RESEARCH REACTOR IN-CORE FUEL MANAGEMENT OPTIMISATION USING THE MULTIOBJECTIVE CROSS-ENTROPY METHOD

Evert B. Schlünz
Radiation and Reactor Theory
Necsa, Building 1900, P.O. Box 582, Pretoria, 0001, South Africa
Department of Logistics
Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa
bernard.schlunz@necsa.co.za

Pavel M. Bokov
Radiation and Reactor Theory
Necsa, Building 1900, P.O. Box 582, Pretoria, 0001, South Africa
pavel.bokov@necsa.co.za

Jan H. van Vuuren
Department of Industrial Engineering
Stellenbosch University, Private Bag X1, Matieland, 7602, South Africa
vuuren@sun.ac.za

Abstract

The *in-core fuel management optimisation* (ICFMO) problem has been studied for several decades. Very little research has, however, been aimed at multiobjective optimisation involving the fundamental notion of Pareto optimality. In this paper, the recently developed *multiobjective optimisation using the cross-entropy method* (MOO CEM) algorithm is applied to a multiobjective ICFMO problem for the first time. A derivation of the MOO CEM algorithm is presented for ICFMO, along with a constraint handling technique. The algorithm is applied to a biobjective test problem for the SAFARI-1 nuclear research reactor. The Pareto set approximated by the algorithm is compared to solutions obtained by typical operational reload strategies. The results indicate that the MOO CEM algorithm for multiobjective ICFMO is a robust and efficient method which is able to obtain a good spread of trade-off solutions. The method may therefore greatly aid in the decision making of a reactor operator tasked with designing reload configurations.

Key Words: **in-core fuel management optimisation, multiobjective cross-entropy method, research reactor**

1. INTRODUCTION

The *in-core fuel management optimisation* (ICFMO) problem refers to the problem of finding an optimal fuel reload configuration for a nuclear reactor core. It is therefore a decision problem on where to load fresh and partially-burnt fuel in the core, in order to optimise one or several objectives, subject to certain constraints. Several characteristics of the ICFMO problem are listed in [1] and include a large disjoint feasible decision space, multiple nonlinear objectives and constraints, and significant computational complexity.

ICFMO has been studied for several decades, but the overwhelming majority of the research has been orientated towards power reactors. Research reactors pose different challenges than power reactors when optimising their reload configurations. In many cases, the decision space for a research reactor problem is much larger than for a power reactor problem. This is due to symmetry in power reactor core geometries which allows one to reduce the full problem to an equivalent, smaller problem. As an example, the Angra 1 nuclear power plant core, containing 121 fuel assemblies and potentially having 121! reload configurations, reduces to having an upper bound of $21! \approx 5 \times 10^{19}$ reload configurations due to its 1/8th core symmetry [2]. In contrast, the SAFARI-1 research reactor core, containing 26 fuel assemblies in an asymmetric geometry, cannot be reduced to an equivalent, smaller problem and has an upper bound of $26! \approx 4 \times 10^{26}$ reload configurations [3].

The neutronic evaluation of a reload configuration also differs between research and power reactors. Typically, core calculation systems use only two energy groups for evaluating power reactors. More groups are, however, required for research reactors and this results in increased computation time per evaluation. Furthermore, the objective in most instances, when optimising the reload configuration for a power reactor, is to improve the reactor's fuel efficiency. However, many research reactors have been designed to be used for multiple purposes, resulting in several objectives to be pursued simultaneously, apart from fuel efficiency, when optimising their reload configurations.

A number of solution techniques have been proposed for solving the ICFMO problem, such as mathematical programming methods, expert/knowledge-based systems, simulated annealing, evolutionary algorithms, swarm intelligence algorithms and tabu search [4]. Although some research has been directed towards multiobjective ICFMO, most of these approaches involve the use of linear weighted sum aggregations of the objectives (see e.g. [5, 6]). Such an aggregation approach is arguably not an appropriate way to address multiobjective optimisation problems [7]. Only a few research papers adopt fundamentally multiobjective optimisation approaches involving the notion of Pareto optimality (see e.g. [8]). We advocate, therefore, that more research into multiobjective ICFMO is required.

In this paper, we apply a recently developed metaheuristic search technique, called the *multiobjective optimisation using the cross-entropy method* (MOO CEM) algorithm [9], to a multiobjective ICFMO problem for the first time. This algorithm has been chosen because it was specifically designed to reduce the number of objective function evaluations when solving multiobjective optimisation problems such as dynamic and stochastic problems which require computationally expensive simulations for such evaluations [9]. A constrained multiobjective ICFMO test problem for the SAFARI-1 nuclear research reactor is considered as case study in this paper, using the OSCAR-4 code system [10] for the neutronic evaluation of configurations.

The paper is organised as follows. A brief introduction to multiobjective optimisation is presented, which is followed by a description of the SAFARI-1 test problem and the associated multiobjective optimisation model. Next, an overview of the MOO CEM algorithm is given along with a derivation thereof for the ICFMO problem specifically. This is followed by a description of the constraint handling technique developed for this problem. The optimisation results are then presented and compared to typical operational reload strategies. Finally, some conclusions are drawn at the end of the paper.

2. MULTIOBJECTIVE OPTIMISATION

A *multiobjective optimisation problem* (MOP) consists of simultaneously maximising/minimising two or more objective functions over some decision space. Suppose, without loss of generality, that all the objective functions are to be maximised. This assumption is valid because minimising an objective function is equivalent to maximising the negative of the function. The general formulation of a MOP with k objective functions $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})$ may then be written as

$$\begin{array}{lll} \text{maximise} & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})], & \\ \text{subject to} & g_i(\boldsymbol{x}) \leq 0, & i = 1, \ldots, r, \\ & h_j(\boldsymbol{x}) = 0, & j = 1, \ldots, s, \end{array}$$

where $\boldsymbol{x} = [x_1, \ldots, x_n]$ denotes the vector of decision variables, $g_1(\boldsymbol{x}), \ldots, g_r(\boldsymbol{x})$ are the inequality constraint functions and $h_1(\boldsymbol{x}), \ldots, h_s(\boldsymbol{x})$ are the equality constraint functions.

