Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05555-6

ORIGINAL ARTICLE

A multi-objective particle swarm for constraint and unconstrained problems

Robert Nshimirimana[1,3] · Ajith Abraham[2] · Gawie Nothnagel[3]

Received: 11 January 2020 / Accepted: 19 November 2020

Abstract
Multi-objective particle swarm optimization algorithms (MOPS) are used successfully to solve real-life optimization problems. The multi-objective algorithms based on particle swarm optimization (PSO) have seen various adaptations to improve convergence to the true Pareto-optimal front and well-diverse non-dominated solution. In some cases, the values of the MOPS control parameters need to be fine-tuned while solving a specific multi-objective optimization problem. It is challenge to correctly fine-tune the value of the PSO control parameters when the true non-dominated solutions are not known as in case of a real-life optimization problem. To address this challenge, a multi-objective particle swarm optimization algorithm that uses constant PSO control parameters was developed. The new algorithm called NF-MOPSO is capable of solving different multi-objective optimization problems without the need of fine-tuning the value of the PSO control parameters. The NF-MOPSO enhances the convergence to the true Pareto-optimal front and improves the diversity of Pareto-optimal using the same fixed values for all the PSO control parameters. The NF-MOPSO uses constant values of the PSO control parameters such as acceleration coefficients c_1 and c_2, and inertia weight ω. A Gaussian mutation is applied to the position of particles to increase diversity while a penalty function is used as constraint mechanism. The algorithm has been tested on 45 well-known benchmark test functions using four performance metrics. The test results demonstrate the capability of the NF-MOPSO to solve different multi-objective optimization problems using the same value of the PSO control parameters. The capability of the NF-MOPSO was demonstrated in real-life optimization problem by solving a multi-objective optimization problem of a neutron radiography collimator. The results of collimator optimization showed that the optimizer was able to provide a set of Pareto optimal solutions from which the geometrical design parameters of a collimator could be retrieved for given application.

Keywords Particle swarm · Multi-objective optimization · Constraint · Control parameters · Neutron collimator

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-020-05555-6.

✉ Robert Nshimirimana
robert.nshimirimana@necsa.co.za

Ajith Abraham
ajith.abraham@ieee.org

Gawie Nothnagel
gawie.nothnagel@necsa.co.za

[1] Department of Industrial Engineering, University of Stellenbosch, Stellenbosch, South Africa

[2] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence Auburn, Washington 98071, USA

[3] Radiation Science Department, South African Nuclear Energy Corporation SOC Ltd, Pretoria, South Africa

1 Introduction

Optimization involves making decision about what outcome is “best” among all sets of available outcomes. When the “best” outcome must satisfy multiple objectives or criteria and those criteria are in conflict with each other, the problem is referred to as a multi-objective optimization problem (MOOP) [1, 2]. A MOOP normally has a set of solutions, and it is difficult to determine if one solution is better than the other, and therefore, the MOOP is solved by finding the trade-off solutions that balance the conflicting objectives [3]. The trade-off solutions form the Pareto-optimal front.

Multi-objective particle swarm optimization (MOPS) algorithms are being used in solving real-life MOOP [4–6]; hence, there is a need to improve the performance of a MOPS. The performance of a MOPS is measured based on key performance indicators such as good convergence of non-dominated solutions to the true Pareto-optimal front, and well-diversed Pareto-optimal front. The performance of the MOPS can be affected by the PSO control parameters such as acceleration coefficients c_1 and c_2, and inertia weight ω [7–9]. It has also been demonstrated that the values of the PSO control parameters are problem dependant [10]. Consequently, the values of the PSO control parameters may need to be fine-tuned to improve the performance of a MOPS for each MOOP. In ideal case, the values of the PSO control parameters are fine-tuned for a MOPS in which the true Pareto-optimal front is known as in case of test functions used in MOOP. There is no guarantee that the same values of the PSO control parameters will work for a different MOOP. This creates a challenge when solving a real-life MOOP in which the true Pareto-optimal front is not known [6]. Researchers have stated a need to find a MOPS in which the control parameters are not problem-dependent [2, 6, 10–13]. Currently, the use of values of the PSO control parameters that satisfy theoretically derived convergence conditions are used for the performance of PSO algorithm [6, 14, 15]. Other techniques such as hybridization of MOPS [16, 17], decomposition of MOPS [18–20], fuzzy clustering [21] have recently been used to improve the performance of MOPS. For a single-objective optimization, recent attempt has been made to create a nonparametric particle swarm optimization for global optimization [22]. In this paper, an attempt is made to create a multi-objective particle swarm algorithm called NF-MOPSO that uses the same constant values of the PSO control parameters to solve different MOOP.

The remainder of this paper is organized as follows. Section 2 introduces the basic technique used in particle swarm optimization. In Sect. 3, the proposed algorithm is discussed. In Sect. 4, the capability of NF-MOPSO to solve different test functions is demonstrated. The results of the tests and the comparison of the NF-MOPSO with other state of the art MOPS are discussed in Sect. 5. In Sect. 6, the capability of NF-MOPSO in solving a real-life optimization problem is demonstrated. Finally, the conclusion and future work are presented in Sect. 7.

2 Particle swarm optimization

2.1 Particle swarm optimization technique

PSO is a population-based optimization approach based on swarm intelligence that was first introduced by Kennedy and Eberhart in 1995 [23]. PSO was inspired by innovative distributed intelligence paradigm observed in a social behavior of a flock of birds. Since its development, PSO has become one of the popular optimization techniques for solving complex optimization problems [6]. PSO finds the solutions to the optimization problem by using the population of the swarm. A population of the swarm moves around in the search space searching for the optimal solutions. A member of a population is called a particle. A position occupied by each particle in the search space represents a candidate solution to the optimization problem. While moving to a new position (candidate solution), an efficient collaboration among particles is required to obtain good convergence to the best solutions. Thus each particle is guided by the cognitive and the social learning experience while moving to the new position [24, 25]. Therefore, the velocity of the particle is the carrier of the cognitive and the social learning experienced. The movements of the particles in collaboration will eventually lead the particle to converge to the best solutions.

In PSO, a multi-objective problem with n number of objectives and m number of decision variables is represented by an objective vector function $\vec{f}(\vec{x}^*) = (f_1(\vec{x}^*), f_2(\vec{x}^*), \ldots, f_n(\vec{x}^*))$ and a decision vector $\vec{x}^* = (x_1^*, x_2^*, \ldots, x_m^*)$.

The position of the particles is updated during the search at each time step t by adding the velocities of the particles as

$$x_i^a(t+1) = \varphi(t)x_i^a(t) + v_i^a(t+1) \quad (1)$$

where x_i^a is the position of the particle a for i decision variable, and φ is the mutation operator acting on particle a for i decision variable, and v_i^a is the velocity of the particle a for i decision variable. During the search of the optimal solution, the velocity of each particle in the swarm is updated using

$$v_i^a(t+1) = \omega v_i^a(t) + C_{-\text{comp}} + S_{-\text{comp}} \quad (2)$$

and

$$C_{-\mathrm{comp}} = r^{a}_{1i}(t)c_1\left[x^{y}_{i}(t) - x^{a}_{i}(t)\right] \quad (3)$$

and

$$S_{-\mathrm{comp}} = +r^{a}_{2i}(t)c_2\left[\hat{x}^{y}_{i}(t) - x^{a}_{i}(t)\right] \quad (4)$$

where $C_{-\mathrm{comp}}$ is the cognitive component, $S_{-\mathrm{comp}}$ is the social component, ω is the inertia weight, r_{1i} and r_{2i} are random values in $[0,1]$ for the particle a for i decision variable, c_1 and c_2 are acceleration coefficients, x^{y}_{i} is the local guide for particle a for i decision variable, and $\hat{x}^{y}_{i}(t)$ is the global guide for all particles for i decision variable.

2.2 Effect of control parameters

The performance of PSO depends on different factors that balance the trade-off between the exploration and exploitation of the swarm in the search space. The important issue in PSO is that those factors are problem-dependent. The settings of those factors may perform well for some optimization problems and fail miserably for others. The challenge is to find the optima settings of those factors for a real-life optimization problem. Alongside the PSO control parameters, other main factors that may affect the performance of PSO are the number of particles in the swarm [26, 27], the number of iterations [28, 29], the neighborhood topology [30], and the mutation operator [31, 32]. The effect of the PSO control parameters are discussed below:

2.2.1 Acceleration coefficients

The acceleration coefficients drive efficiency of the PSO algorithm [33]. The ratio between c_1 and c_2 influences the behavior of the swarm towards the personal best or the neighborhood best as follows:

- For $c_1 = c_2 = 0$ and $\omega = 1$, the velocities of the particles are constant which make the particles to keep on flying until they hit a boundary of the search space.
- For $c_1 > 0$ and $c_2 = 0$, there is no exchange of information between the particles, and hence the particles perform a local search.
- For $c_1 = 0$ and $c_2 > 0$, the particles do not have confidence in themselves and are only attracted to the neighborhood best solution.
- If $c_1 \gg c_2$, the cognitive component is much dominant than the social component making the particles to be much more attracted to their personal best solution.
- If $c_1 \ll c_2$, the social component is much dominant than the cognitive component causing a premature convergence to the solution.

2.2.2 Inertia weight

The inertial weight as PSO control parameter was first introduced as variant PSO in 1998 three year after the introduction of the basic PSO algorithm [34]. Since its introduction, the inertia weight is still used to improve the performance of PSO in real-life optimization problems [35, 36]. The inertia weight put its weight on the previous velocity of the particle, thus controlling the influence the cognitive and social components had on the previous velocity. The value of ω influences the exploration and exploitation abilities of the swarm as follows:

- For $\omega \geq 1$, the velocity is increased at each time step, particles do not change direction thus missing the promising local search area when searching for the best solutions.
- For $\omega < 1$, the velocity is decreased at each time step and may eventually reach zero. The particles are stagnated in their local search area.
- For a large value of ω , the exploration ability of the particles is encouraged and the diversity in the solutions is increased.
- For a small value of ω , a local exploration by the particles is encouraged.
- For $\omega \approx 0$, the exploration ability of the particles is eliminated.

2.3 Selection of PSO control parameters

The values assigned to PSO control parameters affect the performance of the algorithm for a given problem. In PSO, it is important to find the proper values of c_1, c_2, and ω when designing a new algorithm to solve a particular problem. The challenge is to find the best values of c_1, c_2, and ω that are suitable for a particular optimization problem. Theoretical analysis studies have been conducted on PSO to provide guidance on assigning the values of c_1, c_2, and ω [7, 37, 38]. Other selection mechanisms are guided by experimental methods such as trial and error, cross-validation, or self-adapting the values of c_1, c_2, and ω [10, 39, 40]. The setting of the right parameters of a multi-objective particle swarm optimizer is not an easy task, and the use of trial and error is labour-intensive and time consuming especially for the design real-life optimization problems [41]. This challenge has made the use of different values of control parameters to solve a particular problem using the same algorithm. Guided by theoretical analysis studies, empirical experiments, or self-adapting methods, the values assigned to PSO control parameters for any MOPS have been taken preferably in the range of $\omega = [0,1]$, $c_1 = [0.2, 2.5]$, and $c_2 = [0.2, 2.5]$ [13, 42–52]. The need to find an algorithm that performs well without fine-tuning the values of the PSO control parameters and without adding computation burden to the algorithm is desirable as stated in [53].

The MOPS uses customized different the values of the PSO control parameters obtained using theoretically analysis or empirically after numerous experiments. In some cases, different values are used for different test functions making each values of PSO control parameter to be tailored to a particular test functions. The MOPS that uses self-adapting methods could not provide good performance for all the tests conducted [41, 50, 51, 54–56]. In this paper, an attempt is made to design a MOPS that uses constant values for $\omega, c_1$and c_2 and perform competitively with the currently MOPS while maintening simplicity and efficiency of PSO.

3 The proposed algorithm

The NF-MOPSO is based on Coello Coello and Lechuga MOPSO algorithm [13]. The main differences between the NF-MOPSO and the MOPSO are the use of a Gaussian function as a mutation operator and the introduction of the PSO control parameters c_1 and c_2. The constant values used in the NF-MOPSO, $c_1 = c_2 = 1.4$ for the acceleration coefficients and $\omega = 0.7$ for the inertia weight are known to guarantee the convergence of the particles [7, 37]. The algorithm is presented in "Appendix 1".

