2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Application of the MOAA for the optimization of CORAIL assemblies for nuclear reactors

Valerio Lattarulo*, Benjamin A. Lindley†, Geoffrey T. Parks‡
Engineering Design Centre
Department of Engineering
University of Cambridge
Trumpington Street, Cambridge, CB2 1PZ, UK
Email: [*vl261,†bal29,‡gtp10]@cam.ac.uk

***Abstract*—The Multi-objective Alliance Algorithm (MOAA), a recently introduced optimization algorithm, is used for the optimization of heterogeneous low-enriched uranium (LEU) + mixed-oxide fuel (MOX) assemblies for pressurized water reactors (PWRs). This is a constrained nuclear problem with two objectives and a mixed-integer solution space. The efficacy of the algorithm is demonstrated through comparisons with NSGA-II for between 300 and 2000 function evaluations. Through the epsilon and hypervolume indicators and the Kruskal-Wallis statistical test, we show that the MOAA outperforms NSGA-II on this problem. The MOAA was also able to find a set of solutions that are better than the 'expert solution' for this problem.**

I. Introduction

Nuclear reactors are complex systems necessary for the initialization and the control of nuclear reactions. In nuclear reactors, it is important to load the fuel such that the power distribution across the reactor is as even as possible, while balancing other objectives. This is a complex optimization problem. Generally, the complexity of a system is related to the difficulty of finding efficient solutions, and complex real-world optimization problems are typically characterized by several conflicting objectives. Deterministic approaches are not particularly suitable for this typology of problems because the objective space is constrained, many variables are involved, and there are many local minima and non-linearities. One possible way to overcome these problems is by using multi-objective (MO) metaheuristic approaches. These methods have already proven for complex MO problems, in many different fields, to be more successful than deterministic gradient-based methods. Some of the best-known algorithms in the state of the art are based on Genetic Algorithms [7] such as NSGA-II [5], SPEA2 [24] and MOEA/D [22]. These methods are generally preferred because they use a population that can be naturally tuned to solve MO problems by finding several Pareto-optimal (PO) solutions in parallel; thus MO evolutionary approaches are the most widely used [3], [4].

The Multi-objective Alliance Algorithm (MOAA) [14] is an evolutionary approach based on the Alliance Algorithm (AA) which is a relatively new single-objective optimization algorithm that has been applied successfully to many different typologies of problems [10], [2], [12], [17]. The first MO version was compared with NSGA-II [5] and SPEA2 [24]. That study [14] revealed a certain complementarity because the three approaches offered superior performance for different classes of problems. Since then, a mixed-integer version of the MOAA with hybrid components has been developed. This was able to outperfom a hybrid version of NSGA-II on a satellite constellation refueling optimization problem [18]. The knowledge acquired in solving benchmark and complex real-world problems led to the development of a new version of the MOAA which has already been tested on MO benchmark problems [15], the optimization of a supersonic airfoil [16] and a benchmark aerodynamic optimization problem [13].

In this paper, the MOAA is used 'out of the box' and applied to the optimization of CORAIL assemblies for nuclear reactors to test whether the approach is able to perform well without needing to tune its parameters or modify the algorithm.

The central goal of this work is to optimize the plutonium (Pu) loading and the assembly power peaking in a PWR. This is a constrained mixed-integer problem with two objectives. Both NSGA-II and the MOAA are applied to this problem to provide a good basis of comparison.

The remainder of this paper is structured as follows: Sect. II introduces the nuclear optimization problem; Sect. III explains the optimization framework; Sect. IV provides an overview of how the MOAA works; Sect. V introduces the indicators and statistical tests used for the evaluation of the algorithms; Sect. VI reports on the MOAA's performance on this problem, comparing it with that of NSGA-II; Sect. VII discusses the overall performance and compares the solutions found by the algorithms with the expert solution; Sect. VIII concludes the paper and proposes possible future work.

II. CORAIL Assemblies for PWRs

Nuclear reactors are typically fuelled with low-enriched uranium (LEU) fuel. The fuel is usually disposed of after a single pass through the reactor. However, it is possible to reprocess it to produce plutonium (Pu), which can also be used as a nuclear fuel when mixed with depleted or natural U to create mixed-oxide fuel (MOX). It is more expensive to fabricate MOX fuel than LEU fuel. Reactors are operated containing some LEU and some MOX fuel assemblies [20]. An alternative is to place LEU and MOX pins within the same assembly [8] (known as a 'CORAIL' assembly).

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

The probability of Pu and U isotopes reacting with neutrons (the 'cross-section') varies with neutron energy. A wide range of neutron energies are present in the reactor. Pu and U isotopes have substantially different reaction probabilities. This leads to differences in neutron flux between LEU and MOX pins, and consequentially can lead to an uneven power distribution within the reactor [20]. This is undesirable as it leads to higher fuel temperatures. To mitigate this, the Pu loading in the MOX pins is often varied between pins in different regions. For example, a MOX assembly surrounded by LEU assemblies typically contains pins highly loaded with Pu in the centre, and lower Pu loading at the periphery (Fig. 1).

The reference design for the CORAIL assembly from [8] contains a single Pu loading, with Pu pins located at the periphery of the assembly. As the purpose of MOX fuel is generally to destroy Pu, while generating power, it is desirable to maximize the Pu loading of the assembly. However, for a core fuelled entirely with CORAIL assemblies, at least 50% of the pins should be LEU to maintain acceptable safety parameters [8][1].