There is typically no single solution $\boldsymbol{x}^*$ that can maximise all the objective functions in a MOP. Solving a MOP instead yields a set of compromise or trade-off solutions, which leads to the notion of Pareto optimality.

Definition 1: Given two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$, $\boldsymbol{u}$ *dominates* $\boldsymbol{v}$ if $u_i \geq v_i$ for all $i = 1, \ldots, n$ and $u_i > v_i$ for at least one $i \in \{1, \ldots, n\}$.
Definition 2: Given a vector $\boldsymbol{u}$ in some subset $\mathcal{S} \subset \mathbb{R}^n$, $\boldsymbol{u}$ is *nondominated* in $\mathcal{S}$ if there does not exist any vector $\boldsymbol{v} \in \mathcal{S}$ such that $\boldsymbol{v}$ dominates $\boldsymbol{u}$.
Definition 3: Given a feasible domain $\mathcal{X}$, a vector of decision variables $\boldsymbol{x}^* \in \mathcal{X}$ is said to be *Pareto optimal* if $\boldsymbol{f}(\boldsymbol{x}^*)$ is nondominated in $\mathcal{X}$.
Definition 4: Given a feasible domain $\mathcal{X}$, the *Pareto set* of $\mathcal{X}$, denoted by $\mathcal{P}^*$, is the set containing all the Pareto optimal vectors of decision variables in $\mathcal{X}$, while the *Pareto front*, denoted by $\mathcal{P}^*_F$, is the set containing all the objective function vectors corresponding to the Pareto set, i.e. $\mathcal{P}^*_F = \{\boldsymbol{f}(\boldsymbol{x}) \,|\, \boldsymbol{x} \in \mathcal{P}^*\}$.
Definition 5: There may exist several sub-fronts into which a set of objective function vectors may be classified, each lower front being dominated by the one above it. The number of the front to which a vector belongs is known as its front depth or *Pareto rank*, and in this paper the numbering starts at zero. The Pareto rank of a vector in the Pareto front is therefore zero.

3. THE MULTIOBJECTIVE ICFMO TEST PROBLEM FOR SAFARI-1

SAFARI-1 is a 20 MW tank-in-pool type materials testing reactor in South Africa. The reactor is utilised for nuclear and materials research, as well as commercial activities such as irradiation services for isotope production and silicon transmutation doping. The OSCAR code system (*Overall System for the CAlculation of Reactors*) is used as the primary calculation tool to support the day-to-day operation planning, core-follow and reload simulation of the SAFARI-1 reactor [10]. The core layout of the 3-dimensional SAFARI-1 model used in OSCAR-4 is presented in Figure 1.

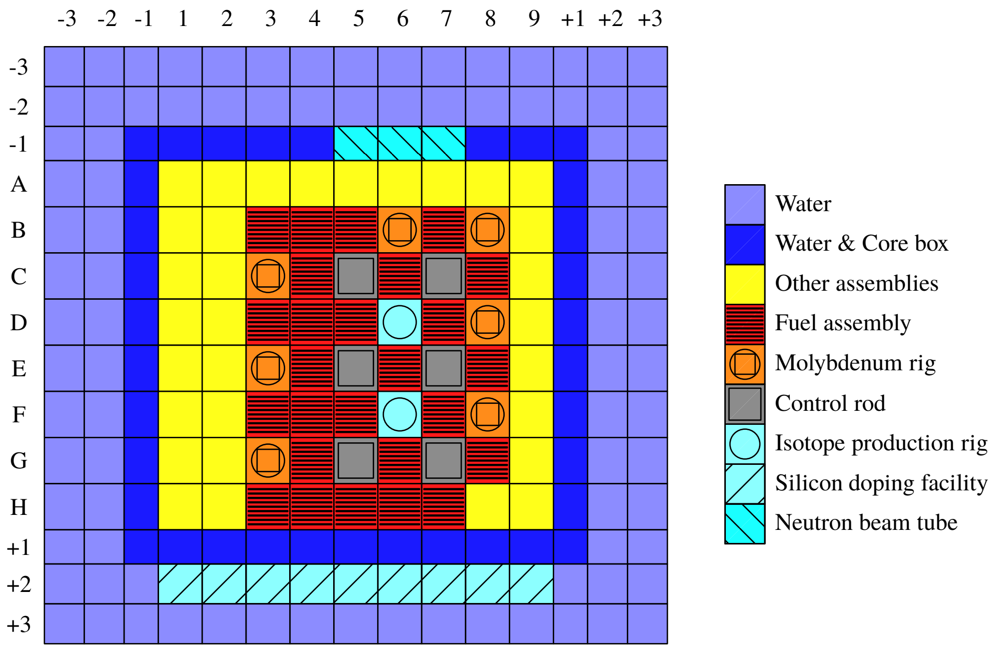

Figure 1. Top view of the core layout of the SAFARI-1 model used in OSCAR-4.

The SAFARI-1 core consists of a 9×8 lattice which houses twenty-six low-enriched fuel assemblies, six control rods of fuel-follower type, seven isotope production rig facilities for molybdenum-99 (^{99}Mo), as well as other core components. There are six neutron beam tubes in SAFARI-1 but only one of them (which is relevant to this paper) is illustrated in Figure 1. The constrained multiobjective ICFMO test problem for SAFARI-1, considered in this paper, is described below.

Two main conflicting objectives prevail in decisions related to reload configurations for this SAFARI-1 test problem. The first objective is to maximise the average thermal neutron flux in the silicon doping facility. The second objective is to maximise the average thermal neutron flux in the beam tube located on the opposite face of the core from the silicon facility. These objectives are subject to the following safety and utilisation constraints:

- The total ^{99}Mo production must be greater than the demand.
- A minimum ^{99}Mo yield is required for each ^{99}Mo rig.
- The relative and absolute power peaking factors must be below the safety limits.
- The total control bank worth must be above the safety limit.
- The shutdown margin must be above the safety limit.

The specific limiting values of these constraints are proprietary knowledge and are therefore not reproduced here. In addition, only twenty-six fuel assemblies are considered for placement in the twenty-six fuel loading positions of the core. The chosen assemblies correspond to assemblies that were loaded into the core during a past operational cycle (designated as C1211-1).

