Application of Multi Objective Genetic Algorithm for Optimization of Core Configuration Design of a Fast Breeder Reactor

Jayalal M L[1], Riyas A[1], Jehadeesan R[1], Devan K[1], Sai Baba M[2]

[1]Indira Gandhi Centre for Atomic Research (IGCAR), Kalpakkam, Chennai, India

[2]National Institute of Advanced Studies (NIAS), Bengaluru, India

Abstract

The optimization problem of nuclear fuel management, reported in the present study aimed at arriving at the optimal number of subassemblies in the two fuel enrichment zones of the core of a 500 MWe Fast Breeder Reactor. The elitist multi-objective approach of Genetic Algorithm, namely Non-dominated Sorting Genetic Algorithm-II (NSGA-II), was employed in the study. The five parameters considered for optimization are: core excess reactivity, liner heat ratings of inner and outer fuel enrichment zones of the core, fissile material inventory, and breeding ratio. The results obtained from the study indicate that the algorithm is able to produce feasible solutions in an efficient manner while preserving the diversity amongst them. The fast convergence and the diversity-preserving feature of the algorithm are described. The major objective of the work is to study the viability of applying the NSGA-II into the nuclear fuel management problems of fast breeder reactors.

Keywords: Genetic Algorithm; Multi Objective Genetic Algorithm; Fast Breeder Reactor; Nuclear Fuel Management; Non dominated Sorting Genetic Algorithm-II (NSGA-II);

1. Introduction

The optimization of the design of a reactor core has multiple objectives and constraints, some of which are in conflict with each other. This would results in the difficulty in optimization of all the parameters simultaneously. Hence, any final solution inevitably represents some sort of compromise in meeting the given objectives. Genetic Algorithm (GA) is a well-known meta-heuristic method that is particularly suited for addressing such problems [1]. In the present study, we consider the application of a suitable flavor of the GA in the optimization of core configuration design of a Fast Breeder Reactor (FBR). Finding out optimal core configuration of FBR, would be the result of a detailed neutronics scoping studies, taking into consideration of several factors like, size of the core, enrichment of the fuel, linear heat rating of the fuel pins, excess reactivity of the core, control rod design, and the inventory of the fuel. Therefore, optimization of the core configuration design is an involved task in terms of computational effort and time. The problem selected in the present study is related to the design of a 500 MWe Fast Breeder Reactor (FBR) core configuration with the aim of identifying the optimal number of fuel sub-assemblies in the core. The scope of the study is limited to the methodology and not in the complexity of the core configuration.

The Fast Breeder Reactors (FBRs) play an important role in the three-stage nuclear power programme of India [2]. The milestone in the second stage is the 500 MWe Prototype Fast Breeder Reactor (PFBR), which is being commissioned at Kalpakkam, India [3]. The reactor core selected in the present study is similar to the core of PFBR, but has certain differences also. The aim of present work is to apply and study the suitability of one category of the evolutionary optimization algorithm namely Genetic Algorithm (GA) in deriving optimal number of sub-assemblies in the two enrichment zones of a 500 MWe FBR core.

Genetic Algorithm (GA) is an optimization tool based on Darwinian Theory of biological evolution. The method was developed by John Holland [4] and later popularized by one of his students, David Goldberg, who successfully applied to various practical engineering problems [5]. GA has several advantages over the traditional optimization techniques. Unlike calculus based optimization techniques, which depend on the derivative information of the objective functions, GA based techniques do not have this dependency. Furthermore, they are more efficient than enumerative schemes and random search algorithms, as they do not require evaluation of a very large number of points in the search space. These advantages brought GA as a suitable and efficient tool in nuclear fuel management applications [6]. The present study is the application of multi-objective Genetic Algorithm (referred to as multi-objective GA in the rest of the paper) - a particular category of Genetic Algorithm- in the optimization of core configuration design of FBRs and assess the advantage it gives to the core designer. The literature survey carried out by us indicates that only limited applications have been reported for the application of multi-objective GA in core design of FBRs [7].

The remaining part of the paper is organized as follows: a brief description about nuclear fuel management and the optimization techniques including GA, applied in that field is given in Section 2.The overview about the 500 MWe FBR core and the optimization problem of core configuration, selected for the study are described in Section 3. The overall scheme of calculation adopted for the optimization study is included in Section 4. Details about the implementation of the multi-objective GA used for the study are presented in Section 5. The mathematical model formulation of the selected optimization problem is outlined in section 6. The results and discussion are given in Section 7, followed by the summary of the study in Section 8.

2. Nuclear fuel management techniques and Genetic Algorithms

The study about finding out optimal number of sub-assemblies of 500 MWe FBR core presented here, comes under the core design optimization of nuclear fuel management problem. The prime aim of the nuclear fuel management is to achieve higher fuel utilization without compromising the safety during operation of the reactor. The complexity involved in such problems call for the application of the optimization techniques like Genetic algorithms (GA. Apart from GA, there are other global optimization techniques applied in the nuclear fuel management. Some of them are: Simulated Annealing [8]. Tabu Search [9, 10], Ant Colony Optimization (ACO) [11, 12], Ant-Q optimization [13], Particle Swarm Optimization (PSO) [14, 15], Artificial Bee Colony Optimization (ABCO) [16, 17], Harmony Search Algorithm (HSA) [18] and Continuous Firefly Algorithm (CFA) [19]. The above listed techniques come under the category of nature inspired intelligent algorithms.

It can be seen from the survey carried out, that there are two major approaches in formulating nuclear fuel optimization model for Genetic Algorithms [7]. The two approaches being: penalty function based Genetic Algorithm (referred to as penalty function based GA) and multi-objective GA. In the first case of the penalty function based GA, the actual multi-objective problem of the fuel management optimization is artificially converted to single objective by adding penalty functions and constraints. In the case of the multi-objective GA, the algorithm’s power of handling multiple objectives together is exploited in an efficient way. In the nuclear fuel management problems where a wide range of solutions with multiple objectives are preferred, the application of multi-objective GA is preferred than the penalty function based GA. The problem of core configuration design that considered in the study is coming under the above category and hence the multi-objective GA has been selected for the present work.

The early applications of the multi-objective GA in the nuclear fuel management were for loading pattern and burnable poison optimization of Pressurized Water Reactor by Parks [20] and later by Pereira [21]. The loading pattern optimization of Boiling Water Reactor by the multi-objective GA was carried out by Kobayashi and Aiyoshi [22, 23]. Quang Do at al. [24] applied the same concept for online refueling simulation of Pressurized Heavy Water Reactor. Hedayat et al. [1] used the multi-objective GA approach to solve the problem of core configuration design of a research reactor. The flavour of multi-objective GA (named as NSGA-II) employed in the present study is the same. There have been some initiatives to apply the multi-objective GA in the refueling scheme of Fast Breeder Reactors also [25, 26].

During the last two decade, a number of different flavours of the multi-objective GA are evolved and applied to solve several real-world optimization problems. There are two basic categories of the multi-objective GA, namely non-elitist multi-objective GA and elitist multi-objective GA [27]. According to the concept of ‘elitism’, a fixed number the GA chromosomes having higher fitness values are considered as elite chromosomes and are retained in the new generation. The earlier implementations of multi-objective GA, mentioned in the literature are of the non-elitist category. The first such algorithm namely, Vector Evaluated Genetic Algorithm (VEGA) was suggested and worked out by Schaffer [28]. The more recent implementations of the multi-objective GA are the elitist multi-objective GA. In general, the elitist multi-objective GAs are more efficient, since the elitism helps to preserve the best solutions in the past generation and speedup the convergence of the algorithm. Among the elitist multi-objective GA implementations, some got wide popularity due to their efficiency in producing better Pareto fronts and are listed as: distance-based Pareto Genetic Algorithm [29], Strength-Pareto Evolutionary Algorithm (SPEA) [30], and Pareto-archived Evolution Strategy [31] and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [27, 32]. Deb [27, 32] showed that NSGA-II outperforms the other three algorithms described, in terms of finding a diverse set of solutions and in converging nearer to the true Pareto-optimal set with less degree of computational complexity.