A concept of "domination" [3, 57] is used to find Pareto optimal solutions. Assuming minimization a decision vector $\vec{x}^a$ is said to dominate $\vec{x}^b$ (denoted $\vec{x}^a \prec \vec{x}^b$) if and only if $f_k(\vec{x}^a) \leq f_k(\vec{x}^b)$, $\forall k = 1, \ldots, q$ and $\exists k = 1, \ldots, q : f_k(\vec{x}^a) < f_k(\vec{x}^b)$, and decision vector $\vec{x}^a$ is said to weakly dominate $\vec{x}^b$ (denoted $\vec{x}^a \preccurlyeq \vec{x}^b$) if and only if $f_k(\vec{x}^a) \leq f_k(\vec{x}^b)$, $\forall k = 1, \ldots, q$. A solution found by a particle in a search space can be either classified as a "dominated solution" or a "nondominated solution." A dominated solution is a solution whose one or more objectives can be improved without causing damage to others objectives in the same solution. A non-dominated solution is a solution whose objectives cannot be improved without degrading other objectives. A Pareto optimum solution is a nondominated solution. The Pareto-optimal solutions form part of the Pareto front, and they are the ideal solutions to a MOOP. A Pareto-optimal front is reached when there is no more possible good compromise between conflicted objectives.

3.1 Gaussian mutation operator

A Gaussian mutation operator described in Eq. (5) is uses in NF-MOPSO to improve diversity in the optimal solutions.

$$\varphi_i^a(t) = m_G + (\delta\sigma_i) \tag{5}$$

and

$$\sigma_i(t) = \left(x_{\max,i} - x_{\min,i}\right)\mathrm{e}^{-t} \tag{6}$$

where φ_i^a is the Gaussian mutation operator of a particle a for i decision variable, $m_G = 0$ is the Gaussian mean, δ is a random value in $[0, 1]$ taken from a Gaussian distribution with 0 as mean and 1 as a standard deviation, and σ_i is the standard deviation of the Gaussian mutation operator for i decision variable. The value φ_i^a is gradually decreased at each step time, and the threshold probability $P_{\varphi_i^a}$ of not using φ_i^a for i decision variable is increased at each time step following Eq. (7).

$$P_{\varphi_i^a}(t) = \frac{t}{t_{\max}} \tag{7}$$

with $t_{\max}$ as the maximum time step.

φ_i^a modifies randomly the position of the particle at each time step. The mutation is used on a particle if and only if the randomly generated probability $P_{x_i^a} \in [0, 1)$ of a particle a for i decision variable is $\geq P_{\varphi_i^a}$. Thus at each time step, the position of the particles is updated following the rule in Eq. (8).

$$x_i^a(t+1) = \begin{cases} x_i^a(t) + v_i^a(t+1) & \text{if } P_{x_i^a}(t) < P_{\varphi_i^a}(t) \\ \varphi_i^a(t)x_i^a(t) + v_i^a(t+1) & \text{if } P_{x_i^a}(t) \geq P_{\varphi_i^a}(t) \end{cases} \tag{8}$$

3.2 Constraint handling

The NF-MOPSO uses the death penalty [58] as constraint handling mechanisms. The death penalty makes the search for the optimum solutions to be performed only in the feasible regions. The death penalty method is applied during the initialization of the particle position and during the update of the position of particles. During initialization, the randomly generated position of a particle is retained if and only if it satisfies all the constraint conditions. The creation of new particles continues until the number of desirable particles is reached and all the particles satisfy all the constraint conditions. During the update of the position of the particle, the following penaly function defined in Eq. (9) is applied.

$$x_i^a(t+1) = \begin{cases} x_i^a(t+1) & \text{if constraint satisfied} \\ \text{random}(x_{Li}, x_{Ui}) & \text{while constraint not satisfied} \end{cases} \tag{9}$$

where x_{Li} is the lower bound for i decision variable, and x_{Ui} is the upper bound of for i decision variable.

The new position of the particle is retained only if it satifies the objective constraint conditions; otherwise, a new position of the particle is randomly generated until a position that satifies the objective constraint is found. The new position of the particle is generated within the boundary of the search space to satisfy the boundary constraint as well. It is important to note that the initial number of particles involved in the optimization process is not reduced, particles are only assigned new position if their updated positions do not satify the constraint condition. The penalty function in Eq. (9) allows only the feasible particles to be evaluated for dominance and also to be store in the external archive.

3.3 External archive

The NF-MOPSO uses an external archive to store the nondominated solutions found so far by the particles. The update of the external archive is based on the rules defined in Coello Coello and Lechuga's MOPSO. The non-dominated solutions are grouped together in the external archive, based on the values of their objectives. A hypercube is used to arrange the storage of the non-dominated solutions in the external archive. The NF-MOPSO does not use an adaptive grid as in MOPSO, instead a fixed size is used for the grid. The size of the grid depends on the range of the feasible objective space in each dimension. Prior information about the objectives space is required. The index H_k (storage location) for each non-dominated solution in the hypercubes is calculated using Eq. (10).

$$H_k = \frac{S_k}{U_k} \tag{10}$$

where S_k is the value of the non-dominated solution for the k objective function, and U_k is the grid unit size calculated based on the maximum range in the solution space of k objective function as

$$U_k = \frac{R_k}{N_g} \tag{11}$$

where R_k is the maximum range in the solution space of k objective function, and $N_g = 30$ is the number of grid in the hypercubes.

3.4 Local guide and global guide

The local guide of the particle represented by decision vector $\vec{x}^y$, is chosen among the next position of the particle or the current local guide using the Pareto-dominance in Eq. (12).

$$\vec{x}^y(t+1) = \begin{cases} \vec{x}^y(t) & \text{if } \vec{x}^y(t) \prec \vec{x}^a(t+1) \\ \vec{x}^a(t+1) & \text{if } \vec{x}^a(t+1) \prec \vec{x}^y(t) \\ \vec{x}^a(t+1) \text{ or } \vec{x}^y(t) & \text{if } \vec{x}^a(t+1) \not\preccurlyeq \vec{x}^y(t) \end{cases} \tag{12}$$

The new position of a particle is given 50% chance of being selected as a local guide if neither the new position or the current local guide dominate each other. The global guide particle is selected from the nondominated solutions in the external archive. In order to obtain a well distributed Pareto-optimal front, the global guide is selected from the less dense hypercube (contains less number of solutions). A roulette wheel is applied to all the hypercubes to select the less dense hypercube. The selection starts by calculating the probability P_h a hypercube has, as a provider of a global guide given in equation

$$P_h = \frac{\alpha_h}{n_h} \tag{13}$$

where $\alpha_h = 10$ is the hypercube density factor, n_h is the current number of non-dominated solutions in the hypercube. The hypercube is selected if

$$P_h \leq T_v \tag{14}$$

and

$$T_v = r_h \alpha_h \tag{15}$$

where $T_v \in [0, \alpha_h)$ is the selection threshold value, and $r_h \in [0, 1)$ is the generated random value. At each time step, a 1000 draws are performed, and then the hypercube with the highest number of selections after the 1000 trials is selected. The global guide is chosen randomly from non-dominated solutions stored at the selected hypercube. The local best of the selected solution becomes the global guide of the swarm.

4 Test and results

The performance of a MOOP is measured using mathematical benchmark test functions [59] and evaluated using performance metrics [60–62]. The benchmarking of the NF-MOPSO was done using 45 well-known constraint and unconstraint multi-objective test functions. The test functions used to evaluate the performance of NF-MOPSO falls into one or more of the following categories: unconstraint, constraint, two objectives, or three objectives test functions. The test functions include among others, the MOP test suite [63], the ZDT test suite [64], and the DTLZ test suite [65]. The test functions cover a long range of scenarios encountered in real-life optimization problem such as convex, nonconvex, continuous, discontinuous, and discrete Pareto-optimal fronts. Other scenarios covered in the test are deception, collateral noise, and bias search space. The number of decision variables used in the performance test varies from 2 and 30 decision variables.

The results of the test were evaluated both graphically and numerically. Four performance metric indicators namely the generational distance (GD), the inverted generational distance (IGD), the spacing (S), and the error ratio (ER) were used in numerically evaluation of the NF-MOPSO.

4.1 Performance metrics

Performance metrics are quality indicators that provide a quantitative evaluation of a MOOP performance [60]. The perfomance metrics measure the accuracy and diversity in the solutions find by the MOOP and also make it possible for a new MOOP to be tested for strengths and weakness with comparison to existing MOOP [66]. Hence, more than one metric is needed to evaluate the overall performance of the MOOP [64, 67]. The NF-MOPSO was evaluated quantitatively using four performance metric indicators. The performance metrics used to evaluate NF-MOPSO are described below:

The GD metric [68] presented in Eq. (16) measures how far, on average, the Pareto-optimal front produced by the algorithm is to the true Pareto-optimal front. A small value of GD indicates a good performance.

$$\mathrm{GD} = \frac{\left(\sum_{q=1}^{|A|} d_q^p\right)^{\frac{1}{p}}}{|A|} \tag{16}$$

where A is the number of solutions in the Pareto-optimal front produced by the algorithm, $p = 2$, d_q is the minimum Euclidean distance between the qth solution from A and the nearest solution in the true Pareto-optimal front.

The IGD metric [69] shown in Eq. (17) measures how far, on average distance, the true Pareto-optimal front is to the Pareto-optimal front produced by the algorithm. The IGD is the inverse of the GD. The IDG metric measures both the convergence and the spread of the Pareto-optimal front produced by the algorithm [16]. A small value of IGD indicates good performance.

$$\mathrm{IGD} = \frac{\left(\sum_{q=1}^{|T|} d_q^p\right)^{\frac{1}{p}}}{|T|} \tag{17}$$

where IGD is the inverted generational distance, T is the number of solutions in the true Pareto-optimal front, $p = 2$, d_q is the minimum Euclidean distance between the qth solution from T and the nearest solution in the Pareto-optimal front produced by the algorithm.

The S metric [70] in Eq. (18) provides an indication of the spread of the Pareto-optimal solutions produced by the algorithm. The S metric measures the variance of the distance of each Pareto-optimal solution to its closest neighbor. A small value of S indicates a good performance.

$$S = \frac{1}{|A|-1}\sum_{u=1}^{|A|}\left(\bar{d} - d_u\right)^2$$

$$\text{where } \bar{d} = \sum_{u=1}^{|A|}\frac{d_u}{A} \text{ and}$$

$$d_u = \min_j\left(\left|f_1^i(\vec{x}) - f_1^j(\vec{x})\right| + \left|f_2^i(\vec{x}) - f_2^j(\vec{x})\right|\right), \quad i,j = 1,\ldots,A \tag{18}$$

where S is the Spacing metric, A is the number of Pareto-optimal solutions produced by the algorithm, $\bar{d}$ is the mean of all d_u, d_u is the minimum distance between the uth solution and all other solutions in the Pareto set.

The ER metric [63] presented in Eq. (19) counts the number of solutions in the Pareto-optimal front produced by the algorithm that are not members of the true Pareto-optimal front. A small value of ER indicates a good performance [61].

$$ER = \frac{\sum_{i=1}^{|n|} e_i}{|n|} \tag{19}$$

where ER is the error ratio, n is the number of solutions in the Pareto-optimal front produced by the algorithm, $e_i = 0$ if ith solution from n belongs to the true Pareto-optimal front, and $e_i = 1$ otherwise.

4.2 Results

A multi-objective optimizer software based on NF-MOPSO algorithm was implemented using a java programming language [71]. Table 1 presents the constant values of the PSO parameters used in all optimization test.

For each test function, 30 tests runs were done. For each test function, the results of the evaluation are presented as the mean, the standard deviation, the minimum, and the maximum of each performance metric. Table 2 shows the evaluation test results, and Table 3 shows the summary of the test results. The test results are presented graphically in "Appendix 2" and "Appendix 3" where the Pareto front obtained by the NF-MPSO is visualized graphically with the true Pareto front. The results of the computational time for each test function are shown in Table 4 for two

Table 1 PSO setup parameters used in the test of NF-MOPSO

PSO set up parameter	Value
C1	1.4
C2	1.4
ω	0.7
Number of particles	100
The size of the external archive	500
Number of time steps	250–5000
Number of hypercubes in the external archive	30
Mean for Gaussian mutation operator	0
Coefficient of the velocity clamping function	1

objectives test functions, and in Table 5 for three objectives test functions.

The mean values presented in Table 2 were compared with the results from other algorithms namely the MOPSO [13], the SMOPSO [42], the HTL-MOPSO [16], MMOPSO [19], the dMOPSO [72], OMOPSO [73], and the EMOPSO [51]. The best values (smallest mean of GD, IGD, Spacing or ER) among the compared MOPS are highlighted in boldface for each test function in the results tables. A percentage score was used as a comparing factor. The MOPS was given a percentage score based on the number of its best values in performance metric indictor. The percentage scores presented in Tables 6, 7, 8, 9, 10, 11 and 12 indicate the number of time where one algorithm performed better than the other on test functions used in the comparison.

