Available online at www.sciencedirect.com

ScienceDirect

Nuclear Science and Techniques 19 (2008) 314-320

NUCLEAR SCIENCE AND TECHNIQUES

Finding an optimization of the plate element of Egyptian research reactor using genetic algorithm

WAHED Mohamed[1] IBRAHIM Wesam[2,*] EFFAT Ahmed[2]

[1] *Mathematics Department, Faculty of Science, Zagazic University, Egypt*

[2] *Atomic Energy Authority, Cairo, Egypt*

Abstract The second Egyptian research reactor ET-RR-2 went critical on the 27th of November 1997. The National Center of Nuclear Safety and Radiation Control (NCNSRC) has the responsibility of the evaluation and assessment of the safety of this reactor. The purpose of this paper is to present an approach to optimization of the fuel element plate. For an efficient search through the solution space we use a multi objective 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. The aim of this paper is to propose a new approach for optimizing the fuel element plate in the reactor. The fuel element plate is designed with a view to improve reliability and lifetime and it is one of the most important elements during the shut down. In this present paper, we present a conceptual design approach for fuel element plate, in conjunction with a genetic algorithm to obtain a fuel plate that maximizes a fitness value to optimize the safety design of the fuel plate.

Key words Genetic algorithm, Non-dominated sorting, Fuel element plate, Egypt nuclear reactor

CLC numbers TL411, TL361

1 Introduction

Genetic Algorithms (GA's)[1,2] are adaptive systems inspired by natural evolution. The Standard GA (SGA) randomly creates an initial population of solutions, also called Chromosomes. Crossover operator recombines these solutions over a certain number of generations until a stop criterion is reached. The Chromosomes to recombine parents are selected according to their fitness values: better solutions have larger probability to be chosen to cross with other solutions and generate offspring—children—that share the genetic material from both parents. Mutations may occur with very low rate. This reproduction method is called *random mating*[1,3]. In random mating, an individual mates with any other regardless of its parenthood or likeness.

Additionally, there are many different types of multi objective genetic algorithms. Some multi-objective genetic algorithms can be found, for example, in Refs.[4,5]. Most multi objective genetic algorithms use either the selection mechanism or some sort of Pareto-based ranking to produce non-dominated solutions. In the proposed method, the ranking scheme presented by Fonseca and Fleming[6] is employed. In the multi-objective GA (MOGA)[7], each individual is ranked according to its degree of dominance. The more population members dominate an individual, the higher the ranking of the individual is. Here an individual's ranking equals the number of individuals that it is dominated by plus one.

In this present paper, we develop a GA optimization and apply it to the Multi Processor Reactor (MPR) fuel plate design for a safety shut down using Matlab.

Supported by Atomic Energy Authority, Egypt.

* Corresponding author. *E-mail address:* mimi9_m@yahoo.com

Received date: 2008-07-23

2 Genetic algorithm

GA's were introduced in the 1970s by Holland[8], who did not use it to solve a particular problem but to investigate the effects of natural adaptation in stochastic search algorithms. GA's consist of a population of possible problem solutions that get refined over time through selection, recombination and mutation. Traditionally they were binary encoded but now real-value encoded GA's is just as frequent. GA's and the closely related evolutionary algorithms are a class of non-gradient methods, which has grown in popularity ever since Holland[8]. In this section we begin with a brief overview of genetic algorithm[9,10]. The goal of GA is essentially to find a set of parameters that maximize or minimize the output of a function. In GA's the mutation operator is the source of the random variation and recombination happening to investigate intermediate solutions. The selection is the primary operator that drives the GA individuals towards optimality. GA's have been described in Refs.[9,11] and have flexible operators in the sense that predetermined domain knowledge can be incorporated. A flow diagram of a simple GA optimization is presented in Refs.[11,12].

A central role was played 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 proportional 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 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[9]. This typical kind of selection allows for an individual to be selected several times and thus results in a loss of diversity. Because of the randomness in selection most techniques cannot guarantee survival of the current best solution. Neither can most traditional recombination nor mutation operators. *Elitism* provides this guarantee by explicitly selecting the best individual or group of individuals. Both the typical selection and the additional elitism technique can lead to duplicates, however, of good individuals. Population diversity decreases with duplicates, but the search can benefit when it comes to the recombination of individuals with good genes[12,13], The simple genetic algorithm can be written as:

- For each Chromosome S_i, i=1, 2,…*pop_size*, compute the fitness value

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

- Compute the total fitness of the population

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

- Compute the probability of a Selection p_i for each Chromosome S_i

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

- Compute the cumulative probability of a Selection q_i for each Chromosome S_i

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

- 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, …, pop_size, such that $q_{i-1} < r < q_1$.