The Pu loading of the CORAIL design of Fig. 1 is limited by the power peaking constraint. For high Pu loadings, increasing the Pu loading increases the reactivity of the MOX pins and therefore their power relative to the LEU pins. By optimizing the position of the MOX pins, and possibly by using different Pu loadings in different MOX pins (as with the pure MOX assembly), it may be possible to reduce the power peaking for a given Pu loading, thereby improving the reactor thermal margins. For this study, two Pu loadings are allowed. This problem was recently investigated by [21].

III. Optimization Framework

It is desired to maximize the Pu loading and minimize assembly power peaking $Power\,Peak$ (i.e. the ratio of the maximum to the average pin power) by changing the relative locations of the MOX and LEU pins in the CORAIL assembly, and changing the Pu loading in the MOX pins. The minimum number of LEU pins is specified as half the total[2]. The LEU enrichment is fixed at 5% and reactor-grade Pu is assumed.

Let W_1 and W_2 be the Pu loadings in MOX pin types 1 and 2. Let N_1 and N_2 be the number of MOX pins per assembly. Let N_3 be the number of LEU pins per assembly. As there are 264 fuel pins per assembly, $N_1+N_2+N_3 = 264$. Using octant symmetry, there are 39 unique fuel pin positions (Fig. 1); some of these are weighted by 0.5 due to their position on symmetry lines. These are assigned fuel types 1, 2 or 3, corresponding to MOX types 1 and 2, and LEU fuel, respectively. There are therefore 39 integer variables and 2 continuous variables (W_1 and W_2), with N_1, N_2 and N_3 being derived quantities.

The Pu loading (to be maximized) is defined as: $MOXT = W_1 \cdot N_1 + W_2 \cdot N_2$.

[1]The reasons for this are beyond the scope of this paper, but cores containing MOX fuel generally operate at reduced margins for some safety parameters compared to LEU-only cores.

[2]As well as being a safety constraint, this prevents the algorithm converging on a 100% MOX assembly.

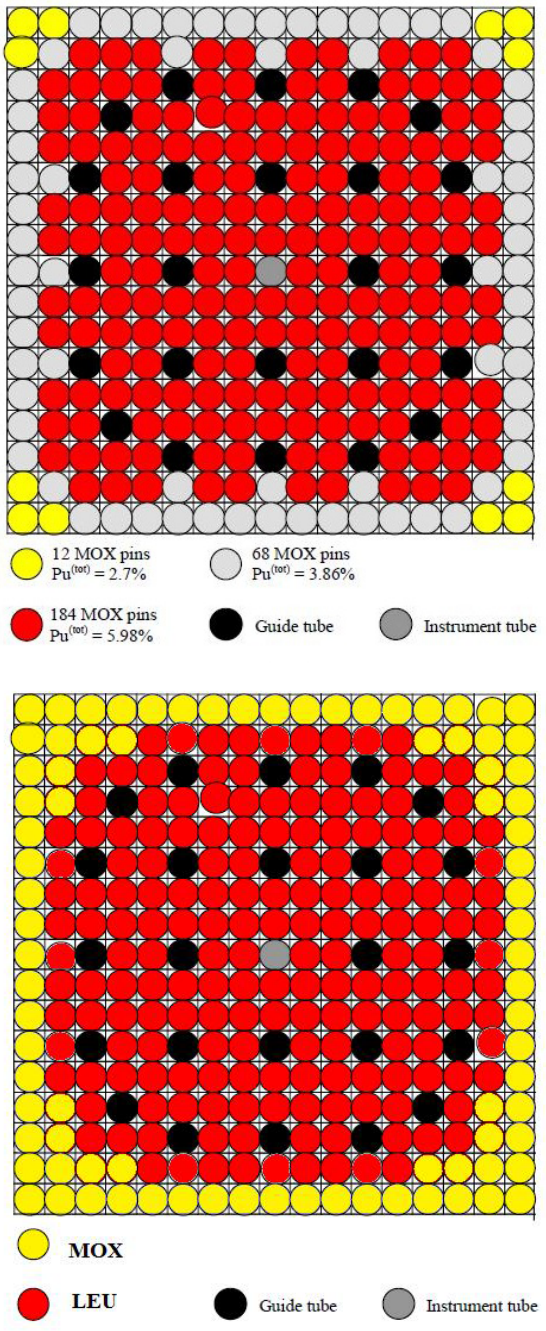

Fig. 1. MOX assembly surrounded by LEU assemblies from [20] (top) and CORAIL assembly with MOX pins at the assembly periphery adapted from [8] (bottom).

The two objectives are: minimize $Power\,Peak$ and minimize $-MOXT$.

The constraint is: $N_3 \geq 16.5$ (i.e. $264/8$).

$Power\,Peak$ is calculated using the reactor physics code WIMS 10 [19], which solves the neutron transport equation to find the neutron flux, and therefore the pin power, for every fuel pin in the assembly. The total fuel assembly power is fixed and irrelevant: only the power peaking is considered here.

Fig. 2 shows the optimization cycle: the algorithm creates a new solution; the solution is given as input to WIMS; the values of the two objectives are calculated; these are given back to the algorithm. This cycle is repeated until a certain number of function evaluations is reached.

