



Novel genetic algorithm for loading pattern optimization based on core physics heuristics



E. Israeli, E. Gilad*

The Unit of Nuclear Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

ARTICLE INFO

Article history:

Received 6 July 2017

Received in revised form 6 February 2018

Accepted 24 March 2018

Available online 7 April 2018

Keywords:

Loading pattern

Optimization

Genetic algorithm

Physical heuristics

ABSTRACT

A genetic algorithm based on novel genetic operators is implemented for the problem of nuclear fuel loading pattern optimization. This is achieved using rank selection or tournament selection and novel crossover operator and fitness function constructions, e.g., 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 its implementation exhibits good results. A comprehensive study is performed on the effect of different adaptive mutation strategies on the performances of the algorithm. 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.

© 2018 Elsevier Ltd. All rights reserved.

1. 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 a third of the (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 should 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 burnup, while maintaining the ability to control and shutdown the reactor within the required safety margins (Turinsky, 2005; Turinsky et al., 2005; Jayalal et al., 2014; Israeli and Gilad, 2017a).

The LP optimization problem is of great importance for electricity utilities as well as for research reactors operating with limited nuclear fuel repository. This study is of true inter-disciplinary nature in the sense that a combination of expertise in both evolution-

ary algorithms and nuclear reactor physics is required. This field of research is active and relevant, and has been for many years, but the successful application of modern evolutionary algorithms for solving such problems is only just beginning (Turinsky, 2005; Jayalal et al., 2014; Israeli and Gilad, 2017a).

A well known method used for addressing the optimization problem of in-core fuel management is the so called evolutionary algorithm, specifically the genetic algorithm (Goldberg, 1989; Parks, 1996). However, many studies dealing with this problem thus far use fairly basic and traditional implementations of the genetic algorithm and disregard important and relevant problem related information, such as the geometrical structure of the core e.g., (DeChaine and Feltus, 1995; Chapot et al., 1999; Toshinsky et al., 1999; Hongchun, 2001; Gang et al., 2002; Ortiz and Requena, 2004; Do and Nguyen, 2007; Alim et al., 2008; Khoshahval et al., 2011; Norouzi et al., 2013; Zameer et al., 2014) (see also Jayalal et al., 2014 and Refs. therein). Other examples include the use of fitness proportionate roulette wheel instead of tournaments and linear ranking for the selection and using binary chromosomes.

Almost all studies in this field impose symmetry restrictions on the problem. The main reason for using symmetry constrains is an operational one; the different primary coolant loops of the nuclear

* Corresponding author.

E-mail address: gilade@bgu.ac.il (E. Gilad).

power plant must maintain similar thermal–hydraulic conditions (e.g., flow rate, temperature, pressure) during nominal operation, imposing symmetry on the reactor core power production distribution.

On the other hand, research reactors (RRs) operating at low power, whether cooled by one or more loops, are free of this operational constraint of symmetry. The same is true for Integral Reactors (IRs) in general, for Small Modular Reactors (SMRs) in particular, and especially for reactor designs characterized by a single coolant loop (IAEA, 2014; Aydogan, 2016). Indeed, other operational and safety requirements, e.g., low power peaking factor (PPF) or excess reactivity control, bare significant constrains on the core loading pattern, but they do not necessarily impose symmetry.

Obviously, there exist non-symmetric LPs that satisfy high initial excess reactivity while maintaining low enough PPF that enable safe reactivity control. Actually, most LPs that use burnt fuel from previous irradiation cycles, in both RRs and NPPs, are always slightly non-symmetric, even for equilibrium cores. Imposing symmetry on the problem, e.g., by considering 1/4, 1/6, or 1/8 core LPs, eliminates a priori any (even slightly) non-symmetric LPs which potentially perform better than symmetric LPs.

In this work, a genetic algorithm is developed and implemented by using up-to-date 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 that consider the chromosomes as permutations (which is a specific feature of the LP problem); highly adaptive mutation strategies based on the instantaneous genetic variance of the population; and the “stage fitness function” that separates the different objectives of the optimization (Israeli, 2016; Israeli and Gilad, 2017a,b).

The new algorithm is first applied to simple benchmark problems for qualification and the study of the algorithm’s components separately, including the effect of boundary conditions on the symmetry of the obtained best solutions for that simple benchmark. Then the algorithm is applied to a more realistic problem of loading pattern optimization. The rest of the article is organized as follows: the genetic algorithm is described in Section 2, the nuclear problem and the core simulator are described in Section 3, the results for the simple benchmarks and for the realistic problem are given in Sections 4 and 5, and the conclusions are discussed in Section 6.

2. Algorithm

The population for the evolutionary process consists of a portion of the search space. That is, the individuals in the population are members of the search space of the optimization problem at hand. A solution in the evolutionary process is an LP of the core, i.e., a spatial arrangement of the FAs in the core. Some solutions are better than others for the purposes of optimization. A good solution in the evolutionary process is characterized by a high FF value. In this study a restriction is imposed on the allowed solution LPs, i.e., they are required to maintain the original fuel bank in the initially given LP.

The evolutionary algorithm (EA) developed in this study is based on a standard genetic algorithm (GA) with required modifications. The essentials of the basic genetic algorithm are summarized in Algorithm 1 (Israeli, 2016; Israeli and Gilad, 2017a,b).