Recombination of individuals is done to investigate the performance of new individuals that resemble 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. Usually the parents are removed to make room for the offspring carrying some of their genes. Recombination is often done by crossover[12]. 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[12]. 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 selected Chromosomes randomly
- For each pair of coupled Chromosomes, generate a random integer number pos[1, …, m–1], m is number of bits in each Chromosome, the number pos indicates the position of crossing point where the following two Chromosomes are crossed over as (b_1, b_2, b_3, …, b_{pos}, b_{pos+1}, b_m) and (c_1, c_2, …, c_{pos}, c_{pos+1}, c_m) are replaced by a pair of their offspring (b_1, b_2, …, b_{pos}, c_{pos+1}, c_m) and (c_1, c_2, …, c_{pos}, b_{pos+1}, b_m).

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. Exploration by mutation is often slow compared to recombination, but in problems where recombination is disruptive mutation can be an important way to explore the landscape. Even if most of the search is being performed by recombination, mutation can be vital to provide the diversity which recombination needs. The probability of mutation is usually a variable GA parameter[4,12]. The goal of GA is essential to find a set of parameters that maximize or minimize the output of a function. GA's have been described and have flexible operators in the sense that predetermined domain knowledge can be incorporated. The mutation operation proceeds in the following manner:

For each Chromosome in the population applied:

- Generate a random number $r \in [0,1]$
- If $r<p_m$ mute this bit by changing its value from 0 to 1 or *vice versa.*

3 The multi-objective optimization problem

Multi-objective genetic algorithms usually try to find all the non-dominated solutions of an optimization problem with multiple objectives. A general multi-objective design problem is expressed as:

$$\text{Maximize } f(x) = \{f_1(x), f_2(x), \dots, f_m(x)\}$$

Subject to $g(x) = \{g_1(x), g_2(x), \dots, g_j(x)\} \le 0$ and $h(x) = \{h_1(x), h_2(x), \dots, h_k(x)\} = 0$, where $\boldsymbol{x}$ is a vector to be determined, and where $x = \{x_1, x_2, \dots, x_M\} \in X$, $y = \{y_1, y_2, \dots, y_M\} \in Y$ and $f_1(.)$, $f_2(.)$, …, $f_n(.)$ are n objective functions to be maximized. If a feasible solution is not dominated by any other feasible 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 $\boldsymbol{x}$ and $\boldsymbol{y}$, it is said that the solution $\boldsymbol{x}$ is dominated by the solution $\boldsymbol{y}$:

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

4 Genetic algorithm for multi-objective problem

We have considered some modified operations such as evaluation, selection, and elitist strategy in the previous sections in order to construct a genetic algorithm for multi-objective optimization problems. We can construct a multi-objective genetic algorithm (MOGA) by employing those operations for multi-objective optimization[9].

The solution procedure of the proposed algorithm is summarized in the following steps:

1. Set k =0;
2. Initialize the two populations $P_s(k)$ and $P_r(k)$ randomly, where $E(k)$ is initialed by zero;
3. If one reference point $r \in M$ is reached, then go to 7;
4. Double the number of trials to obtain a reference point $r \in M$. If it is reached, then go to 7;
5. Increase the precision parameter of the algorithm. If it is reached, go to 7;
6. Read the solution of the dual problem as a reference point. Go to 7;
7. Evaluate the population $P_r(k)$ by using the objective function, and sort $P_r(k)$;
8. Update the elitist point $E(k)$ when the reference point has the best fitness;
9. Check feasibility. If the search point S of the population $P_s(k)$ is feasible, evaluate $P_s(k)$ using the objective function. Go to 12;
10. Create a random point $z \in M$ from a segment line between s and $E(k)$ as: $z = \delta s + (1-\delta) E(k)$, where $\delta \in [0,1]$is a random number;
11. Evaluate the point z. If the fitness of z is better

than that of $E(k)$, then replace $E(k)$ and the most fit reference point r by z. Also, replace s by z with some probability of replacement;