5 Discussion

The aim of this paper is to develop a competitive no fine-tuning PSO parameters algorithm for multi-objective optimization. Based on the results presented in Sect. 4.2, the following interpretations can be made:

5.1 Performance of NF-MOPSO on test functions

The results of the summary of performance test in Table 3 showed that NF-MOPSO has good average performance on all the 45 test functions used in the evaluation. The performance of the NF-MOPSO was achieved using the same value the PSO control parameters on all test functions. For 180 evaluations shown in Table 2, 93% have a performance value less or equal to 0.1. Graphical analysis of the test results in "Appendix 2" and "Appendix 3" confirmed the results in Table 2, and shows that the NF-MOPSO was able to find the theoretical true Pareto-optimal front and Pareto-optimal solutions of all the test functions used in the investigation. There is no single test function on which the NF-MOPSO performed worse on all performance metrics used in the test. The comparison test shown in Tables 6, 7, 8, 9, 10, 11, and 12 showed that the NF-MOPSO performed equally or better than other the recently improved algorithms.

The existence of many local Pareto fronts in the ZDT4 test function has made other algorithms to perform poorly and not finding the true Pareto front [16, 74, 75]. It has been reported in [76] and in [49] that MOPSO failed to find a true Pareto front for ZDT4. The improved MOPSO called OMOPSO also failed to find the true Pareto front for ZDT4 as reported in [74, 76] and in [19, 77], where the OMOPSO scored mean value of 4.44 for GD, 0.1 for IGD, 0.08 for Spacing, and 0.1 for ER. Other MOPS that were reported to fail to find the true Pareto front for ZDT4, are TV-MOPSO [16], DDMOPSO [19], and EMOPSO, with the means score value of 5.19, 1.66, and 8.11 for IGD, respectively. The NF-MOPSO performed well on ZDT4 test function based on the evaluation test results of a mean value of 0.008 for GD, 0.01 for IGD, 0.02 for Spacing, and 0.09 for ER. The plot in Fig. 1 confirms the good performance of the NF-MOPSO by showing that the true Pareto front of ZDT4 was found.

The frequently used values for c_1, c_2, and ω in literature may not guarantee a better performance for most the test function; and the use of fixed values for PSO parameters did not yield good results in the past [51, 73]. In case of NF-MOPSO, the values of the PSO control parameters performed well on all the 45 test functions evaluated. The good performance of the no-fine-tuned PSO control parameters ($c_1 = c_2 = 1.4$ and $\omega = 0.7$) of NF-MOPSO is due to the fact that it has been proven both theoretically and verified empirically that those values derived convergence conditions which guarantee that an equilibrium state will be reached and particles will converge to a solution [7, 37]. The use of Gaussian mutation operator that does not require the fine-tuning of a mutation rate (as it is in the case of MOPSO) also contributed the good performance of NF-MOPSO.

The use of randomly generated values of the PSO control parameters by some MOPS bring the uncertainty in the performance an algorithm; on the other hand, the use of empirically derived values for PSO control parameters recommended by the author may provide poor performance. For example the use of values of $c_1 = c_2 = 2.0$ and $\omega = 0.4$ as suggested by Coello Coello for MOPSO, generated few solutions for ZDT4 as reported in [49]. The weakness of the original MOPSO has been observed in finding uniform a distributed Pareto optimal solutions [13], hence it has failed to find the Pareto optimal solutions for ZDT4 [73]. Thus it was recommended to use additional

Table 2 Performance results of NF-MOPSO

Function	Performance metric															
	GD				IGD				Spacing				ER			
	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max
Belegundu	2.56E−04	7.62E−05	1.58E−04	3.45E−04	7.57E−04	9.22E−05	6.34E−04	8.72E−04	7.14E−04	7.87E−05	6.27E−04	8.00E−04	8.88E−03	4.59E−04	8.47E−03	9.62E−03
Bihn1	5.61E−03	9.94E−04	4.62E−03	7.01E−03	8.53E−03	7.62E−04	7.85E−03	9.81E−03	3.71E−02	6.52E−03	2.95E−02	4.65E−02	3.78E−03	2.35E−04	3.50E−03	4.07E−03
Bihn2	1.70E−02	7.95E−04	1.59E−02	1.80E−02	3.42E−02	5.38E−03	2.66E−02	4.17E−02	2.49E−01	6.15E−02	1.70E−01	3.15E−01	9.08E−02	1.30E−02	7.99E−02	1.12E−01
Bihn3	5.00E−07	0.00E + 00	5.00E−07	5.00E−07	1.00E−06	0.00E + 00	1.00E−06	1.00E−06	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
Bihn4	6.31E−03	6.98E−04	5.61E−03	7.19E−03	7.06E−03	3.01E−04	6.75E−03	7.38E−03	4.82E−03	1.10E−03	3.79E−03	6.40E−03	6.69E−03	1.16E−04	6.54E−03	6.80E−03
DEB1	1.32E−05	6.02E−07	1.25E−05	1.40E−05	9.34E−05	1.07E−05	7.71E−05	1.03E−04	1.41E−05	2.17E−06	1.07E−05	1.63E−05	4.81E−02	1.06E−03	4.76E−02	5.00E−02
DEB2	1.00E−04	2.72E−06	9.60E−05	1.04E−04	2.27E−04	3.75E−05	1.81E−04	2.70E−04	2.59E−05	8.93E−06	1.60E−05	3.55E−05	4.28E−02	9.92E−04	4.17E−02	4.35E−02
Deb3	4.67E−05	3.14E−06	4.24E−05	4.96E−05	6.13E−05	4.33E−06	5.42E−05	6.56E−05	9.75E−06	1.23E−06	7.99E−06	1.14E−05	5.38E−02	1.60E−03	5.26E−02	5.56E−02
Dtlz1	1.47E−02	1.45E−03	1.33E−02	1.70E−02	6.76E−03	3.37E−04	6.37E−03	7.27E−03	6.61E−02	3.41E−03	6.13E−02	7.00E−02	2.06E−02	1.19E−03	1.89E−02	2.17E−02
Dtlz2	2.28E−02	2.82E−04	2.24E−02	2.31E−02	1.51E−02	1.13E−03	1.43E−02	1.69E−02	2.53E−01	1.83E−02	2.39E−01	2.84E−01	5.72E−03	2.32E−04	5.32E−03	5.88E−03
Dtlz3	8.43E−02	7.65E−02	3.28E−02	2.20E−01	1.30E−02	1.70E−03	1.12E−02	1.54E−02	4.05E−01	2.81E−01	2.21E−01	9.00E−01	3.81E−02	4.93E−02	8.06E−03	1.25E−01
Dtlz4	3.36E−02	8.16E−04	3.28E−02	3.48E−02	1.33E−02	3.43E−03	7.45E−03	1.56E−02	3.12E−01	2.18E−02	2.89E−01	3.40E−01	5.75E−03	1.73E−04	5.59E−03	6.02E−03
Dtlz5	4.64E−03	8.47E−04	3.54E−03	5.83E−03	2.82E−04	2.56E−05	2.65E−04	3.27E−04	1.38E−04	9.38E−05	6.45E−05	2.73E−04	5.28E−02	3.24E−03	4.76E−02	5.56E−02
Dtlz6	8.33E−03	2.05E−03	4.94E−03	1.04E−02	1.06E−03	2.05E−04	9.06E−04	1.42E−03	1.94E−03	8.26E−04	7.48E−04	2.98E−03	3.53E−02	3.87E−03	3.23E−02	4.17E−02
Dtlz7	2.41E−02	1.13E−03	2.24E−02	2.54E−02	7.66E−03	7.89E−05	7.57E−03	7.75E−03	2.00E−01	2.25E−02	1.68E−01	2.29E−01	5.40E−03	1.52E−04	5.21E−03	5.56E−03
Dtlz8	1.76E−01	2.83E−03	1.72E−01	1.78E−01	6.47E−01	1.57E−02	6.27E−01	6.64E−01	8.35E−02	3.04E−02	4.69E−02	1.13E−01	1.56E−02	2.66E−03	1.32E−02	1.96E−02
Dtlz9	1.45E−02	2.37E−03	1.12E−02	1.67E−02	6.41E−03	9.53E−04	5.18E−03	7.56E−03	3.51E−02	9.51E−03	2.76E−02	5.08E−02	2.23E−02	3.81E−03	1.72E−02	2.63E−02
Fonseca1	1.72E−04	7.27E−05	1.10E−04	2.94E−04	3.65E−04	1.83E−04	1.97E−04	6.62E−04	3.57E−05	1.87E−05	1.80E−05	5.84E−05	4.76E−02	0.00E + 00	4.76E−02	4.76E−02
Fonseca2	7.68E−04	3.09E−05	7.33E−04	8.14E−04	9.76E−04	3.36E−04	6.03E−04	1.38E−03	1.60E−05	2.65E−06	1.33E−05	2.02E−05	4.81E−02	1.06E−03	4.76E−02	5.00E−02
Jimenez	6.22E−02	5.48E−03	5.66E−02	6.98E−02	8.42E−02	1.21E−02	7.06E−02	9.90E−02	2.38E−01	6.79E−02	1.83E−01	3.41E−01	8.03E−03	7.42E−04	7.19E−03	8.85E−03
Kita	4.27E−02	1.46E−02	2.82E−02	6.05E−02	1.94E−03	1.24E−04	1.82E−03	2.13E−03	3.10E−02	3.11E−02	7.54E−03	7.43E−02	2.96E−02	1.52E−02	1.43E−02	4.55E−02
Kursawe	1.55E−02	2.58E−02	2.79E−03	6.16E−02	1.34E−02	2.45E−02	2.13E−03	5.73E−02	3.72E−03	2.42E−03	2.14E−03	7.88E−03	1.98E−02	2.50E−02	8.06E−03	6.45E−02
Laumanns	1.84E−03	4.07E−05	1.80E−03	1.90E−03	6.46E−02	2.53E−04	6.43E−02	6.48E−02	2.46E−04	1.38E−04	1.56E−04	4.86E−04	1.38E−02	5.25E−04	1.32E−02	1.43E−02
Lis	1.23E−03	6.57E−04	2.84E−04	1.94E−03	4.78E−04	1.02E−04	3.40E−04	5.77E−04	1.30E−05	4.29E−06	8.72E−06	1.79E−05	6.43E−02	3.99E−03	6.25E−02	7.14E−02
Murata	3.90E−03	2.15E−03	9.77E−04	6.03E−03	7.80E−04	1.55E−04	5.32E−04	9.36E−04	1.31E−03	8.94E−04	1.69E−04	2.28E−03	4.06E−02	1.95E−02	2.08E−02	6.12E−02
Obayashi	1.00E−03	1.53E−04	8.16E−04	1.17E−03	4.35E−04	2.00E−05	4.07E−04	4.57E−04	1.03E−04	3.04E−05	6.47E−05	1.45E−04	5.05E−02	1.18E−03	5.00E−02	5.26E−02
Oka1	1.44E−02	1.70E−03	1.21E−02	1.65E−02	4.55E−02	3.39E−03	4.01E−02	4.95E−02	1.36E−02	1.18E−02	5.54E−03	3.44E−02	2.74E−02	3.53E−03	2.17E−02	3.03E−02
Osyczka1	1.60E−03	7.54E−05	1.52E−03	1.69E−03	3.77E−03	3.39E−03	3.15E−03	4.59E−03	1.02E−03	1.31E−04	8.28E−04	1.19E−03	6.13E−02	5.07E−03	5.56E−02	6.67E−02
Poloni	2.30E−02	9.07E−03	8.37E−03	3.32E−02	4.07E−02	2.24E−02	3.84E−03	6.33E−02	5.29E−02	2.99E−02	1.33E−02	9.28E−02	5.69E−02	4.06E−02	2.91E−02	1.28E−01
Rendon1	1.40E−02	8.95E−04	1.25E−02	1.48E−02	9.27E−03	2.06E−05	9.24E−03	9.30E−03	1.95E−02	5.40E−03	1.46E−02	2.75E−02	5.58E−03	3.70E−04	5.29E−03	6.17E−03
Rendon2	1.13E−02	1.87E−03	8.47E−03	1.33E−02	4.90E−03	7.90E−04	3.88E−03	5.83E−03	5.49E−03	1.50E−03	3.38E−03	7.00E−03	1.48E−02	1.26E−03	1.28E−02	1.59E−02
Schaffer1	9.98E−04	2.44E−04	7.75E−04	1.36E−03	1.16E−03	1.97E−04	1.03E−03	1.50E−03	2.66E−04	5.64E−05	2.06E−04	3.28E−04	1.43E−02	3.26E−04	1.37E−02	1.45E−02
Schaffer2	1.08E−03	3.68E−04	7.48E−04	1.61E−03	1.02E−02	1.16E−02	6.18E−04	2.77E−02	5.37E−04	1.02E−04	3.90E−04	6.64E−04	1.16E−02	2.14E−04	1.14E−02	1.19E−02
Srinivas	1.61E−02	2.68E−03	1.24E−02	1.90E−02	1.07E−02	1.28E−03	9.61E−03	1.27E−02	7.19E−01	2.07E−01	5.05E−01	1.06E + 00	3.37E−03	1.93E−04	3.19E−03	3.66E−03
Tamaki	8.48E−02	2.09E−03	8.24E−02	8.81E−02	6.16E−01	1.64E−02	5.98E−01	6.36E−01	2.60E−01	9.60E−03	2.49E−01	2.70E−01	4.43E−03	8.48E−05	4.33E−03	4.52E−03
Tanaka	3.00E−03	1.30E−03	1.37E−03	4.27E−03	1.24E−03	2.28E−04	8.63E−04	1.44E−03	6.53E−04	2.55E−04	2.00E−04	7.94E−04	6.12E−02	4.21E−03	5.56E−02	6.67E−02
Viennet1	1.58E−02	1.23E−03	1.37E−02	1.67E−02	1.77E−02	1.66E−02	1.02E−02	4.73E−02	3.55E−01	1.94E−02	3.40E−01	3.88E−01	1.25E−02	2.44E−02	1.54E−03	5.60E−02
VIENNET2	5.90E−03	1.89E−03	3.70E−03	8.42E−03	5.36E−03	1.39E−05	5.34E−03	5.38E−03	5.31E−03	1.24E−03	3.62E−03	6.63E−03	3.35E−02	3.02E−03	3.03E−02	3.85E−02
VIENNET3	7.02E−04	3.10E−05	6.60E−04	7.34E−04	4.28E−03	4.74E−06	4.28E−03	4.29E−03	3.68E−04	6.82E−05	2.60E−04	4.37E−04	9.25E−03	3.07E−04	8.85E−03	9.52E−03
VIENNET4	3.40E−02	2.50E−03	3.15E−02	3.77E−02	1.26E−02	3.08E−04	1.22E−02	1.30E−02	2.31E−01	4.46E−02	1.88E−01	2.94E−01	4.97E−02	1.25E−02	3.21E−02	6.35E−02
ZDT1	9.26E−03	2.05E−02	7.12E−05	4.59E−02	4.30E−03	9.20E−03	1.74E−04	2.08E−02	1.64E−04	3.25E−04	1.67E−05	7.46E−04	5.06E−02	6.65E−03	4.76E−02	6.25E−02
ZDT2	5.31E−05	1.25E−06	5.16E−05	5.50E−05	1.62E−04	1.35E−05	1.44E−04	1.76E−04	1.37E−05	1.11E−06	1.30E−05	1.56E−05	4.81E−02	1.06E−03	4.76E−02	5.00E−02
ZDT3	4.37E−04	4.89E−05	3.95E−04	5.19E−04	4.22E−04	3.98E−05	3.93E−04	4.91E−04	1.92E−05	5.42E−06	1.24E−05	2.76E−05	3.73E−02	6.37E−04	3.70E−02	3.85E−02

