MYRRHA CORE LOADING PATTERN OPTIMIZATION USING METAHEURISTICS

David Jaluvka\textsuperscript{a,b}, Gert Van den Eynde\textsuperscript{a}, Stefan Vandewalle\textsuperscript{b}

\textsuperscript{a} SCK\textsuperscript{*}CEN: Boeretang/200, Mol, Belgium, 2400, djaluvk\textsuperscript{a}@sckcen.be
\textsuperscript{b} University of Leuven: Celestijnenlaan/200A, Heverlee, Belgium, 3001

The goal of the nuclear reactor in-core fuel management is to analyze and propose new core configurations to be loaded into a reactor core at the beginning of fuel cycles. Each new core loading has to comply with the nuclear facility’s operating objectives, limits, and safety requirements. Given a stock of available nuclear fuel, a search among the possible candidates is usually done in order to identify a fuel loading pattern that also optimizes the fuel usage during the operation. This can be achieved using metaheuristic optimization techniques.

In this paper, a special computer program that addresses the problem of in-core fuel management optimization of MYRRHA, a new irradiation facility currently under development at SCK\textsuperscript{*}CEN, is briefly described. A simple MYRRHA in-core fuel management optimization problem is solved using two different metaheuristic methods – Genetic Algorithm (GA) and Ant Colony Optimization (ACO). The obtained results are compared and the efficiency of the applied algorithms is discussed afterwards.

The MYRRHA in-core fuel management optimization problem discussed in the paper aims at a maximum irradiation performance of the machine expressed in terms of the neutron flux achieved at experimental channels. It is shown that both GA and ACO provide feasible solutions that outperform an intuitively designed loading pattern.

I. INTRODUCTION

At every nuclear facility operating in cycles with a spatially separated fuel bundles, a nuclear fuel loading pattern of elements to be loaded into the reactor core at the beginning of each fuel cycle has to be designed in such a way that all operating objectives, limits, and safety requirements are met and, in the ideal case, the fuel consumption is minimized at the same time. The problem of identifying a loading pattern that complies with all design criteria and maximizes the fuel utilization is called In-Core Fuel Management Optimization Problem (ICFMOP).

ICFMOP is a well-known nuclear engineering optimization problem classified as an NP-hard, nonlinear, non-convex, multimodal, combinatorial optimization problem with a large search space. In the past, the problem has been usually tackled based on expert knowledge. Some classical mathematical programming methods like Linear Programming\textsuperscript{1} (LP) or Mixed-Integer Nonlinear Programming\textsuperscript{2} (MINLP) have also been applied to the core reloading problem. Most recently, various metaheuristic optimization methods have been used for solving ICFMOP, among them Simulated Annealing\textsuperscript{3} (SA), Genetic Algorithm\textsuperscript{4,5} (GA), Ant Colony Optimization\textsuperscript{6-11} (ACO), Particle Swarm Optimization\textsuperscript{12} (PSO), Tabu Search\textsuperscript{13} (TS), and others.

Metaheuristics are a group of black-box iterative optimization methods that find a
solution to highly complex optimization problems by searching over a large set of feasible solutions. Despite the high computational burden – a consequence of a large number of evaluated trial solutions, metaheuristics have established themselves as highly efficient and extensively applied techniques for solving ICFMOP for power generating reactors, delivering the best results so far.

The novelty of the presented work is the application of metaheuristic optimization methods to the in-core fuel management optimization problem for a specific class of nuclear reactors, i.e., for the material testing reactors or MTRs. Unlike the case of conventional light water cooled and moderated reactors designed primarily for power production, MTRs aim at providing excellent irradiation conditions for material tests and other irradiation experiments. This implies specific criteria for designing and optimization of MTR fuel loading patterns.

In this paper, a simple constrained ICFMOP of MYRRHA, an MTR currently under development at SCK•CEN in Mol, Belgium, is solved using two metaheuristic optimization methods – GA and ACO. While some results have already been published on GA applied to the MYRRHA ICFMOP (Ref. 14 and 15), the results obtained using ACO are completely new.

In the first section of the paper, the MYRRHA reactor, some specific aspects of the MYRRHA in-core fuel management and the tool for solving MYRRHA ICFMOP are briefly described. An illustrative MYRRHA ICFMOP is defined in the second section followed by the description of the applied GA and ACO algorithms and by the discussion of the obtained results. Conclusions are presented in the last section of the paper.