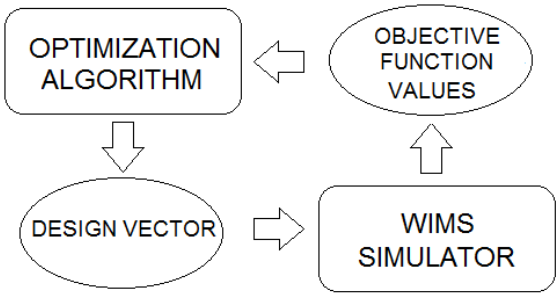

Fig. 2. Optimization cycle.

IV. Multi-objective Alliance Algorithm

The MOAA is a metaheuristic optimization algorithm inspired by the metaphorical idea of a number of tribes struggling to conquer an environment that offers resources that enable them to survive. The tribes are characterized by two features: the skills and resources necessary for survival. Tribes try to improve skills by forming alliances, which are also characterized by the skills and resources needed, but these now depend on the tribes within the alliance. The two main search elements of the algorithm are the formation of alliances and the creation of new tribes. One MOAA cycle ends when the strongest possible alliances of existing tribes have been created. The algorithm then begins a new cycle starting with new tribes whose creation is influenced by the previous strongest alliances.

A tribe t is a tuple (x_t, s_t, r_t, a_t) composed of: a point in the solution space x_t; a set of skills dependent on the values of the N_S objective functions evaluated at x_t; a set of resource demands dependent on the values of the N_R constraint functions; an alliance vector a_t containing the IDs of the tribes allied to tribe t.

An alliance is a mutually disjoint partition of tribes. Each alliance a forms a new point x_a in the solution space defined by the tribes in the alliance. The sets of skills s_a and resource demands r_a of the alliance consist of the objective and constraint functions S and R evaluated at x_a.

A. Algorithm Steps

The procedure followed by the MOAA can be divided into several steps. This version of the algorithm has been already described in detail in other papers. For this reason, only a general description (without equations) of the steps is provided here. A detailed description of these steps can be found in [16]. A general definition of the framework is provided in [11], a copy of which is available from the first author on request.

1) Solution Generation: In the MOAA's first cycle the tribes (solutions) are chosen randomly (with a uniform distribution). In subsequent cycles: some tribes are copies of previously found PO solutions; others are modifications (using a normal distribution with an adaptive standard deviation σ) of PO solutions.

An important feature is the adaptive nature of σ: this parameter adaptively decreases in order to produce high diversity at the start of the optimization and low diversity at the end. This mechanism enhances the initial exploration of the solution space and the final convergence of the solutions already found.

2) Verification: In this phase an alliance/tribe (A/T) tries to forge an alliance (a point in solution space x_a) with another tribe. When the alliance is created, x_a is made up of components drawn from the tribes within the alliance plus some variation. Thus, the new point is created by using uniform recombination between all the tribes of the alliance with variation applied randomly to some of the variables.

The standard deviation for the variation depends on the difference between the highest and lowest values of the corresponding variable among the tribes within the alliance; the variation for an alliance of tribes that are close together is small (local search) and for far-apart tribes it is large (global search). Generally at the start of an optimization the tribes within an alliance are far apart and then they start to come closer together. This behavior can be viewed as an initial global search followed by progressively more localized search.

The new alliance will only be confirmed if at least one skill in s_a of x_a is better than one skill in s_{t_1} of the solution representing the A/T seeking to forge the alliance and one skill in s_{t_2} of the tribe chosen to become an ally.

The resource function $R(x)$ is used to handle the constraint $N_3 \geq 16.5$.

3) Alliance and Data Structure Update: There are two possible outcomes from the Verification Phase: the chosen tribe joins the A/T, forming a new alliance, or the tribe does not join and the new alliance is not confirmed. Next there is an update of the data structures necessary for the low level system to function (all the tribes need to be informed of any change in the environment). The cycle termination conditions are also checked. The cycle finishes when each A/T has tried to form a new alliance with every other tribe and remains unchanged. If this condition is not met, the algorithm continues to try to form new alliances.

4) Selection of the Strongest Alliances and Termination: At the end of the interactions between tribes, many alliances will have been formed but only the strongest A/Ts will conquer the environment. Therefore the A/Ts selected are the non-dominated points in objective space. These correspond to the best solutions to the problem found thus far. They can be used as the input to another MOAA cycle or, if the algorithm has ended, they represent the final results.

There is a limit n to the number of best solutions saved in the archive of PO solutions. If the number of non-dominated solutions exceeds this, then all the solutions with at least one neighbor within a neighborhood distance d (in objective space)

are eliminated. The initial value of d is 0 and then changes adaptively. This formulation recognizes that d should depend on: the previous value of d, the current number of solutions, the number of function evaluations that can be afforded, and the actual ranges of the Pareto front.

The MOAA is terminated when a specified limit E_{tot} on the number of solution evaluations is reached. The algorithm output is then the best solutions and the Pareto front found.

5) Extended Archive: The MOAA also uses an Extended Archive which saves some dominated solutions that could help maintain diversity among solutions and the convergence of the algorithm. It accomplishes this task essentially by finding every large gap in the Pareto front (larger than the average gap between solutions multiplied by a factor d_f) in all the dimensions of the objective space and saving all the non-dominated solutions (and solutions that nearly satisfy the constraints) found in these gaps.

The factor d_f changes adaptively, increasing over time, as the gaps between the solutions become smaller, reaching similar values: by increasing its value only large gaps (in comparison with the average) are taken into consideration.

