M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16-20, 2017, on USB (2017)

Novel Core Physics Heuristics in Advanced Genetic Algorithms for In-Core Fuel Management

Ella Israeli & Erez Gilad

The Unit of Nuclear Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel
ellaisra@post.bgu.ac.il, gilade@bgu.ac.il

Abstract - *In this work, modern and improved genetic algorithms are implemented for the problem of the in-core fuel management. This is achieved using state-of-the-art selection and crossover operators and novel fitness function constructions, e.g., rank selection or tournament selection instead of the traditional roulette wheel selection operator; improved crossover and mutation operators by considering the chromosomes as permutations (which is a specific feature of the loading pattern problem); and the "stage fitness function" that separates the different objectives of the optimization. Another novel feature of the algorithm is the consideration of the geometric nature of the problem and the desired loading pattern solutions. A new geometric crossover is developed to utilize this geometric knowledge and is used with good results. The new algorithm is implemented and applied to two benchmark problems and used to study the effect of boundary conditions on the symmetry of the obtained best solutions and the trade-off relationship between the studied objectives of the optimization, e.g., k_{eff} and PPF.*

I. INTRODUCTION

The majority of nuclear reactors are operated in cycles with periodic complicated and expensive refueling outages. The fuel in the reactor core is not homogeneously burned and usually most depleted fuel assemblies (FAs) are replaced during refueling. The loaded fresh FAs, together with the remaining depleted FAs, are rearranged to form a new core configuration (loading pattern, or LP). The new core configuration must maximize the energy production until the subsequent refuelling outage (long cycle) while still satisfying all safety limitations and operational constraints. For example, the core excess reactivity should be maximized to ensure a long cycle and high fuel burn-up, while maintaining the ability to control and shut-down the reactor within the required safety margins [1, 2].

The in-core FAs LP optimization problem is of great importance for electricity utilities as well as for research reactors operating with limited nuclear fuel repository. This research is inter-disciplinary in the sense that a unique combination of expertise in both evolutionary algorithms and nuclear reactor physics is required. Finally, this field of reasearch is active and relevant and the successful application of modern evolutionary algorithms for solving such problems is only just beginning [1].

A well known method used for addressing the optimization problem of in-core fuel management is the so called evolutionary algorithm, specifically genetic algorithm [3]. However, many studies dealing with this problem thus far use fairly basic and traditional implementations of the genetic algorithm, disregard the geometrical structure of the core and impose symmetry restrictions on the problem, e.g., [4–12].

In this work, genetic algorithms are implemented and improved by using state-of-the-art selection and crossover operators and novel fitness function (FF) constructions, e.g., rank selection or tournament selection instead of the traditional roulette wheel (RW) selection operator; improved crossover and mutation operators by considering the chromosomes as permutations (which is a specific feature of the LP problem); and the "stage fitness function" that separates the different objectives of the optimization [13].

The new algorithm is implemented and applied to benchmark problems and used to study the effect of boundary conditions on the symmetry of the obtained best solutions.

II. METHODOLOGY

Two different cores were considered for the different stages of this study. The first core was used for the initial development of the algorithm and some basic benchmarks, hence it was chosen to be as simple as possible, yet not too simple. The second core was used for more advanced stages of the research, hence it is more realistic and complex.

1. Simplified PWR Core

This core is a simplification of a typical advanced PWR with 17×17 rectangular lattice containing 257 FAs of three different ^{235}U enrichment levels. The axial composition of a FA is assumed to be homogeneous and all FAs are assumed to be fresh. Axial boundary conditions are assumed to simulate the axial reflector whereas the radial boundary conditions are either void or reflective. The number of FAs of each type is assumed to be constant. This core was mainly used with a single objective FF for optimizing k_{eff}. A schematic view of a typical initial core layout is given in Fig. 1.

2. MOX PWR Core

This core is a 100% MOX PWR core designed in order to maximize Pu consumption [14]. It is an optimized equilibrium LP with optimized BP loading. It has 193 FAs, an 18-month fuel cycle and a 3-batch fuel management scheme with five fuel types: fresh fuel with 0, 16, and 24 WABA rods, once burned and twice burned. The LP has 1/8 core symmetry. A quarter core section is shown in fig. 1. Since this core is an equilibrium core and contains once and twice burned FAs, a three-dimensional spatial burnup distribution is accounted for

during both the cross section generation stage and the full core three-dimensional simulations. This core was used with a double objective FF for optimizing k_{eff} and PPF.

3. The Core Simulator

The core simulator used is DYN3D [15], which is a three-dimensional core model, developed at Helmholtz Zentrum Dresden-Russendorf (HZDR), for dynamic and depletion calculations in light water reactor cores with quadratic or hexagonal FA geometry. The two- or multi-group neutron diffusion equation is solved by nodal expansion methods. In this work, the code is used only for static (i.e., eigenvalue) calculations without thermal-hydraulic feedback.