Table 2 (continued)

Function	Performance metric															
	GD				IGD				Spacing				ER			
	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max
ZDT4	8.74E−03	1.08E−02	7.78E−04	2.77E−02	1.29E−02	1.05E−02	1.79E−03	3.00E−02	2.57E−02	3.98E−02	3.04E−04	9.63E−02	9.68E−02	3.55E−02	4.55E−02	1.30E−01
ZDT6	5.18E−03	7.59E−03	4.83E−04	1.86E−02	6.37E−04	6.12E−04	1.44E−04	1.71E−03	1.17E−03	2.38E−03	7.09E−06	5.41E−03	5.56E−02	2.19E−03	5.26E−02	5.88E−02

techniques such as crowding or fine-tuned parameters to improve on the performance of MOPSO. The additional techniques such as a hybrid methods used in HTL-MOPSO, SMPSO, and MMPSO, compromise the quality of simplicity and implementation of PSO, and bring additional computational time. The OMOPSO (which is an improvement of the original MOPSO) uses crowding, mutation, and e-dominance techniques along with randomly generated values of the PSO control parameters. Another improvement version of the MOPSO is the EMOPSO that uses clustering technique and self-adaptive values for PSO parameters. The results in Tables 7 and 8 shows that the NF-MOPSO outperformed both the OMOPSO and EMOPSO based on the performance metrics used in the evaluation. The authors of both OMOPSO and EMOPSO reported finding it difficult to find fixed values of PSO parameters using their approach [51].

5.2 Sensitivity of PSO control parameters on NF-MOPSO

The sensitivity of NF-MOPSO on the values of the PSO control parameters was tested using 100 sets of values that were randomly generated in the range of $c_1 = c_2 = [0.2, 2.5]$ and $\omega = [0, 1]$. The range was chosen based on the reason discussed in Sect. 2.3. The sensitivity of the values of the PSO control parameters was examined on ZDT4 test problem as an example of one of the most difficult multifrontal problems. The results presented in Fig. 2 shows that the performance of NF-MOPSO is sensitive to the values of the PSO control parameters as expected [51], different performance mean values (good or bad) were produced by different sets for different performance metrics.

The NF-MOPSO uses constant values of PSO parameters that are known to guarantee a convergence to a solution in order to avoid the ambiguity that are observed when other values (such as randomly generated values) are used. The results presented in Table 13 (data colored bold for mean < 0.1, italic for $0.1 \leq$ mean < 1 , and bold italics for mean ≥ 1) shows that in the range of $c_1 = c_2 = [0.2, 2.5]$, the set of values closed to $c_1 = c_2 = 1.4$ have the best performance in all four performance metrics used in the evaluation.

5.3 Running time of NF-MOPSO

Using an 8-Core Intel CPU with 8 GB RAM, the running time of NF-MOPSO was on average 3.95 s for two objectives test functions, and 95 s for three objective test functions (Tables 4, 5). The best and worst running time was 0.27 and 44 s, respectively,, for two objectives test functions; 12.67 and 368.73, respectively, for three

Table 3 Summary of the performance results of NF-MOPSO

Function	Performance metric															
	GD				IGD				Spacing				ER			
	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max
All functions	1.76E−02	3.14E−02	5.00E−07	1.76E−01	3.84E−02	1.30E−01	1.00E−06	6.47E−01	8.03E−02	1.49E−01	0.00E+00	7.19E−01	3.18E−02	2.41E−02	0.00E+00	9.68E−02

objectives test functions. The use of the values of PSO parameters guarantee a convergence to a solution, and by maintaining the simplicity and easy implementation that characterizes the original PSO, make the computational time of NF-MOPSO to be better than some recent MOP such as DDMOPSO (48.96 s), and also competitive with recent MOPS such as dMOPSO (2.20 s) and MMOPSO (2.1 s) for two objectives test functions [19].

The result in Tables 4 and 5 shows that the running time is proportional to the number of objectives n and the number of iterations (time step t). The number of objectives is native to the objective function, thus cannot be fine-tuned. Thus, it is clear that the computational time of NF-MOSPO on many objective optimization problems (MOOP with more than three objectives [78, 79]) will increase. The number of iterations is the only PSO set-up parameter that is fine-tuned in NF-MOPSO. Thus the optimal number of iterations is problem-dependent [80], the number of iterations is one of the PSO parameters that can be easily fine-tuned unlike the PSO control parameters. The rule of thumbs is that less number of iteration may cause the search to terminate prematurely [81], and large number of iterations may improve results of optimization at higher cost time [82]. The NF-MOPSO uses 500 or 5000 number of iterations which is still competitive with the number of iterations used in OMOPSO (20,000 iterations), EMOPSO (2000 iterations), dMOPSO (15,000 iterations for two objectives and 45,000 iterations for three objectives), HTL-MOPSO (500–1000), SMOPSO (2000 and 7000 iterations), and MMOPSO (60,000 and 178,000 iterations). The running time of the NF-MOPSO can further be improved by integrating parallel computation mechanism to take advantage of all computing resources such us a number of cores available on a computer.

6 Application in real-life optimization

The practicability and availability of NF-MOPSO were tested in the optimization of the geometric design of a neutron radiography collimator [83, 84]. Neutron radiography is a two-dimensional projection imaging technique (similar to X-ray radiography) that utilizes penetrating radiation (neutrons) to retrieve qualitative and quantitative information of the internal structure of samples under investigation. The integrity of the investigation relies on the quality of a radiograph. A neutron collimator is one of the most important components that define the quality (contrast, resolution) of an image produced by a neutron radiography system [85–88]; thus the design of the neutron collimator needs to be simulated and optimized in virtual environment to improve the quality of a neutron radiography system [89–92].

Table 4 The average computational time (in second) of NF-MOPSO for two objectives functions

Test function	No. of variables	No. of objectives	No. iterations	Average runtime (s)
Belegundu	2	2	500	2.63
Bihn1	2	2	500	0.90
Bihn2	2	2	500	1.00
Deb1	2	2	500	1.33
Deb2	2	2	500	0.93
Deb3	2	2	500	1.30
Fonseca1	3	2	500	0.63
Fonseca2	3	2	1000	1.10
Jimenez	2	2	5000	11.27
Kita	2	2	2000	10.73
Kursawe	3	2	5000	5.77
Laumanns	2	2	500	0.47
Lis	2	2	1000	1.23
Murata	2	2	5000	6.43
Obayashi	2	2	500	0.70
Oka1	2	2	5000	11.47
Osyczka1	2	2	5000	44.00
Poloni	2	2	500	0.60
Rendon1	2	2	500	0.63
Rendon2	2	2	500	0.60
Schaffer1	1	2	500	1.33
Schaffer2	1	2	500	1.03
Srinivas	2	2	500	0.83
Tanaka	2	2	500	0.27
Zdt1	30	2	500	1.90
Zdt2	30	2	500	1.90
Zdt3	30	2	500	1.53
Zdt4	10	2	500	1.00
Zdt6	10	2	500	0.90

Table 5 The average computational time (in second) of NF-MOPSO for three objectives functions

Test function	No. of variables	No. of objectives	No. iterations	Average runtime (s)
Bihn3	2	3	500	19.90
Bihn4	2	3	500	17.47
Dtlz1	12	3	1500	65.27
Dtlz2	12	3	500	19.67
Dtlz3	12	3	500	12.67
Dtlz4	12	3	1000	37.77
Dtlz5	12	3	5000	187.93
Dtlz6	12	3	5000	169.37
Dtlz7	12	3	1000	35.00
Dtlz8	30	3	1000	36.47
Dtlz9	30	3	5000	368.73
Tamaki	3	3	5000	178.30
Viennet1	2	3	5000	182.30
Viennet2	2	3	3000	103.80
Viennet3	2	3	1000	41.80
Viennet4	2	3	1000	47.73

Table 6 Comparison of score between NF-MOPSO and MOPSO

Test function	GD		Spacing		ER	
	NF-MOPSO	MOPSO	NF-MOPSO	MOPSO	NF-MOPSO	MOPSO
Kursawe	4.27E-02	**3.65E−02**	**3.72E−03**	1.09E−01	**2.96E−02**	1.33E−01
Laumanns	1.55E-02	**8.45E−03**	**2.46E−04**	9.75E−02	**1.98E−02**	2.54E−01
Score (%)	0	100	100	0	100	0

Table 7 Comparison of score between NF-MOPSO and OMOPSO

Test function	IGD	
	NF-MOPSO	OMOPSO
ZDT1	**4.30E−03**	1.00E−02
ZDT2	**1.62E−04**	3.40E−02
ZDT4	**1.29E−02**	3.00E−02
DTLZ6	**1.06E−03**	6.50E−03
Score (%)	100	0

Table 8 Comparison of score between NF-MOPSO and EMOPSO

Test function	IGD	
	NF-MOPSO	EMOPSO
ZDT1	4.30E−03	**2.20E−03**
ZDT2	**1.62E−04**	7.00E−04
ZDT3	**4.22E−04**	2.00E−03
ZDT4	**1.29E−02**	8.11E+00
ZDT6	**6.37E−04**	9.80E−02
Score (%)	80	20

Table 9 Comparison of score between NF-MOPSO and dMOPSO

Test function	Spacing	
	NF-MOPSO	dMOPSO
DTLZ2	2.53E−01	**2.42E−02**
DTLZ6	**1.94E−03**	3.56E−03
DTLZ7	2.00E−01	**1.03E−01**
FONSECA2	**1.60E−05**	4.12E−03
ZDT1	**1.64E−04**	4.82E−03
ZDT2	**1.37E−05**	4.23E−03
ZDT3	**1.92E−05**	1.67E−02
ZDT4	2.57E−02	**6.00E−03**
ZDT6	**1.17E−03**	2.76E−03
Score (%)	67	33

Table 10 Comparison of score between NF-MOPSO and HTLMOPSO

Test function	IGD	
	NF-MOPSO	HTLMOPSO
DTLZ1	**6.76E−03**	2.06E−02
DTLZ2	**1.51E−02**	6.02E−02
DTLZ5	**2.82E−04**	3.78E−03
DTLZ7	**7.66E−03**	4.35E−02
ZDT1	4.30E−03	**3.88E−03**
ZDT2	**1.62E−04**	3.85E−03
ZDT3	**4.22E−04**	4.82E−03
ZDT4	1.29E−02	**3.99E−03**
ZDT6	**6.37E−04**	3.08E−03
Score (%)	78	22