V. Indicators and Statistical Test

The performance measures chosen to evaluate the algorithms are the *epsilon* and *hypervolume* indicators provided in the PISA package [1]. The epsilon indicator [25] makes use of the Pareto-dominance concept and measures, given a reference set of points (ideally the true Pareto front, if available), the minimum amount ϵ necessary to translate all the points of the found Pareto front to weakly dominate the reference set. The hypervolume indicator [9] calculates the difference between the hypervolume of the space dominated by the found Pareto front and the hypervolume of the space dominated by a reference set (again, ideally the true Pareto front). This indicator needs a reference point which is dominated by all the found points in order to bound the hypervolume.

The statistical test chosen for the evaluation of results is the Kruskal-Wallis test, provided in the PISA package [1]. This is a non-parametric rank-based test that can be used to compare two independent sets of sampled data. It outputs p-values that estimate the probability of rejecting the null hypothesis of the study question when that hypothesis is true. Here the p-values can be interpreted as the probability that the MOAA is superior to NSGA-II only by chance.

VI. Results

MOAA and NSGA-II were tested for runs with limits of 300, 1000, 1600 and 2000 function evaluations. Each test was repeated 20 times. The small number of function evaluations is justified by the considerable computational cost necessary to solve the neutron transport equation to find the neutron flux. The parameters used for NSGA-II are specified in [5]. The MOAA parameters used are shown in Table I. These parameters have already proven their effectiveness in other studies with standard benchmark functions [15], such as the ZDT [23] and DTLZ [6] families, and also with real-world optimization problems [16], [13].

The PO solutions found in each of the 20 runs for all four tests are shown in Fig. 3, while the PO solutions amongst all 20 runs are shown in Fig. 4. The figures show how the MOAA is able to gradually outperform NSGA-II. In the 300 evaluation tests, the MOAA and NSGA-II solutions are mixed. In this case NSGA-II is more consistent and is able to find similar Pareto fronts, while the MOAA finds different types of Pareto front. The 1000 evaluation tests show a visible improvement in the MOAA's solutions: the algorithm begins to find more solutions than NSGA-II and most of these solutions have better quality. The other two evaluation tests (1600 and 2000) show a gradual improvement of the MOAA Pareto fronts: these solutions are well-spaced, well-spread and generally have equal or better convergence. In contrast, NSGA-II presents clusters of solutions in only a few focused areas and many other areas of the Pareto fronts are almost empty.

Figure 4 shows that the best PO solutions found by the MOAA generally dominate all the PO solutions found by NSGA-II: in some cases there is an evident gap between the two Pareto fronts. The graphs also show that the MOAA finds more well-spaced solutions with better convergence, while the NSGA-II Pareto fronts have many gaps, and in some areas the Pareto front seems disconnected. However, the MOAA results show that the Pareto front clearly is connected.

Results from each of the 20 runs were used to compute the mean and standard deviation of the epsilon and hypervolume indicators along with their corresponding p-values. Lower values of these metrics are indicative of better performance. The reference set for these comparisons was composed of the PO solutions found by both algorithms.

Table II shows the mean and standard deviation of the epsilon indicator. Comparable performance for both algorithms is observed for 300 function evaluation runs. However, for the other cases, the mean gradually increases for NSGA-II and decreases for the MOAA. The increase of the mean for NSGA-II might seem counterintuitive but it is related to the creation of clusters: this mechanism gradually improves the convergence of localized areas of the Pareto front but the solutions outside these clusters become quickly dominated and parts of the Pareto fronts are sometimes lost. The result of this behavior is an increase in the values of both the indicators because they take into account how the solutions are distributed and if parts of the Pareto front (especially the edges) are missing.

The standard deviation is also, in general, smaller for the MOAA results. This shows that the MOAA is able to gradually find qualitatively better solutions than NSGA-II.

The hypervolume indicator metrics are shown in Table III. NSGA-II exhibits better performance for 300 function evaluation runs. Thereafter, the mean hypervolume values decrease for the MOAA and tend to increase for NSGA-II for the reasons explained above. Again, the standard deviation is also, in general, smaller for the MOAA results.

These two indicators show that with increasing function evaluations the MOAA steadily outperforms NSGA-II in terms

of both convergence and diversity of solutions.

The 20 values of the epsilon and hypervolume indicators obtained in all the tests by the MOAA (the blue circles) and NSGA-II (the red squares) sorted from best to worst are shown respectively in Fig. 5 and Fig. 6. In all the graphs, except the ones for 300 function evaluation runs, there is a clear gap between the MOAA and NSGA-II lines. This confirms visually the values previously indicated in the tables.

To quantify the efficacy of these two metrics in characterizing their relative success, we compute the p-values obtained using the Kruskal-Wallis statistical test. Lower p-values indicate that one algorithm is better than the other. These values are shown in Table IV. In this case we compute the probability that the MOAA provides a better set of PO solutions than NSGA-II purely by chance. For the 300 function evaluation runs the null-hypothesis is satisfied for both the epsilon and hypervolume indicators: this essentially means that it is unclear which algorithm performs better, and confirms the figures and tables previously shown. In the other cases, it is possible to observe a gradual improvement in the p-values: they decrease towards 0, meaning that the MOAA improves its performance in comparison to NSGA-II. These results, again, confirm all the values and graphs previously shown. Thus, for higher function evaluation runs on this problem, it may be concluded that the MOAA outperforms NSGA-II.