4. THE CROSS-ENTROPY METHOD

The *cross-entropy method* (CEM) for optimisation is a relatively new metaheuristic search technique dating back to 1999. The method was motivated by an adaptive algorithm for estimating rare event probabilities in computer simulation models and involves variance reduction by means of importance sampling [11]. The method has since been extended to include other types of problems, such as combinatorial optimisation problems and continuous optimisation problems [12]. Due to space limitations, only a conceptual description of the CEM for optimisation is presented here. The reader is referred to [13] for a detailed tutorial on the CEM.

The CEM uses the *Kullback-Leibler divergence* as a measure of "distance" between two sampling distributions. Rubinstein [12] showed that many optimisation problems can be translated to rare-event estimation problems, since the probability of finding an optimal solution using a pure random search is typically very small (i.e. has a rare-event probability). As such, the CEM can be utilised to change the sampling distribution of the random search at each iteration so that the rare-event is more likely to occur. The method estimates a sequence of sampling distributions (using an updating rule based on the cross-entropy term of the Kullback-Leibler divergence) that converges to a distribution which concentrates its probability mass in the neighbourhood of an optimal solution.

In order to apply the CEM to an optimisation problem, the following information is required. One first has to specify how the samples will be generated, i.e. which family of sampling distributions will be used. Secondly, one needs to calculate the updating rules in order to estimate the next sampling distribution in the sequence, based on cross-entropy minimisation.

4.1. The MOO CEM Algorithm

Recently, the CEM for optimisation was extended from single-objective optimisation to multiobjective optimisation by Bekker [9, 14]. It should be noted that the MOO CEM algorithm presented in [14] was primarily designed for continuous optimisation problems and integer programming problems. Combinatorial optimisation problems, such as the assignment problem and travelling salesman problem, are not well suited to the MOO CEM formulation as it stands. This is reflected in [9] since significantly different algorithmic formulations are provided there for the *vehicle routing problem with soft time windows* (VRPSTW) and the *buffer allocation problem* (BAP). A conceptual description of a generic version of the MOO CEM for combinatorial optimisation (based on the formulations presented in [9] for the VRPSTW and BAP) is presented here. The reader is referred to [9] for a detailed description of the method.

The basic principles of the MOO CEM algorithm remain the same as those of the single-objective CEM. Sample vectors are generated as before from a sampling distribution and, instead of only one function, each of the k objective functions are evaluated using the samples. However, since the goal of multiobjective optimisation is to approximate the Pareto set (containing nondominated solutions), the algorithm should now strive towards uncovering nondominated solutions. To that end, a Pareto ranking is performed (using the algorithm suggested in [9]) on the solutions of the first iteration in order to determine the best combinations of objective function values. The Pareto rank ρ of the objective function vector corresponding to a solution then indicates which front it belongs to, where $\rho = 0$ corresponds to a nondominated solution.

All the solutions with a Pareto rank not exceeding a pre-specified threshold value ρ_E are moved to an elite set, denoted by $\mathcal{E}$. The elite set represents the dominating set (i.e. the "best" solutions) and is continually updated during the algorithm's progression. During subsequent iterations, the elite set is appended to the new iteration's sample solutions and a Pareto ranking is performed on the combined set of solutions, denoted by C_t at iteration t. The elite set is then cleared and repopulated with the solutions in the combined set with a Pareto rank not exceeding ρ_E.

The combined set is used to calculate the updating rule for estimating the next sampling distribution in the sequence. This updating rule requires solution of the stochastic program (which is problem-specific)

$$\underset{\boldsymbol{v}}{\text{maximise}}\ \frac{1}{N_c}\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}} \ln p(\boldsymbol{x}_q;\boldsymbol{v}), \tag{1}$$

where N_c is the number of solutions in C_t, ρ_q is the Pareto rank of solution $\boldsymbol{x}_q$, and $p(\cdot;\boldsymbol{v})$ is the sampling distribution parameterised by the vector $\boldsymbol{v}$ [9]. The indicator function $I_{\{\rho_q \leq \rho_E\}}$ in (1) is defined as

$$I_{\{\rho_q \leq \rho_E\}} = \begin{cases} 1 & \text{if } \rho_q \leq \rho_E, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

A derivation of the MOO CEM algorithm for the ICFMO problem is presented in the next section, where the choice of sampling distribution family and the calculation of the updating rules are addressed. The derivation is similar to those for the travelling salesman problem in [13] and the BAP in [15].

5. THE MOO CEM ALGORITHM FOR MULTIOBJECTIVE ICFMO

Consider a reactor core with n loading positions, labelled $1, 2, \ldots, n$, as well as m available fuel assemblies labelled $1, 2, \ldots, m$, with $m \geq n$. Let $\mathcal{X}$ be the set of all possible fuel reload configurations. Each reload configuration may be represented by a partial permutation vector $\boldsymbol{x} = [x_1, \ldots, x_n]$ where $x_i = j$ denotes that fuel assembly j is placed into position $i \in \{1, \ldots, n\}$. Therefore, any one assembly can only be placed into one position, and the cardinality of this solution space is $|\mathcal{X}| = m!/(n-m)!$. The unconstrained multiobjective ICFMO problem may then be formulated as

$$\left.\begin{aligned} \text{maximise} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})], \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{X}. \end{aligned}\right\} \tag{3}$$

In order to apply the MOO CEM algorithm to the ICFMO problem, the family of sampling distributions and the associated updating rules are required. The first step in deriving these, is to relate (3) to an equivalent maximisation problem. Let $\widetilde{\mathcal{X}} = \{\boldsymbol{x} = [x_1, \ldots, x_n] \mid x_i \in \{1, \ldots, m\}, i = 1, \ldots, n\}$ be the set of vectors corresponding to solutions that may place the same fuel assembly into multiple positions, i.e. the requirement of having a partial permutation vector is relaxed. Note that $|\widetilde{\mathcal{X}}| = n^m$

and that $\mathcal{X} \subset \widetilde{\mathcal{X}}$. Furthermore, define the multiobjective function $\widetilde{\boldsymbol{f}}$ on $\widetilde{\mathcal{X}}$ as

$$\widetilde{\boldsymbol{f}}(\boldsymbol{x}) = \begin{cases} \boldsymbol{f}(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \mathcal{X}, \\ -\boldsymbol{\infty} & \text{otherwise,} \end{cases}$$

where $-\boldsymbol{\infty}$ denotes a k-dimensional vector (corresponding to the k objectives) containing the value $-\infty$ in each dimension. Problem (3) is therefore equivalent to the problem

$$\text{maximise} \quad \widetilde{\boldsymbol{f}}(\boldsymbol{x}), \quad \text{subject to } \boldsymbol{x} \in \widetilde{\mathcal{X}}. \tag{4}$$