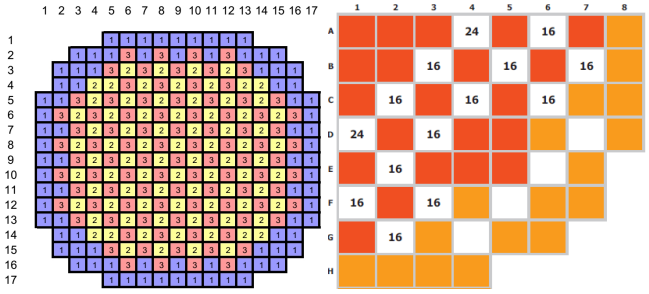

Figure 1. Schematic layout of a typical LP of a simplified PWR core (left) and MOX PWR core [14] (right). Different colors represent different enrichment (left) or burnup levels (right).

III. ALGORITHM

The evolutionary algorithm (EA) developed in this study is based on a standard EA with the required modifications. The essentials of the basic EA are summarized in Algorithm 1.

Algorithm 1 basic evolutionary algorithm

```
1:  procedure EA
2:      Generation zero: gen = 0
3:      Create an initial random population pop of size N
4:      Calculate its variance popVar
5:      Calculate fitness for every individual
6:      while popVar > threshold AND gen < maxGen do
7:          Store the best individual for later reinsertion
8:          Select N/2 pairs of individuals for crossover
9:          Crossover chosen pairs to generate N offsprings
10:         Randomly mutate a fraction of the population
11:         Reinsert best individual from previous generation
12:         Store new population as newPop
13:         gen = gen + 1
14:         Calculate the newPop variance popVar
15:         Calculate fitness for every individual
16:     end while
17: end procedure
```

An LP of a nuclear reactor core is simply an array of cells that contain materials of different types, e.g., fuel, absorber, reflector. It is a two dimensional matrix as shown in Fig. 1. It is represented by a *core vector* whose entries represent the different locations of the FAs in the core. The core vector entries are integers representing the corresponding fuel types.

The chromosome is a vector of the core's length and is logically divided into n segments, where n is the number of fuel types. Each segment is as long as the number of FAs of that type. The chromosome is a permutation of the core vector entries and the location of a core index in the chromosome determines the fuel type it holds: The core indices in the first part of the chromosome are of the first fuel type, the ones in the second part contain fuel number two, etc. This chromosome structure is chosen in order to preserve the predetermined quantities of the different materials and elements of the core.

In order to begin the evolutionary process an initial population of solutions is needed. This initial population is created randomly in order not to affect the search with unintentional bias. An example LP from a random initial population can be seen in Fig. 2. The algorithm terminates the search when most of the population has converged to a single solution, that is, when the population's variance has gone under a chosen threshold, or after a set number of generations if not converged. The population's variance is calculated using a function that estimates how different the chromosomes in the population are from one another. It counts chromosomal differences throughout the population, i.e., for every chromosome in the population it counts the number of differences from subsequent chromosomes. Approximately:

$$population\ variance = \frac{number\ of\ differences}{number\ of\ genes\ compared}\ . \quad (1)$$

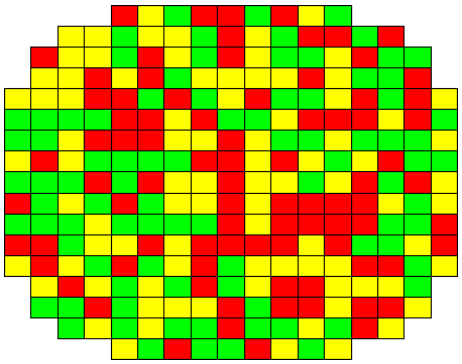

Figure 2. A random LP from a first generation of an evolutionary process.

1. The Fitness Function

Through the process of genetic evolution, the population of solutions migrates toward an optimal LP. But in order for that to be possible there must be a way of grading the LPs. For that purpose a function that determines a solution's "fitness" is constructed, namely the FF. The FF grades the solutions of the current solution population so the ones that fit the purposes better can be selected to act as parents for the next generation of solutions and carry on their superior genetic data.

A. The Single Objective FF

The algorithm is initially tested using a simplified PWR core (Sec. 1.) and a simple single objective of maximizing k_{eff}. This objective is chosen for its relative simplicity, which allows for an estimation of the approximate optimal solution. Such an approximation can be seen in Fig. 3. The configuration seen in the estimated LP maximizes k_{eff} by positioning the fissile material as far away from the core's boundaries as possible, minimizing neutron leakage.