The flavour of the multi-objective GA considered in the present study is NSGA-II. Before going in to the details of the implementation of the algorithm, a brief description about the problem of core configuration design, selected for the study is given in the next section.

3. Optimization of the core configuration design

The aim of the study is to find out the optimal number of fuel sub-assemblies in the two fuel enrichment zones of the reactor core which gives the maximum fuel economy, while satisfying the operational and safety related constraints. The reactor core model used for the study is similar to that of Prototype Fast Breeder Reactor (PFBR) [33]. The cross sectional view of the core of PFBR is shown in Fig. 1. The core is composed of several types of subassemblies like fuel, control, blanket, and shielding. The fuel subassembly contains the mixed oxide fuel (UO_2-PuO_2) with axial blanket and shield. The active core where most of the nuclear heat is generated consists of 181 fuel subassemblies [2]. The active core (i.e. the fuel region) is divided in to two radial fissile enrichment zones: inner (referred to as core-1 in the rest of the paper) and outer (referred to as core-2 in the rest of the paper) zones. The core-1 consists of 85 subassemblies with ~21% PuO_2. Core-1 also houses 9 Control and Safety Rods (CSR) and 3 Diverse Safety Rods (DSR) for reactivity control and reactor shutdown. The core-1 is surrounded by 96 core-2 subassemblies with relatively higher enrichment ~28% PuO_2. The variation in enrichment in the radial direction helps in radial flux flattening. In the axial direction, the fuel sub-assemblies mainly comprises of fuel material, upper axial blanket and lower blanket. The blanket sub-assemblies contain depleted uranium and the breeding happens in these sub-assemblies. The steel reflectors minimize leakage of neutrons from the core. The B_4C sub-assemblies shown in the figure are the neutron shielding sub-assemblies.

The present study explores the suitability and advantages of the application of the multi-objective GA in designing a similar 500 MWe FBR core. For finding out the optimal core configuration, the number of sub-assemblies placed in core-1 and core-2 are being changed in every iteration of the algorithm. Then, the evaluation of each configuration is done based on the objectives and constraints of the optimization problem. The above-mentioned steps are to be repeated by the algorithm without any manual intervention. The 2D geometrical model used in the study is R-Z model of the core and is shown in Fig. 2. The model shown in the figure (Fig. 2) differs from the actual PFBR core (shown in Fig. 1) by way of not considering the Control and Safety Rods (CSR) and Diverse Safety Rods (DSR).This approximation allows varying the number of sub-assemblies in core-1 and core-2 in an easier way. In essence, the presence of CSR and DSR in the core and their positioning in the core-1 are excluded from the study for achieving the automatic variation of the radii of core-1 and core-2. The positioning of CSR and DSR gives the scope for another optimization study and in that case, the number of core-1 and core-2 subassemblies arrived from the present study may vary slightly.

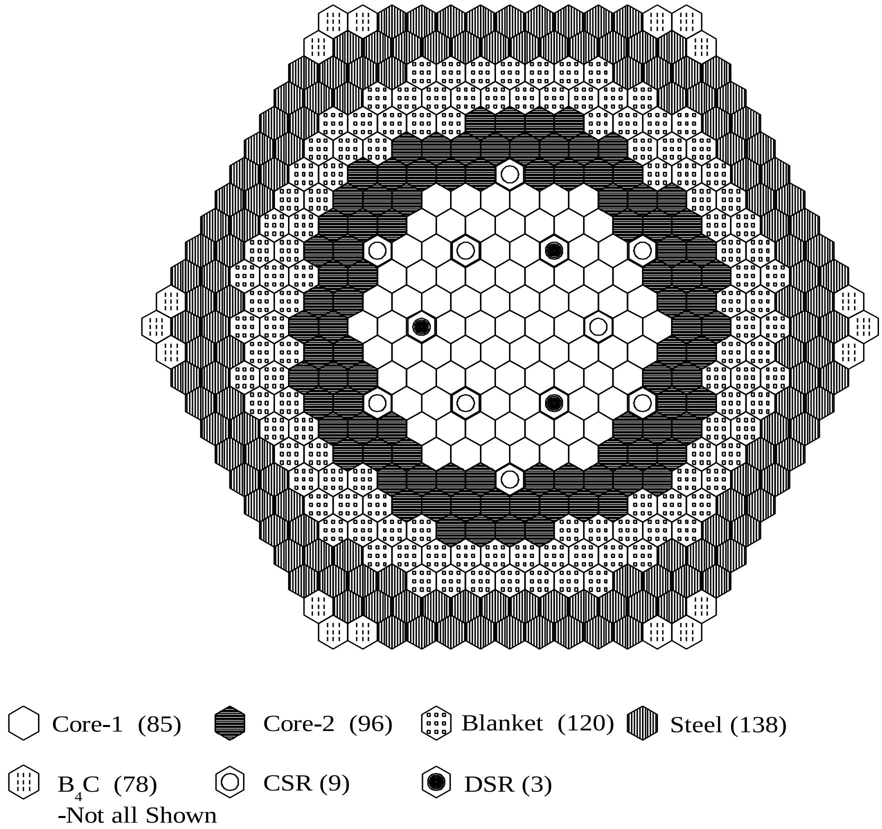

Fig. 1. Cross sectional view of 500MWe PFBR core.

As part of the optimization procedure, when the number of sub-assemblies in core-1 and core-2 are changed, the diameter of the core in radial direction changes accordingly, while the height of the core remains unchanged. The radial thickness of the portions above and below the core-1 and core-2 (consists of axial blankets, axial plenum, and stainless steel reflector as shown in Fig. 2) also vary accordingly. However, the radial thickness of the portions of the core beyond core-2 (consisting of radial blanket, radial blanket plenum, radial blanket foot and stainless steel reflector, see Fig. 2) remains unchanged. The radial thickness variations of the core regions during the optimization procedure are more clearly illustrated in the given schematic representations (Fig. 3). The figures 3(a) and 3(b) represent the radial thickness of different core regions at two randomly selected different iterations of the optimization procedure. In the figures, '$R1$' and '$R2$' denote the radii of core-1 and core-2 respectively. The terms '$C1$' and '$C2$' represent the radial thickness of blanket and steel reflectors respectively. The total radius of the core is denoted by '$R3$' i.e., $R3=R1+R2+C1+C2$. During different iterations of the optimization procedure, '$R1$' and '$R2$' (hence $R3$ also) are varied independently but '$C1$' and '$C2$' are kept fixed. Therefore, the optimization procedure that finds the optimal number of fuel sub-assemblies in core-1 and core-2, would consider the corresponding total core geometry also.