Algorithm 1 basic genetic algorithm (Israeli and Gilad, 2017a)

```

1: procedure GA
2:   Generation zero:  $g = 0$ 
3:   Create an initial random population of size  $P$ 
4:   Calculate the genetic variance of the population
5:   Calculate the fitness  $F_i$  for every individual,  $i = 1 \dots P$ 
6:   while (genetic variance > threshold) AND ( $g < maxG$ ) do
7:     Store the best individual for later reinsertion (Elitism)
8:     Select  $\frac{P}{2}$  pairs of individuals for crossover, according to their fitness
9:     Crossover chosen pairs to generate  $P$  offsprings
10:    Randomly mutate a fraction  $\mu$  of the population
11:    Replace random individual with best one from previous generation (Elitism)
12:     $g = g + 1$ 
13:    Calculate the genetic variance of the new population
14:    Calculate the fitness  $F_i$  for every individual,  $i = 1 \dots P$ 
15:  end while
16: 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. 4. 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 (Israeli and Gilad, 2017a).

2.1. Chromosome representation

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, and so on and so forth (Israeli and Gilad, 2017a), as demonstrated in Fig. 1. Each entry in the chromosome vector is called a **gene**. This chromosome structure is chosen in order to preserve the predetermined quantities of the different materials and elements of the core (Israeli and Gilad, 2017a). Moreover, this representation gives simple and intuitive physical meaning to the genetic variance of the population, i.e., low genetic variance indicates that many chromosomes are similar in the sense that they position the same FAs in the same locations in the core.

In this representation the same LP can be represented by different chromosomes (any permutation of the genes within a single fuel type segment codes for the same LP). The genes need not be sorted in each segment. Although duplicate chromosomes of this kind artificially increase the genetic variability of the population, they bear no negative impact on the final results. In order to

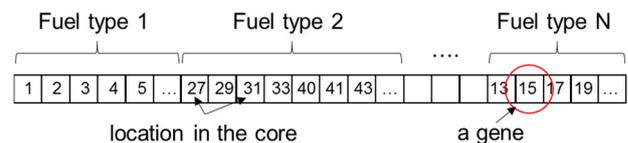


Fig. 1. A schematic description of the chromosome structure. The number of entries in the chromosome vector equals the number of FAs in the core. It is logically divided into N segments, where N is the number of fuel types. This structure preserves the predetermined quantities of each FA type.

eliminate these duplicated chromosomes, some kind of sorting and comparing needs to be carried out over the entire population each generation, which might result in significant computational overhead.

2.2. Initialization and termination

In order to begin the evolutionary process an initial population of solutions is needed. This initial population is created randomly as 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 genetic variance has descended under a chosen threshold, or after a pre-determined number of generations if not converged (Israeli and Gilad, 2017a).

2.3. Genetic variance of the population

The population's genetic variance is a measure of the degree of diversity within the population of chromosomes. It is calculated by counting chromosomal differences throughout the population, i.e., for every chromosome in the population the number of different genes with respect to subsequent chromosomes is counted. More rigorously, the population genetic variance is defined as in Israeli and Gilad (2017a)

$$\text{population genetic variance} = \frac{\text{number of differences}}{\text{number of genes compared}}$$

Given a population of size P of chromosomes of length M , the number of unique pairs is

$$\binom{P}{2} = \frac{1}{2}P(P-1) \quad (1)$$

and the total number of gene comparisons is $\frac{1}{2}MP(P-1)$.

The genetic variance is close to unity for a random population and is zero for a population of identical chromosomes. This quantity is a very good measure of the population's composition and diversity during evolution. For example, for large selection pressure (e.g., as can occur in the case of fitness proportionate selection) the genetic variance will decrease rapidly, indicating the dominance of a single chromosome and rapid convergence of the population towards this solution. For moderate selection pressure the decrease in genetic variance is slower. The best way to increase the genetic variance is through mutations, which introduce new genetic material into the existing gene pool.

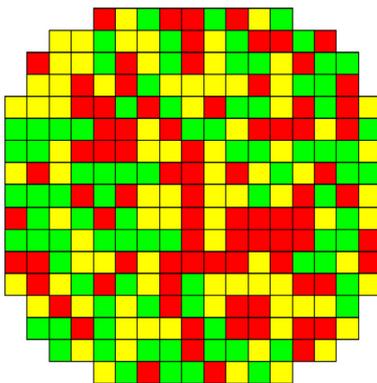


Fig. 2. A random loading pattern from the first generation of an evolutionary process (Israeli and Gilad, 2017a,b).

There is great importance in monitoring the genetic variance of the population during the evolution process (Whitley, 2001). Larger genetic variance means that the optimization process samples larger portions of the search space and is less likely to converge to **local** optimum (Friedrich et al., 2009). In many cases, when the genetic variance drops rapidly, the population essentially becomes homogeneous and there is no meaning for its size anymore (Leung et al., 1997). Online monitoring of the genetic variance of the population enables better control over the process of evolution and convergence of the algorithm through real-time parameter's change, also known as adaptive control (De Jong, 1975; Grefenstette, 1986; Eiben et al., 1999; Lobo et al., 2007).

2.4. 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 in accordance with their degree of optimality. 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 that ones who better fit the optimization purposes can be selected to become parents for the next generation of solutions and carry on their superior genetic data (Israeli and Gilad, 2017a).

2.4.1. The single objective FF

The algorithm is initially tested using a simplified PWR core (see Section 3.1) 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 (Israeli and Gilad, 2017a). In case of void boundary conditions, such an approximation is an LP in which as much fissile material is positioned as far away from the core's boundaries as possible, minimizing neutron leakage. Hence, the anticipated LP for maximal k_{eff} will position the high enriched FAs in the center of the core, surrounded by less enriched FAs, and so on (Israeli and Gilad, 2017b).

One example of the many single objective k_{eff} FFs is one in which LPs are graded according to the distance of their k_{eff} values from some preset upper limit (Israeli and Gilad, 2017a):

$$FF(k_{eff}) = \frac{1}{k_{max} - k_{eff} + \zeta}, \quad (2)$$

where ζ is used to regulate the scale of FF and control the selection pressure. Higher ζ values result in weaker dominance of the best chromosomes. The value of k_{max} is chosen as the upper limit for the k_{eff} of the LP in order to prevent an LP's k_{eff} value from exceeding the limit. In a similar manner, a PPF FF is defined,

$$FF(PPF) = \frac{1}{PPF - PPF_{min} + \zeta} = \frac{1}{PPF - 1 + \zeta}, \quad (3)$$

where ζ is the same as in Eq. (2).

2.4.2. 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. These 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 (Israeli, 2016; Israeli and Gilad, 2017a).

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 enriched FAs together to create areas rich in fissile material and increase the chances of

fission. Another geometrical quality of the k_{eff} maximizing core is minimizing neutron leakage from the core. 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 (Israeli and Gilad, 2017a).

The PPF is defined as the ratio between the local power density at the reactor's hotspot and the average power density in the reactor core. Hence, the configuration for PPF minimization distributes the FAs with different 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. Others optimize one objective and place 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 to decrease the complexity of the multi-directional search, limiting it to one clear direction at a time (Israeli and Gilad, 2017a).

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 to one objective, as much as possible for the current population, before switching to the other objective. The problem with this method is that the two objectives are reciprocally inter-related, so the resulting optimized LP of one stage is most likely very bad in regards to the other optimization objective. So the final LP resulting from the algorithm has no reason to be dually optimized, but rather is dependent on the optimization objective of the last stage alone.