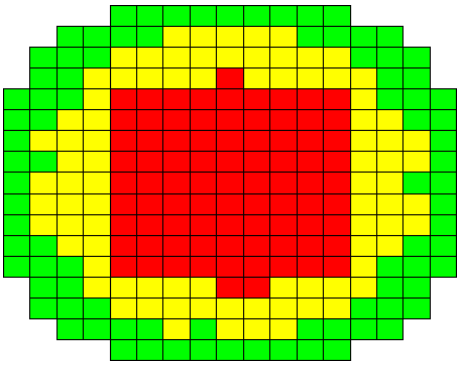

Figure 3. An estimation of the optimal k_{eff} maximizing LP.

One example of the many single objective k_{eff} FFs is one that grades LPs according to the distance of their k_{eff}'s from some pre-set upper limit:

$$FF = \frac{1}{1.5 - k_{\text{eff}} + FF\,parameter} \ . \tag{2}$$

where $FF\,parameter$ is used to regulate the scale of FF and control the selection pressure. Higher $FF\,parameter$ values result in weaker dominance of the best chromosomes. The value of 1.5 is chosen as the upper limit for the k_{eff} of the LP in order to prevent an LP k_{eff} from exceeding the limit.

B. The Multi-Objective FF

The FF can have more than one objective. The objectives chosen for this research are the maximization of k_{eff} and the minimization of PPF. The two core parameters, k_{eff} and PPF, are reciprocally interrelated. That is, a core that is characterized by high k_{eff} value is most likely to exhibit high PPF value, and vice versa. This can be easily confirmed by considering the physical meaning of each of these parameters.

The effective neutron multiplication factor, k_{eff}, is the average number of neutrons generated from a single fission event that eventually induce another fission event. Therefore, the configuration for k_{eff} maximization concentrates high enrichment fuels together to create areas reach in fissile material and increase the chances for fission. another geometrical quality of the k_{eff} maximizing core is minimizing neutron leakage from the core, keeping as much of the neutron population in the core, taking part in the reactor chain reaction. In the case of void boundary conditions, for example, the estimated best LP positions as much fissile material at core center, as far away from the boundaries as possible, reducing neutron leakage. The PPF is defined as the ratio between the local power density at the reactor hotspot and the average power density in the reactor core. Hence, the configuration for PPF minimization distributes the different fuel enrichments more evenly throughout the core, in an attempt to create a flat power density profile. As is evident from the aforementioned physical reasoning, the two objectives have trade-off relations.

So, taking both objectives into consideration in the construction of the FF is no simple feat. Some implementations take the form of the weighted composite FF, creating a single FF and giving each objective a weight in it. Some optimize one objective and put constraints on the others. In this study, it has been chosen to examine the idea of optimizing the different objectives in stages. That is, optimizing one while keeping the other in check, and then doing the same for the other. This is done in order to try and decrease the complexity of the multi-directional search, by limiting it to one clear direction at a time.

The first attempt at this form of FF is the "zigzag" FF. The zigzag FF switches between the two objectives every few generations. The number of generations between FF swaps is dubbed a stage. The original idea behind this approach is optimizing the population in regards to one objective, as much as possible for the current population, before switching to the other objective. The problem with this method is that if the population at the end of a stage is completely converged to a relative optimized solution, it has no genetic variance. Ergo, it is incapable of further optimization.

Improving upon this idea, the stages' lengths are controlled, limiting the premature stage convergence by simply cutting it off. The stage length is a definable parameter of the algorithm. Limiting stage length alone, though, is not enough. In a stage of optimizing one objective, the other is being limited to a "neighborhood" of allowed values in the vicinity of the best one in the current population. The size of the neighborhood is also an adjustable parameter and can be decreased through the progression of the evolution. The limit for the silent one of the two objectives in each generation is set by the following formula:

$$\epsilon = \frac{max(FF) - min(FF)}{FF\ limit} \ , \tag{3}$$

$$FF\ limit = a \times \left(\frac{gen\ num}{max\ gens} \right)^{b} \ , \tag{4}$$

where a and b are controlled parameters of the algorithm, $gen\ num$ is the number of the current generation and $max\ gens$ is the maximum number of generations allowed in the evolution. The variable ϵ is then the margin that we allow the silent FF to move in. That is, when k_{eff} is the silent objective the minimum k_{eff} allowed is $max(k_{\text{eff}}) - \epsilon$, and when PPF is the silent one the maximum PPF allowed is $min(PPF) + \epsilon$. We can see that as the evolution progresses, the size of the allowed environment diminishes and reaches $1/a$ of the max-min difference of that stage's silent FF.

The cons of this method are that it restricts the search very much to the search space area arrived at at the end of the first stage. It simply does not allow for a wide enough search. So a more compact form of the zigzag approach is being tested. It is the stage approach, in which the objective switch is performed only once during the evolution. Once the population reaches a

search space region rich enough in one objective's optimized solutions, a limit can be set for it and the other can be improved. This method is different from simply setting a constraint upon one of the objectives. Setting a constraint too early on in the evolution limits the optimization of the non constrained objective. With this method, the problem is averted.