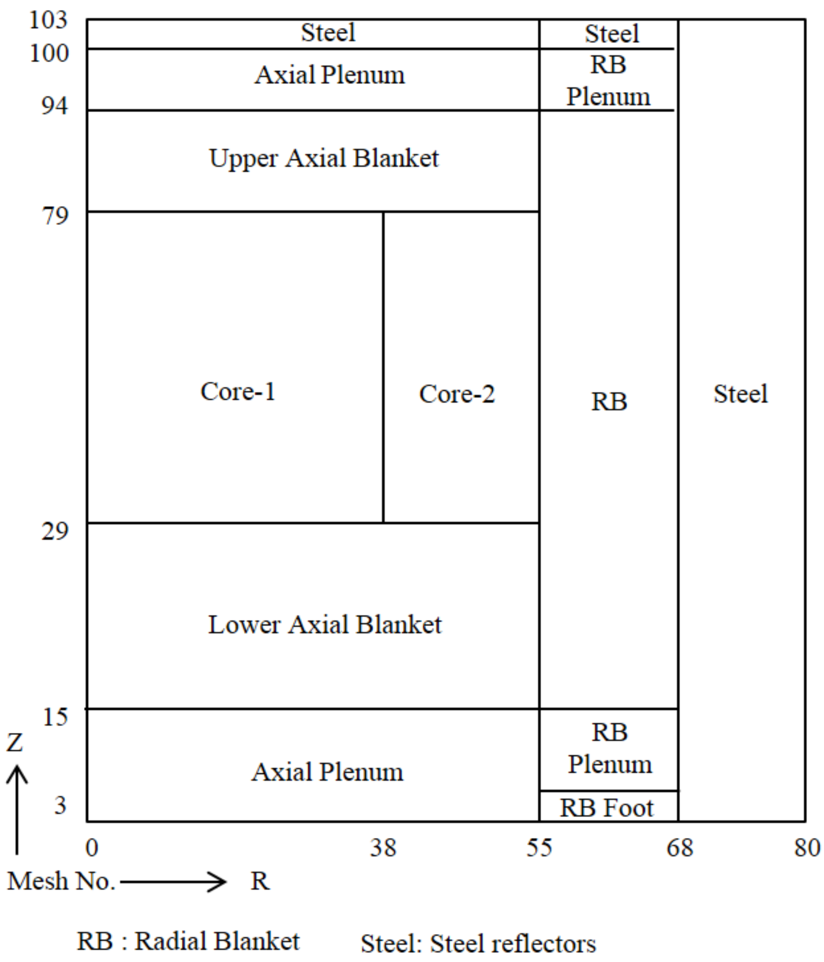

Fig. 2. R-Z model of 500MWe FBR core used for the study (control rods i.e. CSR and DSR are not considered in the model)

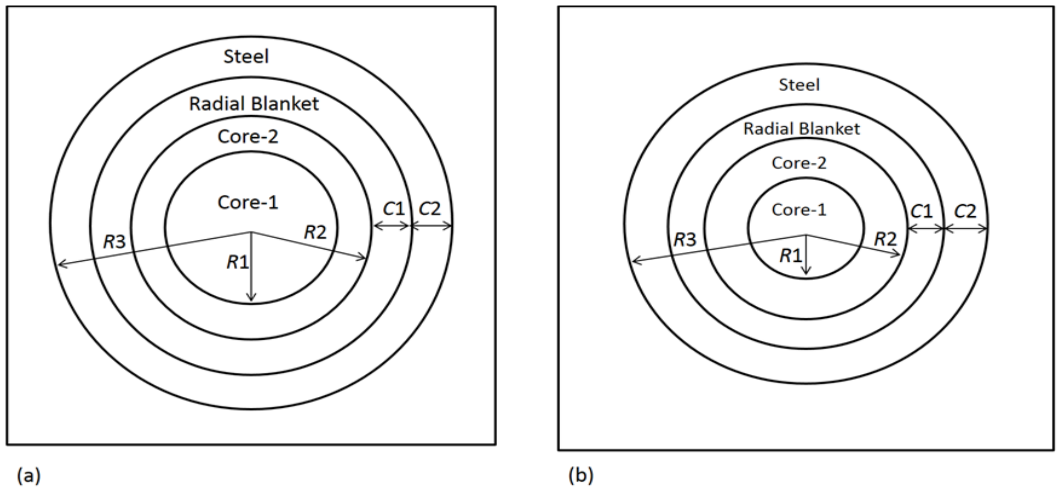

Fig. 3. The schematic representations showing the radial thicknesses variations of the core regions for two different iterations in the optimization procedure. (a) Represents the iteration for bigger core geometry with higher values of R1, R2 and R3. (b) Represents the iteration for smaller core geometry with lower values of R1, R2 and R3.

The optimization procedure followed in the study allows the algorithm to perform the search for the optimal number of fuel sub-assemblies in core-1 and core-2 by considering certain objectives and constraints. The parameters which form the objectives as well as constraints for the study are core excess reactivity, the linear heat rating of core-1 and core-2, the required fuel inventory, and the breeding ratio of the core. The excess reactivity of the core indicates the effective neutron multiplication factor to be provided in the core in order to override all the reactivity losses during an operational cycle. The linear heat rating is the power generated per unit length of the fuel pin. The objective is to limit its value such that the temperature in the fuel pin does not exceed the melting point of the fuel. The fuel inventory represents the amount of fissile material used in the core and the objective is to get a core configuration with minimal fuel inventory. The breeding ratio indicates the ratio of fissile material obtained to the fissile material spent. For a FBR core, more breeding ratio represents better core configuration design. Various steps followed in the optimization procedure are described in the next section.

4. Optimization procedure: Overall scheme of calculation

The first step in applying GA to nuclear fuel optimization is to determine the representation method which is suitable for the problem. As part of GA representation, a candidate solution (in the present study, the number of sub-assemblies of core-1 and core-2) is encoded as a digital chromosome which has enough information to reproduce the original solution. While being executed, GA generate a collection of trial solutions i.e. a population of chromosomes, and the fitness values of each chromosome is evaluated. For example, in the present study, two integer numbers that represents number of sub-assemblies in core-1 and core-2 forms one chromosome. The fitness value for each such chromosome is calculated by running the neutronics simulation codes. Similar to the natural selection process of biological evolution, chromosomes which have higher fitness values will have better chance of getting selected as 'parents' which participate in reproduction process [34, 35]. The 'offspring' solutions are produced from the parents using the genetic operations like crossover and mutation. These steps of standard GA procedure are repeated until the search process of finding the optimal solution is converged.

The flowchart illustrating the overall scheme of calculation followed in the optimization procedure is given in Fig. 4. The scheme of calculation includes GA module (that includes steps of the standard GA procedure), interface module, and neutronics simulation codes. As shown in the flowchart, the interface module provides two-way communication between the GA module and the neutronics simulation codes. The GA module is developed in 'C' programming language. Most of the neutronics codes used in the nuclear fuel management are in FORTRAN programming language and are specific to the type of the reactor. The interface module should be compatible with the neutronics codes and also should able to create the input files without user intervention. Similarly, the required output values generated by the neutronics simulation codes should be searched and read by the interface module and given back to the GA module for further calculations. The 'R' programming language [36] which supports several efficient pattern searching and file-handling operators is used for developing the interface module. Further, the 'R' programming language supports calling functions of the GA module during the runtime, as Dynamic Link Libraries (DLLs) that facilitates the smooth communication between the modules.