Improving upon this idea, stage length is controlled, limiting the premature convergence of the population. Stage length is a definable parameter of the algorithm. Limiting stage length alone, though, is not enough, since, as mentioned above, the two objectives k_{eff} and PPF have trade-off relations. Therefore, in a stage of optimizing one objective, the other is being limited to a “neighborhood” of allowed values in the vicinity of the best chromosome in the current population, in regards to the non optimized objective of the stage. The size of the neighborhood is also an adjustable parameter and can be decreased through the progression of the evolution. The neighborhood size for the silent one of the two objectives in each generation is set by the following formula (Israeli, 2016; Israeli and Gilad, 2017a):

$$\varepsilon = \frac{\max (FF_s) - \min (FF_s)}{\zeta}, \quad (4)$$

$$\zeta = \alpha \times \left(\frac{g}{G}\right)^\gamma, \quad (5)$$

where FF_s is the silent objective, α and γ are controlled parameters of the algorithm, g is the number of the current generation and G is the maximum number of generations allowed in the evolution. The parameter ε is then the margin allowed for the silent FF to move in. That is, when k_{eff} is the silent objective the minimum k_{eff} allowed is $\max (k_{eff}) - \varepsilon$, and when PPF is the silent one the maximum PPF allowed is $\min (PPF) + \varepsilon$. Any solution whose silent fitness is outside the allowed neighborhood is discarded by zeroing its FF. As the evolution progresses, the size of the allowed environment ε diminishes and reaches α^{-1} of the max–min difference of that stage's silent FF.

The disadvantage of this method is that it restricts the search very much to the search space area arrived at the end of the first stage. It simply does not allow for a sufficiently wide spread 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 optimized solutions for one objective, a limit can be set on that objective and the other can be improved. This method is different from simply setting a constraint upon one of the objectives from beginning of evolution on account of the preliminary optimization stage. This optimization prior to placing the constraint on one of the objectives allows the constrained objective to leave the original search space area to which it would have been limited otherwise. In other words, setting a constraint too early on in the evolution limits the optimization of the non constrained objective. With this method, the problem is averted (Israeli and Gilad, 2017a).

2.5. Selection operator

Each chromosome has a probability to be selected to become a parent to the next generation of LPs 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 $P(c) = FF(c) / \sum_c FF(c)$. 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**. This is a problematic approach, the problem of which lies in the selection pressure caused. Since the selection probability the FP selection method induces on the different solutions is proportional to their respective FFs, the selection pressure is also dependent upon the **relative differences** between those FFs (Goldberg and Deb, 1991).

This effect results from the fact that in large difference FF populations, the high FF solutions are likely to be selected as parents for next generations and take over the gene pool, causing the population to converge towards themselves. Weaker solutions, on the other hand, are not selected at all and disappear from subsequent generations, contributing further to the population's variance drop (Blickle and Thiele, 1995). This convergence is premature if the high FF solutions are not globally optimal but only better relative to the current population.

On the other side of the selection pressure scale, in a population comprised of solutions of very **similar FFs**, 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 selection probabilities very close to each other. This renders the evolutionary process powerless to gain any progress (de la Maza and Tidor, 1991, 1993).

One possible solution is LR selection which equates the weight of FF differences in selection probability calculation by basing it on **rank**, rather than directly on FF value (Baker, 1985; Whitley, 1989). With LR, the chromosomes are ranked according to their relative FFs and given a selection probability linearly according to their relative rank, not their FFs. Selection probability for every chromosome c is calculated using a parameter, $expVal$, which represents the expected number of copies of chromosome c in the selection table (Blickle and Thiele, 1995; Baker, 1985; Grefenstette and Baker, 1989). The parameter $expVal$ is calculated according to

$$expVal(c) = 2 - m + \frac{2(m-1)(rank-1)}{groupSize-1}, \quad (6)$$

where m is the maximum expected number of copies for the best individual and is in the range of $1 < m \leq 2$. The rank is defined such that the worst individual has $rank = 1$ and the best has $rank = groupSize$. According to its definition (Eq. (6)), higher values of m result in greater selection pressure toward the best solution. The $groupSize$ parameter is the size of the group of chromosomes from which parents are selected. Said group may include the entire population, or a smaller group within it, as in the case of tournament selection, seen hereafter. Selection probability for chromosome c is then $P(c) = expVal(c)/groupSize$. So, the LR probability of a chromosome to be selected is proportional to the expected number of its copies in the selection table, which in turn is proportional to the chromosome's relative rank in the population.

Selection methods used in this study include the standard FP scheme Roulette Wheel and a Tournament selection scheme in which set sized tournament groups of chromosomes are randomly chosen out of the current population, from each of which one chromosome is selected as parent, until the parent pool is filled (Goldberg and Deb, 1991; Blicke and Thiele, 1995). In the Tournament selection scheme, the selected chromosome is either the best of the tournament group or it is chosen with a selection probability as described above, i.e., either FP or LR. Tournament size is adjustable and allows the algorithm selection pressure control, which influences the convergence rate (Eiben et al., 1999; Lobo et al., 2007), as does the value of m in $expVal$ (Baker, 1985; Grefenstette and Baker, 1989).

2.6. Crossover operator

Crossover is the genetic operator responsible for creation of new solutions out of selected parents. It swaps gene segments between two parent chromosomes, mixing their genetic data to create offspring. The crossover operator created for this study is of **geometrical** nature. It consists of a geometric crossover mechanism based on swapping rectangular or square segments of neighboring FAs (geometrically adjacent FAs) between two selected LP parents. Since the chromosomal indices are not directly translated to core locations, a segment of the LP is a collection of non-consecutive chromosomal indices. The new crossover operator enables the manipulation of genetic data in the chromosome in a way that allows control over swapped segment's shape and size.

Surprisingly, crossover segment shape has never before been used as a parameter of the algorithm, though it stands to reason it should have significant impact due to the geometric nature of the problem. It can be chosen as one of the following (Israeli and Gilad, 2017a):

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 in the LP. A segment of the LP is a collection of non-consecutive indices in the chromosome.
3. Rectangle of core neighbors – A set of chromosomal indices forming a rectangular segment in the LP.
4. Square of core neighbors – A set of chromosomal indices forming a square shape around a randomly chosen index in the LP.

Controlling crossover segment size provides control over the extent of crossover genetic information exchange. Large segment swaps at beginning of evolution allows for the unoptimized chromosomes of the early generations to exchange large portions of genetic information, exploring the search space. On the other hand, the gradual decrease of the segment size toward later generations

allows for finer genetic alterations in the good solutions found. 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 neighbors:

$$a = \begin{cases} \lfloor \eta(\frac{I+J}{2}) \rfloor & \text{if } g < g_x \\ \lfloor \eta(\frac{I+J}{2}) \times \chi(g) \rfloor & \text{if } g \geq g_x \end{cases} \quad (7)$$