2. Selection Operator

Each chromosome has a probability to be selected according to its fitness. In this study, both fitness proportionate (FP) and linear ranking (LR) selection probabilities are considered. With FP, the probability of a chromosome c to be chosen (in the selection process) is determined according to $P(c) = FF(c)/\sum FF(c)$. It is an outdated selection method, hardly used any more in GAs, for its inherent flaws, hereby explained. The ramification of this probability equation is that each chromosome gets a selection probability proportional to its FF value relative to the current population. The problem with this lies in the selection pressure caused. Since the selection probability the FP selection method gives to the different solutions is proportional to their respective FFs, the selection pressure is also dependent upon the relative differences between those FFs. That is, big differences cause complete convergence of the population; prematurely, if the high FF solution is not necessarily the best solutions possible but only better relative to the current population. On the other hand, differences that are too small do not impress upon the population enough selection pressure for a progression in any direction, and result in a search dead end. This effect results from the fact that solutions that have very high FF values relative to the current population are much more likely to be selected as parents for the next generation and thus take over the gene pool and cause convergence, while weaker solutions are not selected at all and disappear from subsequent generations. On the other side of the selection pressure scale, in a population comprised of solutions of very similar FF, as is usually the case in the first generation of the evolution, the better (albeit slightly) ones do not receive any substantial selective advantage and are given a selection probability very close to the others; a situation that renders the evolutionary process powerless to gain any progress in any direction.

A possible solution is the LR selection. With LR, the chromosomes are ranked according to their relative FFs and given a selection probability linearly according to their relative rank. The selection probability for every chromosome c is calculated using a parameter, $expVal$, that represents the expected number of copies of c in the selection table. The parameter $expVal$ is calculated according to

$$expVal(c) = 2 - m + \frac{2(m-1)(rank-1)}{groupSize - 1}\ , \qquad (5)$$

where m is the maximum expected number of copies for the best individual and is in the range of $1 < m \leqslant 2$. According to its definition (Eq. 5), higher values of m result in greater selection pressure on the best solution. The $groupSize$ parameter is the size of the group of chromosomes. It can be the entire population, or a smaller group within it, as in the case of tournament selection, seen hereafter. The selection probability for chromosome c is then $P(c) = expVal(c)/groupSize$. So, with LR the probability of a chromosome to be selected is proportional to the expected number of its copies in the selection table, which is proportional to its relative rank in the population. This method eliminates the convergence pressure's dependence upon the FF differences between the solutions of the population by basing the selection probability on the ranks rather than directly on the FFs themselves.

The selection method is either RW or Tournament, where one chromosome is selected from each tournament. The selected chromosome is either the best of the tournament group or it is chosen with a selection probability as described above, either FP or LR. Tournament size is adjustable and allows the algorithm control over the selection pressure, which influences the convergence rate, as does the value of m in $expVal$.

3. Crossover Operator

The crossover is the genetic operator responsible for the creation of new solutions out of the selected parent solutions.It swaps segments of entries between two chromosomes, mixing the genetic data of the parents and creating their offspring. The crossover operator used in this study is of geometrical nature, and consists of a geometric crossover mechanism, developed to manipulate the genetic data given by the chromosome in a way that allows control over the swapped core segment shape and size. The segment's shape has never before been used as a variable of the algorithm, even though it may have significant impact due to the geometric nature of the problem. It can be chosen as one of the following:

1. Chromosome consecutive segment – A set of indices in the chromosome between two randomly chosen cut points. Since the chromosomal indices are not *directly* translated to core locations, it is a random collection of fuel locations in the core.
2. Core consecutive segment – A set of indices between two randomly chosen indices.
3. Rectangle of core neighbours – A set of indices forming a rectangular shape.
4. Square of core neighbours – A set of indices forming a square shape around a randomly chosen index.

Decreasing the swapped segment's size along the evolution serves the purpose of controlling the amount of genetic information exchange along the evolution. On the one hand, it allows the unoptimized chromosomes of the early populations to exchange large segments of genetic information in search of the best solutions. On the other hand, the gradual decrease of the segment size allows for finer, rather than large and crude, genetic alterations in the good solutions found. The segment size is controlled using slightly different formulae, each adapted for its crossover form. A representative example is the formula for the case of the square of core neighbours:

$$recSide = floor\left(maxRec \times \frac{I+J}{2} \times \frac{crossDecRate}{crossDecRate + genDepend(g - crossGen)}\right)\ , \qquad (6)$$

where *maxRec* is the maximum portion of the core diameter size that the segment square side can reach; *genDepend* is a boolean parameter that determines if the segment's maximum size decreases over the generations or remains constant; and *crossDecRate* is a parameter that influences the decrease rate. The larger *crossDecRate* is the slower the decrease.