A simple method for generating a random solution $\boldsymbol{x} = [x_1, \ldots, x_n] \in \widetilde{\mathcal{X}}$ is to draw $x_1, x_2, \ldots, x_n$ independently according to fixed distributions $\boldsymbol{p}_i = [p_{i,1}, \ldots, p_{i,m}]$, $i = 1, \ldots, n$. The distributions $\boldsymbol{p}_i$ can be combined into the $n \times m$ probability matrix $\boldsymbol{P} = [p_{i,j}]$, where all the elements of $\boldsymbol{P}$ are greater than zero, and

$$\sum_{j=1}^{m} p_{i,j} = 1, \qquad i = 1, \ldots, n. \tag{5}$$

The sampling distribution $p(\cdot; \boldsymbol{v})$ of $\boldsymbol{x}$ is parameterised by the matrix $\boldsymbol{P}$ and therefore denoted by $p(\cdot; \boldsymbol{P})$.

In order to update the parameters for this equivalent optimisation problem, the stochastic program (1) has to be solved, where the indicator function (2) now refers to $\widetilde{\boldsymbol{f}}$ and not to $\boldsymbol{f}$. Furthermore, the program must be solved under the additional constraint set (5). Let $\widetilde{\mathcal{X}}_{i,j}$ be the set of all solutions in $\widetilde{\mathcal{X}}$ for which $x_i = j$, and define the indicator function

$$I_{\{\boldsymbol{x} \in \widetilde{\mathcal{X}}_{i,j}\}} = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \widetilde{\mathcal{X}}_{i,j}, \\ 0 & \text{otherwise.} \end{cases}$$

The natural logarithm of the probability mass function $p(\cdot; \boldsymbol{P})$ of $\boldsymbol{x}$ corresponds to the one in the derivation of the travelling salesman problem [13] and is given by

$$\ln p(\boldsymbol{x}; \boldsymbol{P}) = \sum_{i=1}^{n} \sum_{j=1}^{m} I_{\{\boldsymbol{x} \in \widetilde{\mathcal{X}}_{i,j}\}} \ln p_{i,j}.$$

Since the stochastic program is a maximisation problem with equality constraints (5), the method of Lagrange multipliers may be used to solve it. Let $\lambda_1, \ldots, \lambda_n$ denote the Lagrange multipliers for the n equality constraints. Then the appropriate Lagrangian function is

$$\begin{aligned} \mathcal{L}(\boldsymbol{P}, \lambda_1, \ldots, \lambda_n) &= \frac{1}{N_c} \sum_{q=1}^{N_c} I_{\{\rho_q \le \rho_E\}} \ln p(\boldsymbol{x}_q; \boldsymbol{P}) + \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{m} p_{i,j} - 1 \right) \\ &= \frac{1}{N_c} \sum_{q=1}^{N_c} I_{\{\rho_q \le \rho_E\}} \left(\sum_{i=1}^{n} \sum_{j=1}^{m} I_{\{\boldsymbol{x}_q \in \widetilde{\mathcal{X}}_{i,j}\}} \ln p_{i,j} \right) + \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{m} p_{i,j} - 1 \right). \end{aligned} \tag{6}$$

The next step is to differentiate $\mathcal{L}$ in (6) with respect to $p_{i,j}$ and set the result equal to zero. For all $j = 1, \ldots, m$, this yields

$$\frac{\partial \mathcal{L}}{\partial p_{i,j}} = \frac{1}{N_c} \sum_{q=1}^{N_c} I_{\{\rho_q \le \rho_E\}} \left(I_{\{\boldsymbol{x}_q \in \widetilde{\mathcal{X}}_{i,j}\}} \frac{1}{p_{i,j}} \right) + \lambda_i = 0,$$

so that

$$\frac{1}{N_c}\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}} I_{\{\boldsymbol{x}_q \in \widetilde{\mathcal{X}}_{i,j}\}} = -\lambda_i p_{i,j}. \tag{7}$$

Summing over all $j = 1, \ldots, m$ in (7) yields

$$\frac{1}{N_c}\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}} \sum_{j=1}^{m} I_{\{\boldsymbol{x}_q \in \widetilde{\mathcal{X}}_{i,j}\}} = -\lambda_i \sum_{j=1}^{m} p_{i,j}, \tag{8}$$

and by substituting (5) into (8), it follows, after simplification, that

$$\frac{1}{N_c}\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}} = -\lambda_i. \tag{9}$$

By substituting (9) into (7), the updating rule

$$p_{i,j} = \frac{\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}} I_{\{\boldsymbol{x}_q \in \widetilde{\mathcal{X}}_{i,j}\}}}{\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}}} \tag{10}$$

is obtained. This updating rule may be interpreted as follows. The probability $p_{i,j}$ is updated by counting the number of sample vectors $\boldsymbol{x}_q$ which have fuel assembly j located in position i within those whose Pareto ranks are at most ρ_E, and dividing it by the total number of sample vectors whose Pareto ranks are at most ρ_E.

The sample generation and parameter updating for the equivalent problem (4) may be performed in the following way. Generate x_1 from the first row of $\boldsymbol{P}$, then independently generate x_2 from the second row of $\boldsymbol{P}$, etc. The updating formula (10) may then be applied to the combined set of solutions (the elite set from the previous iteration and the new samples from the current iteration).

Unfortunately, the majority of solutions generated in this way would not be useful — their elements would not form a partial permutation, and therefore their function values $\widetilde{f}$ would be $-\infty$. The generation of such undesirable solutions may, however, be avoided by using Algorithm 1.