where η is the maximum fraction of the core diameter size (approximated by $\frac{I+J}{2}$, I and J are the number of columns and rows, respectively, in the LP, and $\chi(g)$ is a decreasing factor that limits the segment's maximum size from a pre-determined generation, g_x , onward. The functional form of the decreasing factor is chosen as

$$\chi(g) = \frac{\sigma}{\sigma + (g - g_x)}, \quad g \geq g_x, \quad (8)$$

where σ is a parameter that influences the decrease rate. The larger σ is the slower the decrease.

Option 1 is the standard crossover operator widely used in nuclear fuel management problems (Jayalal et al., 2014), whereas options 2–4 are novel and allow control over segment shape. Segments' sizes are adaptive throughout the evolution. An example is shown in Fig. 3, where a FA is randomly chosen. Then, the algorithm randomly chooses the square size; in this case, a 3×3 square (with the selected FA in the middle of the square neighborhood). Some of the options allow segment size to decrease as a function of generation, relating the advantages mentioned above.

2.6.1. Fixing the chromosome after crossover

For GAs whose solutions are of permutational nature a simple segment swapping crossover results, in most cases, in an inapplicable chromosome that is no longer a permutation or a legal chromosome.

As mentioned in Section 2.1, the genes in the chromosomes are indices in the core and chromosomes are permutations of those indices; so any chromosomal recurrences caused by the crossover mechanism must be fixed. That is, there can not be a core cell denominator that appears both in the first and second parts of the chromosome, since it's counterpart in the respective LP should then need to contain two different FAs at the same site in the core.

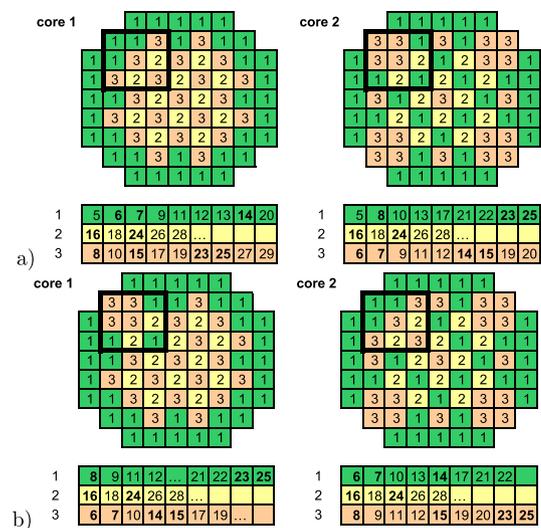


Fig. 3. Illustration of "square of core neighbors" crossover. The original parents cores (top) and the corresponding chromosomes, with each fuel type, e.g., 1, 2, 3, represented by different shades. Shown are the parents before the segment swap (a) and after (b). Only relevant parts of the chromosome are shown. The segment for crossover is the 3×3 rectangular neighborhood randomly chosen around the central cell. Segment parts that are not in the core are omitted.

The correction is done by a separate function which, outside the chosen segment, swaps recurring indices between the two chromosomes. A more detailed and elaborated description of the genetic operators can be found in (Israeli, 2016) and in (Israeli and Gilad, 2017b).

2.7. Mutation

The evolutionary search is limited to the regions of reach through population's existing genetic pool. If new genetic data is not available, the algorithm's reach is limited to the local optima of those regions only. The mutation operator is used to avoid this stagnation by introducing new genetic data into the chromosomes of the population (Goldberg, 1989).

At chromosomal level, the mutation operator swaps two random genes within a chromosome. Since in this study the location of the genes in the chromosome is the trait that carries genetic significance, this action introduces new information into the genetic pool. For example, if none of the chromosomes contain the gene "a" in location "A" then due to the nature of the crossover operator, neither can any of the offspring. The only possibility to attain this LP is through the mutation of that gene. At population level, the mutation operator mutates a randomly chosen portion of the population at each generation. The portion size is normally distributed around an adjustable predetermined value that is an algorithm parameter, namely the mutation rate μ . This mutation rate parameter has a great effect on search stagnation (David Schaffer et al., 1989; Israeli and Gilad, 2017b).

The population can be mutated in several ways. The simplest one is the constant mutation rate. A more sophisticated approach is an adaptive mutation rate (De Jong, 1975; Grefenstette, 1986; Eiben et al., 1999; Lobo et al., 2007) according to the genetic variance of the population, which prevents the population from becoming homogeneous. Once the genetic variance of the population decreases below some threshold, the mutation can be introduced either as peaks (very short and large spikes) in the mutation rate or as gradual increase. Another popular approach is the time-dependent (decreasing) mutation rate (Fogarty, 1989; Bäck, 1992; Smith and Fogarty, 1996). The idea behind this approach (Srinivas and Patnaik, 1994) is allowing the exploration of the search space at beginning of evolution, through the insertion of abundant new genetic data, and convergence of the population as evolution progresses. The formula used to decrease the mutation rate depends on the difference between current generation, g , and the generation in which decrease begins g_μ . The pace of decrease is controlled by μ_r parameter; high μ_r values lead to a slower decrease pace,

$$\mu(g) = \begin{cases} \mu_0 & \text{if } g < g_\mu \\ \mu_0 \times \frac{\mu_r}{\mu_r + (g - g_\mu)} & \text{if } g \geq g_\mu \end{cases} \quad (9)$$

3. Methodology

Two different cores are considered in this study. The first core is used for basic benchmarks of the algorithm performances during its development, hence it is chosen to be as simple as possible, yet not too simple (e.g., Pereira et al., 1999). The simple benchmarks approach enables the evaluation and characterization of different components of the algorithm separately and the validation of algorithm performance against known (or expected) solutions.

The second core is used during later and more advanced stages of the study to evaluate the algorithm performance and to solve a more practical and complicated loading pattern problem, hence it is more realistic and complex.

3.1. Benchmark problem #1

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, i.e., 3.1, 2.4, and 1.6 w/o ^{235}U . 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 can be changed in the range between void and fully reflective. The number of FAs of each type is assumed to be constant. This core is mainly used with a single objective FF (optimizing k_{eff}) for the qualification of the algorithm during its development stages and for studying the effect of boundary conditions on the symmetry of the solutions. A schematic view of a typical initial core layout and initial loading pattern is given in Fig. 4.

3.2. Benchmark problem #2

This core is a 100% MOX PWR core designed to maximize Pu consumption (Fridman and Kliem, 2011). It is arranged in an optimized equilibrium LP with optimized burnable poison (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 wet annular burnable absorber (WABA) rods, and once and twice burned FAs. The LP has 1/8 core symmetry. A quarter core section is shown in Fig. 5. Since this core is an equilibrium core and contains once and twice burnt FAs, a three-dimensional spatial burnup distribution is accounted for during the full core three-dimensional simulations. This core was used with a double objective FF for optimizing k_{eff} and PPF.

The unit cell calculations including the generation of few-group cross section data sets were performed with the HELIOS lattice code, utilizing the current coupling and collision probabilities methodology for solving the transport equation in a two-dimensional (2D) unstructured mesh and the ENDF/B-VI evaluated data files with 190 neutron and 48 gamma energy groups (Fridman and Kliem, 2011).

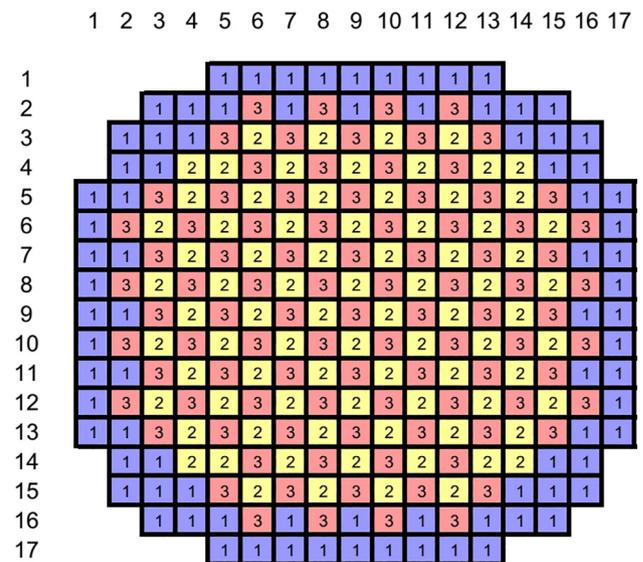


Fig. 4. A schematic layout of a typical initial loading pattern of a simplified PWR core (core #1). Fuel types 1/2/3 (also distinguished by different colors) represent different enrichment levels of 3.1/2.4/1.6 w/o ^{235}U , respectively (Israeli and Gilad, 2017a,b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

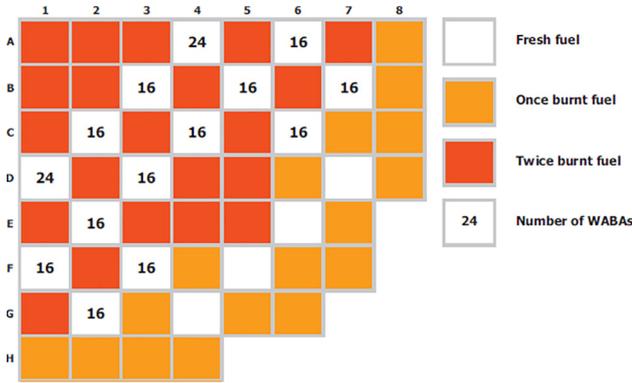


Fig. 5. A schematic layout of a quarter core layout and equilibrium loading pattern of a MOX PWR core (Fridman and Kliem, 2011; Israeli and Gilad, 2017a) (core #2).

3.3. The core simulator

