove heat from the reactor core if the steam generators were completely unavailable. Under normal conditions, the CWIS function is performed by injection through the valves V_3 and V_6 but, for redundancy, crossover valves V_4, V_5, V_7, V_8 and V_9 provide alternative flow paths if some failure were to occur in one of the nominal paths [12].

This stand-by safety system has to be inspected periodically to test its availability. The test interval (T_I) specified by the technical specifications (T_S) both for the pumps, and the valves is 30 days "720 h". In this study the system components have been divided into three groups characterized by different test strategies. All the components belonging to the same group undergo testing with the same periodicity, Fig. 3.

Assuming a mission time of one year, the range of variability of the three $T_{I's}$ is [0, 8640] hours. Therefore, any solution to the optimization problem can be encoded using the following array of decision variables: $x = \{T_1, T_2, T_3\}$.

The goal of the work is to optimize the effectiveness of the $T_{I's}$ of the HPIS with respect to three different criteria: i) mean availability, ii) cost, and iii) workers' time of exposure to radiation. 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". The Boolean reduction of the corresponding structure function has allowed us to determine the system minimal cut sets (MCS) and from these we can compute the mean system unavailability f_1 as a function of the elementary unavailabilities of the components in the MCS. As for the mean unavailability f_1 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 [11, 15]:

$$f_1 = x_i + \frac{1}{2} y_i \tau_i + (x_i + y_i \tau_i) \frac{d_i}{\tau_i} + \frac{t_i}{\tau i} + \gamma_0 \tag{14}$$

where: x_i is the probability of failure on demand; y_i is the failure rate for the i-th component; τ_i is the test interval for the i-th component; t_i is the mean downtime due to testing; d_i is the mean downtime due to corrective maintenance and γ_0 is the probability of human error.

Equation (14) is valid for $x < 0.1$ and $y\tau < 0.1$ which are reasonable assumptions when considering safety systems. The relevant parameters' values are taken from [11] and [7].

For the cost objective C, we assume that it is the sum of two major contributions:

1. f_2 = costs associated with surveillance and maintenance.
2. $C_{accident}$ = costs associated with consequences related to accidents possibly occurring at the plant.

For a given mission time, T_M, the surveillance and maintenance costs amount to:

$$f_2 = \sum_{i=1}^{N_c} \left[C_{ht,i} \cdot \left(\frac{T_M}{\tau_i} \right) \cdot t_i + C_{hc,i} \cdot \left(\frac{T_M}{\tau_i} \right) \cdot d_i \cdot (x_i + y_i \tau_i) \right] \tag{15}$$

As concerns the accident cost contribution, $C_{accident}$, this is intended to measure the costs associated to damages of accidents which are not mitigated due to the CWIS failing to intervene. To this aim we have referred to a small LOCA event tree found in the literature [7]. Actually, the CWIS plays an important role in many other accident sequences generating from other initiators such as intermediate LOCA, station blackout, turbine trip, etc. In our example, for simplicity we consider only the contribution due to small LOCA's, recognizing that by so doing we significantly underestimate the accident cost contribution related to the CWIS. The characteristics of the plant damage states (PDS), resulting from the various small LOCA accident sequences and the economic damages of the associated consequences, were also taken from [7]. The accident sequences considered for the quantification of the accident costs are those which involve the failure of the HPIS.

These costs obviously 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 [7, 10] for all systems except for the SDC and MSHR, and for the HPIS for which the unavailability is calculated as above explained and which depends on the test intervals of the components. Finally, for the accident costs associated to the relevant plant damage states we adopted the mean value of the uniform distributions given in Ref. [7].

During testing operations, the technicians may be subjected to radiation exposure. With reference to the SAR recommendation [7, 9], 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

$$f_3 = \sum_{i=1}^{N_c} \left[\left(\frac{T_M}{\tau_i} \right) \cdot t_i + \left(\frac{T_M}{\tau_i} \right) \cdot d_i \cdot (x_i + y_i \tau_i) \right] \tag{16}$$

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

Equation (16) is similar to that of Eq. (15) for the surveillance and maintenance costs.

The genetic parameters used are:

(17) $N_{ger} = 100$, $N_{pop} = 100$, Mutation Pr $ob. = 0.05$, Mission time $(T_M) = 8760$.