Algorithm 1: Generation of solutions (partial permutations)

```
1 Generate a random permutation $[\pi_1, \ldots, \pi_n]$ of the set of positions $\{1, \ldots, n\}$.
2 Define $\boldsymbol{P}^{(1)} = \boldsymbol{P}$ and set $a = 1$.
3 Generate $x_{\pi_a}$ according to the distribution formed by the $\pi_a$-th row of $\boldsymbol{P}^{(a)}$, that is
  $[p_{(\pi_a,1)}, \ldots, p_{(\pi_a,m)}]$. Thus, fuel assembly $x_{\pi_a}$ is placed into position $\pi_a$.
4 Obtain $\boldsymbol{P}^{(a+1)}$ from $\boldsymbol{P}^{(a)}$ by first setting the $x_{\pi_a}$-th column of $\boldsymbol{P}^{(a)}$ to $\mathbf{0}$, and then renormalising the
  rows to sum up to 1.
5 If $a = n$ then stop and return $\boldsymbol{x}$; otherwise set $a = a + 1$ and return to step 3.
```

Note that the updating formula (10) is unaffected if Algorithm 1 is used to generate solutions [15]. Algorithm 1 simply speeds up the generation process described above. Furthermore, since only partial permutations are generated in this way, solutions now correspond to valid reload configurations, i.e. $\boldsymbol{x} \in \mathcal{X}$. The updating formula for $p_{i,j}$ then becomes

$$p_{i,j} = \frac{\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}} I_{\{\boldsymbol{x}_q \in \mathcal{X}_{i,j}\}}}{\sum_{q=1}^{N_c} I_{\{\rho_q \leq \rho_E\}}}, \tag{11}$$

where $\mathcal{X}_{i,j}$ is the set of all solutions in $\mathcal{X}$ for which $x_i = j$, and $I_{\{\boldsymbol{x}_q \in \mathcal{X}_{i,j}\}}$ is the corresponding indicator function. It has been suggested that the probability matrix (or vector parameterising the sampling distribution) should not be updated from $\boldsymbol{P}_{t-1}$ to $\boldsymbol{P}_t$ directly via (11), but smoothed instead using a smoothing parameter α [9].

The MOO CEM algorithm for the ICFMO problem is finally presented in Algorithm 2. The inner loop in the algorithm promotes exploration of the search space while the outer loop promotes intensification as the search progresses. A faster Pareto ranking algorithm, called the Fast Nondominated Sorting Algorithm [16], was used instead of the algorithm suggested in [9].

Algorithm 2: MOO CEM algorithm for the multiobjective ICFMO problem

```
Set $\mathcal{E} = \emptyset$, $\rho_E = 2$ and choose $N$
while outer loop stopping criteria not met do
    Initialise probability matrix $\boldsymbol{P}_0$ such that all $p_{i,j} = 1/m$ for $i = \{1, \ldots, n\}$ and $j = \{1, \ldots, m\}$
    Set $t = 1$
    while inner loop stopping criteria not met do
        Generate $N$ independent reload configurations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ using Algorithm 1 and $\boldsymbol{P}_t$
        Evaluate the objective function values $\boldsymbol{f}(\boldsymbol{x}_q)$ for each reload configuration $\boldsymbol{x}_q$
        Append the reload configurations in $\mathcal{E}$ to the new sample reload configurations to form $C_t$,
          and rank $C_t$ according the Fast Nondominated Sorting Algorithm [16]
        Calculate the number of solutions in $C_t$ as $N_c = |\mathcal{E}| + N$
        Use the solutions in $C_t$ and determine the intermediate $\hat{\boldsymbol{P}}_t$ using the updating formula (11)
        Calculate $\boldsymbol{P}_t$ by smoothing $\hat{\boldsymbol{P}}_t$ according to $\boldsymbol{P}_t = \alpha\hat{\boldsymbol{P}}_t + (1 - \alpha)\boldsymbol{P}_{t-1}$
        Set $\mathcal{E} = \emptyset$ and move the solutions of $C_t$ having $\rho_q \leq \rho_E$ to $\mathcal{E}$ in order to obtain an
          updated $\mathcal{E}$
        Set $t = t + 1$
    end
    Set $\rho_E = \max\{0, \rho_E - 1\}$ and trim $\mathcal{E}$ according to the new threshold value
end
Set $\rho_E = 0$ and trim $\mathcal{E}$ according to the new threshold value in order to obtain the (final)
  nondominated set
```

6. CONSTRAINT HANDLING

Algorithm 2 is applicable to unconstrained multiobjective ICFMO problems, as formulated in (3). However, in practice, ICFMO problems are usually constrained optimisation problems, such as the SAFARI-1 test problem described in Section 3. Therefore, a constraint handling technique is applied to transform the constrained ICFMO problem to an unconstrained problem so that Algorithm 2 may be used to solve it.

A multiplicative penalty function approach has been developed and incorporated into the multiobjective ICFMO problem model. In this approach, any constraint violation incurs a penalty value related to the magnitude of that violation. The total scalar penalty value of all the constraint violations is then used to penalise the objective function vector $\boldsymbol{f}(\boldsymbol{x})$ via multiplication. The major advantage of this approach is that a single scalar penalty value is used to penalise all k objective functions, irrespective of their orders of magnitude.