The core simulator is used to evaluate the physical quantities of the solutions (LPs), e.g., k_{eff} and PPF using full core three-dimensional model, in order to calculate their FF. The core simulator used in this study is DYN3D (Grundmann et al., 2000), which is a few-group diffusion code for three-dimensional steady-state and transient core calculations in square and hexagonal fuel element geometry with thermal hydraulic feedback. The code, developed at Helmholtz Zentrum Dresden-Russendorf (HZDR), can also perform detailed depletion calculations. The two- or multi-group neutron diffusion equation is solved by nodal expansion methods. A thermal-hydraulic model (FLOCAL) of the reactor core and a fuel rod model are implemented in DYN3D. The reactor core is modeled by parallel coolant channels which can describe one or more fuel elements. In this work, the code is used only for static (i.e., eigenvalue) calculations without thermal-hydraulic feedback.

3.4. Result presentation

The code written and used for this study is an amalgam of different genetic operators and schemes, each with many different parameters, each with many different possible values. In the research on which this article is based, many different combinations of those parameters are tested. Since specifying **all** operator parameter choices for each result presented would burden the text, only data relevant for each result or comparison is presented.

4. Results for benchmark problem #1

4.1. Adaptive geometric crossover

Influence of the adaptive geometric crossover, that limits segment size beyond a certain generation (g_x , see Eqs. (7) and (8)), is manifested in a dramatic improvement of the results. An evolution with $g_x < 50$ leads to rapid convergence to bad solutions. Good results start appearing above $g_x \geq 50$ and peak around $g_x = 200$, as shown in Figs. 6 and 7. The graphs present the averaged results of several realizations using geometric crossover with decreasing segment size (#4 from Section 2.6).

Results shown in said graphs indicate that the larger g_x is, i.e., the later the crossover swapped segment size is restricted, the slower the convergence rate is. This phenomenon can be explained through the understanding that swapping small segments only allows for making small adjustments to current population. In other words, it is oriented towards finding local optima of the current search space area. Hence, limiting the size of the swapped seg-

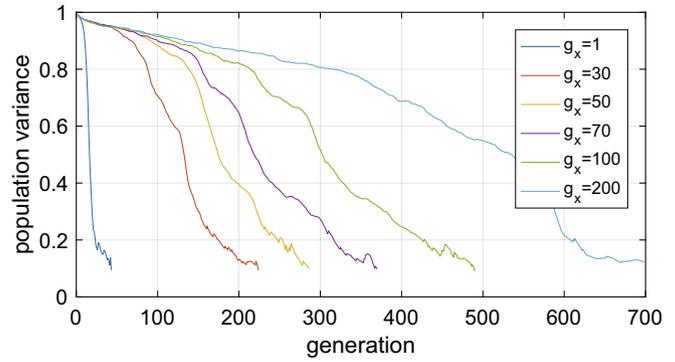


Fig. 6. The effect of an adaptive “square of core neighbors” crossover on convergence. Evolution with $g_x < 50$ leads to rapid convergence to bad solutions.

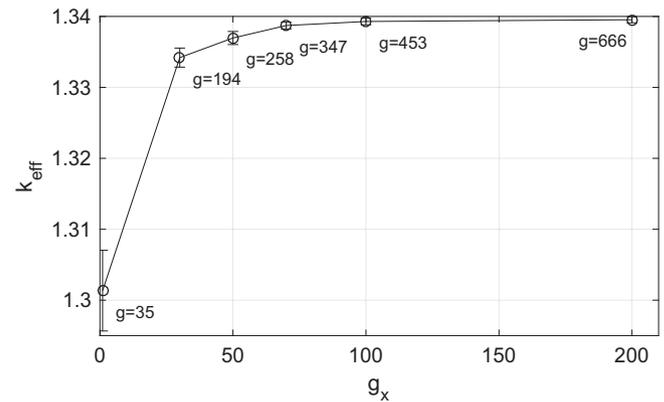


Fig. 7. The effect of an adaptive “square of core neighbors” crossover on k_{eff} . For every g_x in the graph, g is the average generation in which the evolution halted (due to variance drop, see Fig. 6).

ment in the early stages of evolution hinders population’s ability to explore the search space.

4.2. Population variance and selection pressure

Higher m values in Eq. (6) increase the selection pressure, which dramatically affects the convergence rate and the genetic diversity of the population, as shown in Fig. 8. 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 optima and sample larger areas of the search space. The graph presents the averaged results of several realizations using LR selection.

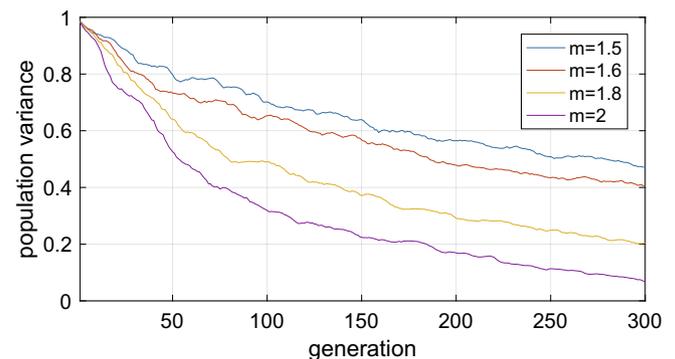


Fig. 8. Population variance convergence for different selection pressures (m values).

4.3. k_{eff} and selection pressure

It can also be noted that the selection pressure induced by the value of m in the *expVal* formula (Eq. (6)) has significant 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 local optima results, while a pressure too low hinders the population's convergence. This phenomenon can be seen in Fig. 9. It presents the averaged k_{eff} values of optimization with different m values. Every point on the graph is the average value of several realizations using LR selection.

4.4. Symmetry and boundary conditions

The assumption of symmetrical loading patterns dominantly underlies the entire field of loading pattern optimization of nuclear

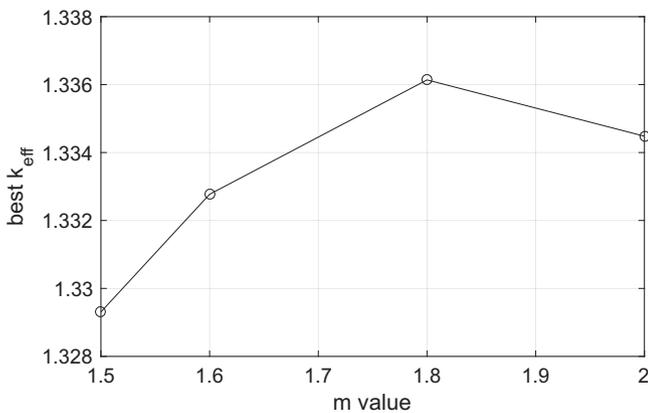


Fig. 9. The averaged k_{eff} for different selection pressures (m values).

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 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 LPs are **not** symmetric and are very counter-intuitive.

Consider a bare reactor core with void boundary conditions. Using our intuition as reactor 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. 10a. 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. 10f, is far from the immediately intuitive symmetric design. The range of optimized LPs corresponding to the range of different boundary conditions is also shown in Fig. 10b–e. As shown, increasing the reflector gain from 0 (void) to 1 (reflective) results in the gradual transition of the bulk of more enriched FAs towards the boundary, while keeping the clear separation between the differently-enriched FAs.

This is one of the predicted possible results of the exercise, another one being the splitting of the single large central cluster of highly enriched FAs to two smaller clusters, one remaining in

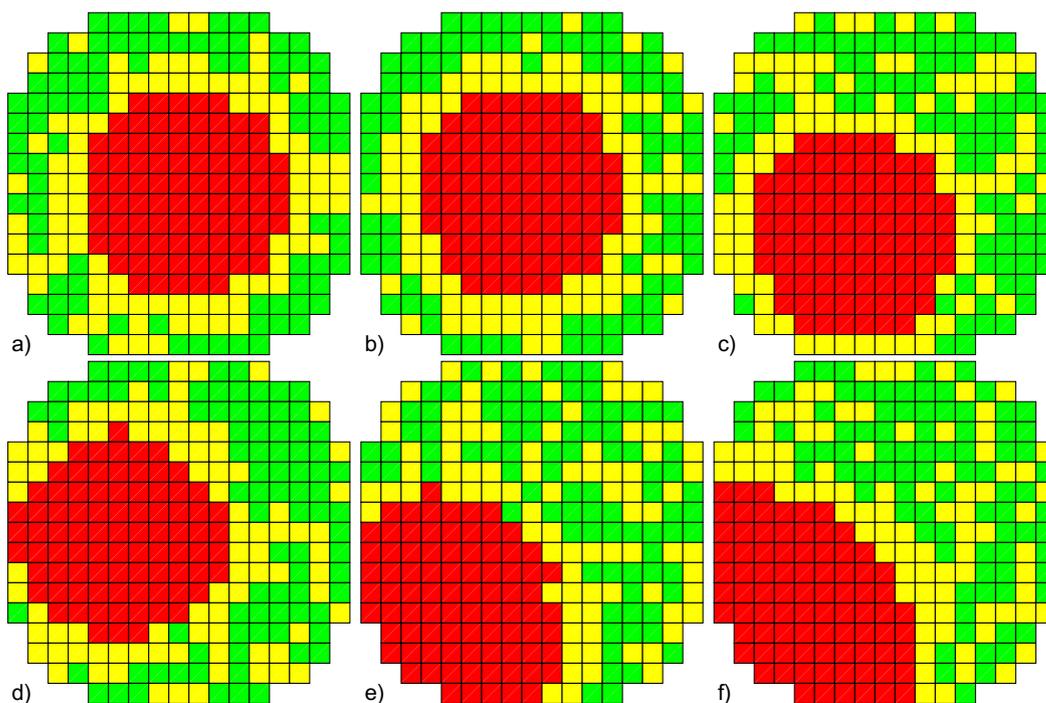


Fig. 10. Boundary conditions' effect on the symmetry of the best LP for maximal k_{eff} . The albedo values are 0 (void), 0.8, 0.85, 0.87, 0.9, and 1 (reflective) for plots a–f, respectively. Different colors indicate different enrichment, with red (green) indicating high (low) enrichment. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the center while the other one migrates to the corner of the core. In retrospect the physical reasoning for the solution is simple: it exhibits the optimal balance between neutron leakage due to proximity to core boundaries on the one hand and the boundaries neutronic reflection on the other. This explanation is corroborated by the eventual results.

It can be seen that the fully reflective boundaries core solution has 90 degrees symmetry, and the question might be asked why is this orientation the one that emerges. The answer being that the orientation determination is completely chaotic and depends (with extremely high sensitivity, therefore chaotically) on initial conditions and the stochastic progress of the evolution. That is, at some point along the evolution a cluster of highly enriched FAs is assembled at some position in the core. The proximity of near core boundaries, and other clusters, affect the cluster to grow near to them. At the end, the final position of the high enriched cluster cannot be a priori predicted since the core structure itself has 90 degrees symmetry.

Given that modern neutron reflector designs minimize the leakage to approximately 3%, this could imply that there may be slightly asymmetric LPs which are operationally valid and superior to symmetric ones, both in multi-loop NPPs and in SMRs or RRs cores.

5. Results for benchmark problem #2

5.1. Population variance and mutation

As previously mentioned, the genetic variance of the population is a very good measure of population composition and diversity during evolution. The best way to increase genetic variance is through mutations, which introduce new genetic material into the existing gene pool. High genetic variance means the optimization process samples larger portions of the search space and is less prone to local optimum convergence.

Experimenting with different mutation strategies and parameters yields interesting results and reveals new insights into the relationship between mutation rate, population variance, and convergence. Firstly, the constant mutation rate strategy is applied. Examples of the resulting optimizations are shown in Fig. 11. One can see that the behavior of population variance is not consistent. It is very much affected by the random construction of initial population and stochastic changes throughout evolution. This makes it hard to control the convergence of the population and exposes the population to feedback effects that lead to premature convergence, as explained later. Some tactics are attempted to avoid the problems of the constant mutation rate strategy, which are presented here with their effect on the evolution.

One effect of different mutation strategies on the genetic variance of the population is shown in Fig. 12. As demonstrated, high

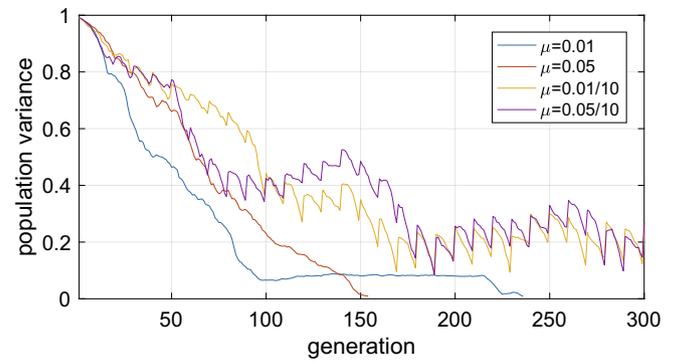


Fig. 12. Population variance convergence for different mutation rates and strategies (μ values).

(low) mutation rate leads to higher (lower) genetic variance throughout the evolution. Also shown is the effect of two different adaptive mutation mechanisms on genetic variance. The first has a constant mutation rate throughout evolution, and the second has a nominal low mutation rate with peak events every 10 generations.