Table 11 Comparison of score between NF-MOPSO and SMOPSO

Test function	Spacing		GD	
	NF-MOPSO	SMOPSO	NF-MOPSO	SMOPSO
Bihn2	2.49E−01	**1.16E−01**	1.70E−02	**2.69E−03**
Deb2	**2.59E−05**	3.40E−03	**1.00E−04**	2.98E−04
Viennet3	**9.25E−03**	3.96E−01	**7.02E−04**	1.11E−02
Score (%)	67	33	67	33

6.1 Neutron collimator optimization problem

The neutron collimator optimization is normally performed to allow the productions of a high neutron intensity, and a large area of homogenous neutron beam at the position of the image formation [89–91]. The measurement of the flux profile (intensity) and the homogeneity of the neutron beam are illustrated in Fig. 3

The ideal neutron radiography beam should have a large area of homogeneity, a high neutron flux at the detector position. A large area of homogeneity provides a large field of view, thus allows the investigation of bigger samples. A high neutron flux gives better contrast and gives the possibility of shortening the time for the neutron scan. The two main objectives (large area and high flux) are in conflict

Table 12 Comparison of score between NF-MOPSO and MMOPSO

Test function	IGD	
	NF-MOPSO	MMOPSO
Dtlz1	**6.76E−03**	1.01E−02
Dtlz2	**1.51E−02**	2.74E−02
Dtlz3	**1.30E−02**	2.75E−02
Dtlz4	**1.33E−02**	2.85E−02
Dtlz5	**2.82E−04**	6.61E−02
Dtlz6	1.06E−03	**6.44E−04**
Dtlz7	**7.66E−03**	2.86E−02
Fonseca2	**9.76E−04**	1.86E−03
Kursawe	**1.34E−02**	1.63E−02
Schaffer1	**1.16E−03**	8.00E−03
Schaffer2	1.02E−02	**8.00E−03**
ZDT1	4.30E−03	**1.87E−03**
ZDT2	**1.62E−04**	1.91E−03
ZDT3	**4.22E−04**	2.10E−03
ZDT4	1.29E−02	**1.84E−03**
ZDT6	**6.37E−04**	1.56E−03
Score (%)	75	25

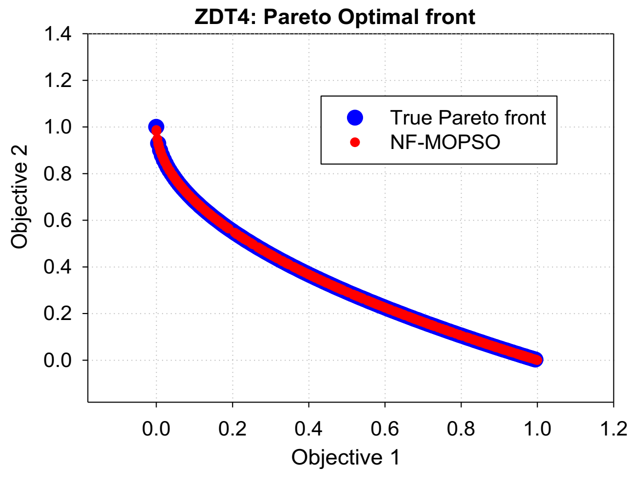


Fig. 1 NF-MOPSO performance results on ZDT4 test function

with each other. This makes a neutron collimator optimization a real-life multi-objective optimization problem.

In this example, the NF-MOPSO was tested in the optimization of the geometrical design of a neutron radiography collimator at the beam tube 2 situated at the SAFARI-1 nuclear research reactor at Nesca [93]. The objective of this optimization is to find the best size and the best position of the collimator aperture that optimize the intensity and the flat part of the flux profile (homogeneity of the neutron beam). The size and the position of the aperture were defined in term of the diameter D and the position L of the aperture relative to the detector (assuming the sample is placed very close to the detector position), respectively, as illustrated in Fig. 4.

The L/D ratio characterizes the ideal neutron beam [86]. A high L/D ratio is desirable, but the high L/D ratio comes with the cost of the neutron beam intensity.

The objectives of the optimization function were the neutron flux and the area of homogeneity. The maximum gray value G_v and the radius of beam homogeneity R_h were used to measure the neutron flux and the area of homogeneity, respectively. A multi-objective optimization problem for a neutron collimator was defined mathematically as:

$$\text{Maximize}\ \vec{f}_{\text{coll}}(\vec{p}_{\text{coll}})\ \text{subject to} \tag{20}$$

$$q_j(\vec{p}_{\text{coll}}) \geq c_j \quad j = \{\text{flux}, \text{homogeneity}\} \tag{21}$$

$$p_{\min} \leq p_i \leq p_{\max} \quad i = \{1, 2, 3, \ldots, m\} \tag{22}$$

where $\vec{f}_{\text{coll}} = (f_{\text{flux}}, f_{\text{homogeneity}})$ is a vector function representing the objective functions for collimator optimization, f_{flux} and $f_{\text{homogeneity}}$ are objectives functions for collimator optimization, $\vec{p}_{\text{coll}}$ is a vector of decision variables representing the collimator parameters D and L, q_j is the inequality constraint function, c_j is the constraint value for objective j, $p_{\min}$ and $p_{\max}$ are the lower and upper boundaries, respectively, for decision variable p_i.

The values of $\vec{f}_{\text{coll}}$ were computed using a radiography simulator presented in [94] in which the NF-MOPSO was integrated with as illustrated in Fig. 17 of "Appendix 4". The allowed positions of the aperture are at a distance between 300 and 900 cm from the entrance window of the neutron beam tube as illustrated in Fig. 4. The maximum possible diameter of the aperture was determined by the diameter of the entrance window of the neutron beam tube. The boundary conditions for the decision space are defined in Table 14 using the allowed position and maximum diameter of the aperture.

6.2 Results and discussion

The NF-MOPSO was used to optimize the collimator problem using the same PSO setup parameter presented in Table 1. The average time for the collimator optimization was approximately 12 h on an 8-Core Intel CPU with 8 GB RAM computer. The high running time is due to the computational cost required to simulate each radiographic image of a real like neutron radiography system using a simulator. The results of optimization are presented in Fig. 5.

The solutions to a geometrical optimization of a collimator are presented in Fig. 5 in form of Pareto optimal front and the corresponding Pareto optimal solutions both

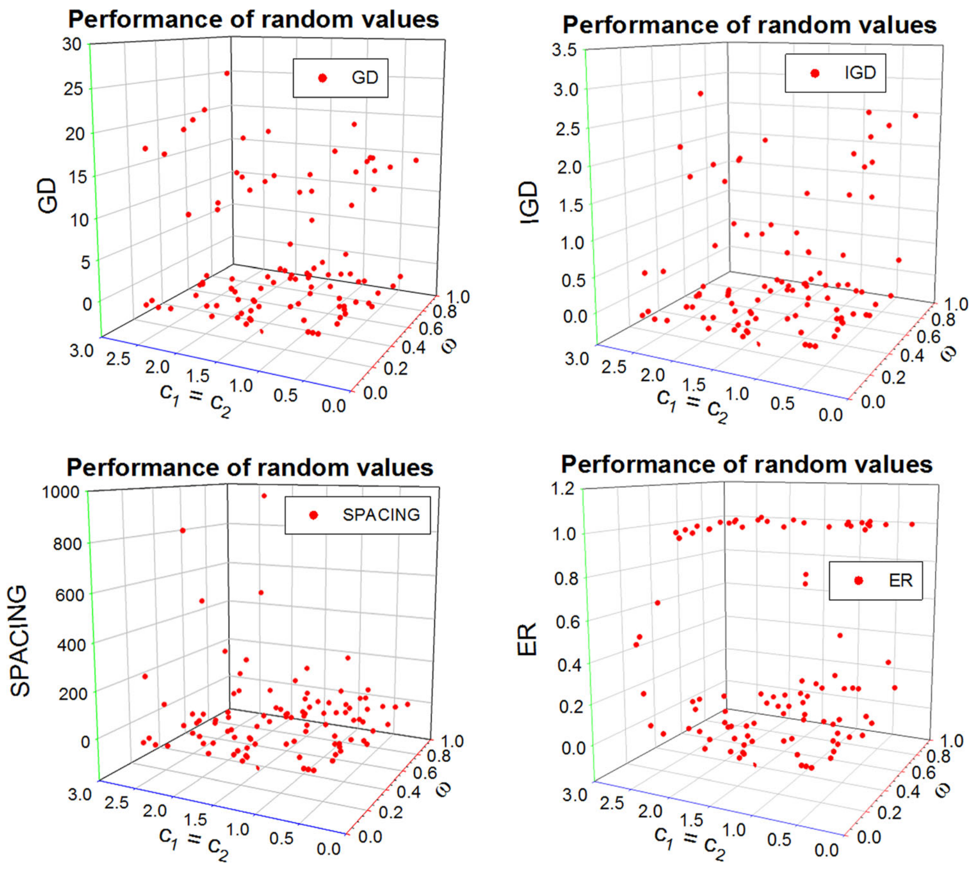


Fig. 2 NF-MOPSO performance on ZDT4 using randomly generated values of the PSO control parameters

Table 13 Performance values of $c_1 = c_2$.

Set no.	PSO parameters			Performance metrics			
	$c1$	$c2$	ω	GD	IGD	Spacing	ER
1	0.2	0.2	0.7	**1.05E-01**	*8.54E−02*	**3.80E−01**	**3.33E−01**
2	0.4	0.4	0.7	**1.49E-01**	*8.27E−02*	**1.55E−01**	**5.19E−01**
3	0.6	0.6	0.7	*4.55E-02*	*1.81E−02*	*1.12E−02*	**3.75E−01**
4	0.8	0.8	0.7	***2.56E+01***	***2.61E+00***	***6.27E+02***	***1.00E+00***
5	1	1	0.7	*8.30E-03*	*9.95E−03*	5.80E−03	**2.00E−01**
6	1.2	1.2	0.7	*2.91E-02*	*5.60E−02*	**1.93E−01**	**1.30E−01**
7	1.4	1.4	0.7	*3.83E-03*	*9.14E−03*	*7.16E−03*	*9.09E−02*
8	1.6	1.6	0.7	*5.03E-03*	*1.12E−02*	*1.14E−02*	*9.09E−02*
9	1.8	1.8	0.7	*2.57E-02*	*1.41E−02*	*5.11E−02*	**1.30E−01**
10	2	2	0.7	***3.32E+00***	**1.12E−01**	***2.20E+01***	**8.18E−01**
11	2.2	2.2	0.7	***1.33E+01***	***1.74E+00***	***1.10E+02***	***1.00E+00***
12	2.4	2.4	0.7	***1.90E+01***	***2.29E+00***	***2.80E+02***	***1.00E+00***
13	2.5	2.5	0.7	***1.73E+01***	***2.65E+00***	***2.89E+01***	***1.00E+00***

obtained using NF-MOPSO. The 300-cm aperture position taken by all the solutions in Fig. 5 (right hand side) are at the lower boundary of the search space ($p_{\min} = 300$ in Table 14). Further analyses were conducted to confirm if the optimization solutions favor the aperture position that is the closest to the neutron source. The analyses were done by repeating the neutron collimator optimization with different search ranges (different values of $p_{\min}$). The results of the analysis are shown in Fig. 6.

The results in Fig. 6 confirmed that all the Pareto optimal solutions provide the aperture position at the lower bound $p_{\min}$ of the search range. The lower bound of the search range is the closest position to the neutron source. The results in Fig. 6 also show that the number of possible optimal solutions decreased as the search range moves further away from the neutron source. An MCNP simulation [95] of the flux profile confirmed the results in Fig. 6 that shows that the best profile conditions will be achieved with the aperture position nearest to the neutron source.

Five L/D ratio taken from five Pareto optimal solutions in Fig. 5 (right hand side) were taken as example inputs to the simulation to visualize the output radiographs of an optimized neutron radiography collimator. The simulated radiography and their analysis presented in Fig. 7 shows that the NF-MOPSO was able to provide Pareto optimal solutions within which one or more solutions can be selected in design of a neutron radiography collimator within the boundary or design conditions.