12. If the stopping rule is satisfied then go to step 18; else set $k = k + 1$;

13. Select the population $P_s(k)$ from $P_s(k–1)$ using ranking selection method;

14. Recombine the new population $P_s(k)$ by using genetic operators;

15. If the number of k/n =0, then go to 6, else go to 9;

16. Recombine the new population $P_r(k)$ by using genetic operators;

17. Select the population $P_r(k)$ from $P_r(k–1)$ using ranking selection method, go to 7;

18. Stop. See Fig.(1).

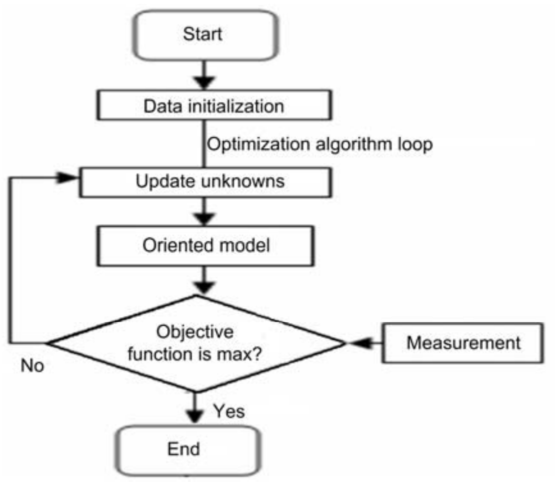

Fig.1 Genetic Algorithm Fitting algorithm.

5 Description of non-dominated sorting genetic algorithm

The basic idea behind "NSGA" is the ranking process executed before the selection operation[12]. 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, usual operators such as selection, crossover, and mutation 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[14]. 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[15,16]. The sharing procedure used in this method can be summarized below[12]. 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…., k as follows:

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

$$d_y = \sqrt{\sum_{p=1}^{P}\left(\frac{x_p^i - x_p^j}{x_p^u - x_p^l}\right)^2} \tag{5}$$

where p is the number of variables in the problem, $x_p{}^l$, $x_p{}^u$ is the lower and upper bound of the variables x_p.

Step 2: The distance d_{ij} is computed with a pre-specified parameter ∂_{share}

$$sh(d_y) = \begin{cases} 1-\left(\dfrac{d_{ij}}{\partial_{\text{share}}}\right)^2, \text{if} \quad d_y \le \partial_{\text{share}} \\ 0, \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{6}$$

The approximate value of ∂_{share} is

$$\partial_{\text{share}} \approx 0.5q^{1/p} \tag{7}$$

where q is the desired number of distinct optimal solution.

Step 3: Increasing j. If $j \le n_k$, go to Step1; if $j > n_k$, calculate count $m_i = \sum_{j=1}^{n_k} sh(d_y)$

Step 4: Sharing the dummy fitness f_k of the i^{th} in the k^{th} non-dominated as shared fitness $f = f_k/m_i$.

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

6 Implement problem

In this paper we design the fuel element (FE) of the MPR reactor during the safe Shutdown Earthquake (S1) using the genetic algorithm. The seismic design verification of the FE should demonstrate that the mechanical and geometrical stability of the FE is kept during and after the (S1) in order to fulfill the following safety requirements[17] (Fig.2):

- To allow the fast insertion of control absorber blades for the safe reactor shutdown.
- To maintain a coolable geometry for residual heat removal from the fuel.

Fig.2 Finite element model for the fuel plate.

6.1 Description of the fuel element

Fig.3 shows a general scheme of the FE, with references to the figure:

- End Box Zone(EBZ): This zone corresponds to the joint between the upper part of the End Box and the lower edges of the FP and the Side Plates(SP).
- Cooling Windows Zone (CWZ): It is the inlet channel zone of the FE. This is the weakest part of the FE.
- Aluminum Active Zone (AAZ): It corresponds to the Inner Fuel Plates (IFP) zone of the FE without fuel mass (FM).
- Meat Active Zone (MAZ): It is the true active zone of the FE.
- Upper Plenum Zone(UPZ): This is the upper part of the fuel element where the Handing Pin is placed.

Hypothesis:

1-Damping: For the seismic verification of the FE, it is adopted a damping ratio of 0.02.

2-Effects of water: The water that covers completely the reactor core produces on every submerged component the following effects:

a) Modification of the vibration characteristic: The influence of the water in the vibration of a body submerged in it has been analyzed by adding the mass of certain volume of water, denominated "equivalent fluid added mass", and solving the problem as if the component vibrates in air. Assuming in a conservative way that the FE is a rigid prism, the coefficient of "added mass" is K_a=1.25.

b) Movement of the water during the shutdown: The influence of the water motion near the FE can be neglected in the seismic verification.