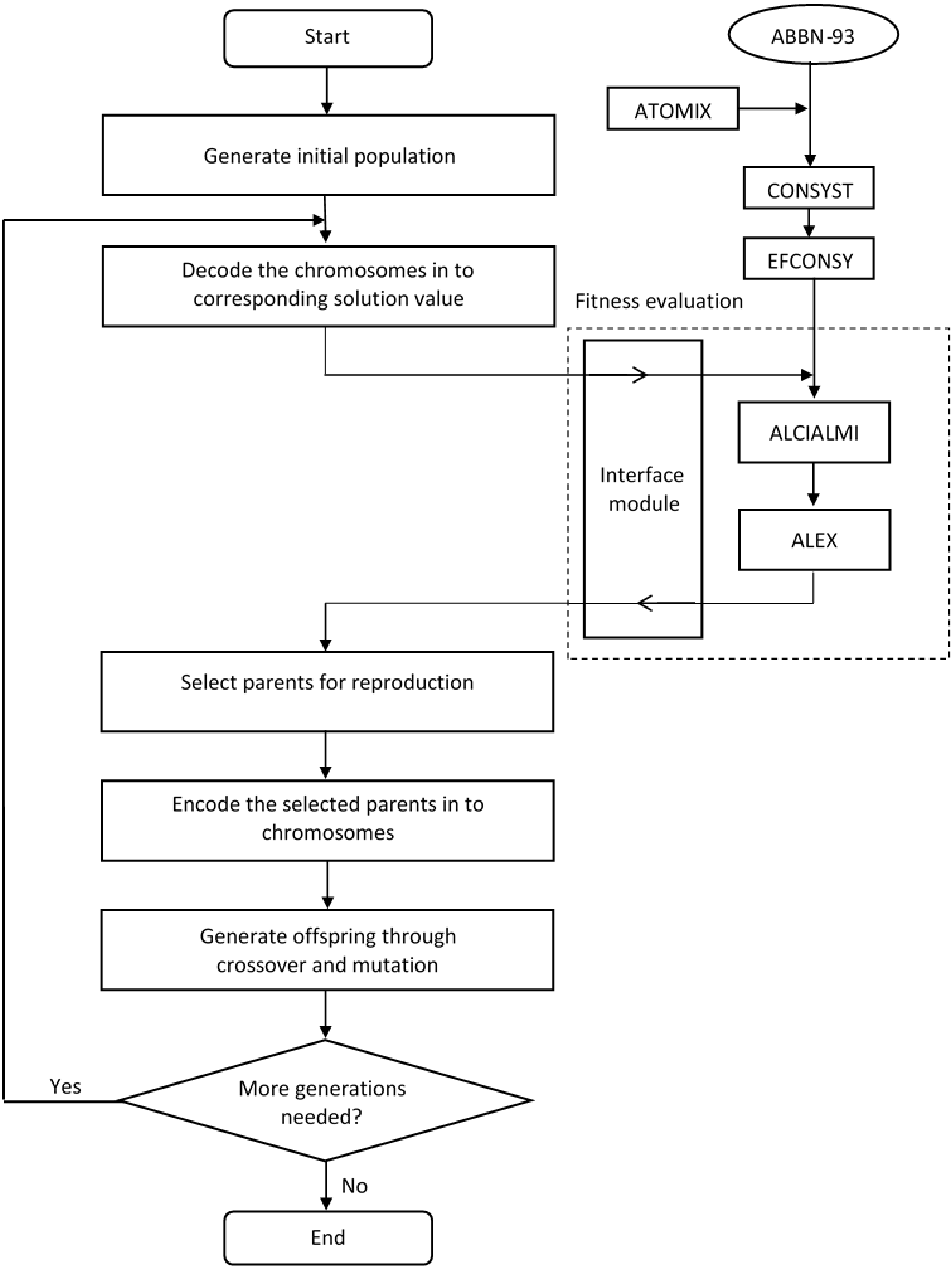

Fig. 4. Flowchart of the overall scheme of calculation followed in the optimization procedure. ATOMIX, CONSYST, EFCONSY, ALCIALMI, and ALEX are the neutronics simulation codes. ABBN-93 is the multi group cross-section library.

There are five neutronics simulation codes used in the present study which are ATOMIX, CONSYST, EFCONSY, ALCIALMI, and ALEX. The number densities of various nuclei present in the different regions of the core are calculated using the code ATOMIX [33]. Using the multi-group library of ABBN-93, the self-shielded cross-sections are calculated using CONSYST and EFCONSY codes [37, 38]. The excess reactivity of the core is calculated using the two-dimensional diffusion theory code ALCIALMI which uses R-Z geometry of the core for calculations. The code ALEX gives the power densities, from which linear heat rating of the fuel pins are calculated. The code ALEX also calculates breeding ratio of the given core configuration. In the present study, the aim is confined to find out the optimal number of fuel sub-assemblies. Therefore, the fuel enrichment values of core-1 and core-2 are fixed throughout the optimization procedure. Subsequently, in every iteration of the optimization procedure, the fuel inventory values are calculated based on the number of sub-assemblies assigned to core-1 and core-2.

During every iteration of the GA, the values representing the number of sub-assemblies in core-1 and core-2 are assigned to each of the chromosomes of the GA population. The neutronics simulation codes are used to calculate the fitness value of each of the chromosome. Since the fuel enrichments of core-1 and core-2 are fixed in the present study, the number density and cross-section calculations need not be repeated for every fitness evaluation iteration. This is represented in the flowchart (Fig. 4) by excluding the codes ATOMIX, CONSYST, and EFCONSY from the fitness evaluation block. The optimization problem of core configuration is implemented with GA as the optimization module and the neutronics simulation codes, ALCIALMI and ALEX, as the fitness evaluation module. The communication among these modules is smoothly achieved by the interface module. The mathematical model of the optimization problem is incorporated in to the GA module. The modular approach followed in the optimization procedure allows easy extension of the scope of the study to the design of other FBR cores of different sizes.

Next, we look in to the details about the implementation of the GA module. In the present work, the GA implementation is based on a specific category of the algorithm i.e. multi-objective GA.

5. Multi-objective GA

Multi-objective GA relies on the concepts of Pareto-optimality and dominance [39]. Essentially, the main task of the multi-objective GA is to find the Pareto-optimal solutions for the given problem with multiple conflicting objectives [27]. The Pareto-optimal solution is the one in which an improvement in one of the objectives requires a degradation of another. The set that consists of all the Pareto-optimal solutions for a given problem forms the Pareto-optimal front (or non-dominated front). In Pareto-optimal front, one solution cannot be considered as better than the other one. The method makes it possible to identify the “trade-offs” between conflicting objectives in a single optimization run. For example, in the case of core configuration optimization of FBR, the objectives of maximizing core excess reactivity and maximizing breeding ratio are conflicting with each other. Higher fuel enrichment (i.e. less fertile material) results in higher core excess reactivity, whereas higher fertile material (i.e. uranium-238) results in higher breeding. Often the core designer needs to consider many possible “trade-off” solutions before choosing one that best suits the need. The multi-objective GA which uses a population-based search is attractive as it leads to find many possible optimal solutions in a single run [40].

The NSGA-II implementation of the multi-objective GA is followed in the study by using the ‘C’ programming language. The implementation procedure is adopted from our work on optimization of fuel bundle burnup of a Pressurized Heavy Water Reactor (PHWR) by Jayalal et al. [41]. The procedure of NSGA-II (Fig. 5) provides an efficient sorting scheme for classifying the population into different fronts and a good diversity preserving mechanism by the crowding distance concept. When we apply NSGA-II to nuclear fuel management application, the fitness evaluation is carried out by calling the neutronics simulation codes, as in the case of the standard GA (shown in Fig. 4). The standard GA procedure like selection, crossover, and mutation are the same in NSGA-II also. However, the additional steps for incorporating concepts of Pareto-optimality and dominance are added in the procedure.