The purpose of the different strategies is to prevent the population from converging too quickly, missing global optima. In Fig. 12 the impact of one of those strategies can be seen. It presents population variance of the constant mutation rate and of the peak mutation rate mutation strategies. The optimization objective of those optimizations is minimizing PPF. It can be clearly seen that the peak mutation succeeds in maintaining higher variance values in later stages of the evolution with respect to constant mutation rate. It does so by increasing mutation rate to $\mu = 10$ every 10 generations, thus maintaining genetic variance. More importantly, peak mutation optimizations achieved lower PPF LPs, as can be seen in Table 1. These results support the claim that higher population variance permits the exploration of the search space, allowing the discovery of better LPs. Fig. 12 and Table 1 present the averaged results of several realizations using different mutation schemes, namely the constant rate and adaptive mutations with base mutation rates of 0.01 and 0.05.

Another aspect of the mutation-variance-convergence relationship reveals itself in the exploration of the decreasing mutation rate and the variance-dependent mutation rate variations of the

Table 1
PPF values for the constant and peak mutation strategies.

Mutation strategy Mutation Rate μ	Constant Mutation Rate	Peak Mutation
0.01	1.4212	1.3622
0.05	1.4329	1.4213

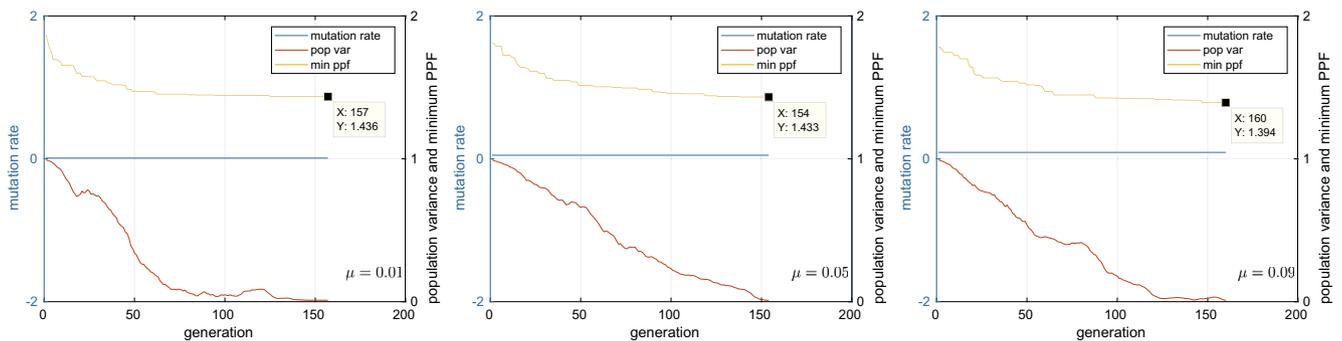


Fig. 11. Constant mutation rate of 0.01, 0.05, and 0.09.

mutation operator. Decreasing mutation rate is used as to allow the population to converge in the final stages of the evolution. The variance-dependent mutation rate increases or decreases the mutation rate (by some pre-defined factor) according to the value of the population variance. That is, two population variance thresholds are determined, high and low. Population variance exceeding the high variance threshold decreases the mutation rate to the smaller one of two possible values: 0.9 of its current value, or 0.02. Alternatively, if the population variance drops below the low threshold, mutation rate is increased to the higher value out of: 2 times the current rate, or 0.9.

This variation of the mutation operator shows clearly the connection and influence of the mutation rate on the variance of the population. Fig. 13 displays the results of an optimization with decreasing, variance dependent, mutation rate. The mutation parameters chosen for this optimization are: a baseline mutation rate of 0.03, a high variance threshold of 0.7, and a low variance threshold of 0.05. It can be seen that population variance is close to 1 in the beginning of evolution and descends along the generations. Note also that as variance is higher than the high threshold of 0.7 in early generations, mutation rate is not only decreased according to the decreasing mutation scheme, but is also multiplied by a factor of 0.9. The decreasing factor is removed once population variance drops under the high threshold. This can be seen in generations 33–34, where population variance drops below 0.7 and mutation rate jumps from 0.018 to 0.02. Another escalation in mutation rate can be seen between the generations 114 and 115, when variance drops below the threshold of 0.05, and mutation rate jumps from a value of 0.011 to 0.022.

It is important to note that the jump in mutation rate slows down the rate of decrease of population variance. The graph of population variance can be seen to go from decreasing rapidly to being convex and slowing down the rate of decrease after both of the mutation rate jumps. This shows that the jump in mutation rate supplies the population with genetic diversity to slow down convergence and allow further optimization.

This relationship between mutation rate and population variance can be seen clearer still in Fig. 14, which presents the results of an optimization with decreasing, variance-dependent, mutation rate, and the following mutation parameters: baseline mutation rate of 0.03, a high variance threshold of 0.8, and a low variance threshold of 0.4.

The phenomenon of population variance drop, mutation jump and variance graph convexity can be seen clearer yet in Fig. 15, which shows the results of an optimization with decreasing, variance-dependent, mutation rate, and the following mutation parameters: baseline mutation rate of 0.03, a high variance threshold of 0.8, and a low variance threshold of 0.05. Comparing the two figures it should be noted that the higher variance threshold of the

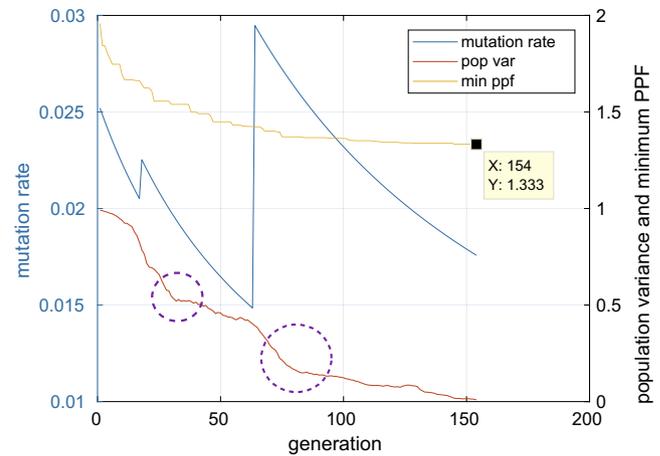


Fig. 14. Variance-dependent, decreasing mutation rate. Baseline mutation rate of 0.03, low (high) variance threshold of 0.4 (0.8).

Fig. 15 optimization causes an earlier mutation rate increase that elevates the entire mutation rate graph. This promotes more genetic diversity throughout the evolution, which, as before, slows down convergence and allows further optimization (resulting in better final solutions).

The effect of raising the lower variance threshold can be observed in Fig. 16, which present the results of optimizations with decreasing, variance-dependent, mutation rate, and the following mutation parameters: baseline mutation rate of 0.03, a high variance threshold of 0.6 and 0.5, respectively, and a raised low variance threshold of 0.1. In both optimizations there can be seen a drop in population variance under the low threshold of 0.1 around the 115th generation, which is answered by multiplying mutation rate by a factor of 2. The effect of the raised mutation rate is apparent, and a “bump” in population variance can be seen around the 150th generation, when variance rises back up above the low threshold. The low threshold can be understood, then, to provide the population with genetic variance toward the end of evolution, when variance is very low. That is, it provides a “last attempt” at escaping local minima in which the population might have been stuck before final convergence.

Fig. 17 shows the benefit of raising both thresholds. Firstly, the entire graph of mutation rate is raised, so the population is provided with more genetic variety throughout the evolution. Secondly, the “last attempt” mutation rate bump occurs earlier in the evolution, and supplies the population with a diversity injection which helps it to avoid stagnation. Specifically, in this optimization, stagnation can be seen to occur between generations 50 and 80, where population variance observably goes down and drops below the low threshold of 0.4 whereas the minimal PPF hardly changes. Mutation rate is then multiplied by 2, injecting the population with new genetic data. Two things then occur that should be noted: first, further optimization of the minimal PPF is found, and second, the decrease rate of the population variance is slowed down and its graph becomes convex.

Summing the analyses, the phenomena are interlinked thus: An individual is produced during the evolution, which is fittest of all others in the current population, reflecting the existence of a local minimum in the search space. It then begins taking over, which causes a decrease in variance and reduces search effectiveness. This is a positive feedback loop that left undisturbed causes premature convergence. The variance threshold method (or double-threshold method) is one way to introduce genetic variance in response to the search state (i.e., in an *adaptive* manner) and curbing the feedback process, to allow escape from local minima and further optimization.

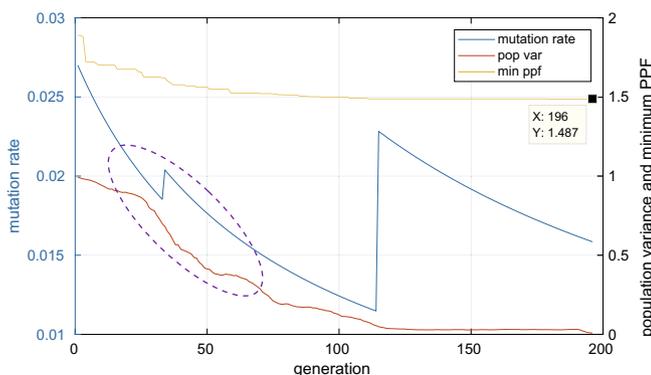


Fig. 13. Variance-dependent, decreasing mutation rate. Baseline mutation rate of 0.03, low (high) variance threshold is 0.05 (0.7).

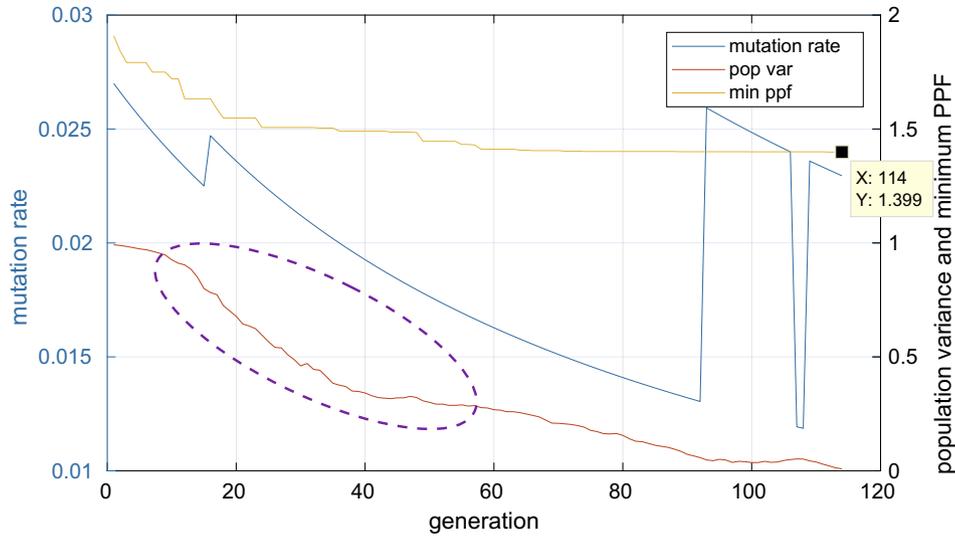


Fig. 15. Variance dependant, decreasing mutation rate. Baseline mutation rate of 0.03, low (high) variance threshold of 0.05 (0.8).

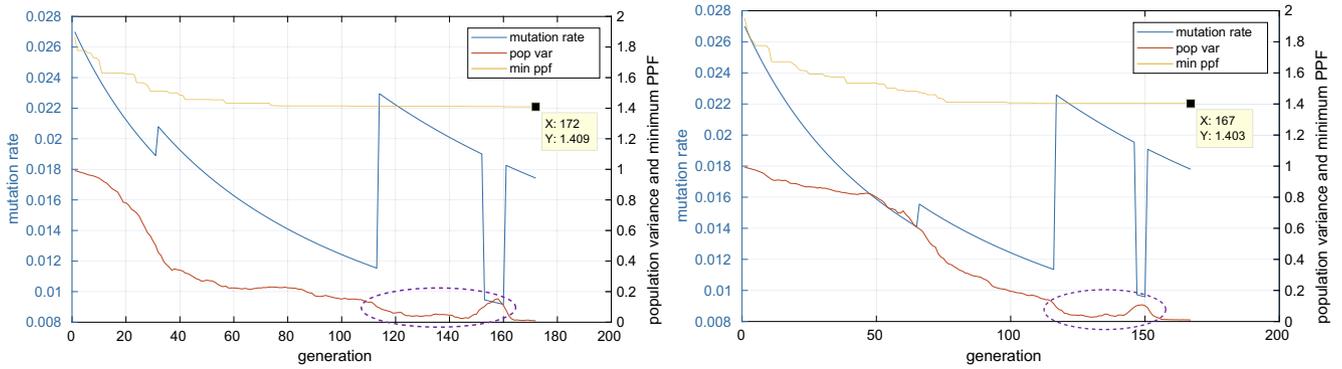


Fig. 16. Variance dependant, decreasing mutation rate. Baseline mutation rate of 0.03, low (high) variance threshold of 0.1 (0.5 left, 0.6 right).

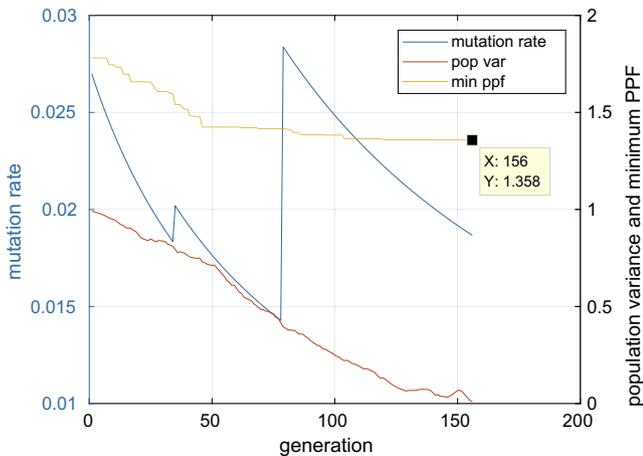


Fig. 17. Variance-dependent, decreasing mutation rate. Baseline mutation rate of 0.03, low (high) variance threshold of 0.4 (0.8).

5.2. Random vs. geometrical crossover