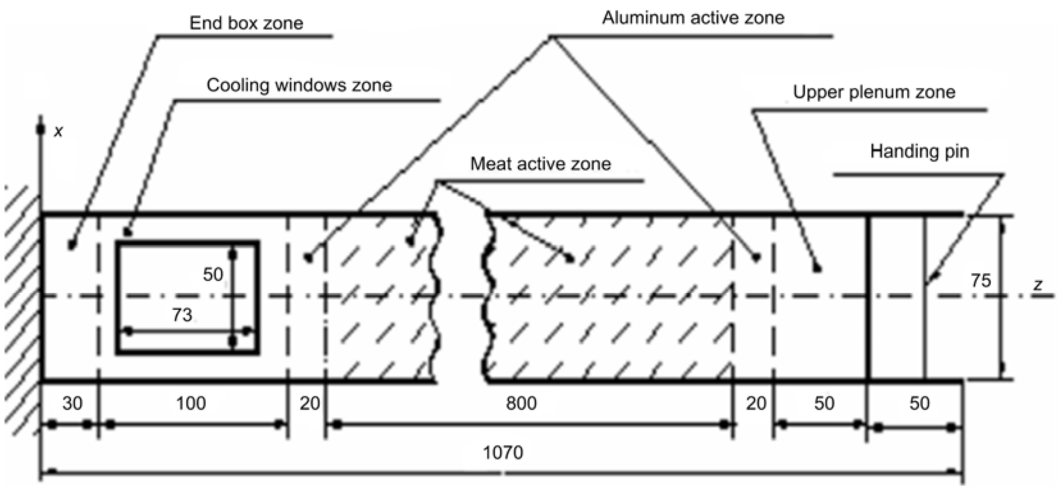

Fig.3 General description of the fuel plate.

6.2 Problem formulation

Now what we are interested in is to design the FE which is 1.070 m long and can sustain the effect of water in the core of the reactor, the coefficient of added mass is 1.25, which minimizes both mass and deformation of the FE. The FE is constructed of A96061-tolimit stress aluminum, which has the following properties: ρ=1.070 m, σ_{cr}=666.78 MPa, S_y=240 MPa (tension), E=75, $\overline{FS}$=0.02, where ρ is the FE length, σ_{cr} is the critical uniform compressive stress, S_y is yield stress, E is the elasticity module[17].

The desired factor of safety (FS) of the FE =0.02 (Fig.2).

First the maximum allowable stress must be calculated. From the maximum distortion energy theory[4], the following two equations can be written:

$$\begin{aligned}\sigma &= (\sigma_{\text{all}}^3 + 3\tau_{\text{all}})^{\frac{1}{2}} \\ \sigma &\le \frac{S_y}{FS} = \frac{240}{0.02} = 12000\end{aligned} \tag{8}$$

Now, the two stresses must be defined in terms of the design variables. The allowable normal stress in the fuel plate is defined as:

$$\begin{aligned}\sigma_{\text{all}} &= -\frac{|M|_{\max}}{S_{\min}} = \frac{|M|_{\max}}{\frac{1}{6}bh^2} = \\ &\frac{606.76}{\frac{1}{6}bh^2} = \frac{3640.56}{bh^2}\end{aligned} \tag{9}$$

There is no need to worry about the shear stress here, because when σ_{all} is maximum, τ_{all}=0. In relation, when it is maximum, σ_{all}=0. The only stress we need to consider is σ_{all}. The first constraint (g_1) can now be defined as:

$g_1(b,h)$: $((3640.56/(bh^2))^2+3*(0)^2-120*10^6)^{1/2}]\le 0$

The second and third constraints are rather obvious. Both the base and the height must be non negative.

$$g_2(b): -b\le 0, \qquad g_3(h): -h\le 0 \tag{10}$$

The last step in the problem definition is to define the two objective functions. The first function will be the mass of the fuel plate.

$$f_1(b,h) = \text{mass} = 4\rho bh = 4.280bh \tag{11}$$

The second objective function is the deformation of fuel plate. The maximum deformation will occur at the center of the plate (L=0.75), where the force is applied

$$f_2(b,h) = \delta = \frac{33}{\frac{1}{12}Ebh^3} = \frac{5.28}{bh^3} \tag{12}$$

It is now possible to define the problem in the standard notation:

$$\begin{aligned}&\min_{b,h} f_1(b,h) = 4.280bh \\ &\min_{b,h} f_2(b,h) = \frac{5.28}{bh^3} \\ &\text{subject to} \\ &g_1(b,h): \sqrt{\left(\frac{3640.56}{bh^2}\right)^2 - 120\times 10^6} \le 0 \\ &g_2(b): -b \le 0 \\ &g_3(h): -h \le 0\end{aligned} \tag{13}$$