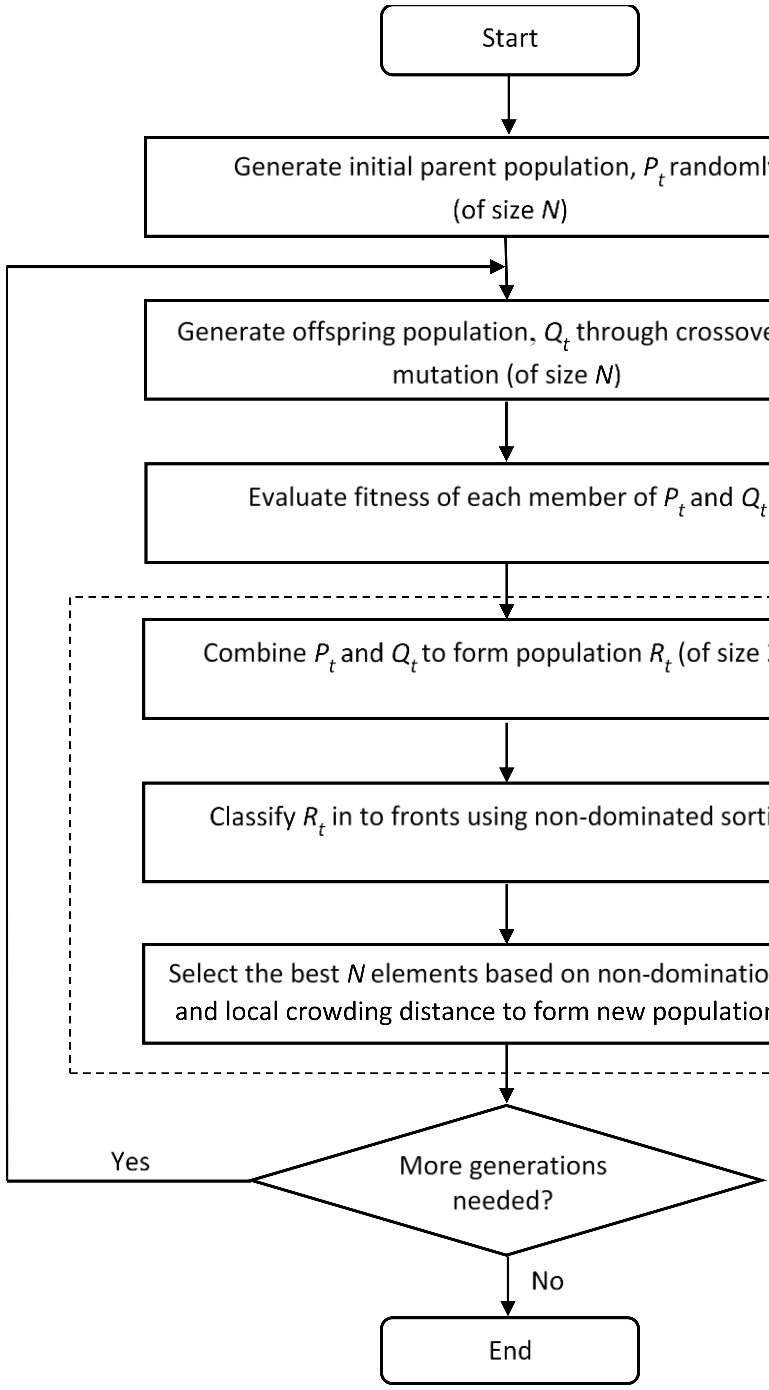

Fig. 5. Flowchart of multi-objective GA (NSGA-II implementation). The procedure consists of the standard GA operations and the additional steps of non-dominated sorting and non-domination ranking.

The first step is to generate the initial parent population, P_t of the size N (Fig. 5). Then, the crossover and mutation operations are performed on P_t to get offspring population, Q_t. This step is done before the fitness evaluation step, for doubling the population size (i.e. 2N) and that size is necessary for the subsequent steps of NSGA-II. The combined population of P_t and Q_t (denoted as R_t) undergoes the non-dominated sorting, in the subsequent step of the algorithm. The non-dominated sorting is used to classify R_t into different Pareto-optimal fronts. According to the concept of dominance, a solution $x^{(1)}$ is said to dominate another solution $x^{(2)}$, if both of the following two conditions are satisfied:

1. The solution $x^{(1)}$ is not worse than $x^{(2)}$ in all the objectives

2. The solution $x^{(1)}$ is strictly better than $x^{(2)}$ in at least one objective

In the first condition, the term 'not worse than' indicates that two solutions can equally be good with respect to an objective. The term 'strictly better than' in the second condition emphasizes that the equally good solutions are not considered in that case. The solutions belonging to the best Pareto-optimal front, $F_1$are the best solutions in the combined population. If the size of F_1 is smaller than N, all the members of F_1 are added to the new population, P_{t+1} . The remaining members of P_{t+1} , are chosen from subsequent Pareto-optimal fronts in the order of their ranking. To choose exactly *N* population members, solutions of the last allowed front are sorted using the crowded comparison operator (normally denoted by $<_c$). The new population P_{t+1} , is used for creating offspring populationQ_{t+1} , using crowded comparison operator, crossover, and mutation.

The crowded comparison operator assumes that every solution'i' has two attributes: a non-domination rank, r_i(corresponding to the Pareto-optimal front to which the solution belongs), and a local crowding distance,d_i (a measure of density of solutions in the neighborhood of the Pareto-optimal front). According to the definition of crowded comparison operator, a solution 'i'wins over another solution 'j', if any of the following conditions are satisfied:

1. If solution'i' has a better rank than solution 'j', i.e. $r_i < r_j$
2. If they have the same rank bt solution 'i'has a better crowding distance than solution 'j', i.e.$r_i = r_j$ and $d_i > d_j$

The crowded comparison operator guides the selection process at various stages of the algorithm towards a uniform spread of solutions along the best-known Pareto front. The main advantage of using the crowded comparison operator is that a measure of population density around a solution is computed without requiring a user-defined niche size or the k^{th} closest neighbor [42, 1]. The sorting of the population based on non-domination ranks along with the crowded comparison operation as a diversity-preserving mechanism, provides NSGA-II a powerful 'elitism' strategy.

The NSGA-II implementation described above allows the optimization procedure in performing the search process efficiently under the given multiple objectives and constraints. In the rest of the paper, the NSGA-II implementation of the multi-objective GA followed in the study is referred simply as multi-objective GA. In the next section, we consider the mathematical model formulation of the given problem that suits for the multi-objective GA implementation.

6. Optimization problem: model formulation for the multi-objective GA

As already mentioned, the present study aims to find the optimal number of sub-assemblies in core-1 and core-2 of a 500 MWe FBR core. The optimal core configuration design is arrived while trying to satisfy the given objectives and constraints. The given optimization problem considered have five objectives and five constraints. The objectives for maximization are related to core excess reactivity (denoted by *RHO*) and breeding ratio (denoted by *BR*). The objectives for minimization are linear heat rating of core-1 (denoted by *LHR*1), linear heat rating of core-2 (denoted by *LHR*2), and percentage deviation of fuel inventory from a selected upper limit value (denoted by *FUI*).The objectives of minimization of linear heat rating are only within a specified limit and can be considered as a special case for the selected problem. The unit of core excess reactivity is percent-milli or pcm (1 pcm = $10^{-5}\frac{\Delta k}{k}$, where '$k$'denotes the effective neutron multiplication factor, Δk denotes its deviation from the unity) and that of linear heat rating is W/cm. The upper and lower limits are defined for the constraints related to the parameters *RHO*, *LHR*1, and *LHR*2. The constraint related to *FUI* has an upper limit and that of *BR* has a lower limit. The limits of the constraints are taken in accordance with the uncertainties involved in their estimation. A solution to the problem can be termed as feasible, only if it satisfies all the five constraints. Therefore, the mathematical formulation of the given optimization problem is given as:

Max (*RHO*, *BR*) and *Min* (*LHR*1, *LHR*2, *FUI*) = *f* (number of subassemblies of core-1, number of subassemblies of core-2)

Such that, 10800 ≤ *RHO* ≤ 11200 pcm, 465 ≤ *LHR*1 ≤ 485 W/cm, 430 ≤ *LHR*2 ≤ 460 W/cm, *FUI* < given upper limit (in % deviation), and *BR* >1.045, where *Max* represents the maximization, *Min* represents the minimization and *f* () represents "function of".