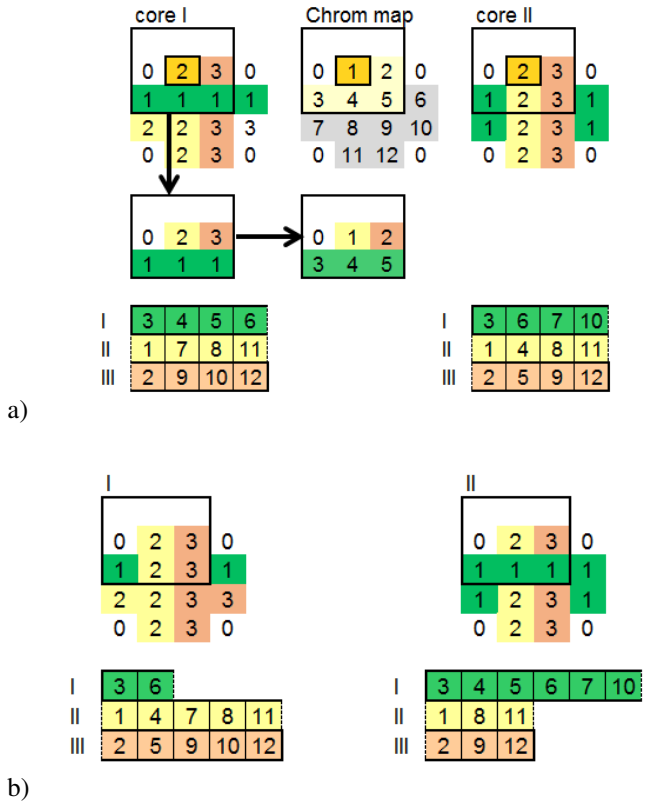

Figure 4. Illustration of "square of core neighbours" crossover.

Option 1 is the standard crossover segment used, whereas options 2-4 are novel and allow control over the segment shape. Segments sizes are adaptive throughout the evolution. An example is shown in Fig. 4, where FA #1 is chosen randomly. Then, the algorithm randomly chooses the square size; in this case, a 3×3 square. Some of the options allow segment size to decrease as a function of generation, allowing the unoptimized chromosomes of the early population to swap large segments of genetic information in the search for the best solutions, but gradually decrease the segment size to allow for finer genetic alterations in the good solutions found.

IV. RESULTS

1. Adaptive Geometric Crossover

The influence of the adaptive geometric crossover, that limits the segment's size beyond a certain generation (*recGen*), is manifested in the dramatic improvement of the results. An evolution with *recGen*<50 leads to rapid convergence to bad solutions. Good results start appearing above *recGen*=50 and peak in the range 200<*recGen*<350, as shown in Fig. 5. The graph presents the averaged results of several realizations. Results refer to evolutions on core #1 and the optimization of k_{eff} only.

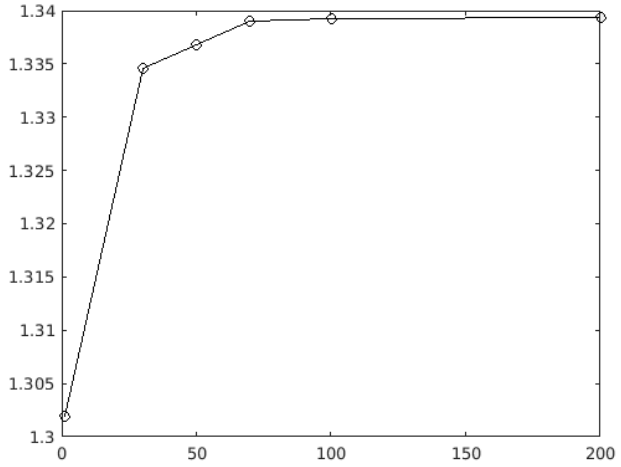

Figure 5. The effect of an adaptive "square of core neighbours" crossover on k_{eff} using the simplified PWR core.

2. Random versus Geometrical Crossover

When optimizing k_{eff} alone, the geometrical crossover has proven most successful. When introducing the objective of minimizing PPF as well, it was necessary to compare the two crossover methods once again. The geometric, "chunk swapping" crossover was initially created with the purpose of assisting the algorithm in achieving the concentric circle pattern of the k_{eff} optimizing core. This pattern is a non-homogeneous one in which each of the circles is a different region with different FA properties. Therefore, the geometric crossover might not suit the PPF objective, the optimizing core of which is very different in nature and structure. The chunk swapping geometric crossover seems much too crude to succeed in building the fine checkers like pattern of the optimal PPF core configuration. It simply lacks the required resolution. This is the reason for the introduction of the decreasing segment size into the crossover. It was done in order to allow finer resolution changes toward the end of evolution, when the LPs found are close to optimal and require small improvements.