Consider the constrained multiobjective ICFMO problem formulation

$$\begin{aligned} \text{maximise} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})], \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq G_i, \qquad i = 1, \ldots, r, \\ & h_j(\boldsymbol{x}) = H_j, \qquad j = 1, \ldots, s, \\ & \boldsymbol{x} \in \mathcal{X}, \end{aligned}$$

where G_i and H_j are the (strictly positive) limiting values for each of the constraint functions and the other symbols have the same meanings as before. Let

$$\mathcal{G}(\boldsymbol{x}) = \sum_{i=1}^{r} \max\left\{0, \frac{g_i(\boldsymbol{x}) - G_i}{G_i}\right\}$$

be the total scaled constraint violation of the inequality constraint functions. Similarly, let

$$\mathcal{H}(\boldsymbol{x}) = \sum_{j=1}^{s} \left| \frac{h_j(\boldsymbol{x}) - H_j}{H_j} \right|$$

be the total scaled constraint violation of the equality constraint functions. We remind the reader about the assumption made in Section 2, that all objectives are to be maximised. Since the objectives in the ICFMO problem correspond to physical quantities in a reactor core, the assumption is made that the objectives are all positive. In order to apply the multiplicative penalty function approach, a distinction is now necessary between the two cases (minimisation and maximisation). Let f^+ denote an objective function that is unaffected, and let f^- denote an objective function after it has been multiplied by -1 in order to change it from a minimisation paradigm to a maximisation paradigm. Therefore, the objective function vector $\boldsymbol{f}$ may be written as $\boldsymbol{f}(\boldsymbol{x}) = [f_1^+(\boldsymbol{x}), \ldots, f_u^+(\boldsymbol{x}), f_{u+1}^-(\boldsymbol{x}), \ldots, f_{u+v}^-(\boldsymbol{x})]$, with u and v denoting the number of unaffected and transformed objective functions, respectively and $u + v = k$.

The multiplicative penalty function is also partitioned into the two cases, ϕ^+ and ϕ^-, respectively. Given a severity factor γ as a free parameter whose value is typically determined empirically, these two cases are

$$\begin{aligned} \phi^+(\boldsymbol{x}) &= 2 - \exp\Big(\gamma(\mathcal{H}(\boldsymbol{x}) + \mathcal{G}(\boldsymbol{x}))\Big), && \text{for } f^+(\boldsymbol{x}), \\ \phi^-(\boldsymbol{x}) &= \exp\Big(\gamma(\mathcal{H}(\boldsymbol{x}) + \mathcal{G}(\boldsymbol{x}))\Big), && \text{for } f^-(\boldsymbol{x}). \end{aligned}$$

A penalised objective function vector $\boldsymbol{f}_\phi$ may then be calculated by multiplying the objective function values with their corresponding penalty function values. The transformed unconstrained multiobjective ICFMO problem becomes

$$\left.\begin{aligned} \text{maximise} \quad & \boldsymbol{f}_\phi(\boldsymbol{x}) = \Big[\phi^+(\boldsymbol{x})\Big[f_1^+(\boldsymbol{x}), \ldots, f_u^+(\boldsymbol{x})\Big], \phi^-(\boldsymbol{x})\Big[f_{u+1}^-(\boldsymbol{x}), \ldots, f_{u+v}^-(\boldsymbol{x})\Big]\Big], \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{X}, \end{aligned}\right\} \tag{12}$$

to which Algorithm 2 may be applied.

7. COMPUTATIONAL RESULTS

The MOO CEM algorithm for ICFMO was applied to the test problem for SAFARI-1, as described in Section 3, using the transformed unconstrained formulation of (12). Five personal computers (Intel® Core$^{\text{TM}}$ i5-2500 CPU @ 3.30GHz, 4GB RAM, 32-bit operating system) were utilised for the calculations in this study. Algorithm 2 was used to solve the test problem independently on each computer, resulting in five concurrent algorithmic executions, each with a different random seed (since the MOO CEM algorithm is a stochastic technique). The optimisation results thus obtained were then compared to typical operational reload strategies.

A typical shutdown and reload period for SAFARI-1 lasts five days and an acceptable computational running time for ICFMO should not exceed three days. Considering that OSCAR-4 requires approximately four minutes to evaluate a single reload configuration, the stopping criterion for Algorithm 2 was fixed at 1080 evaluations. Since so few candidate solutions may be considered, the ranking threshold was fixed at $\rho_E = 2$ for the entire algorithmic progression. Therefore, the outer loop in Algorithm 2, as well as Step 15, was disregarded. The number of solutions generated during each iteration was chosen as $N = 30$, resulting in a maximum of thirty-six iterations (used as the stopping criterion for the inner loop in Algorithm 2). Finally, the smoothing parameter and severity factor were chosen as $\alpha = 0.6$ and $\gamma = 1.0$, respectively.

The initial and final nondominated fronts obtained by the five algorithmic runs are presented in Figure 2. The objective function values in the figure have been scaled according to the best values obtained overall in each objective. In addition, five of the eleven solutions in the initial nondominated sets, as well as eight of the one-hundred and twenty-nine solutions in the final nondominated sets, are infeasible and are thus excluded from these results. Hence, only solutions that adhere to the ^{99}Mo-related and safety constraints (stated in Section 3) are featured. The initial nondominated fronts from all five runs have been grouped together.

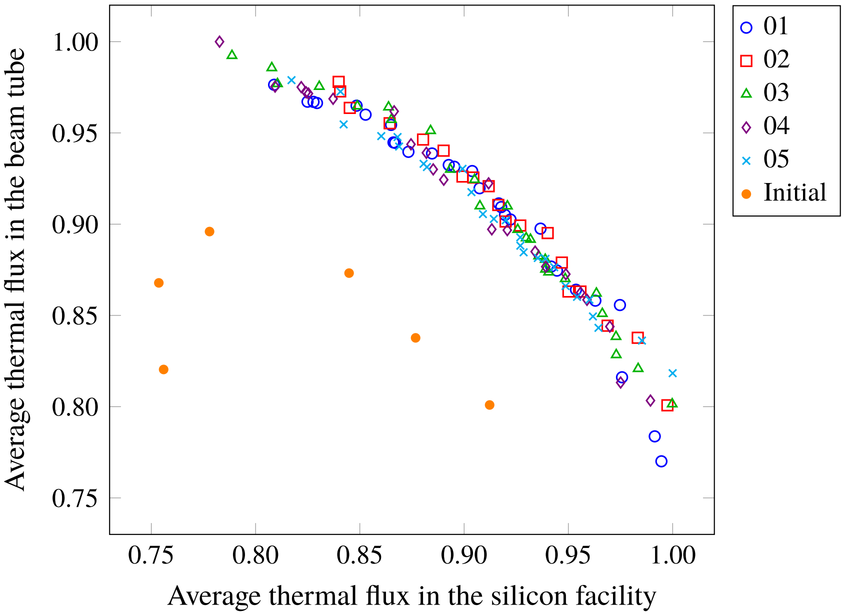

Figure 2. Initial and final nondominated fronts obtained by the five algorithmic runs.