The given objectives are function of the number of subassemblies of core-1 and core-2. The number of subassemblies explored for the core-1 and the core-2 are limited to range, based on the initial trial runs of the neutronics simulation codes. Therefore, the given problem has the two boundary conditions for the input values, as given below:

40 ≤ number of subassemblies of core-1 ≤ 95; 50 ≤ number of subassemblies of core-2 ≤ 108

Based on the above model, the multi-objective GA has been implemented (NSGA-II implementation) to find the optimal number of sub-assemblies in core-1 and core-2. One important step in the multi-objective GA procedure is the handling of constraint violations that helps the algorithm to bias the search through a constrained space. The constraint violations are handled by an approach which is similar to the penalty handling mechanism in the penalty function based GA [27]. The constraint functions are first normalized and then the violation for each constraint is calculated. For the five constraints of the selected problem, corresponding constraint violations are calculated as:

$$C1 = \begin{cases} \frac{RHO-RHO_{mid}}{RHO_{min}-RHO_{mid}}, & \text{if } RHO < RHO_{lb} \\ \frac{RHO-RHO_{mid}}{RHO_{max}-RHO_{mid}}, & \text{if } RHO > RHO_{ub} \\ 0, & \text{otherwise.} \end{cases}$$

$$C2 = \begin{cases} \frac{LHR1-LHR1_{mid}}{LHR1_{min}-LHR1_{mid}}, & \text{if } LHR1 < LHR1_{lb} \\ \frac{LHR1-LHR1_{mid}}{LHR1_{max}-LHR1_{mid}}, & \text{if } LHR1 > LHR1_{ub} \\ 0, & \text{otherwise.} \end{cases}$$

$$C3 = \begin{cases} \frac{LHR2-LHR2_{mid}}{LHR2_{min}-LHR2_{mid}}, & \text{if } LHR2 < LHR2_{lb} \\ \frac{LHR2-LHR2_{mid}}{LHR2_{max}-LHR2_{mid}}, & \text{if } LHR2 > LHR2_{ub} \\ 0, & \text{otherwise.} \end{cases}$$

$$C4 = \begin{cases} \frac{FUI - FUI_u}{FUI_{max} - FUI_u}, & \text{if } FUI > FUI_u \\ 0, & \text{otherwise.} \end{cases}$$

$$C5 = \begin{cases} \frac{BR_l - BR}{BR_l - BR_{min}}, & \text{if } BR < BR_l \\ 0, & \text{otherwise.} \end{cases}$$

where, the terms C1,C2, C3,C4, and C5 represents the constraint violation values related to *RHO, LHR*1, *LHR*2, *FUI*, and *BR* respectively. In the above equations, the subscripts have the following meanings related to the corresponding objective functions:

'*min*' denotes the minimum value possible.

'*max*' denotes the minimum value possible.

'*mid*' denotes the middle value of the feasible range.

'*lb*' denotes the lower bound value of the feasible range.

'*ub*' denotes the upper bound value of the feasible range.

'*l*' denotes the lower limit value.

'*u*' denotes the upper limit value.

Accordingly, the individual constraint violations corresponding to the five constraints of the problem are calculated. Then the overall constraint violation (C_{tot}) is calculated as:

$$C_{tot} = C1 + C2 + C3 + C4 + C5$$

The next step is to modify each objective function value according to the overall constraint violation. The overall constraint violation is multiplied with suitable constant values and the product is added to each of the objective function values to get the modified objective functions' values as:

$$RHO_{mod} = RHO + A1 \times C_{tot}$$

$$LHR1_{mod} = LHR1 + A2 \times C_{tot}$$

$$LHR2_{mod} = LHR2 + A3 \times C_{tot}$$

$$FUI_{mod} = FUI + A4 \times C_{tot}$$

$$BR_{mod} = BR + A5 \times C_{tot}$$

where, the term C_{tot} represents the overall constraint violation. The subscript '*mod*' denotes the modified objective function values obtained. The terms $A1$, $A2$, $A3$, $A4$, and $A5$ are used to denote constant values, assigned to make both terms on the right side of the above equations to have the same order of magnitude.

For a feasible solution, C_{tot} should be 0 and in that case, the modified values of the objective functions are same as that of actual objective function values. For an infeasible solution, a penalty is added to each of the objective functions corresponding to overall constraint violation. Once the modified objective functions are calculated, those values are used by the multi-objective GA for Pareto-optimal fronts sorting. The efficiency of the multi-objective GA (i.e. NSGA-II) in approaching Pareto-optimal fronts is not analyzed, as it is normally done for different Multi Objective GAs comparison.

7. Results and discussion

The results given are based on the multi-objective GA (NSGA-II) implementation of the optimization problem using the real-parameter encoding scheme. The trial runs are carried out on a computer system with Intel Core2 Duo CPU@ 3 GHz and 2 GB RAM. Several trial runs are conducted with randomly generated initial population. Based on the results generated initially, GA parameters selected for the final implementation have been fine-tuned. The important GA parameters relevant for the study are: population size, crossover and mutation methods, crossover and mutation probabilities, selection method and maximum number of generations. The information relating to the GA parameters used is given in the Table 1.

Table 1. Genetic Algorithm related parameters and the methods/values assigned for the implementation of multi-objective GA.

Parameter	Methods/Values
Population size	40
Crossover method	Arithmetical
Crossover probability	0.6
Mutation method	Non-uniform
Mutation probability	0.025
Selection method	Non-dominated sorting of Pareto-optimal fronts
Maximum number of generations	30

Ten trial runs are conducted with randomly generated initial population. Each trial run is started with entirely different initial population, ensuring different initial search space for different trial runs. In the following sections, we further consider the convergence of the algorithm in the objective and the solution spaces. The diversity-preserving feature of the algorithm is also considered.

7.1. Maximum and minimum values of objective functions

The maximum and minimum values of objective functions in the feasible solutions obtained from the algorithm indicate its diversity-preserving ability. The feasible solutions obtained in the final generation (i.e. 30th generation) for the 10 trial runs are considered. The maximum and minimum values considering each of the objective functions are furnished in Table 2. Corresponding number of sub-assemblies arrived as the solutions to the optimization problem are also given in the table. The neutronics simulation codes generate outputs with the accuracy of four decimal places for *LHR*1, *LHR*2, and *BR* and with the accuracy of two decimal places for *RHO*. In the case of *FUI*, the percentage deviation is calculated from the given upper limit and represented with the accuracy of two decimal places.

The prime observation from the results obtained in the present study is that, the multi-objective GA is capable of generating wide range of feasible solutions (while meeting the given constraints) for all the five objective functions. For example, consider the objective of the core excess reactivity (*RHO*) for Trial No. 1 in Table 2. The maximum and minimum values arrived for that objective is 11189.17 pcm and 10860.15pcm respectively. The constraint given to the objective is that it should be in the range between 10800 and 11200 (refer Section 6 for the constraints and their limits). The result arrived shows that the algorithm is able to generate different feasible solutions covering almost the entire range. The same were observed for *LHR1* and *LHR*2. In the case of *FUI*, all the generated results are well below the given constraint i.e. *FUI* (in percentage deviation) should be less than given maximum upper limit. For the objective *BR*, the results found to be well above the given constraint i.e. *BR* should be greater than 1.045. This capability of the multi-objective GA to generate wide range of diverse feasible solutions provides the reactor core designer, the flexibility in deciding the final core configuration. Often the designer of the core needs to consider many possible number of sub-assemblies combinations in core-1 and core-2, before choosing the one that best serve the purpose.

Table 2. Maximum and minimum values obtained in the final generation for the five objective functions and the corresponding solutions.