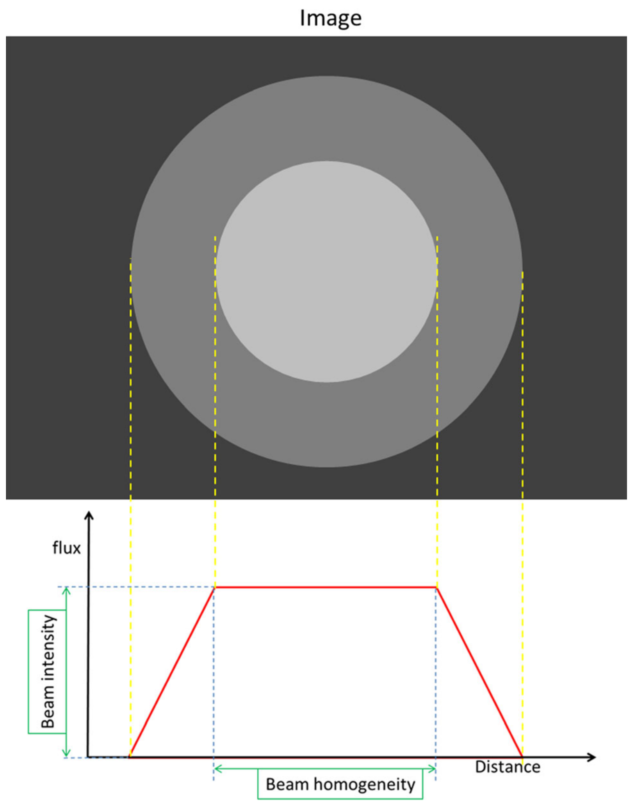


Fig. 3 Illustration of measurement of the beam intensity and homogeneity from the image

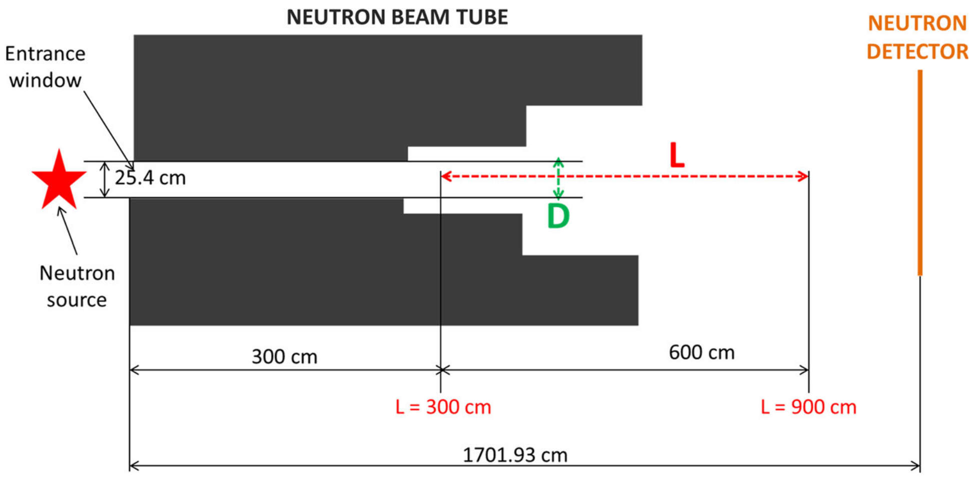
7 Conclusion

A multi-objective particle swarm optimization called the NF-MOPSO was presented. The NF-MOPSO used the same constant values for PSO control parameters throughout the optimization process and for all test functions used in the study. The advantage of using the values of PSO control parameters that guarantee an equilibrium state and convergence to a solution, has been demonstrated in NF-MOPSO. The values of the control parameters c_1 and c_2, and ω in the NF-MOPSO performed well on all test functions used in the study. The NF uses a Gaussian mutation operator with decreasing mutation rate that does not require manual fine-tuning. The search mechanism used in NF-MOPSO make is to be more efficient and easy to implement since it uses the generic PSO techniques. The NF-MOPSO has improved the performance of the original MOPSO of Coello Coello and Lechuga. Furthermore the test results showed that the NF-MOPSO performance and the running time are competitive compared with existing state of the art MOPS. The capability of the NF-MOPSO was also demonstrated in a real-life MOOP, where a collimator optimization problem was solved by the NF-MOPSO without fine-tuning the PSO control parameters. The NF-MOPSO was able to provide Pareto optimal solutions within which one or more solutions can be selected to design of a neutron radiography collimator. Future work will focus on integration of parallel computing in NF-MOPSO to further improve computational time when applied in real-life optimization problems; other focus will explore the possibility of using the same approach used in NF-MOPSO to solve many objectives optimization problems.

Table 14 Boundary of the search space for collimator optimization

Decision variable	$p_{\min}$	$p_{\max}$
Aperture diameter (cm)	0.07	25
Aperture position (cm)	300	900

Fig. 4 Illustration of the geometry of the radiography beam tube at the SAFARI-1 nuclear research reactor



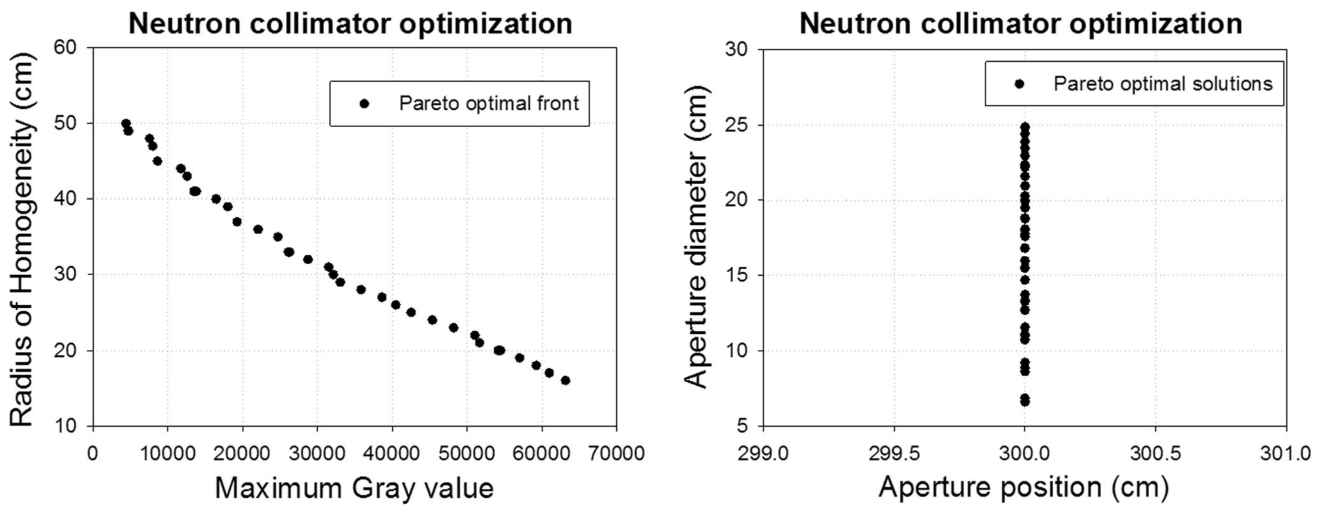
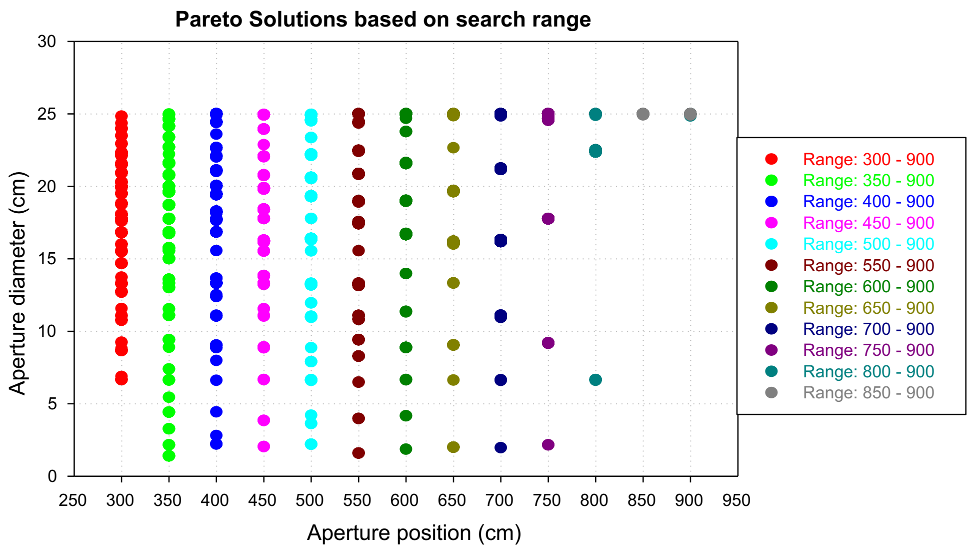


Fig. 5 Pareto optimal front (right) and the corresponding Pareto optimal solutions (left) for neutron radiography collimator

Fig. 6 Pareto optimal solutions obtained from different boundary conditions of the search space



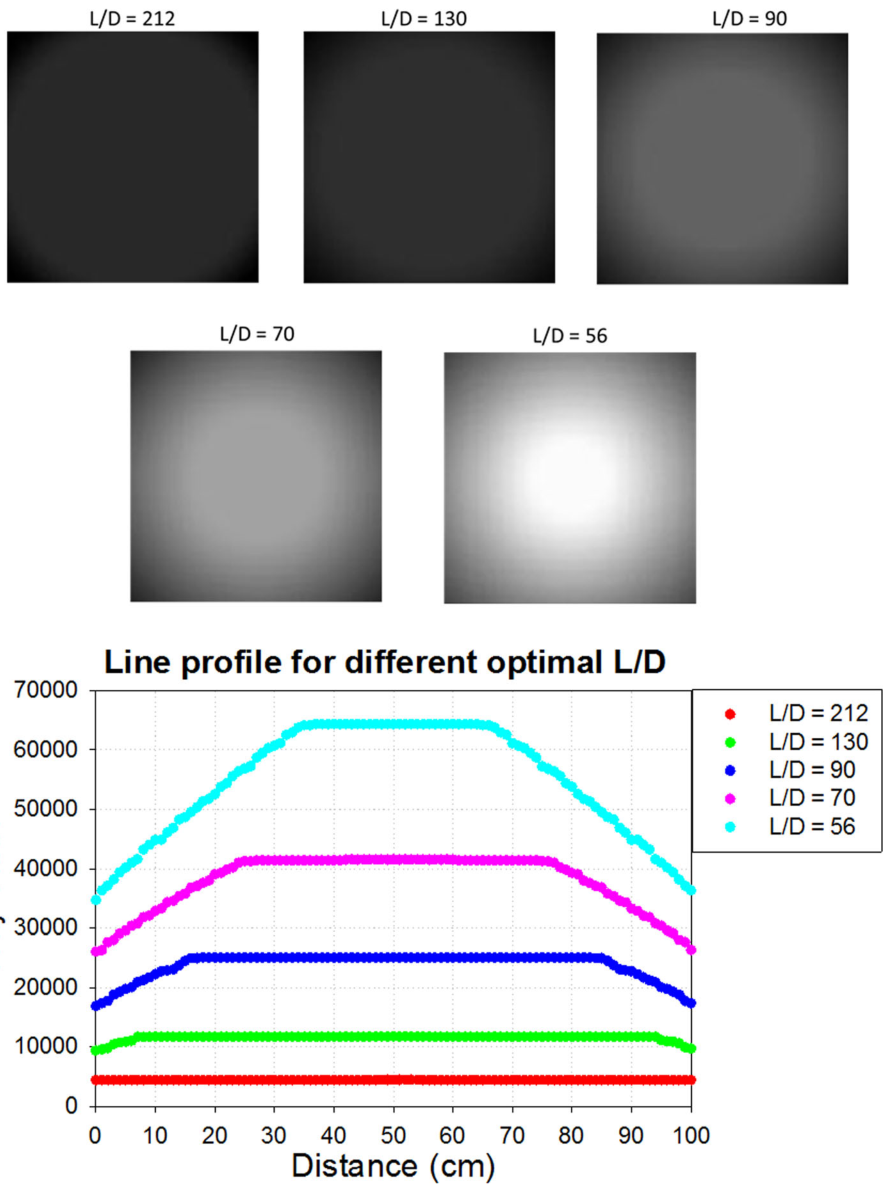


Fig. 7 Visualization and analysis of simulated radiography using five L/D ratio from Pareto optimal solutions

Acknowledgements The author wishes to thank the financial support of the South African Nuclear Energy Corporation and the National Research Fund.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix 1: NF-MOPSO algorithm

See Fig. 8.