TABLE I
MOAA PARAMETERS

Parameter	Value	Description
N	6	Number of tribes
P_1	0.5	Probability 1 for the creation of tribes
P_2	0.2	Probability 2 for the creation of tribes
σ_{init}	0.3	Initial standard deviation
σ_{end}	0.01	Final standard deviation
P_3	$2/V$	Probability for the creation of alliances
σ_a	0.1	Standard deviation for the creation of alliances
N_{tot}	100	Total number of PO solutions
N_f	10	Factor for evaluation neighborhood

TABLE II
COMPARISON WITH EPSILON INDICATOR

E_{tot}	MOAA		NSGA-II	
	Mean	Std	Mean	Std
300	0.3656	0.1178	0.3228	0.1001
1000	0.3640	0.1379	0.4597	0.1166
1600	0.3363	0.1391	0.5024	0.1622
2000	0.3318	0.1379	0.5226	0.1553

TABLE III
COMPARISON WITH HYPERVOLUME INDICATOR

E_{tot}	MOAA		NSGA-II	
	Mean	Std	Mean	Std
300	0.3171	0.1432	0.2563	0.0551
1000	0.2452	0.0795	0.3575	0.0591
1600	0.2375	0.0814	0.3569	0.1213
2000	0.2239	0.0818	0.3536	0.1191

TABLE IV
KRUSKAL-WALLIS STATISTICAL TEST

E_{tot}	p-value (MOAA better than NSGA-II by chance)	
	Epsilon	Hypervolume
300	H0	H0
1000	$2.04 \cdot 10^{-2}$	$6.00 \cdot 10^{-6}$
1600	$6.74 \cdot 10^{-4}$	$2.59 \cdot 10^{-4}$
2000	$1.18 \cdot 10^{-4}$	$1.98 \cdot 10^{-5}$

VII. DISCUSSION

Looking at the clear trade-off found between $Power\,Peak$ and $MOXT$, as expected, the power peaking increases with increasing Pu loading when the Pu loading is high. For $MOXT = \sim 2$, the power peaking is low as the reactivity of the MOX and LEU pins is very similar. The power peaking of $\sim$1.06 is similar to pure LEU assemblies, i.e. assemblies with $[-MOXT, Power\,Peak, N_3] = [0, 1.06, 33]$ which are dominated by the $[-2, 1.06, \sim 16.5]$ solutions, such that solutions with $Power\,Peak < 1.06$ are generally not found.

Raising $MOXT$ to $\sim$4 results in a power peaking of $\sim$1.2. This is around the maximum likely to be acceptable [8].

The MOAA generally converged on a single distribution of pin types throughout the Pareto front (with only 2 pin positions changing type in any of the PO solutions). For low power peaking solutions, the Pu loadings were 5–10% different, implying that a single Pu loading is sufficient to achieve a near-optimal power distribution. This is advantageous as the fuel fabrication cost is expected to increase with the number of different Pu loadings. This is consistent with the conclusions of [21], where the optimization converged on a single Pu loading by reducing the number of type 1 pins.

The pin types are, in general, quite well mixed in PO solutions, although there is a grouping of LEU pins towards the edge of the assembly. However, as the MOX loading increased, the relative difference in Pu loading between type 1 and type 2 pins generally increased to $\sim$20–30%. This implies that the pin positions may be sub-optimal in some cases. This is a consequence of the small number of function evaluations used on this complex problem: the adaptive parameters force the algorithm to an early convergence, exploiting the best area found, without allowing it to better explore the solution space. The pin distribution for a solution from the Pareto front found by the MOAA is shown in Fig. 7.

The expert design from Fig. 1 [8] achieved a power peaking of 1.16 and MOX loading of 2.55 (with $N_3 = 22.5$). Figure 8 shows an enlargement of the best Pareto front found by the MOAA and NSGA-II, with a power peak in the range 1.06–$\sim$1.2, and the expert design vector. It is interesting to observe that the expert solution is only dominated by solutions found by the MOAA. This is a further example of the superiority of the MOAA's performance with increasing function evaluations: the algorithm is able to find solutions that are better than solutions found by an expert. In particular:

- Given the $MOXT$ value obtained for the expert solution, the MOAA is able to find solutions with up to $\sim$4–4.5% improvement in $Power\,Peak$.