Trial No	Max /Min	Values obtained for objective functions					Solutions arrived	
		RHO	LHR1	LHR2	FUI	BR	Core-1	Core-2
1	Max	11189.17	484.7147	456.6290	-11.00	1.0619	87	97
	Min	10860.15	471.5066	439.8517	-8.64	1.0555	82	94
2	Max	11110.48	482.8968	456.6290	-10.91	1.0617	88	96
	Min	10860.15	471.7812	436.6091	-8.18	1.0573	82	94
3	Max	11189.17	484.7147	455.1672	-11.00	1.0619	90	97
	Min	10838.62	468.071	430.2425	-7.27	1.0552	82	94
4	Max	11189.17	484.7147	456.6290	-11.00	1.0619	90	97
	Min	10838.62	469.7777	431.4371	-7.73	1.0554	82	94
5	Max	10997.37	484.7147	454.6181	-11.00	1.0619	86	95
	Min	10868.13	477.2796	444.4523	-9.55	1.0595	83	94
6	Max	11189.17	484.7147	458.6028	-11.00	1.0619	89	97
	Min	10852.56	466.3866	432.1792	-7.18	1.0551	81	94
7	Max	11088.59	484.7147	456.6290	-11.00	1.0619	87	96
	Min	10868.13	473.5101	439.8517	-8.64	1.0574	82	94
8	Max	11178.23	484.7147	458.6028	-11.00	1.0619	89	97
	Min	10852.56	466.7057	432.1792	-7.18	1.0552	81	94
9	Max	11189.17	482.8968	458.6028	-11.00	1.0617	89	97
	Min	10852.56	469.7777	433.4109	-7.73	1.0554	81	94
10	Max	11189.17	484.7147	456.6290	-10.91	1.0619	85	97
	Min	10876.57	471.5066	445.0830	-8.91	1.0555	82	94

7.2. Performance of the multi-objective GA

In general, for a multi-objective optimization problem, no single solution is said to be the optimal or the best. Therefore, the diversity in the generated solutions can be considered as a performance measurement metric. We look in to the performance of the algorithm by considering the diversity of the solution space at various evolution stages of the multi-objective GA. The convergence of the algorithm in the solution space, at various stages of generations, for a single trial run of the multi-objective GA, is shown in Fig. 6. The initial population is generated by randomly assigning the possible combinations of number of subassemblies to core-1 and core-2 (Fig. 6(a)).At the 10^{th} generation itself, the algorithm converges towards a narrow region of the solution space (Fig. 6(b)). During the later generations, the algorithm performs the search in nearer areas of the converged solution space and finds out more different combinations among the feasible solutions (Figs. 6(c) and 6(d)). This is due to the diversity-preserving mechanism of the algorithm by crowded distance sorting. As mentioned earlier, this feature of the algorithm provides the designer of the core, more information about the available possibilities of number of subassemblies in core-1 and core-2, without deviating from the safety and operational constraints of the reactor.

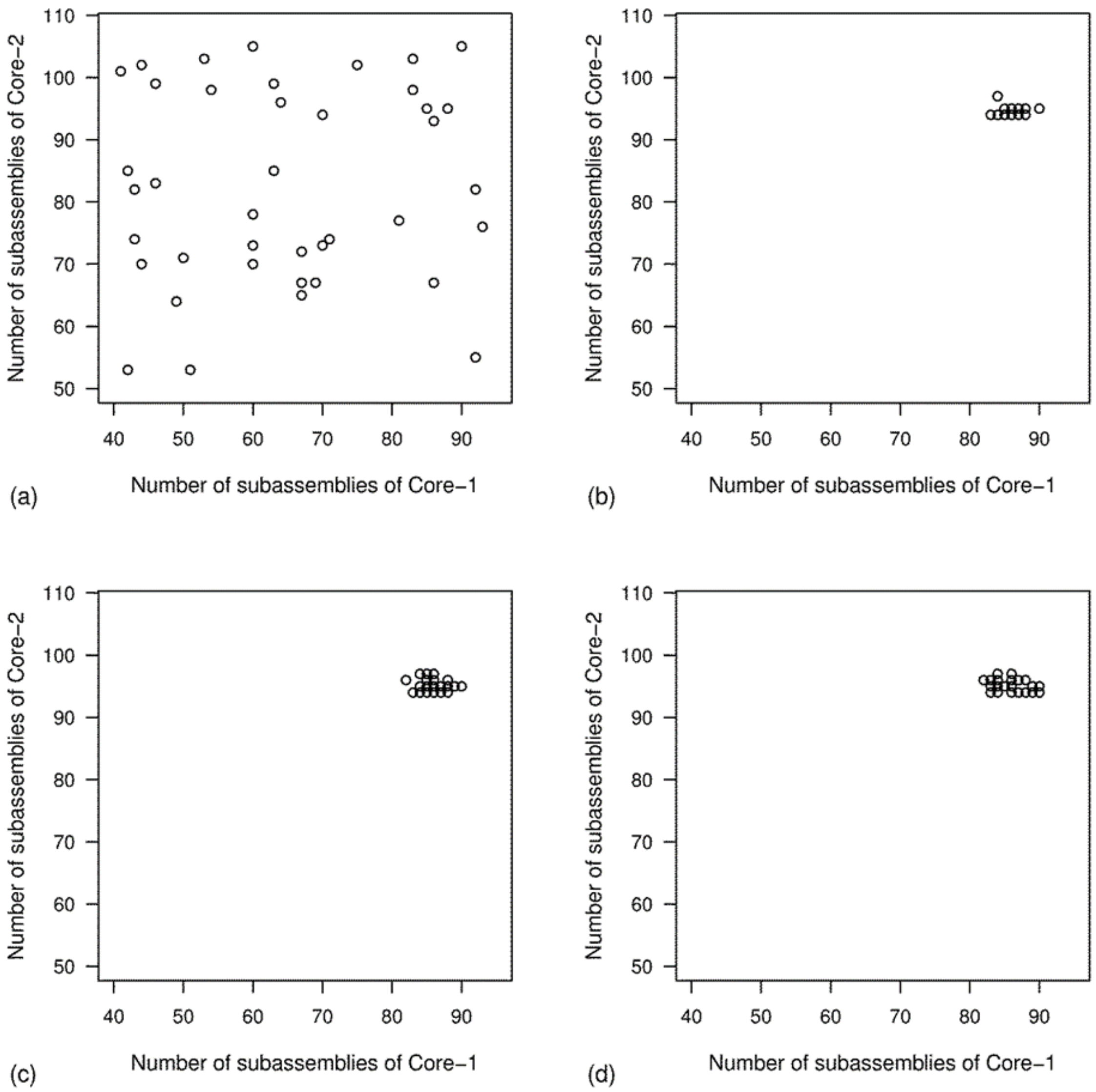

Fig. 6. Convergence of multi-objective GA in the solution space for the core configuration optimization problem: (a) Initial population. (b) Population at 10th generation. (c) Population at 20th generation. (d) Final population.

8. Conclusions

In this work, we have studied the application and suitability of the multi-objective GA in finding out the optimal number of fuel sub-assemblies of the core of a 500 MWe Fast Breeder Reactor. The flavor of multi-objective GA selected for the study is the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) which has an efficient diversity-preserving mechanism by the crowding distance concept. The results obtained show that the algorithm is able to generate diverse optimal solutions with respect to all the objectives in an efficient manner. The efficiency of the algorithm is further illustrated by plotting the algorithm's convergence in the solution space. The diversity-preserving feature of the algorithm is also illustrated by considering the solution space of the problem. The ability to find much better spread of solutions by the multi-objective GA is an important point with respect to the present study. It is resulting in getting more choices for the designer while deciding the number of sub-assemblies in the different fuel enrichment zones of the reactor core. The speed of convergence and the diversity-preserving capability shows the efficiency and suitability of the multi-objective GA (NSGA-II) for the optimization problem of the core configuration considered in the study.