The final nondominated fronts obtained from each algorithmic run are relatively similar, indicating the robustness of the MOO CEM algorithm with respect to different initial solutions, as well as its converging behaviour. Since the Pareto set is not known, the true convergence towards the Pareto front cannot be measured. These results are, however, very promising in respect of the efficiency and effectiveness of the algorithm, considering that only 1080 evaluations per execution run were performed. The initial nondominated fronts are reference points for the improvement upon random solutions that may be obtained during optimisation, and they also give an indication towards the difficulty in finding feasible solutions.

The final probability matrix obtained in the first algorithmic run is shown in Figure 3. The core positions have been labelled in the graph in order to correspond with their labels in the SAFARI-1 model in Figure 1. Similarly, the fuel assemblies have been labelled according to their specific uranium-235 masses. This labelling is intended to make the probability matrix more easily interpretable. Notice how the probability matrix concentrates the probability mass of the heavier (fresher) fuel assemblies almost exclusively to rows B and C (adjacent to the beam tube) and rows G and H (adjacent to the silicon facility). The final probability matrices of the remaining four algorithmic runs have similar patterns to those in Figure 3.

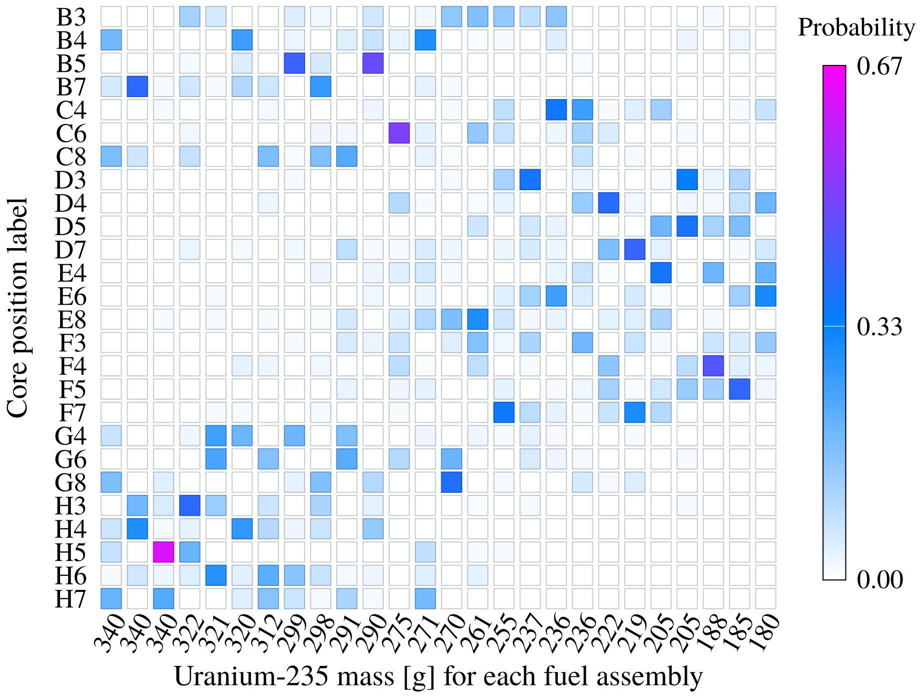

Figure 3. Final probability matrix obtained in the first algorithmic execution.

Given the five final nondominated sets computed during the algorithmic runs, an approximate Pareto set was obtained by performing a nondominated sorting over the combined solutions. The approximate Pareto front is presented in Figure 4, in order to compare the results with solutions obtained from operational strategies.

The first strategy corresponds to the *actual core loading* during the past operational cycle (C1211-1) under consideration, as designed by a SAFARI-1 operator with operational experience of the reactor. The second strategy was elicited from a nuclear engineer who does not have operational experience with SAFARI-1. The objectives and constraints of the problem were explained to the engineer, along with the core layout in Figure 1 and the available fuel assemblies. He was then asked to design a reload configuration and, based on his knowledge, a so-called *engineering solution* was obtained. The third strategy is based on a well-known heuristic rule, known as *highest mass to lowest flux* (HMLF), and it attempts to level the neutron flux profile over the core.

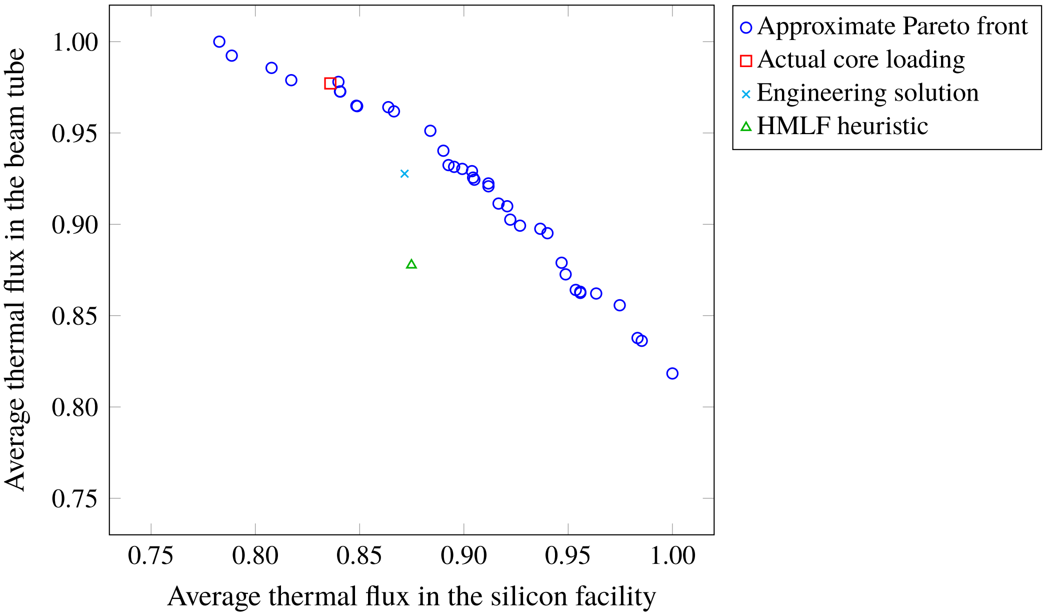

Figure 4. The approximate Pareto front and objective function values obtained from the operational strategies.