II. MYRRHA IN-CORE FUEL MANAGEMENT

Fig. 1. MYRRHA-FASTEF core model.

MYRRHA (Multi-purpose hYbrid Research Reactor for High-tech Applications) is a flexible fast spectrum MTR of 50-100 MWth power output conceived as an accelerator driven system, able to operate in both sub-critical and critical mode[16]. The foreseen fuel cycle length of MYRRHA is 90 days followed by a 30 days long refueling and maintenance period. Each third cycle the maintenance period lasts 90 days. In the MYRRHA-FASTEF design[17], a five batch in-out refueling scheme is assumed with a total number of 69 fuel assemblies (FAs) containing 34.5 wt.% Pu enriched MOX fuel. The simplified core exhibits a one-third core symmetry.

Fig.1 shows the complete MYRRHA-FASTEF core arrangement. The letters indicate different types of core cells: F0 – fresh FA, F1-F4 – FA irradiated in one to four equilibrium cycles assuming the reshuffling scheme depicted in the figure, a – experimental channels/in-pile sections (IPSs), b – control rods (CR), c – scram rods (SR), d-e – dummy subassemblies (SAs), f-i – barrel approximations, j – outer coolant.

The multi-purpose nature of the facility, its flexible design, and various modes of operation imply large number of possible variables to be taken into account during the MYRRHA loading pattern optimization. Therefore, it was decided that the metaheuristic optimization
methods will be used for the MYRRHA in-core fuel management optimization, which will allow for a use of arbitrary objective and constraint functions.

II.A. MYRRHA Loading Pattern Optimization Tool

A special computer program for solving ICFMOPs is being developed as a part of the MYRRHA project. This tool provides an interface between an arbitrary metaheuristic optimization method suitable for tackling ICFMOP and an arbitrary reactor physics code suitable for MYRRHA core modeling – a black-box LP evaluator that returns requested LP characteristics back to the optimization algorithm. In this way, basically any LP characteristic that can be computed in a reasonable time can be optimized.

DIF3D10.0 (Ref. 18 and 19) is used in the core management tool for calculation of effective multiplication factor $k_{eff}$ and 3D neutron flux and power distributions. Maximum fuel pin cladding temperature $T_{clad}$ is computed based on a simple analytical sub-channel model\textsuperscript{20}.

III. ILLUSTRATIVE MYRRHA ICFMOP

Since the MYRRHA operating and safety limits are still under discussion, the presented optimization problem should be perceived as a merely illustrative and very simplified problem formulated with the aim to demonstrate in principle the use of the GA and ACO algorithms for solving MYRRHA ICFMOP.

The goal is to find a LP that maximizes the MYRRHA-FASTEF core irradiation performance expressed in terms of the peak neutron flux $\Phi$ [cm$^{-2}$s$^{-1}$] at the experimental channels (objective) and provides a large enough excess of reactivity for 90 days of operation (constraint).\textsuperscript{a} At the same time the maximum hottest pin cladding temperature $T_{clad}$, a function of the core thermal power $P$ [MW$_{th}$] and radial, axial, and fuel assembly components of the power peaking factor ppf, has to be lower than the limit value $T_{clad}^{lim}$ (constraint).\textsuperscript{b}

Hence, the ICFMOP may be defined as follows:

Maximize $\phi(x)$, \hspace{1cm} (1)
subject to $k_{eff}(x) > 1.015$. \hspace{1cm} (2)

The variable $x$ is stays for a LP.

The constraint on the maximum cladding temperature was omitted in (1-2) since it can be always implicitly satisfied by decreasing $P$ accordingly to the following equation:

$$P = P' \frac{T_{clad}^{lim} - T_{in}}{T_{clad}' - T_{in}}$$ \hspace{1cm} (3)

where $P'$ is the reactor power for which the maximum cladding temperature $T_{clad}'$ was evaluated and $T_{in}$ is the inlet coolant temperature.

A fixed number of FAs is given for each refueling batch assuming a one-third core symmetry $n = (5,5,5,4,4)$, where $n_1$ is the total number of F0s in a one third of the LP, $n_2$ is the total number of F1s in the same third of the LP, etc. Only those cells occupied by FAs (F0-F4) are subject to optimization. The rest of the cells stay fixed in their positions as indicated in Fig. 1.