References

1. Hedayat A, Davilu H, Barfrosh A A, Sepanloo K 2009 Optimization of the core configuration design using a hybrid artificial intelligence algorithm for research reactors. *Nuclear Engineering and Design*. 239: 2786-2799
2. Chetal S C, Balasubramaniyan V, Chellapandi P, Mohanakrishnan P, Puthiyavinayagam P, Pillai C P, Raghupathy S, Shanmugham T K, Sivathanu Pillai C 2006 The design of the Prototype Fast Breeder Reactor. *Nuclear Engineering and Design*. 236: 852-860
3. Chetal S C, Chellapandi P 2013 Indian fast reactor technology: Current status and future programme. Sadhana. *Indian Academy of Sciences*.385: 795-815
4. Holland J H 1995 Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence, *University of Michigan Press*, *First MIT Press Edition*. 1992
5. Goldberg D E. 1989 Genetic Algorithms in Search Optimization and Machine Learning. *Addison-Wesley, Reading*.
6. Poon P W and Parks G T 1993 Application of genetic algorithms to in-core nuclear management optimization, Proc. Joint Int. Conf. *Mathematical Methods Supercomputing Nuclear Applications*. Karls-ruhe, Germany 777-786
7. Jayalal M L, Satya Murty S A V, Sai Baba M. 2014 A Survey of Genetic Algorithm Applications in Nuclear Fuel Management. *Journal of Nuclear Engineering and Technology* 4(1): 45-62
8. Kropaczek D J and Turinsky P J 1991 In-Core Nuclear Fuel Management Optimization for Pressurized Water Reactors utilizing Simulated Annealing. *Nuclear Technology* 95: 9-33
9. Lin C, Yang J, Lin K, Wang Z 1998 Pressurized water reactor loading pattern design using the simple Tabu search. *Nuclear Science and Engineering*.129: 61–71
10. Castillo A, Alonso G, Morales L B, Martín-del-Campo C, Francois J L, del Valle E 2004 BWR fuel reload design using a Tabu search technique. *Annals of Nuclear Energy*.31:151–161
11. Ortiz J J, Castillo A, Montes J L, Perusquia R 2007 A New System to Fuel Loading and Control Rod Pattern Optimization in Boiling Water Reactors. *Nuclear Science and Engineering*.157: 236–244
12. Lin C and Lin B F 2012 Automatic pressurized water reactor loading pattern design using ant colony algorithms. *Annals of Nuclear Energy*. 43: 91–98
13. Machado L and Schirru R The Ant-Q algorithm applied to the nuclear reload problem. *Annals of Nuclear Energy*. 29: 1455–1470
14. Meneses A A M., Machado M D, Schirru R 2009 Particle swarm optimization applied to the nuclear reload problem of a Pressurized water reactor. *Progress in Nuclear Energy*. 51: 319-326
15. Waintraub M, Schirru R, Pereira C M N A 2009 Multiprocessor modeling of parallel Particle Swarm Optimization applied to nuclear engineering problems. *Progress in Nuclear Energy*. 51: 680-688
16. Oliveira I M S and Schirru R 2011 Swarm intelligence of artificial bees applied to in-core fuel management optimization. *Annals of Nuclear Energy*. 38: 1039–1045
17. Safarzadeh O, Zolfaghari A, Norouzi A, Minuchehr H 2011 Loading pattern optimization of PWR reactors using Artificial Bee Colony. *Annals of Nuclear Energy*. 38: 2218–2226
18. Poursalehi N, Zolfaghari A, Minuchehr A 2013a PWR loading pattern optimization using Harmony Search algorithm. *Annals of Nuclear Energy*. 53: 288-298
19. Poursalehi N, Zolfaghari A, Minuchehr A, Moghaddam H K 2013b Continuous firefly algorithm applied to PWR core pattern enhancement. *Nuclear Engineering and Design*. 258: 107-115

20. Parks G T 1996 Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search. *Nuclear Science and Engineering.* 124: 178-187

21. Pereira C M N A 2004 Evolutionary Multicriteria Optimization in Core Designs: Basic Investigations and Case Study. *Annals of Nuclear Energy.* 31: 1251–1264

22. Kobayashi Y and Aiyoshi E 2002 Optimization of boiling water reactor loading pattern using two-stage genetic algorithm. *Nuclear Science and Engineering*. 142 : 119–139

23. Kobayashi Y and Aiyoshi E 2003 Optimization of a boiling water reactor loading pattern using an improved genetic algorithm. *Nuclear Technology.* 143:144-151

24. Quang Do B, Choi H, Hong Roh G 2006 An Evolutionary Optimization of the Refueling Simulation for a CANDU Reactor. *IEEE Transactions on Nuclear Science.* 53(5): 2957-2961

25. Toshinsky V G, Sekimoto H, Toshinsky G I 1999 Multiobjective fuel management optimization for self-fuel-providing LMFBR using genetic algorithm. *Annals of Nuclear Energy.* 26: 783-802

26. Toshinsky V G, Sekimoto H, Toshinsky G I 2000 A method to improve multiobjective genetic algorithm optimization of a self-fuel-providing LMFBR by niche induction among nondominated solutions. *Annals of Nuclear Energy*. 27: 397-410

27. Deb K 2008 Multi-Objective Optimization using Evolutionary Algorithms. *John Wiley & Sons Ltd*..

28. Schaffer J D 1984 Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms. Ph. D. Thesis, Nashville. TN: Vanderbilt University.

29. Osyczka A and Kundu S 1995 A new method to solve generalized multicriteria optimization problems using simple genetic algorithm. *Structural Optimization.* 10(2) : 94-99

30. Zitzler E and Thiele L 1998 An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. Technical Report 43, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK). *Swiss Federal Institute of Technology (ETH).*

31. Knowles J D and Corne D.W. 2000 Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation Journal.* 8(2):149-172

32. Deb K, Pratap A, Agarwal S, Meyarivan T 2002 A Fast and Elitist Multiobjective Genetic Algorithm:NSGA-II. *IEEE Transactions on Evolutionary Computation.* 6(2): 182-197

33. Devan K, Riyas A, Mohanakrishnan P 2011 Approach to equilibrium fuelling scheme of 500 MWe PFBR based 3-D core burnup modeling. *Nuclear Engineering and Design.* 241:1596-1605

34. Michalewicz Z 1994 Genetic Algorithm + Data Structure = Evolution Programs. *Spinger-Verlag.* New York

35. Gen M and Cheng R 1997 Genetic Algorithms and Engineering Design. *Wiley-Interscience Publication*, *John Wiley & Sons.*

36. R Project, http://www.r-project.org/ (accessed September 13, 2016).

37. Manturov G N 1997 ABBN-93 Group Data Library Part I: Nuclear Data for the Calculation of Neutron and Photon Radiation Functions INDC (CCP-409). IAEA. Vienna.

38. Devan K 2003 An Interface between CONSYST/ABBN-93 and IGCAR Diffusion Theory Codes. *RPD/NDS/90, IGCAR Report*.

39. Censor Y 1997 Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization.* 4 : 41-59

40. Reddy M J and Kumar D N 2006 Optimal reservoir operation using Multi-Objective Evolutionary Algorithm. *Water Resource Management.* 20: 861-878 DOI: 10.1007/s11269-005-9011-1.

41. Jayalal M L, Suja R, Rathakrishnan S, Satya Murty S A V, Sai Baba M 2015 Application of Genetic Algorithm methodologies in fuel bundleburnup optimization of Pressurized Heavy Water Reactor. *Nuclear Engineering and Design*. 281:58-71 DOI: 10.1016/j.nucengdes.2014.11.013.

42. Konak A, Coit D W, Smith A E 2015 Multi-objective optimization using genetic algorithms: a tutorial. *Reliability Engineering Systems Safety*. 91: 992-1007.

Conflicts of Interest: There is no conflicts exists.

Funding Statement: There is no grant applicable.