In that sense, the random crossover which swaps a random set of cells might seem more suited for the purpose of PPF optimization. The random pattern of the random crossover cell segment is intuitively better fitting for creating homogeneous patterns. When tested, though, the random crossover did not display any advantage over the geometric one in the single objective optimization of PPF or in the multi-objective optimization of both k_{eff} and PPF .

The results shown in Tab. I are a demonstration of the consistent trend observed, suggesting the geometric crossover is preferable. They are the averaged results of several realizations. As can be seen, the geometric crossover cores have better values, i.e., higher k_{eff} and lower PPF values.

Table I. Random vs. geometrical crossover.

Crossover	Non geometric	Geometric
k_{eff}	1.0076	1.0081
PPF	1.29	1.29

3. Population Variance and Selection Pressure

Higher m values in the *expVal* formula (Eq. 5) increase the selection pressure (see Eq. 5), which dramatically affects the convergence rate and the diversity of the population, as shown in Fig. 6. When the variance of the population is 1 (0), all chromosomes are completely different (identical). This has significant implication on the ability of the algorithm to escape local minima and sample larger areas of the search space.

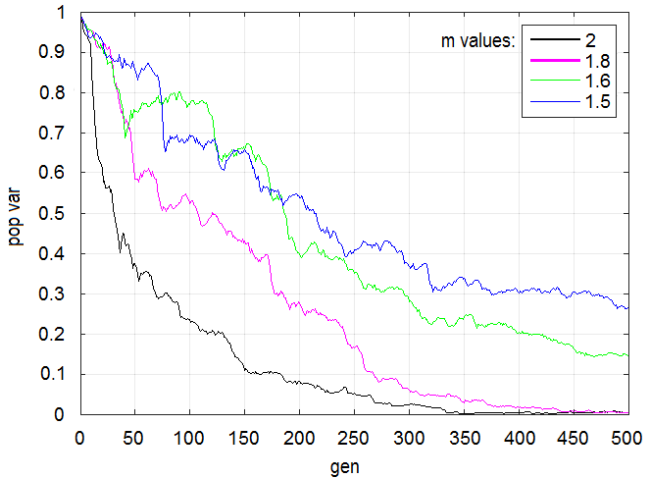

Figure 6. Population variance convergence for different selection pressures (m values).

4. k_{eff} and Selection Pressure

It can also be noted that the selection pressure that increasing or decreasing m in the *expVal* formula induces has a great deal of influence over the results obtained. Greater selection pressure causes greater pressure of convergence toward the better solutions. Too great a pressure results in premature convergence, to non optimal results, while too small a pressure does not create convergence enough. This phenomena can be seen in Fig. 7. It presents the averaged k_{eff} values of optimizations with different m values. Every point on the graph is the average value of several realizations.

5. Symmetry and Boundary Conditions

The assumption of symmetrical loading patterns dominantly underlies the entire field of loading pattern optimization of nuclear power plants. This is of course for a good reason, i.e., the primary coolant loops (and other components of the steam supply system) are symmetrically arranged around the reactor pressure vessel and the nuclear core within. Hence, symmetry in the power and mass flow distribution is a real necessity. Moreover, symmetrical loading patterns are much

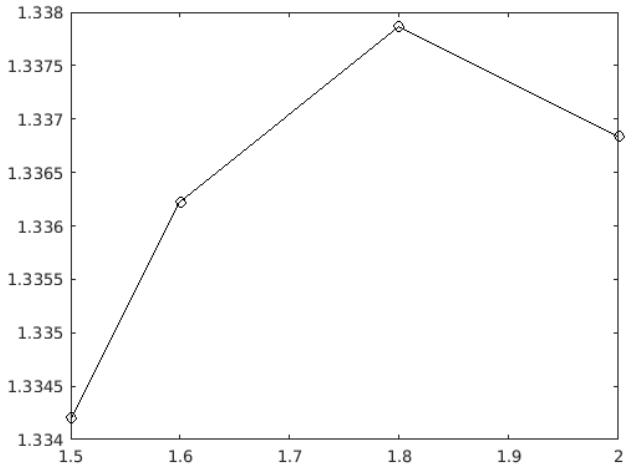

Figure 7. The averaged k_{eff} for different selection pressures (m values).

more intuitive, and nuclear engineers in charge of the plant fuel management indeed rely to some extent on this intuition and on their experience in designing core loading patterns.

However, the symmetry requirement imposed on the core by coolant loops and steam supply systems is removed once other types of reactors or critical facilities are considered, e.g., research reactors. In this section, and as an academic exercise, it is demonstrated that in some cases the best loading patterns are *not* symmetric and are very counter-intuitive.

Consider a bare reactor core with void boundary conditions, i.e., a neutron that crosses the core's boundaries to the outside does not return into the core. Using our intuition as core physicists, the spatial arrangement of FAs that maximizes k_{eff} is the one that minimizes the neutron leakage. This implies the positioning of as much fissile material as possible away from the core boundaries, i.e., in its center, as shown in Fig. 8a. This is a good example where human intuition works well, since this is only a very simple problem and the best solution is the intuitive symmetrical one.