The total number of different core configurations is approximately 2.60E13.

III.A. Solution Methods

In this paper, GA and ACO are applied to the illustrative MYRRHA in-core fuel

\textsuperscript{a} For the sake of simplicity, only the beginning-of-cycle (BOC) steady-state ICFMOP is assumed here.

\textsuperscript{b} This limit corresponds to the lower boundary of the oxygen concentration in the liquid lead-bismuth coolant that should prevent dissolution of the fuel cladding protective oxide layer and thus minimize the corrosive damage caused by the flowing coolant to the cladding.
management problem. In both cases a common constrain handling technique based on the objective function penalization was used, according to which, whenever the constraint (2) is violated, the objective function value is decreased by a penalty term proportional to the extent of the constraint violation. The optimization problem (1-2) can be then redefined as:

Maximize $f_a(x)$, \hspace{1cm} (4)

where $f_a$ is a new (augmented) function to be optimized

$$f_a(x) = \phi(x) - \lambda \delta (1.015 - k_{eff}(x)).$$ \hspace{1cm} (5)

$\lambda$ is the penalization factor, and $\delta = 0$ if $k_{eff}(x) > 1.015$ otherwise $\delta = 1$.

III.A.1. Genetic Algorithm

GA is an evolutionary metaheuristic optimization method based on a progressive improvement of the fittest individuals from generation to generation through applying selection, crossover, and mutation operators.\cite{GA1}

In the GA jargon, the objective function is called fitness, the optimization iteration is called generation, and the LP is an individual represented by its chromosome $x$, a vector consisting of $n$ genes $x = (x_1, x_2, ..., x_n)$.

A GA proceeds in the following steps: (1) an initial population (generation $g = 0$) of $p$ individuals is generated randomly; (2) $f_a(x)$ is calculated for each individual $x$ in the population; (3) $p/2$ pairs of parents are selected from the current population by applying the selection operator; (4) two offspring are produced for each pair of parents by applying the crossover operator with the crossover rate $c$; (5) each offspring is mutated with a mutation rate $m$ by applying the mutation operator; (6) $f_a(o)$ is calculated for each offspring $o$; and (7) a new generation $g \leftarrow g + 1$ is formed by applying the population replacement strategy; the steps 3–7 are repeated until the convergence criterion is satisfied, in this case, until the maximum number of generations is reached. A more detailed description of the algorithm can be found in Refs. 14 and 15. The GA configuration used in the calculations is given in Table I.

In order to prevent infeasible individuals in the population that violate the constraints on the given number of FA types in LP after applying crossover operator, a special procedure based on a simple ranking system was introduced into the GA. The detailed description of this procedure can be found in Ref. 15.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>Integer string</td>
</tr>
<tr>
<td>Selection operator</td>
<td>Tournament selection of size 2</td>
</tr>
<tr>
<td>Crossover operator</td>
<td>Uniform crossover of random size</td>
</tr>
<tr>
<td>Mutation operator</td>
<td>Random swap</td>
</tr>
<tr>
<td>Initial population</td>
<td>Randomly generated</td>
</tr>
<tr>
<td>Population replacement strategy</td>
<td>Generational</td>
</tr>
<tr>
<td>Population size $p$</td>
<td>100 or 50 individuals</td>
</tr>
<tr>
<td>Crossover rate $c$</td>
<td>1.00</td>
</tr>
<tr>
<td>Mutation rate $m$</td>
<td>0.05</td>
</tr>
<tr>
<td>Number of generations</td>
<td>100 or 200 (+ 1 initial)</td>
</tr>
</tbody>
</table>

In Fig. 2, the 2D ↔ 1D mapping used for the solution encoding/decoding is demonstrated on the LP example from Fig. 1.
III.A.2. Ant Colony Optimization

ACO was firstly applied to the ICFMOP by L. Machado and R. Schirru in 2002 (Ref. 6). Since then a few more papers have been published on the topic.7-11

ACO is another Metaheuristics inspired by nature suitable for tackling discrete optimization problems. It was introduced in the early 1990’s and it is based on ants’ foraging behavior. When searching for food, ants leave a chemical pheromone trail on the ground the quantity of which may depend on the quantity and quality of the food source. Other ants, which can smell the pheromone, will then tend to choose paths marked by strong pheromone concentration which enables them to find the shortest path between food sources and the ant hill.22

In ACO, a trial solution to the problem is constructed step-by-step for each ant from a group of ants based on the current pheromone distribution by means of a so-called state transition rule. After all ants have constructed their solution, the pheromone distribution is updated and the optimization continues in a new iteration.