The set of genetic parameters used are:

$$\begin{aligned}&N_{\text{ger}} = 100, \\ &N_{\text{pop}} = 100, \\ &\text{Mutation prob.} = 0.05, \\ &\text{Mission time} = 8760\end{aligned}$$

Fig.1 shows the fitting algorithm, the graphic results in Fig.4 shows the results obtained through the genetic algorithm procedure for maximizing the two objective functions of mean unavailability, reciprocal of costs and reciprocal of exposure time, simultaneously, which show the best fitness function for all of them. We report the values of the objective functions as [1.15898, 0.00105] in correspondence of all the non-dominated solutions 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.

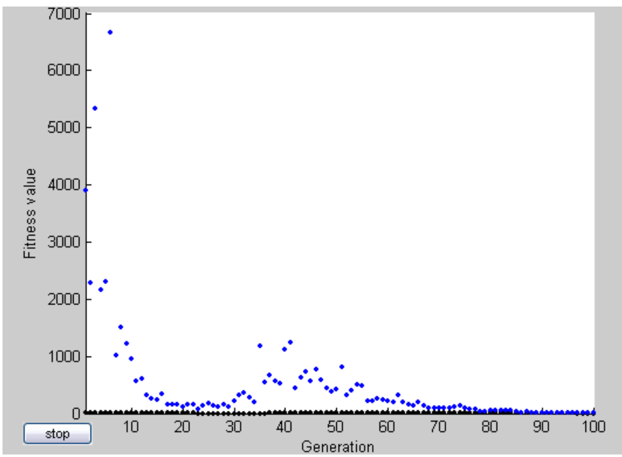

Fig.4 Multi-objective optimization results.

7 Conclusion

In this paper we performed a multi-objective optimization by means of genetic algorithms. The genetic algorithm 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 multi objective genetic algorithm approach has been applied for determining the optimal test intervals of the components of a safety system in a nuclear research reactor. The optimization performed with respect to availability, economic and workers' safety objectives has shown the potentials of the approach and the benefits which can be derived from a more informative multi objective framework.

In this paper the main results of the fuel elements plate of the MPR control operation (1997-2008) are considered and main stages of their modernization directed to increase reliability and lifetime of control member are given. This paper describes a new approach for optimizing the fuel element plate in the MPR core. This approach is based on the nonlinear multi-objective method using genetic algorithm which has already been successfully implemented for structural optimization.

References

1 Roughgarden J. The ory of population genetics and evolutionary ecology. Prentice-Hall, 1979.

2 Goldberg D E. Genetic algorithms in search, optimization and machine learning. Michigan: Addison-Wesley Publishing Company, Inc., 1989.

3 Coello C. An empirical study of evolutionary techniques for multi objective optimization in engineering design. Dissertation of Department of Computer Science, Tulane University, 1996.

4 Deb K. Evaluation Computing, 1999, **7:** 205-230.

5 Zitzler E, Thiele L. IEEE Trans Evaluation computing, 1999, **3:** 257-271.

6 Fonesca C, Fleming P. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kaufmann Publisher, 1993, 416-423.

7 Gruneninger T, Wallace D. Multi-modal optimization using genetic algorithms. MIT CADlab - Technical Report 96.02, 1996.

8 Holland J H, Adaptation in natural and artificial systems. Cambridge, Massachusetts: MIT Press, 1975.

9 Murato T. Genetic algorithms for multi-objective optimization, OSKA Prefecture University, 1997.

10 Fonseca C M, Fleming P J. IEEE Trans Systems, Man, & Cybernetics Part A: Systems & Humans, 1998, **28:** 26-37.

11 Back T. Evolutionary computation, 1993, **1:** 1-12.

12 Srinivas N, Deb K. J Evol Comput, 2004, **2:** 221-248.

13 Wahed M. J Concert Application of Math, 2007, **5:** 219-229.

14 Horn J. Multi criterion decision making. Handbook of evolutionary computation, IOP Publishing Ltd, 1997.

15 Ball M, Fleischer M, Church D. A product design system employing optimization-based tradeoff analysis. Proceedings of ASME DETC Design Theory and Methodology Conference, Baltimore, USA, September 10-13, 2000.

16 Shaaban N, Takahashi H. J Nuclear Science and Technology, 2006, **43:** 816-818.

17 Invap A. Final safety analysis report of ETRR-2. Document #0767-5325-3IBLI-001-1A, 1999.