Now, consider the same reactor core with completely reflective boundary conditions. In this exercise, the best (i.e., k_{eff} maximizing) LP produced by our algorithm, shown in Fig. 8b, is far from the immediately intuitive symmetric design. Given that modern LWR reflector designs minimize the leakage to approximately 3%, this could imply that there may be asymmetric LPs which are superior to the symmetric ones.

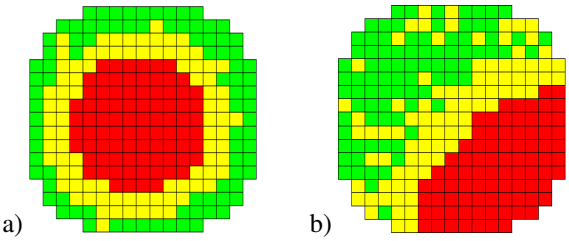

Figure 8. Boundary conditions' effect on the symmetry of the best LP. Different colors indicate different enrichment, with red (green) indicating high (low) enrichment.

6. Stage FF

Fig. 9 features a histogram of the evolution of one of the optimizations. It presents the population's best individual in every generation of the evolution in respect to k_{eff} and PPF separately. In every generation there is both an LP that holds the highest k_{eff} value and one that holds the lowest PPF value. The histogram presented allows one to follow those values in the population through the evolution.

Through this histogram the effect of the stage FF on the evolution is plainly displayed. It is the result of an optimization with a stage FF variation in which the objective change takes place once half the population reaches the PPF threshold. One can observe in the histogram the rather random distribution in both values at the beginning of evolution, the objective switch at the 169$^{\text{th}}$ generation, as well as the process of convergence to a single parameter value (at around the 300$^{\text{th}}$ generation). These phenomena are apparent both in the histogram and in the value graphs of the parameter values and the population variance. One can see the evident shift to the left (the lower PPF region) of the populations' values during the PPF minimization stage, and its swing back to the right after the optimization moves to k_{eff}.

Another purpose the stage FF serves is allowing investigation into the structure of the problem. It allows checking the limits of the PPF of the LPs while still maintaining a critical core. To work with lower PPF thresholds, the population must be allowed enough generations to migrate to a PPF optimized area of the search space though. The averaged results of several realizations can be seen in Tab. II.

Table II. Stage FF comparison of different stage lengths.

Stage	**50**	**150**	**300**	**1001**
PPF limit	**1.29**	**1.27**	**1.22**	**1.29**
k_{eff}	1.0069	1.0056	1.0014	1.001
PPF	1.28	1.27	1.22	1.20

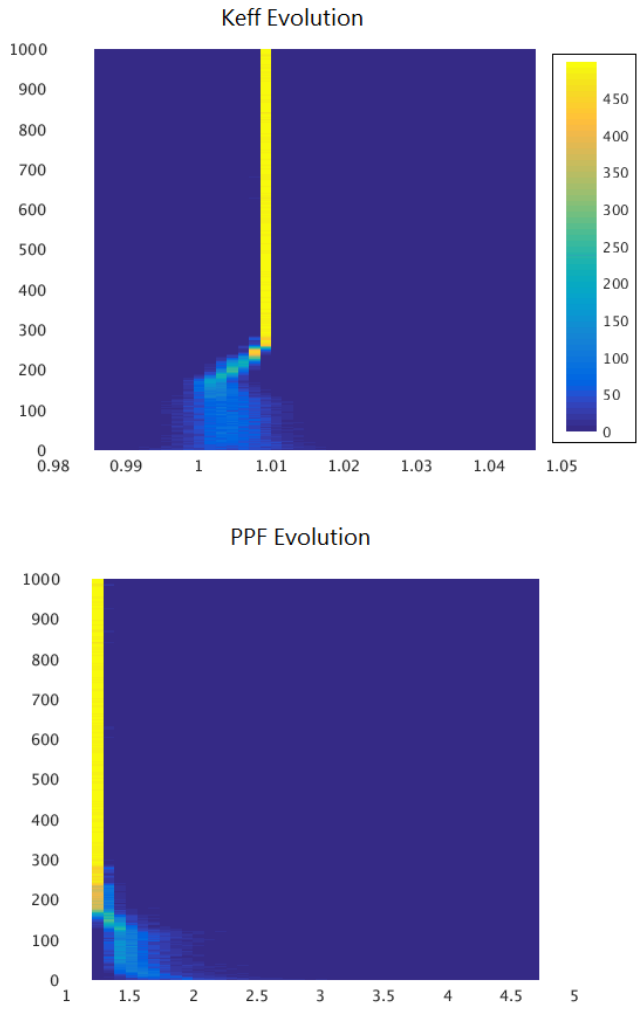

Figure 9. A histogram of the population's best individual in every generation of the evolution in respect to k_{eff} and PPF separately. The population's size in this simulation is 500, the number of generations is 1000, and the histogram describes the evolution of the distribution of the population over the k_{eff} and PPF values.