Depending on the form of the state transition rule and the pheromone update rule, several ACO variants have been proposed including Ant-Q algorithm6,11, Ant Colony System7 (ACS), Max-Min Ant System11 (MMAS), or Rank-Based Ant System11 (RAS).

The ACO algorithm used in this work is a simplified variant of the ACS from Ref. 7: (1) an initial pheromone amount is set to a constant value \( \tau_0 \) for all possible movements between two successive LP positions; (2) solutions are constructed for \( k \) ants by applying the state transition rule (see Eq. 6 and 7); (3) \( f_a(x) \) is calculated for every solution \( x \) in the iteration; (4) the pheromone distribution is updated according to the pheromone update rule (see Eq. 8 and 9); the steps 2, 3, and 4 are repeated until the convergence criterion is satisfied, in this case, until the maximum number of iterations is reached.

The state transition rule for selecting the FA type \( s \) to be loaded into the position \( p + 1 \) next to the FA type \( r \) previously loaded into the position \( p \) is defined as follows:

\[
\tau(r, s, p) = \begin{cases} 
\arg\max\{(\tau(r, s, p))^\alpha \times [\eta(r, s, p)]^\beta \} & \text{if } \eta \leq q_0, \\
\text{Roulette} & \text{if } \eta > q_0.
\end{cases}
\] (6)

where for roulette:

\[
\text{Roulette} = \left\{ \begin{array}{ll} 
\frac{[\tau(r, s, p)]^\alpha \times [\eta(r, s, p)]^\beta}{\sum_{s \in f_k} [\tau(r, s, p)]^\alpha \times [\eta(r, s, p)]^\beta} & \text{if } s \in f_k(r), \\
0 & \text{if } s \notin f_k(r).
\end{array} \right.
\] (7)

\( \tau(r, s, p) \) is the amount of the pheromone associated to the transition from the FA type \( r \) to \( s \) moving from the position \( p \) in the core, \( \eta(r, s, p) \) is the heuristic information associated to this transition, \( q \in [0,1] \) is a random number, \( q_0 \in [0,1] \) is a parameter, \( \alpha \) and \( \beta \) are weighting factors, and \( f_k(r) \) is the list of available FA types having a FA of the type \( r \) placed in the position \( p \).

All pheromone trails between all possible pairs of successive nodes at all positions in the reactor core are updated first according to:

\[
\tau(r, s, p) \leftarrow (1 - \rho) \tau(r, s, p) + \rho \tau_0.
\] (8)

Then, only the nodes of the iteration best solution \( x_{best} \) are updated according to:

\[
\text{TABLE II. Ant Colony Optimization parameters.}
\]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ants ( k )</td>
<td>100 or 50 ants</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>101 or 201</td>
</tr>
<tr>
<td>Evaporation rate ( \rho )</td>
<td>0.2</td>
</tr>
<tr>
<td>( \tau_0 )</td>
<td>1.0</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>0.9</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>1.0</td>
</tr>
<tr>
<td>( \beta )</td>
<td>0.0</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>3.7E-16</td>
</tr>
</tbody>
</table>
where $\gamma$ is a scale factor and $\rho$ is so-called evaporation rate. Both equations 8 and 9 constitute the pheromone update rule.

The use of the local heuristics $\eta$ is somehow problematic in case of ICFMOP for multiple reasons. First, it is extremely difficult to define a local heuristics for a movement between two successive positions in the loading pattern as this depends not only on the two particular FA types $x_i$ and $x_{i+1}$ loaded into these positions but also on the actual position of the FA couple in the complete LP scheme represented by the variable $p$ in Eq. 6 and 7. Second, in some works it has been shown that the use of local heuristic information may not necessarily lead to better results when applied to ICFMOP. Therefore, the local heuristics was eliminated in our algorithm by setting $\beta$ to zero. Moreover, the local heuristics $\eta$ from Eq. 6 and 7 would depend on the problem representation, i.e., on the order/track in which a problem solution is constructed. In the presented algorithm, the FA type is first selected for the core position denoted by the small number 1 in the LP from Fig. 2, then for the position denoted by the small number 2, and so on.