When optimizing k_{eff} alone, the geometrical crossover proves most successful. When introducing the objective of minimizing PPF as well, it is necessary to compare the two crossover methods once again. The geometric, “chunk swapping” crossover is 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, a priori, 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 initially seems to lack the required resolution. This is the reason for the introduction of decreasing segment size into crossover. It is introduced with the purpose of allowing 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 at first 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 does 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 Table 2 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

Table 2
Random vs. geometrical crossover (Israeli and Gilad, 2017a).

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

k_{eff} for the same PPF values. The results shown all belong to LPs produced via the same GA procedure and parameters, with only crossover method changed. Optimizations are run using non geometric and geometric crossovers, namely 1 and 4 from Section 2.6, both with decreasing segment size.

5.3. Stage FF

When testing the zigzag FF described in Section 2.4.2 a sort of “breathing” effect of the LPs produced along the evolution is observed. That is, the best produced LPs seem to change in configuration according to the optimization objective currently used in the evolution. The cause of the phenomenon is the two very different configurations of the two objectives’ optimal LPs and the reciprocal-relations between them. In the stages of k_{eff} optimization the LPs produced resemble the optimal k_{eff} core, i.e., concentrating the high enrichment fuel in core center, while in the PPF

optimization stages the LPs seem to spread high enrichment FAs more evenly throughout the core and surround them with low enrichment FAs. The breathing effect is the continuous “swinging” of the best produced LPs between the two different patterns throughout the evolution, without convergence. The effect is demonstrated in Fig. 18, where a representative section of the evolution is shown. This indicates the flaw in the FF, which prevents convergence, as discussed in Section 2.4.2.

Fig. 19 features two histograms of the evolution of one of the optimizations. It presents the distribution of the population in every generation of the evolution with 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 histograms presented allow one to follow those values in the population through the evolution.

Through these histograms the effect of the stage FF on the evolution is plainly displayed. The histograms are the result of an optimization with a stage FF variation in which the objective change takes place once half the population reaches PPF threshold. One can observe the rather random distribution in both values at beginning of evolution, the objective switch at the 169th generation, as well as the process of convergence to a single parameter value (at around the 300th generation). These phenomena are apparent in the histograms, where an evident shift to the left (the lower PPF region) of the populations’ values during the PPF minimization stage, and a swing back to the right after the optimization moves to k_{eff} can be seen. The LP obtained using the stage FF, which cor-

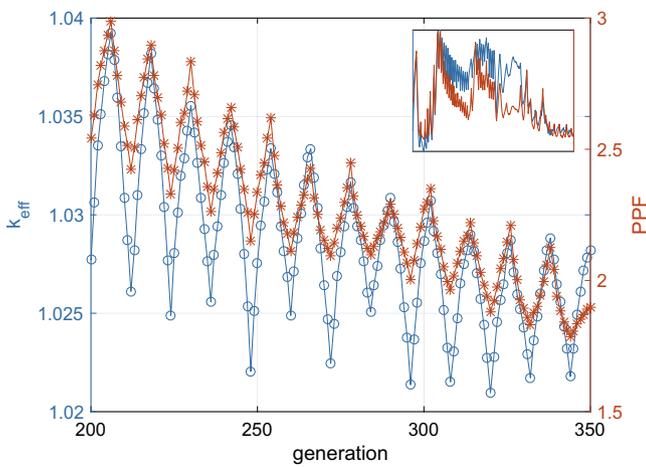


Fig. 18. The “swinging” effect of the zigzag FF demonstrated in the evolution process over a representative section of the evolution. The inset shows the full length evolution demonstrating lack of convergence.

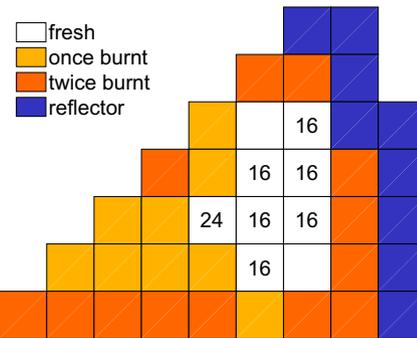


Fig. 20. The LP obtained using the stage FF, which corresponds to the optimization shown in Fig. 19 with $k_{eff} = 1.0088$ and $PPF = 1.29$. The number in the fresh FAs indicates the number of WABA rods.

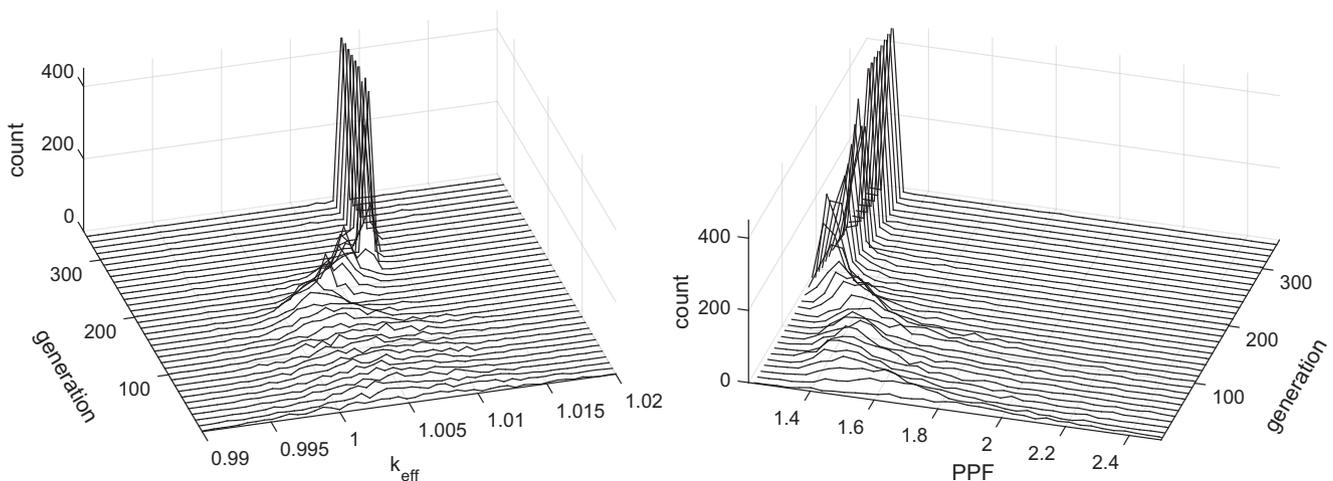


Fig. 19. Two histograms representing the evolution of the distribution of the population with respect to k_{eff} and PPF separately. The population’s size in this simulation is 500 and the number of generations is 1000.

Table 3
Stage FF comparison of different stage lengths (Israeli and Gilad, 2017a).

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

responds to the optimization shown in Fig. 19 is shown in Fig. 20 with $k_{eff} = 1.0088$ and $PPF = 1.29$.

Another purpose the stage FF serves is allowing investigation into the structure of the problem. It allows checking the limits of the PPF of LPs while maintaining a high multiplication factor. To work with lower PPF thresholds, the population must be allowed enough generations to migrate to a PPF optimized area of the search space. The averaged results of several realizations can be seen in Table 3.

6. Conclusions

New genetic algorithms are developed by using up-to-date selection operators, novel crossover operator 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 adaptive geometric crossover (Sections 2.6 and 4.1) proves better than the standard (non-geometric) one. Results shown in Figs. 6 and 7 indicate that obtaining globally good solutions is more easily achieved by delaying the limitation of the segment size to later stages of the evolution (larger g_x , see Eqs. (7) and (8)), after the population has already travelled sufficiently to close in on global optima areas.

Additionally, the proposed geometric crossover takes the geometrical nature of the problem at hand into consideration, utilizing regularly overlooked information. Results show that using the adaptive geometric crossover results in better LPs not only for the simpler k_{eff} optimization but also for the dual objective optimization of both k_{eff} and PPF.

The genetic variance of the population is defined (Section 2.3) and proposed as a valuable measure for monitoring and controlling the convergence of the algorithm. The behavior of the genetic variance throughout the evolution process is successfully used in studying other parameters of the algorithm, e.g., adaptation of the geometric crossover (Fig. 6), selection pressure (Fig. 8), and mutation strategies (Fig. 12).

It is shown that higher genetic variance diminishes the likelihood of converge to **local** optimum and that when genetic variance drops rapidly, the population essentially becomes homogeneous so there is no meaning for its size anymore. The results mentioned above indicate that the genetic variance of the population can be used for real-time parameters' adaptation for better control of the algorithm performances.

An important phenomenon worth noting relates to mutation rate and its relation to the rate of change of the population variance. When an individual superior to the rest of the current population is formed during the evolution, it is likely to take over and begin a positive feedback loop of variance decrease that results in premature convergence to the local minimum found. In other words, local minima are potential stagnation points of the search. Variance-dependent mutation rate is one way of curbing those feedback effects. It does so by injecting the population with genetic diversity **when needed**, thus increasing the potential of further optimization and escape from local minima. The comprehensive research, selected results of which are presented in this article,

shows that adaptive mutation schemes are highly beneficial to the optimization process and provide additional and useful control measures over the stochastic optimization process.

Increased selection pressure (higher m values in Eq. (6)) dramatically affects the convergence rate and the genetic diversity of the population (Fig. 8). However, too great a pressure results in premature convergence to local optima results, while a pressure too low hinders the population's convergence. This phenomenon can be seen in Fig. 9.

The effect of different mutation strategies (constant and peak mutation rate) on the algorithm performances are studied (Fig. 12). The results indicate that as long as the mutation strategy is able to sustain the genetic variance of the population above some level (especially in later stages of the evolution), thus preventing the complete genetic homogenization of the population, the obtained results are improved (e.g., Table 1). These results support the claim that higher population variance permits wider exploration of the search space, allowing the discovery of better LPs.

The intuitive, yet untested, assumption as to LPs' symmetry has been put to the test. In the modern GA optimization world, advanced methods discard, if possible, any potentially limiting influences of human intuition. An example of such limiting influences can be seen in the case of the completely reflective boundary conditions (Fig. 10). It shows that in some cases, placing synthetic symmetry restrictions upon the LPs can prevent the creation of the optimal one. The example also shows 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.