V. CONCLUSIONS

Improved genetic algorithms are developed by using state-of-the-art selection and crossover operators and novel FF constructions. The algorithm is implemented and applied to benchmark problems and used to study the effect of boundary conditions on the symmetry of the obtained best solutions.

The conclusions arising from the theoretical debate and from the results shown strengthen the claims as to the new algorithm's adaptability to the problem and the newly developed genetic operators' superiority over commonly used, "off the shelf" ones. According to results, the new geometric crossover proves better than the standard one. Even though the problem at hand is of geometrical nature, it is often overlooked. Not only for the simpler k_{eff} optimization does it produce better LPs, but also for the dual objective optimization of both k_{eff} and PPF.

The intuitive, yet untested, assumption as to the LP's symmetry has been put to the test. In the modern GA optimization world, advanced methods should discard, if possible, any potentially "harmful" influences of human intuition. An example of the harm of such influences can be seen in the case of the bare reactor core with void boundary conditions. It shows that in some cases, placing synthetic symmetry restrictions upon the LPs can prevent the creation of the optimal one. The example shows also that the matter of symmetry in the LP must be considered. It is not as inherently and immediately justified to assume symmetry in all cases as one might think. It is a question that promotes more research.

Lastly, the multi-objective optimization problem has been tackled in a new way, in an attempt to simplify it and minimize the objectives from interrupting one another. The stage FF has made it possible to determine the desired direction of movement through the search space of the problem with relative ease, confining each objective to a stage of its own.

REFERENCES

1. P. J. TURINSKY, "Nuclear fuel management optimization: a work in progress," *Nuclear Technology*, **151**, 3–8 (2005).
2. P. J. TURINSKY, P. M. KELLER, and H. S. ABDEL-KHALIK, "Evolution of nuclear fuel management and reactor operational aid tools," *Nuclear Engineering and Technology*, **37**, *1*, 79–90 (2005).
3. D. E. GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st ed. (1989).
4. G. T. PARKS, "Multiobjective PWR reload core design by nondominated genetic algorithm search," *Nuclear Science and Engineering*, **124**, 178–187 (1996).
5. J. L. C. CHAPOT, F. C. D. SILVA, and R. SCHIRRU, "A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem," *Annals of Nuclear Energy*, **26**, 641–655 (1999).
6. V. G. TOSHINSKY, H. SEKIMOTO, and G. I. TOSHINSKY, "Multiobjective fuel management optimization for self-fuel-providing LMFBR using genetic algorithms," *Annals of Nuclear Energy*, **26**, 783–802 (1999).
7. W. HONGCHUN, "Pressurized water reactor reloading optimization using ganetic algorithms," *Annals of Nuclear Energy*, **28**, 1329–1341 (2001).
8. P. GANG, P. FENG, and F. RONG, "Application of genetic algorithm in research and test reactor core loading pattern optimization," in "PHYSOR 2002, Seoul, Korea," (2002).
9. J. J. ORTIZ and I. REQUENA, "An order coding genetic algorithm to optimize fuel reloads in a nuclear boiling water reactor," *Nuclear Science and Engineering*, **146**, *1*, 88–98 (2004).
10. F. ALIM, K. IVANOV, and S. H. LEVINE, "New genetic algorithms (GA) to optimize PWR reactors Part I: Loading pattern and burnable poison placement optimization techniques for PWRs," *Annals of Nuclear Energy*, **35**, 93–112 (2008).
11. F. KHOSHAHVAL, H. MINUCHEHR, and A. ZOLFAGHARI, "Performance evaluation of PSO and GA in PWR core loading pattern optimization," *Nuclear Engineering and Design*, **241**, 799–808 (2011).
12. A. ZAMEER, S. M. MIRZA, and N. M. MIRZA, "Core loading pattern optimization of a typical two-loop 300 MWe PWR using Simulated Annealing (SA), novel crossover Genetic Algorithms (GA) and hybrid GA(SA) schemes," *Annals of Nuclear Energy*, **65**, 122–131 (2014).
13. E. ISRAELI, *In-Core Fuel Management using Genetic Algorithms*, Master's thesis, Ben-Gurion University of the Negev, Israel (2016).
14. E. FRIDMAN and S. KLIEM, "Pu recycling in a full Th-MOX PWR core. Part I: Steady state analysis," *Nuclear Engineering and Design*, **241**, 193 (2011).
15. U. GRUNDMANN, U. ROHDE, and S. MITTAG, "DYN3D - Three-dimensional core model for steady state and transient analysis of thermal reactors," in "PHYSOR 2000, Pittsburgh, PA, USA," (2000).