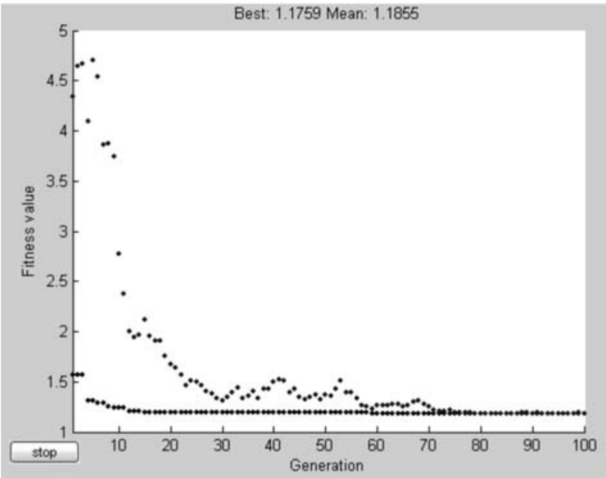

Fig. 4. Multi-objective optimization for the three functions.

The result is shown in Fig. 4 which shows the best fitness function for all of them. After terminated optimization a maximum number of generations exceeded the three functions together. We found that:

– the best function value for the first function found was 1.1759,
– the best function value for the second function found was 4.6378,
– the best function value for the third function found was 49.826

at the point value 0.07959, 0.08025, 0.98274, 0.04148, 0.09098.

It is clear that there exist a linear relationship between cost and exposure time. This is due to the fact that during failure of safety systems, frequencies and accidental costs are such that the contribution to cost due to accidents is negligible compared to that of surveillance and maintenance. So, we can see that, the test intervals in the genetic algorithm's archive give an indication that the CWIS can made more available, on average, by increasing the frequency of the inspections but, as reasonable, this leads to large exposure times of inspectors and also renders the system more expensive.

Conclusion

Engineering design is clearly about making many decisions often under uncertainty and with multiple conflicting criteria. In this paper, the design problem is reformulated as a multi-objective optimization problem. The Egypt Second Research Reactor, ETRR-2, has been used as a case study. We present an approach to optimization the components of the CWIS of the 22 MW open pool multipurpose reactor (MPR) at the Egyptian Atomic Energy Authority, using a non-dominated sorting genetic algorithm. The proposed multi-objective genetic algorithm approach has been applied for determining the optimal test intervals of the components of a safety system in a nuclear reactor. The optimization performed with respect to availability, economic and works' safety objectives has shown the potentials of the approach and the benefits which can derive from a more informative multi-objective framework.

References

1. Abd el-Razek ID (2000) The development of the MPR project. In: Proc of the 5th Arab Conf in the Peaceful Uses of Atomic Energy, November 2000, Lebanon, I:41–66
2. Alexandre A, de Vasconcelos A (2002) Multi-objective algorithms applied to solve optimization problems. IEEE T Magn 38:152–161
3. Andersson J, Pohl J, Krus P (1998) Design of objective functions for optimization of multi-domain systems. ASME Annual Winter Meeting, June 1998, FPST Division, Anaheim, California, USA
4. Cantoni M, Marseguerra M, Zio E (2002) Genetic algorithms and Monte Carlo simulation for optimal plant design. Reliab Eng Sys Safety 68:29–38
5. Fonesca C, Fleming P (1993) Proc of the 5th Int Conf on Genetic Algorithms, 4:416–423
6. Furdu I, Patrut B (2006) Genetic algorithm for ordered decision diagrams optimization. In: Proc of the 7th Joint Conf on Mathematics and Computer Science, July 3–6, 2008, Cluj, Romania, pp 437–444
7. Habib A, Ashry H, Shokr A (2005) Improving reactor safety systems using component redundancy allocation technique. Nukleonika 50;3:105–112
8. Holland JH (1975) Adaptation in natural and artificial system. University of Michigan Press, Ann Arbor, MI
9. INVAP-AEA (1999) Final safety analysis report of ETRR-2. Document #0767-5325-3IBLI-001-1A
10. Marseguerra M, Zio E (2002) Optimizing maintenance and repair policies via a combination of genetic algorithms and Monte Carlo simulation. Reliab Eng Sys Safety 35:69–83
11. Martorell S, Carlos S, Sanchez A, Serradell V (2000) Constrained optimization of test intervals using a steady-state genetic algorithm. Reliab Eng Sys Safety 23:215–232
12. Parks GT (1994) Multi-objective pressurized water reactor reload core design using genetic algorithm search. Nucl Sci Eng 24:178–187
13. Shaaban N, Takahashi H (2006) An overview of genetic algorithm. J Nucl Sci Technol 43:816–818
14. Srinivas N, Deb K (2004) Multi-objective optimization using non-dominated sorting in genetic algorithms. J Evolutionary Computation 2;3:221–248
15. Yang JE, Hwang MJ, Sung TY, Jin Y (1999) Application of genetic algorithm for reliability allocation in nuclear power plants. Reliab Eng Sys Safety 12:229–238