Finally, the multi-objective optimization problem has been addressed in a new way, in an attempt to simplify it and minimize the objectives from interrupting one another (Fig. 19). 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.

Acknowledgments

The authors would like to extend their gratitude to Prof. (Emeritus) Alex Galperin, Ben-Gurion University of the Negev, for helpful discussions. The research was partially funded by the Israeli Ministry of Energy, contract number 216-11-008.

References

- Alim, Fatih, Ivanov, Kostadin, Levine, Samuel H., 2008. New genetic algorithms (GA) to optimize PWR reactors part I: Loading pattern and burnable poison placement optimization techniques for PWRs. *Ann. Nucl. Energy* 35, 93–112.
- Aydogan, Fatih, 2016. Advanced small modular reactors. In: Pioro, Igor L. (Ed.), *Handbook of Generation IV Nuclear Reactors*, Woodhead Publishing Series in Energy, chapter 20. Woodhead Publishing, pp. 661–699.
- Bäck Thomas, The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In: MännerR., Manderick, B., (eds.), *Proceedings of the 2nd Conference Parallel Problem Solving from Nature*, pages 85–94. Amsterdam: North-Holland, 1992.
- Baker James E., Adaptive selection methods for genetic algorithms. In: Grefenstette John J., ed., *Proceedings of International Conference on Genetic Algorithms and their application*, pp. 101–111. Lawrence Erlbaum Associates, 1985.
- Blickle Tobias, Thiele Lothar, A comparison of selection schemes used in evolutionary algorithms. *techreport*, Swiss Federal Institute of Technology (ETH), Gloriastrasse 35, 8092 Zurich, Switzerland, December 1995. TIK-Report, Nr. 11, Version 2 (second ed.).
- Chapot, Jorge Luiz C., Da Silva, Fernando Carvalho, Schirru, Roberto, 1999. A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem. *Ann. Nucl. Energy* 26, 641–655.
- David Schaffer, J., Caruana, Richard A., Eshelman, Larry J., Das, Rajarshi, 1989. A study of control parameters affecting online performance of genetic algorithms for function optimization. In: David Schaffer, J. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 51–60.

- DeChaine, Michael D., Feltus, Madeline Anne, 1995. Nuclear fuel management optimization using genetic algorithms. *Nucl. Technol.* 111 (1), 109–114.
- De Jong, Kenneth A., 1975. Analysis of the Behavior of a Class of Genetic Adaptive Systems PhD thesis. University of Michigan.
- de la Maza Michael, Tidor Bruce, Boltzmann weighted selection improves performance of genetic algorithms. Technical report, MIT Artificial Intelligence Laboratory, 1991. A.I. Memo 1345.
- de la Maza, Michael, Tidor, Bruce, 1993. An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. In: Forrest, Stefanie (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, CA, pp. 124–131.
- Do, Binh Quang, Nguyen, Lan Phuoc, 2007. Application of a genetic algorithm to the fuel reload optimization for a research reactor. *Appl. Math. Comput.* 187 (2), 977–988.
- Eiben, Ágoston Endre, Hinterding, Robert, Michalewicz, Zbigniew, 1999. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.* 3 (2), 124–141.
- Fogarty, Terence C., 1989. Varying the probability of mutation in the genetic algorithm. In: David Schaffer, J. (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 104–109.
- Fridman, Emil, Kliem, Sören, 2011. Pu recycling in a full Th-MOX PWR core. Part I: steady state analysis. *Nucl. Eng. Des.* 241, 193.
- Friedrich, Tobias, Oliveto, Pietro S., Sudholt, Dirk, Witt, Carsten, 2009. Analysis of diversity-preserving mechanisms for global exploration. *Evol. Comput.* 17 (4), 455–476. PMID:19916783.
- Gang Peng, Feng Peng, Rong Fu, Application of genetic algorithm in research and test reactor core loading pattern optimization. In: PHYSOR 2002, Seoul, Korea, 2002.
- Goldberg, David E., 1989. Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Goldberg, David E., Deb, Kalyanmoy, 1991. A comparative analysis of selection schemes used in genetic algorithms. In: Rawlins, Gregory J.E. (Ed.), Foundations of Genetic Algorithms, vol. 1. Morgan Kaufmann Publishers, pp. 69–93.
- Grefenstette, John J., 1986. Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybern.* 16 (1), 122–128.
- Grefenstette, John J., Baker, James E., 1989. How genetic algorithms work: a critical look at implicit parallelism. In: David Schaffer, J. (Ed.), Proceedings of the third international conference on Genetic algorithms. Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 20–27.
- Grundmann Ulrich, Rohde Ulrich, Mittag Siegfried, DYN3D – Three-dimensional core model for steady state and transient analysis of thermal reactors. In: PHYSOR 2000, Pittsburgh, PA, USA, 2000.
- Hongchun, Wu., 2001. Pressurized water reactor reloading optimization using genetic algorithms. *Ann. Nucl. Energy* 28, 1329–1341.
- International Atomic Energy Agency. Advances in small modular reactor technology developments. techreport, International Atomic Energy Agency, Vienna International Centre, PO Box 100, 1400 Vienna, Austria, 2014. A Supplement to IAEA Advanced Reactors Information System (ARIS).
- Israeli, Ella, 2016. In-Core Fuel Management using Genetic Algorithms Master's thesis. Ben-Gurion University of the Negev, Israel.
- Israeli Ella, Gilad Erez, Novel core physics heuristics in advanced genetic algorithms for in-core fuel management. In: M&C 2017 – International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering, Jeju, Korea, April 16–20, 2017.
- Israeli, Ella, Gilad, Erez, 2017b. Novel genetic algorithms for loading pattern optimization using state-of-the-art operators and a simple test case. *J. Nucl. Eng. Radiat. Sci.* 3, 030901-1.
- Jayalal, M.L., Satya Murty, S.A.V., Sai Baba, M., 2014. A survey of genetic algorithm applications in nuclear fuel management. *J. Nucl. Eng. Technol.* 4 (1), 45–62.
- Khoshahval, Farrokh, Minuchehr, Hamid, Zolfaghari, Ahmad, 2011. Performance evaluation of PSO and GA in PWR core loading pattern optimization. *Nucl. Eng. Des.* 241, 799–808.
- Leung, Yee, Gao, Yong, Zong-Ben, Xu., 1997. Degree of population diversity – a perspective on premature convergence in genetic algorithms and its markov chain analysis. *IEEE Trans. Neural Networks* 8 (5), 1165–1176.
- Lobo, Fernando G., Lima, Cláudio F., Michalewicz, Zbigniew (Eds.), 2007. Parameter Setting in Evolutionary Algorithms, Studies in Computational Intelligence. Springer Verlag.
- Norouzi, A., Aghaie, M., Mohamadi Fard, A.R., Zolfaghari, A., Minuchehr, A., 2013. Nuclear reactor core optimization with parallel integer coded genetic algorithm. *Ann. Nucl. Energy* 60, 308–315.
- Ortiz, Juan Jose, Requena, Ignacio, 2004. An order coding genetic algorithm to optimize fuel reloads in a nuclear boiling water reactor. *Nucl. Sci. Eng.* 146 (1), 88–98.
- Parks, Geoffrey T., 1996. Multiobjective PWR reload core design by nondominated genetic algorithm search. *Nucl. Sci. Eng.* 124, 178–187.
- Pereira, Cláudio Márcio do Nascimento Abreu, Schirru, Roberto, Martinez, Aquilino Senra, 1999. Basic investigations related to genetic algorithms in core designs. *Ann. Nucl. Energy* 26 (3), 173–193.
- Smith, J., Fogarty, T.C., 1996. Self adaptation of mutation rates in a steady state genetic algorithm, Proceedings of the IEEE Conference on Evolutionary Computation. IEEE Press, Piscataway, NJ, USA, pp. 318–323.
- Srinivas, M., Patnaik, L.M., 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. Syst. Man Cybern.* 24 (4), 656–667.
- Toshinsky, Vladimir G., Sekimoto, Hiroshi, Toshinsky, Georgy I., 1999. Multiobjective fuel management optimization for self-fuel-providing LMFBR using genetic algorithms. *Ann. Nucl. Energy* 26, 783–802.
- Turinsky, Paul J., 2005. Nuclear fuel management optimization: a work in progress. *Nucl. Technol.* 151, 3–8.
- Turinsky, Paul J., Keller, Paul M., Abdel-Khalik, Hany S., 2005. Evolution of nuclear fuel management and reactor operational aid tools. *Nucl. Eng. Technol.* 37 (1), 79–90.
- Whitley, Darrell, 1989. The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In: David Schaffer, J. (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, pp. 116–121.
- Whitley, Darrell, 2001. An overview of evolutionary algorithms: practical issues and common pitfalls. *Inf. Softw. Technol.* 43 (14), 817–831.
- Zameer, Aneela, Mirza, Sikander M., Mirza, Nasir M., 2014. 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. *Ann. Nucl. Energy* 65, 122–131.