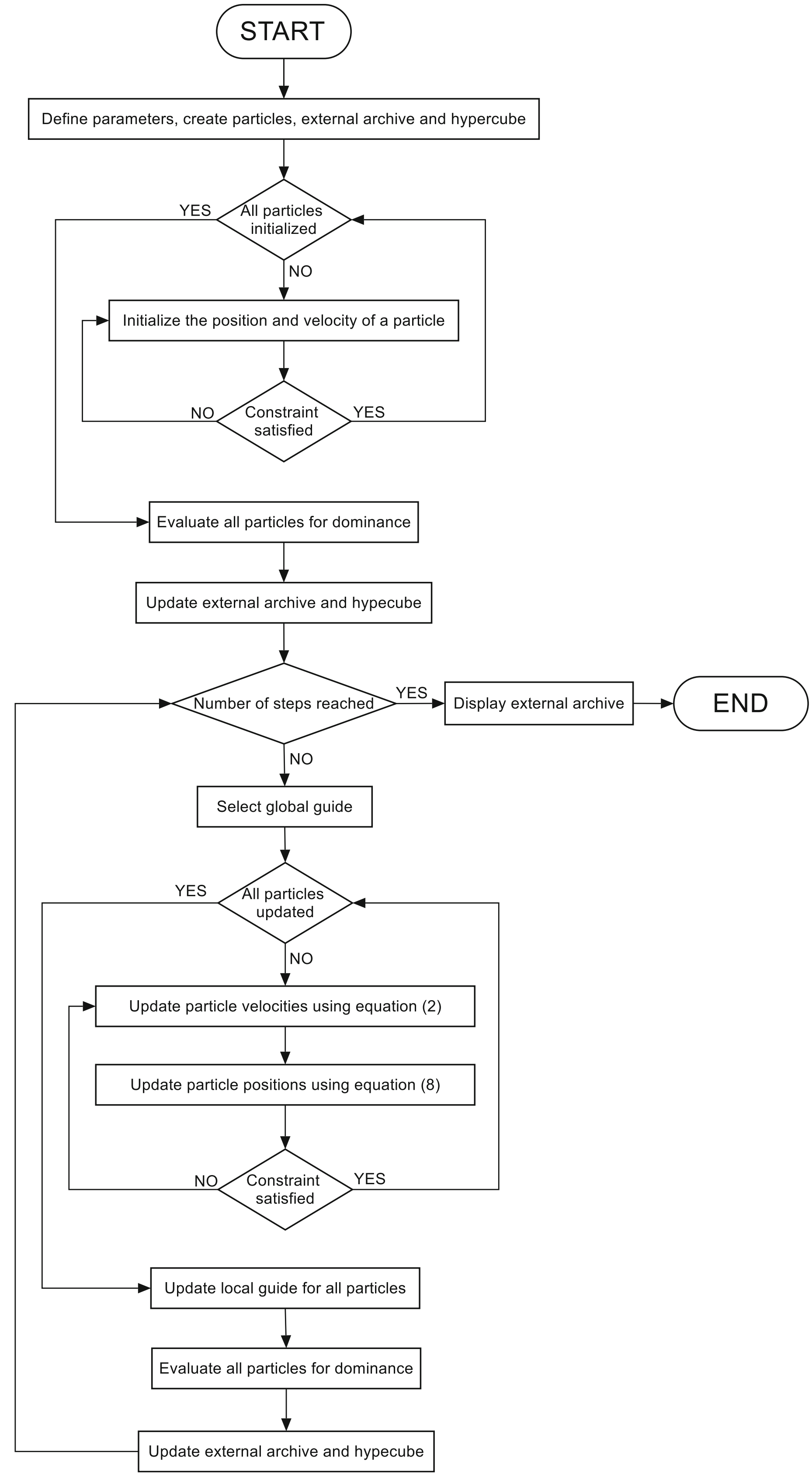


Fig. 8 NF-MOPSO algorithm
START
Define parameters, create particles, external archive and hypercube
All particles initialized
YES
NO
Initialize the position and velocity of a particle
Constraint satisfied
NO
YES
Evaluate all particles for dominance
Update external archive and hypecube
Number of steps reached
YES
Display external archive
END
NO
Select global guide
All particles updated
YES
NO
Update particle velocities using equation (2)
Update particle positions using equation (8)
Constraint satisfied
NO
YES
Update local guide for all particles
Evaluate all particles for dominance
Update external archive and hypecube

Appendix 2: Performance results of two objectives test functions

See Figs. 9, 10, 11, 12, 13, 14.

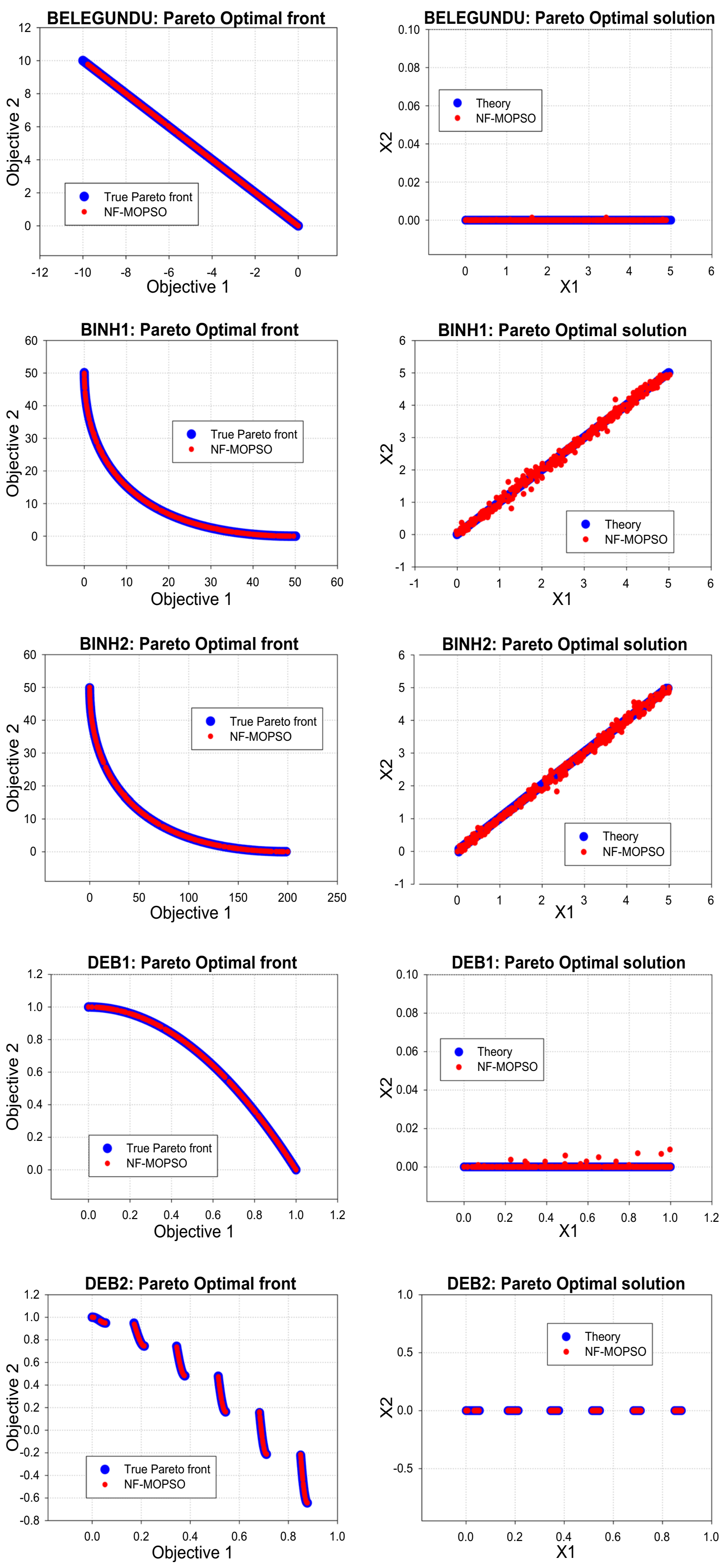


Fig. 9 Graphical performance results for two objectives test functions of NF-MOPSO vs true

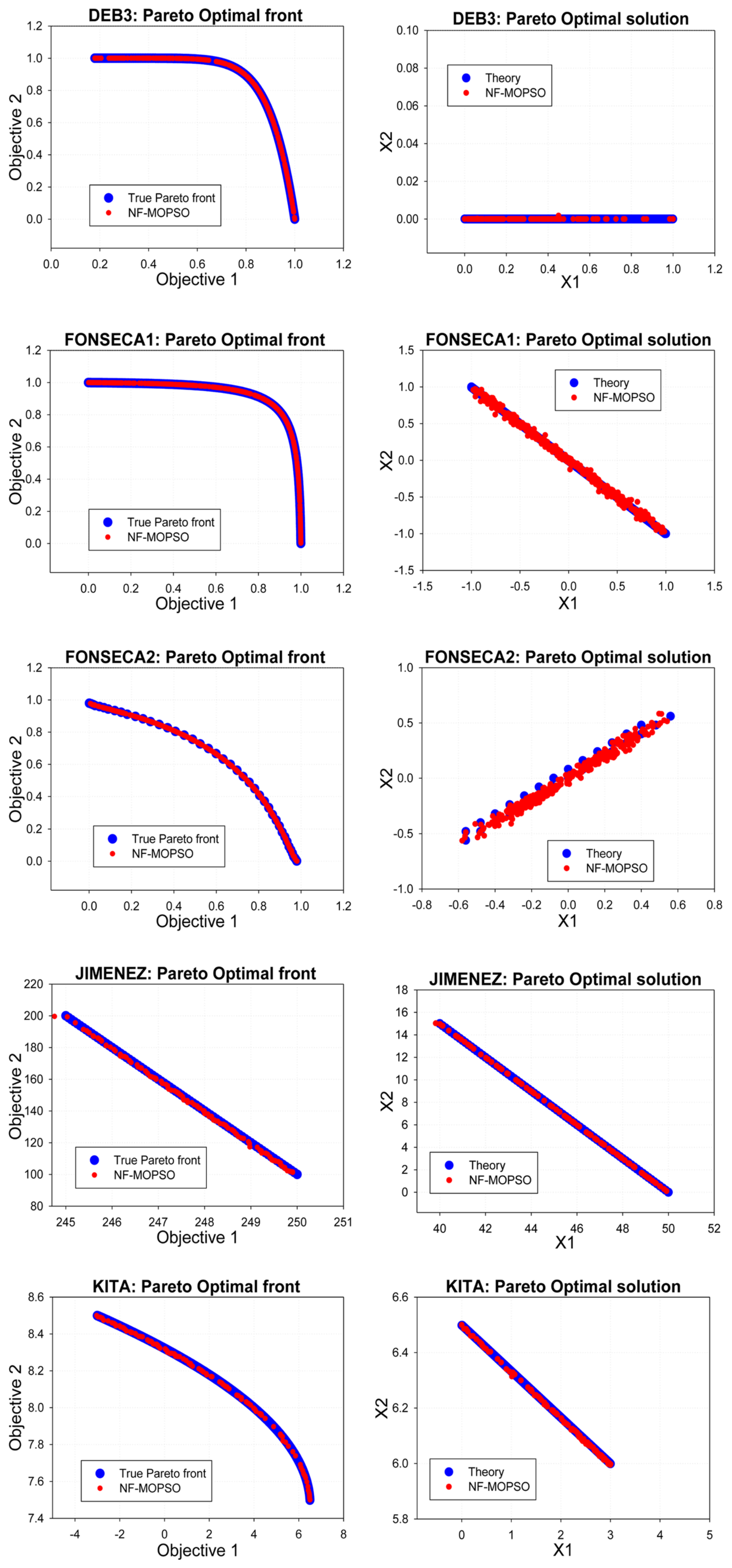


Fig. 10 Graphical performance results for two objectives test functions of NF-MOPSO vs Theory

Fig. 11 Graphical performance results for two objectives test functions of NF-MOPSO vs Theory

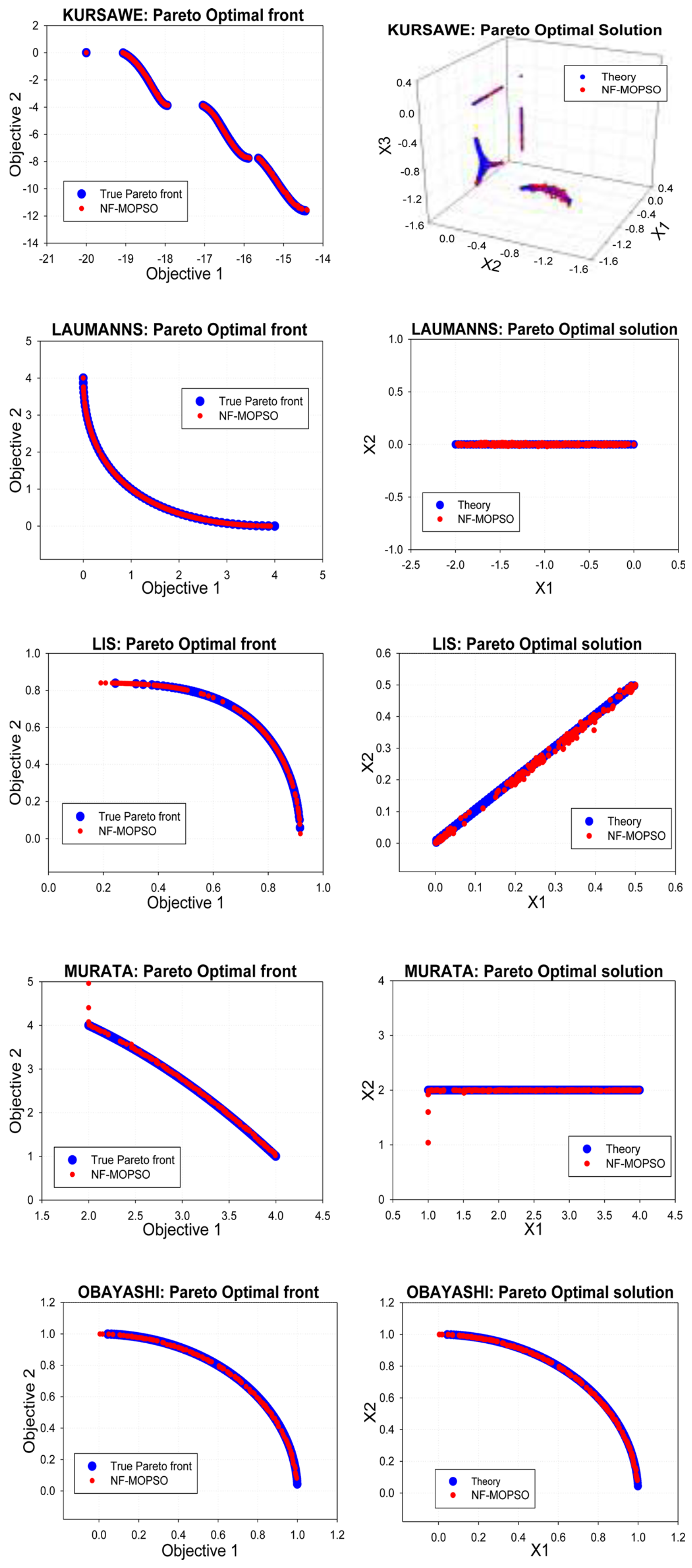


Fig. 12 Graphical performance results for two objectives test functions of NF-MOPSO vs Theory