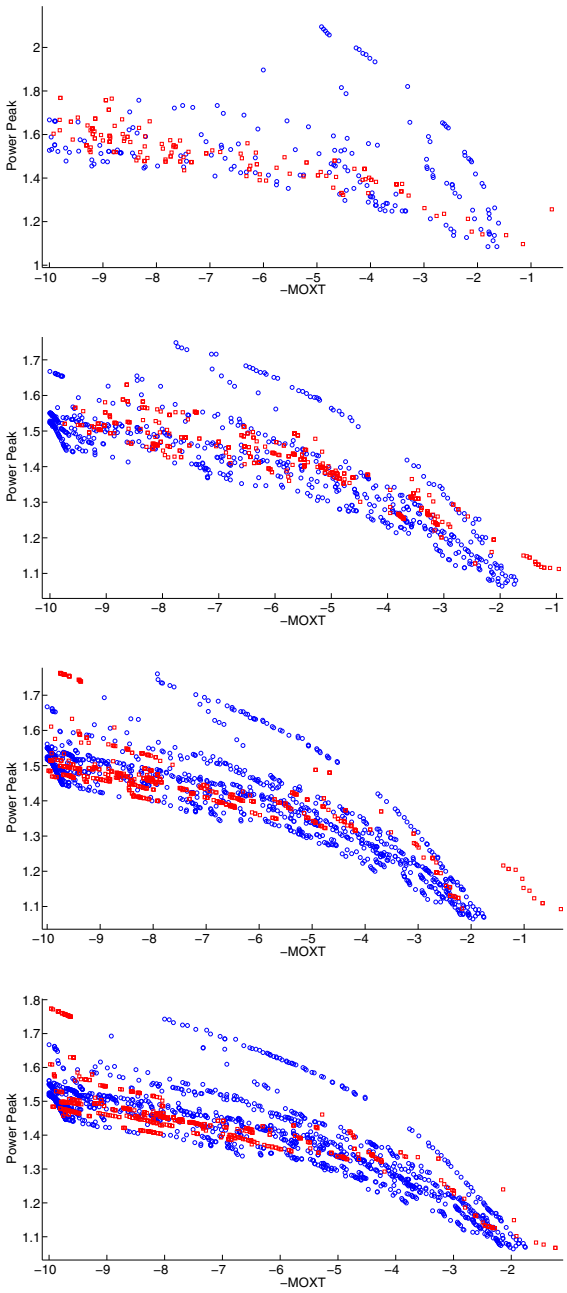

Fig. 3. Comparison of the results from 20 runs in the four tests. MOAA solutions are shown in blue; NSGA-II solutions in red. From top to bottom: 300, 1000, 1600, 2000 evaluations.

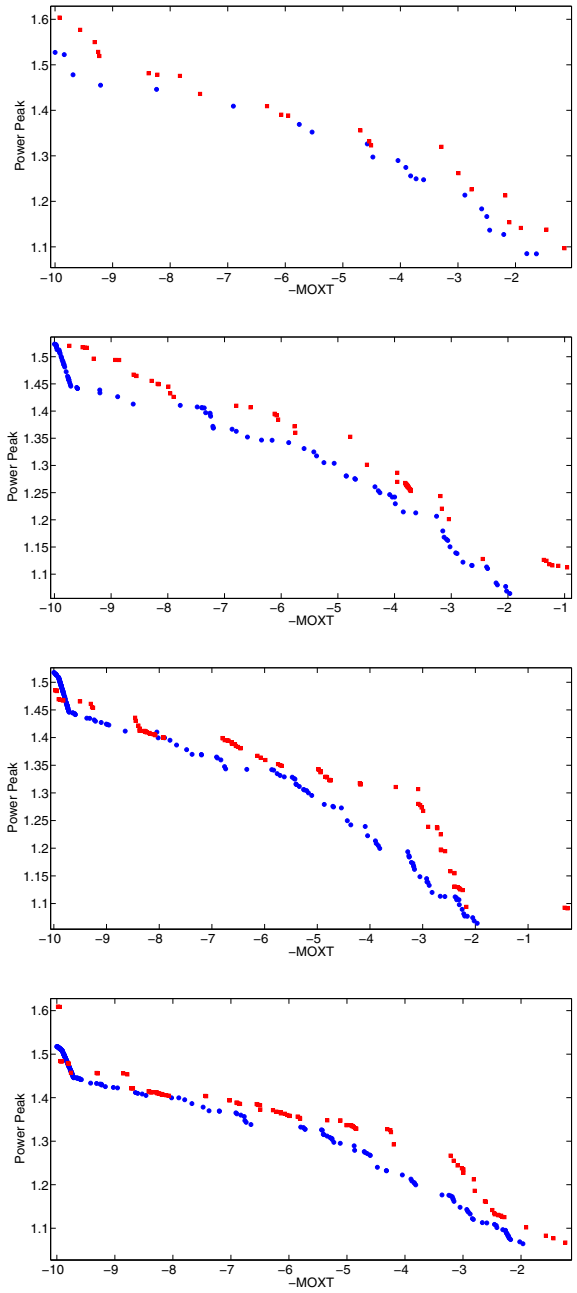

Fig. 4. Comparison of the non-dominated solutions found in the four tests. MOAA solutions are shown in blue; NSGA-II solutions in red. From top to bottom: 300, 1000, 1600, 2000 evaluations.

- Given the $Power\,Peak$ value obtained for the expert solution, the MOAA finds solutions with up to ∼23.5–24% improvement in $MOXT$.
- The set of PO solutions found by the MOAA that improve both the objectives yield an improvement of 10–19.5% in $MOXT$ and 1–3.5% in $Power\,Peak$.

This variety of solutions allows the decision-maker to choose the most suitable trade-off solution. The MOAA was able to find such solutions, in part, due to a reduction in the number of LEU pins (N_3) towards the constraint, which increases the number of MOX pins and hence the Pu loading for a given power peaking. Again, this is consistent with the findings of [21].