It is not surprising to notice that the actual core loading (based on the operational experience of a SAFARI-1 operator) performs so well, lying almost on the approximate Pareto front. Experience can never be discounted when optimising real-world problems. However, the value of multiobjective optimisation towards decision support for an operator is clearly illustrated in the graph. Apart from outperforming the engineering solution and heuristic rule strategies, the optimisation algorithm is able to suggest a whole spread of trade-off solutions. This approximate Pareto set allows much more flexibility for a reactor operator, with the shape of the Pareto front providing information as to the capabilities of the reactor.

8. CONCLUSION

In this paper, a multiobjective cross-entropy method has been applied to the multiobjective ICFMO problem for the first time. A derivation of the MOO CEM algorithm for ICFMO was presented, along with a multiplicative penalty function constraint handling technique. This technique transforms a constrained MOP to an unconstrained MOP so that the MOO CEM algorithm may be used to solve it. The MOO CEM algorithm for ICFMO was applied to a constrained biobjective test problem for the SAFARI-1 nuclear research reactor. It was found that the algorithm is rather robust, obtaining very promising results within its limited computational budget.

The approximate Pareto set obtained by the algorithm was compared to solutions obtained from typical operational reload strategies. The algorithm outperformed the engineering solution and heuristic rule strategies for SAFARI-1, whilst simultaneously providing a good spread of trade-off solutions from which an operator may choose a reload configuration. The actual core loading, which followed a strategy based on the operational experience of a SAFARI-1 operator, was found to produce a solution lying almost on the approximate Pareto front. However, such a strategy does not provide the flexibility in decision making, or the additional information regarding possible performances, that the biobjective results indeed provide.

It is concluded that the MOO CEM algorithm for multiobjective ICFMO may be used as an effective and efficient method for solving constrained ICFMO problems approximately. The good spread of trade-off solutions obtained by the algorithm may greatly aid in the decision making of a reactor operator tasked with designing reload configurations.

ACKNOWLEDGMENTS

The first author (E.B. Schlünz) was financially supported in part by Stellenbosch University via a postgraduate merit bursary, and by the National Research Foundation (NRF) of South Africa (Grant 88003). The second and third authors (P.M. Bokov and J.H. van Vuuren) were financially supported by the NRF (Grant 70730 and Grant 70593, respectively). Any opinion, finding and conclusion or recommendation expressed in this material is that of the author(s) and the NRF does not accept any liability in this regard. The second author would also like to thank professor A.P. Engelbrecht from the University of Pretoria for introducing him to the world of multiobjective optimisation.

References

[1] P. J. Turinsky. "Nuclear fuel management optimization: A work in progress." *Nuclear Technology*, **151**: pp. 3 – 8 (2005).

[2] A. A. M. Meneses, M. D. Machado, and R. Schirru. "Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor." *Progress in Nuclear Energy*, **51**: pp. 319 – 326 (2009).

[3] E. B. Schlünz, R. H. Prinsloo, and P. M. Bokov. "In-core fuel management optimisation within the OSCAR-4 code system." In: *Transactions of the European Research Reactor Conference 2013*. RRFM-2013. St. Petersburg, Russia, April 21 - 25 (2013).

[4] A. A. M. Meneses, A. M. M. de Lima, and R. Schirru. "Artificial intelligence methods applied to the in-core fuel management optimization." In: *Nuclear Power* (P. Tsvetkov, editor). Sciyo. URL http://www.intechopen.com/books/nuclear-power. Available online (2010).

[5] N. Poursalehi, A. Zolfaghari, and A. Minuchehr. “Multi-objective loading pattern enhancement of PWR based on the discrete firefly algorithm.” *Annals of Nuclear Energy*, **57**: pp. 151 – 163 (2013).

[6] E. B. Schlünz, P. M. Bokov, and R. H. Prinsloo. “Application of a harmony search algorithm to the core fuel reload optimisation problem for the SAFARI-1 nuclear research reactor.” In: *Proceedings of the 41st Annual Conference of the Operations Research Society of South Africa*. ORSSA-2012. Muldersdrift, South Africa, September 16 - 19 (2012).

[7] T. J. Stewart. “The essential multiobjectivity of linear programming.” *ORiON*, **23(1)**: pp. 1 – 15 (2007).

[8] A. Hedayat *et al.* “Optimization of the core configuration design using a hybrid artificial intelligence algorithm for research reactors.” *Nuclear Engineering and Design*, **239**: pp. 2786 – 2799 (2009).

[9] J. Bekker. *Applying the cross-entropy method in multi-objective optimisation of dynamic stochastic systems*. PhD dissertation, Stellenbosch University, Stellenbosch, South Africa (2012).

[10] G. Stander *et al.* “OSCAR-4 code system application to the SAFARI-1 reactor.” In: *International Conference on Reactor Physics*. PHYSOR ’08. Interlaken, Switzerland, September 14 - 19 (2008).

[11] R. Y. Rubinstein. “Optimization of computer simulation models with rare events.” *European Journal of Operational Research*, **99**: pp. 89 – 112 (1997).

[12] —. “The cross-entropy method for combinatorial and continuous optimization.” *Methodology and Computing in Applied Probability*, **1(2)**: pp. 127 – 190 (1999).

[13] P. T. de Boer *et al.* “A tutorial on the cross-entropy method.” *Annals of Operations Research*, **134(1)**: pp. 19 – 67 (2005).

[14] J. Bekker and C. Aldrich. “The cross-entropy method in multi-objective optimisation: An assessment.” *European Journal of Operational Research*, **211**: pp. 112 – 121 (2011).

[15] K. Alon *et al.* “Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment.” *Annals of Operations Research*, **134**: pp. 137 – 151 (2005).

[16] K. Deb *et al.* “A fast and elitist multiobjective genetic algorithm: NSGA-II.” *IEEE Transactions on Evolutionary Computation*, **6(2)**: pp. 182 – 197 (2002).