The values of the ACO parameters used in the calculations are given in Table II.

### III.B. Results
The constrained ICFMOP formulated at the beginning of Section III was solved first with the constant penalization factor $\lambda = 1.02E16$. The results of the optimization are given in Fig. 3 and 4 for both algorithms. It can be seen from the figures that the maximum objective (augmented) function values are reached faster for ACO. This means that with ACO many high quality results may be obtained already after 10 iterations while for GA around 50 iterations are needed. However, slower convergence of GA may imply more homogeneous search over the search space and consequently a lower probability to be trapped in a local optimum and also a higher probability to find the global optimum.

Similar results were obtained for different forms of the penalization factor $\lambda$. Fig. 5 shows the results obtained for the GA with the dynamic penalization factor $\lambda(i) = 2.04E14 \cdot i$, where $i$ is the number of the current iteration during the optimization, and Fig. 6 shows the results obtained for the ACO with the adaptive penalization factor in the form $\lambda(i + 1, l) = 1.03^{-1} \lambda(i, l)$ if all best solutions in the last $l$ iterations did not satisfy the constraint, or $\lambda(i + 1, l) = 1.05 \lambda(i, l)$ if all best solutions in the last $l$ iterations did satisfy the constraint, or $\lambda(i + 1, l) = \lambda(i, l)$ otherwise; $\lambda(0, l) = 5.00E15$ and $k = 10$. 

Fig. 5. Evolution of the iteration best solution for GA (dynamic penalization) and ACO (adaptive penalization) with 100 trial solutions per iteration.

Fig. 6. GA (dynamic penalization) and ACO (adaptive penalization) solutions in the iterations 0, 5, 10, 25, 50, and 100.
The results of all performed optimizations are summarized in Table III. Results for the GA and ACO algorithms with 50 trial solutions evaluated during 201 iterations are also included in the table. Only the static penalization was used in this case. From the table it follows that all tested optimization algorithms gave feasible results ($k_{eff} > 1.015$) of approximately the same quality ($\Phi \approx 2.983\text{E}15 \text{cm}^{-2}\text{s}^{-1}$). However, the number of the unique solutions evaluated during the execution was almost two times larger in the case of GA than in the case of ACO, which means approximately two fold increase in the GA computational time comparing to ACO.

It should be noted that for several reasons the performance of GA and ACO algorithms in solving nuclear reactor ICFMOP cannot be compared based only on the results presented above. Both algorithms are of the stochastic nature and, therefore, several executions for each algorithm variant would be needed in order to have statistically more reliable data. Also, the algorithms described here are rather very simplified/basic versions of GA and ACO and their performance can be easily improved by some modifications. For instance, a GA with an elitist generation replacement strategy would converge much faster than a GA with the generational replacement strategy assumed here.\cite{15}

Table III contains also the characteristics of the LP based on a simple in-out refueling scheme (IO) and the characteristics of an intuitively designed LP (G). The latter LP gives larger neutron flux values than the best optimized LP, however, the constraint on the required BOC excess of reactivity is not satisfied in this case.

Finally, the best performing GA and ACO LPs are shown together with the simple in-out LP and the guessed LP in Fig. 7.

IV. CONCLUSIONS

Two metaheuristic optimization methods,
Genetic Algorithm and Ant Colony Optimizations were implemented in the developed MYRRHA core management tool and successfully used for solving an illustrative MYRRHA in-core fuel management optimization problem.

Both methods delivered comparable feasible solutions for all three considered forms of the objective function penalization – static, dynamic, and adaptive. However, the number of LP evaluations during the optimization was almost two times lower for ACO than for GA.

An intuitively designed LP was proposed as a solution to the MYRRHA ICFMOP. Although this LP was characterized by an excellent objective function value, it did not satisfy the imposed constraint. The GA and the ACO provided feasible solutions of a comparable quality to the guessed LP.

In the future, the GA and ACO algorithms will be used for solving more realistic MYRRHA loading pattern optimization problems that will account for an average fuel burnup and a core irradiation performance throughout the whole fuel cycle.

ACKNOWLEDGMENTS

This research was performed in the framework of the SCK•CEN Academy PhD programme.

REFERENCES

12. A. A. M. Meneses and R. Schirru, “Particle Swarm Optimization Applied to the Combinatorial Problem in order to Solve the Nuclear Reactor Fuel Reloading Problem.”