VIII. Conclusions

In this paper, the MOAA was applied to the optimization of CORAIL assemblies for PWRs in order to maximize the Pu loading (minimize $-MOXT$) and minimize the assembly power peaking. This was a constrained mixed-integer problem

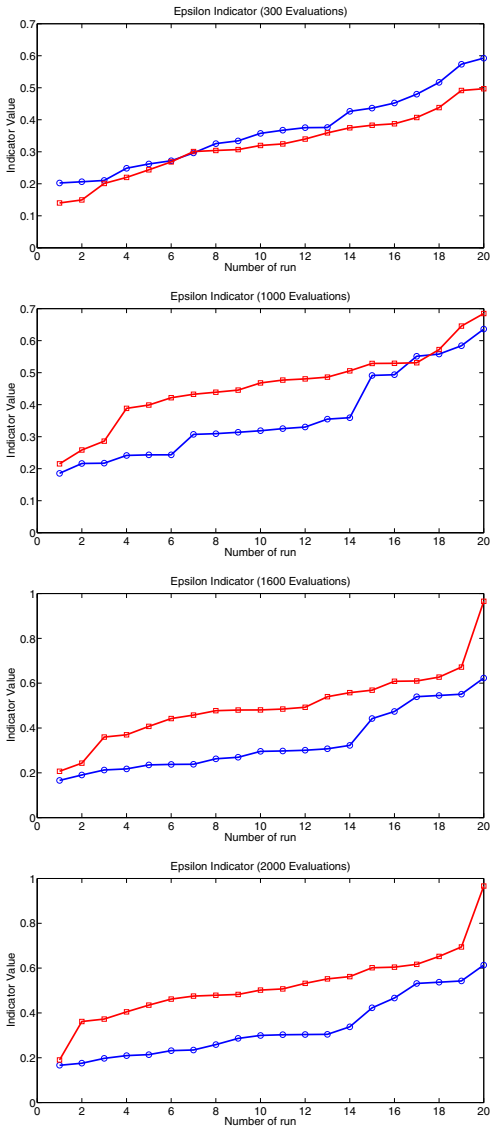

Fig. 5. Comparison of epsilon indicator values for all the tests. MOAA values are shown in blue; NSGA-II values in red. From top to bottom: 300, 1000, 1600, 2000 evaluations.

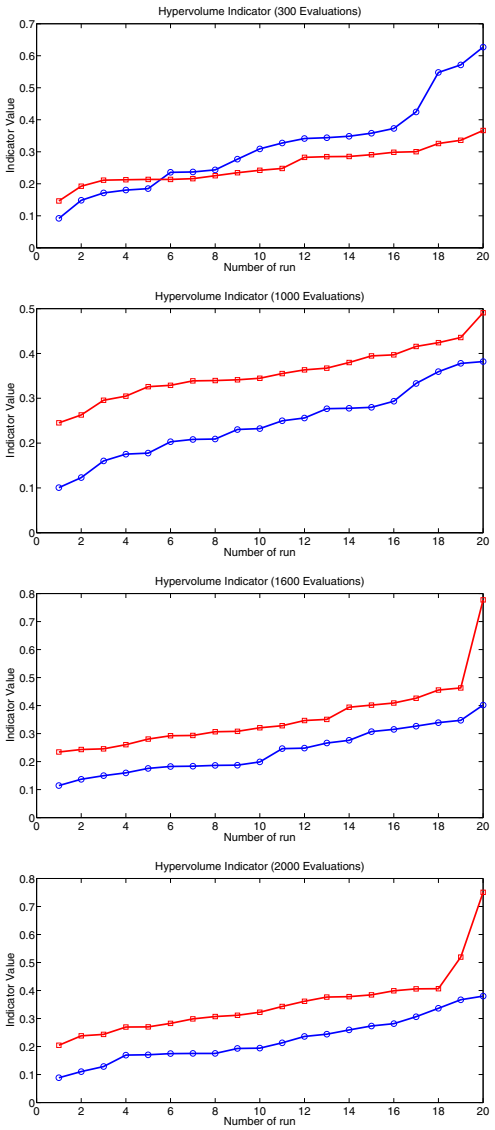

Fig. 6. Comparison of the hypervolume indicator values for all the tests. MOAA values are shown in blue; NSGA-II values in red. From top to bottom: 300, 1000, 1600, 2000 evaluations.

with 39 integer variables and 2 continuous variables. The results were compared with those given by NSGA-II and with an expert solution for this problem. The following are the key conclusions of this study:

1) The results obtained by the MOAA and NSGA-II for 300 function evaluations were similar but with more function evaluations the MOAA increasingly outperforms NSGA-II. This is confirmed by the p-values for the epsilon and hypervolume indicators which demonstrate the MOAA's efficacy on longer runs.
2) The Pareto front found by the MOAA is better spread, well-distributed, with increasingly better convergence, and more solutions are found. In contrast, NSGA-II finds fewer solutions, and there is a tendency for these to cluster, leading to several gaps in the Pareto front.
3) The Pareto front found by the MOAA contains solutions that fully dominate the expert solution for this problem; this is not true for the solutions found by NSGA-II.
4) The usage of the MOAA 'out of the box', without performing any modification or parameter tuning, has demonstrated the validity of this approach as a general problem solver.

In future work, the performance of the MOAA will be investigated for other variants of this problem with increased numbers of variables and contraints in order to understand the robustness and scalability of this approach for this category of problem. Moreover, further analysis of the optimal solution found by the MOAA will be undertaken to check whether the pin positions are, in fact, insensitive to the value of $MOXT$, as indicated by the MOAA results.

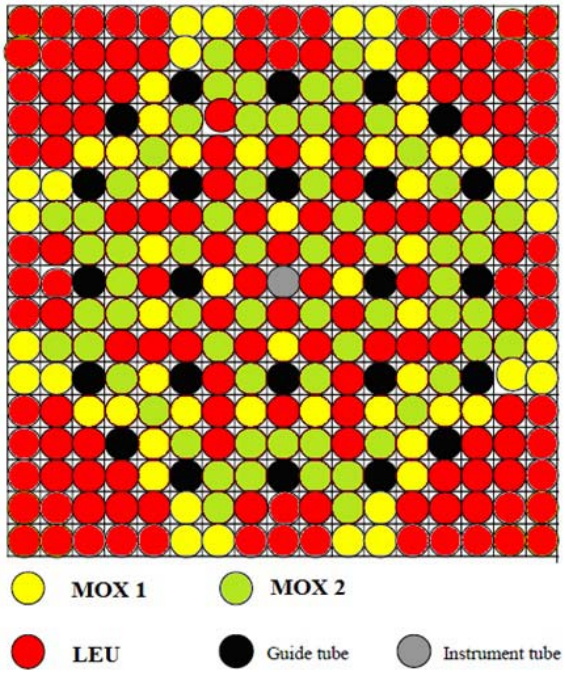

Fig. 7. Optimal pin distribution for a CORAIL assembly found by the MOAA. Type 2 pins are more highly loaded than type 1 pins.

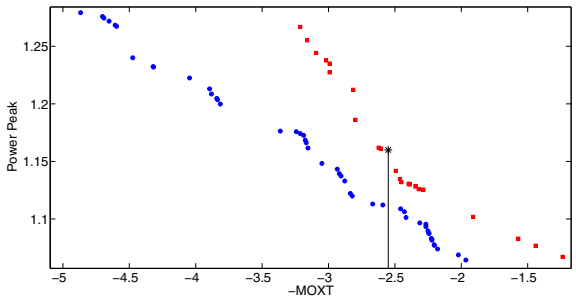

Fig. 8. Comparison of the MOAA (blue) and NSGA-II (red) solutions with the expert design (asterisk).

IX. Acknowledgement

We would like to acknowledge the ANSWERS team at AMEC for providing access to and guidance on the use of WIMS.

References

[1] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA—A platform and programming language independent interface for search algorithms. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, Faro, Portugal, 2003.

[2] V. Calderaro, V. Galdi, V. Lattarulo, and P. Siano. A new algorithm for steady state load-shedding strategy. In *12th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pages 48–53, Brasov, Romania, 2010.

[3] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag, Secaucus, NJ, USA, 2006.

[4] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK, 1 edition, June 2001.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197, 2002.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. TIK Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), July 2001.

[7] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[8] T. K. Kim. Assessment of CORAIL-Pu multi-recycling in PWRs. Technical Report ANL-AAA-018, Argonne National Laboratory, 2002.

[9] J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), February 2006.

[10] V. Lattarulo. Application of an innovative optimization algorithm for the management of energy resources. BSc thesis, University of Salerno, 2009.

[11] V. Lattarulo. Multi-Objective Alliance Algorithm. Technical Report CUED/C-EDC/TR.157, Department of Engineering, University of Cambridge, 2011.

[12] V. Lattarulo. Optimization of biped robot behaviors by 'alliance algorithm'. Master's thesis, University of Hertfordshire, 2011.

[13] V. Lattarulo, T. Kipouros, and G. T. Parks. Application of the Multi-objective Alliance Algorithm to a benchmark aerodynamic optimization problem. In *International Conference on Evolutionary Computation (CEC)*, Cancun, Mexico, 2013.

[14] V. Lattarulo and G. T. Parks. A preliminary study of a new multi-objective optimization algorithm. In *International Conference on Evolutionary Computation (CEC)*, Brisbane, Australia, 2012.

[15] V. Lattarulo and G. T. Parks. Testing of Multi-Objective Alliance Algorithm on benchmark functions. In *GECCO 2013*, Amsterdam, The Netherlands, 2013.

[16] V. Lattarulo, P. Seshadri, and G. T. Parks. Optimization of a supersonic airfoil using the Multi-Objective Alliance Algorithm. In *GECCO 2013*, Amsterdam, The Netherlands, 2013.

[17] V. Lattarulo and S. G. van Dijk. Application of the "Alliance Algorithm" to energy constrained gait optimization. In *15th Annual RoboCup International Symposium*, Istanbul, Turkey, 2011.

[18] V. Lattarulo, J. Zhang, and G. T. Parks. Application of the MOAA to satellite constellation refueling optimization. In *Evolutionary Multi-Criterion Optimization (EMO 2013)*, Sheffield, UK, 2013.

[19] T. Newton, G. Hosking, L. Hutton, D. Powney, B. Turland, and E. Shuttleworth. Developments within WIMS10. In *Proc. Physics of Reactors Topical Meeting (PHYSOR 2008)*, Interlaken, Switzerland, 2008.

[20] OECD Nuclear Energy Agency. Plutonium management in the medium term – a review by the OECD/NEA working party on the physics of plutonium fuels and innovative fuel cycles (WPPR). Technical Report NEA44517, 2003.

[21] Z. Yang. PWR heterogeneous fuel assembly optimisation. Master's thesis, Cambridge University Engineering Department, 2013.

[22] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.*, 11(6):712–731, December 2007.

[23] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, June 2000.

[24] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001.

[25] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.*, 7(2):117–132, 2003.