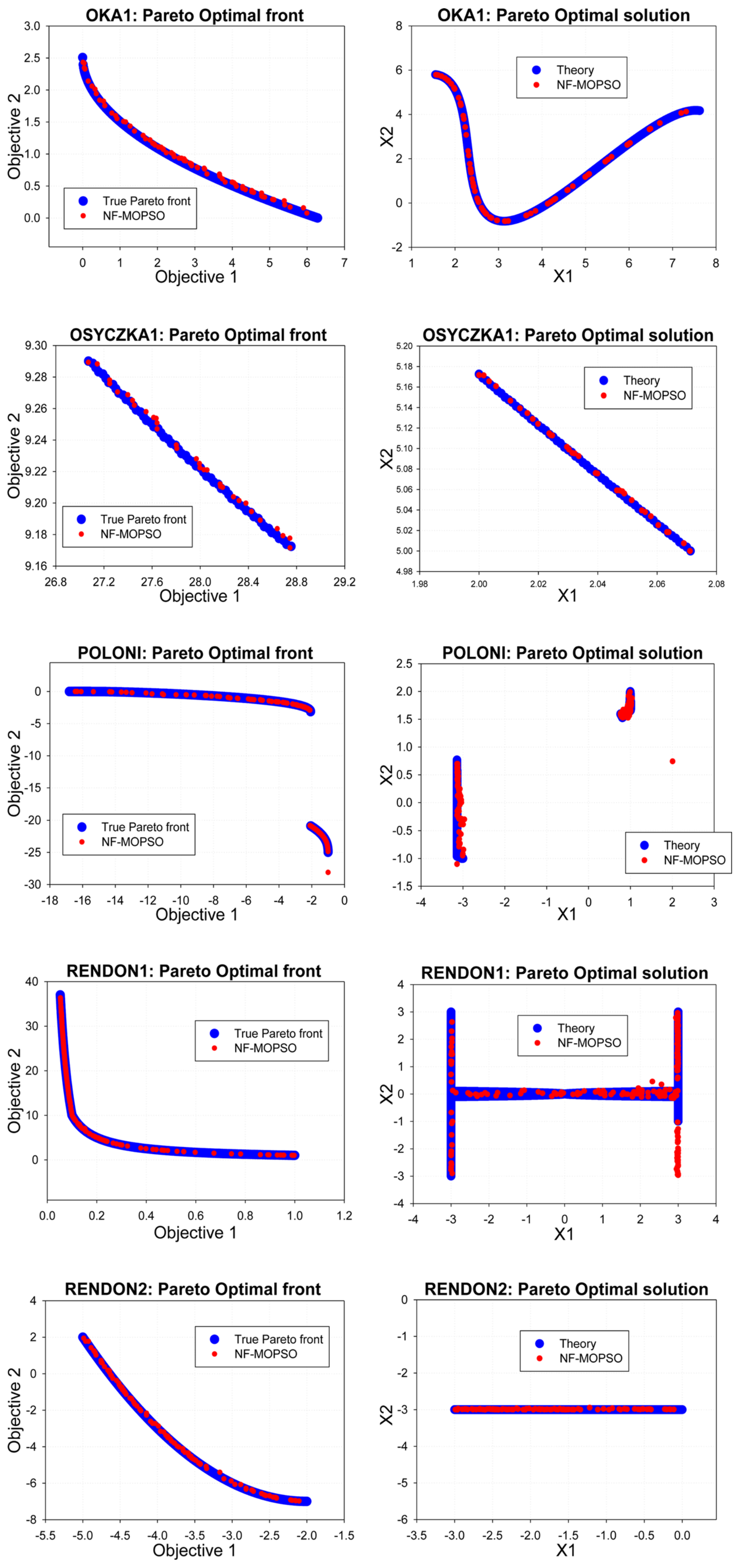
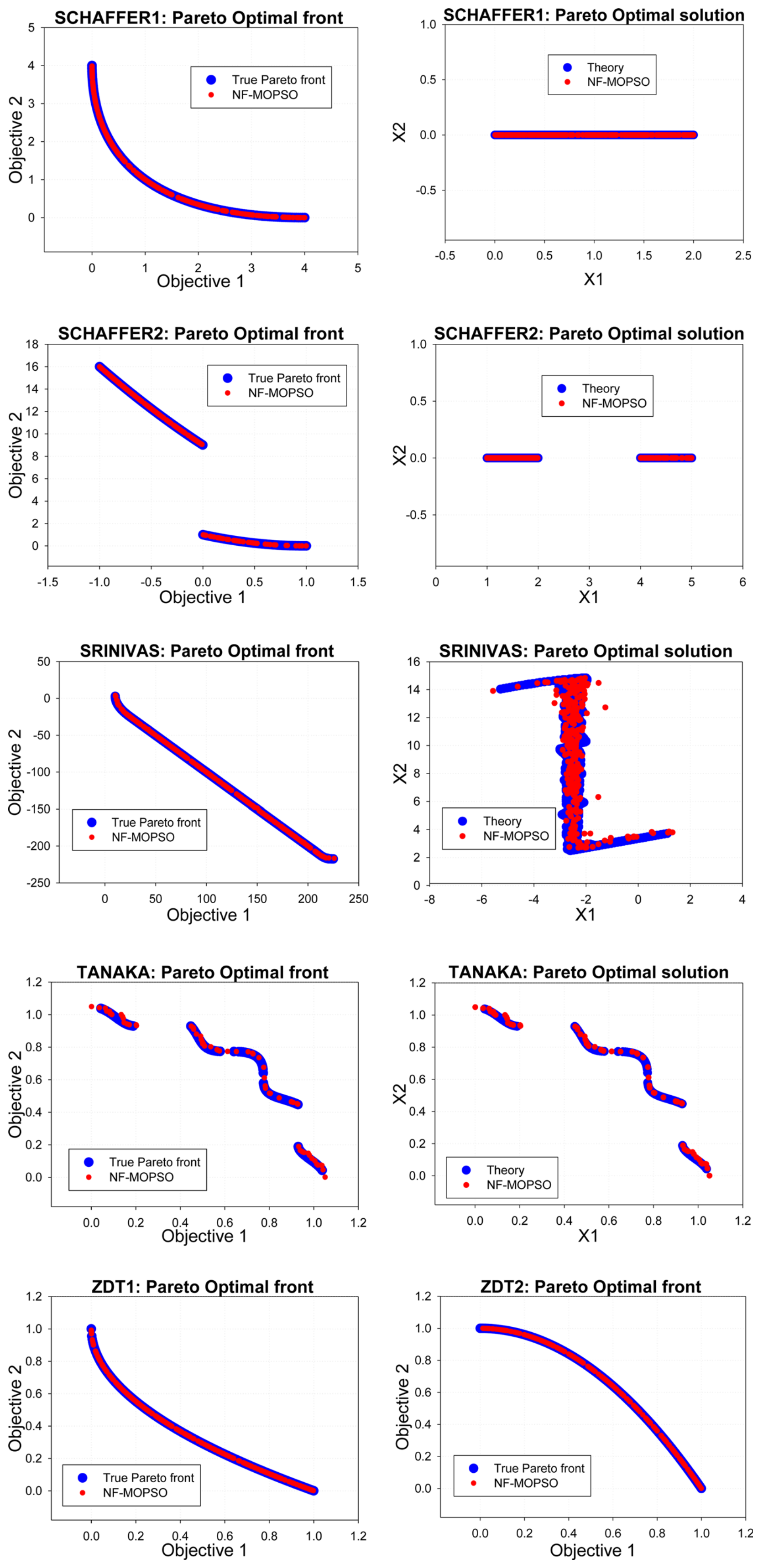


Fig. 13 Graphical performance results for two objectives test functions of NF-MOPSO vs Theory



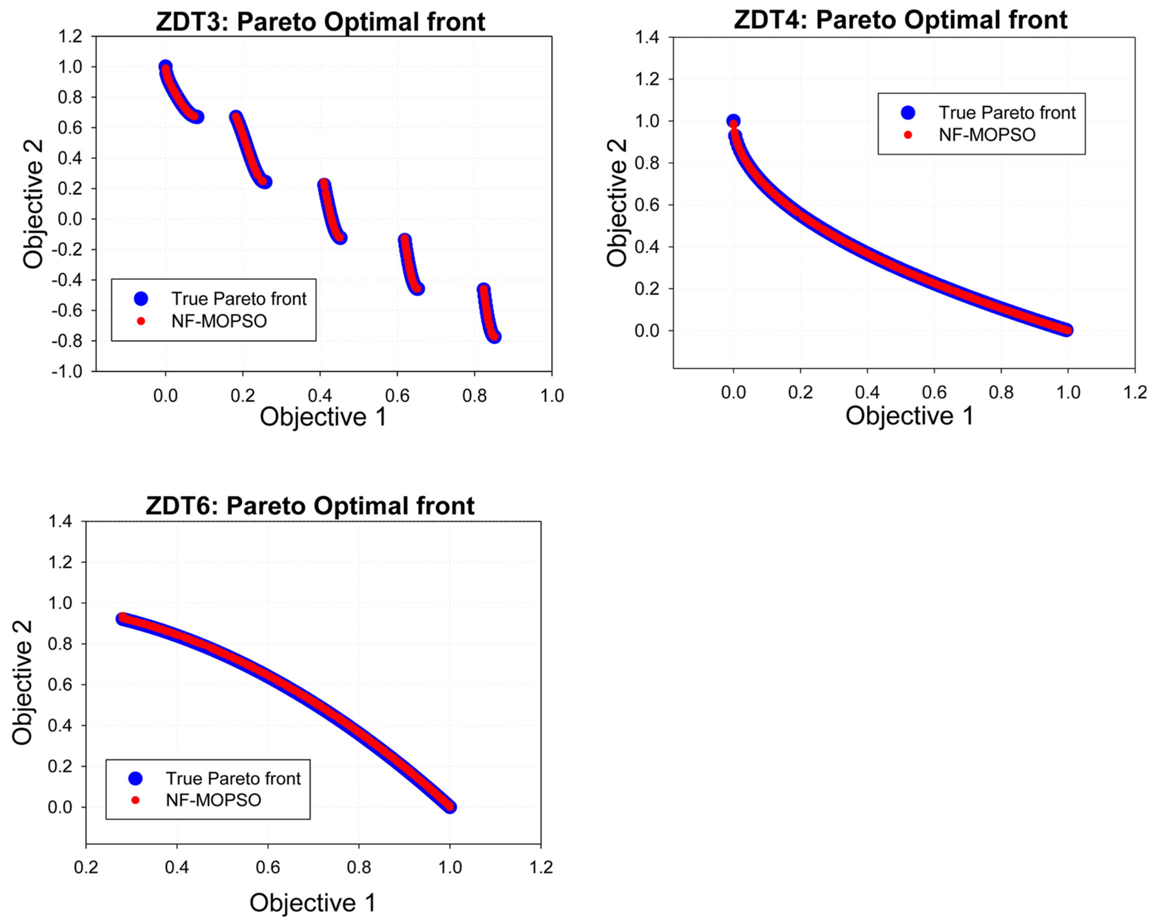


Fig. 14 Graphical performance results for two objectives test functions of NF-MOPSO vs Theory

Appendix 3: Performance results of three objectives test functions

See Figs. 15, 16.

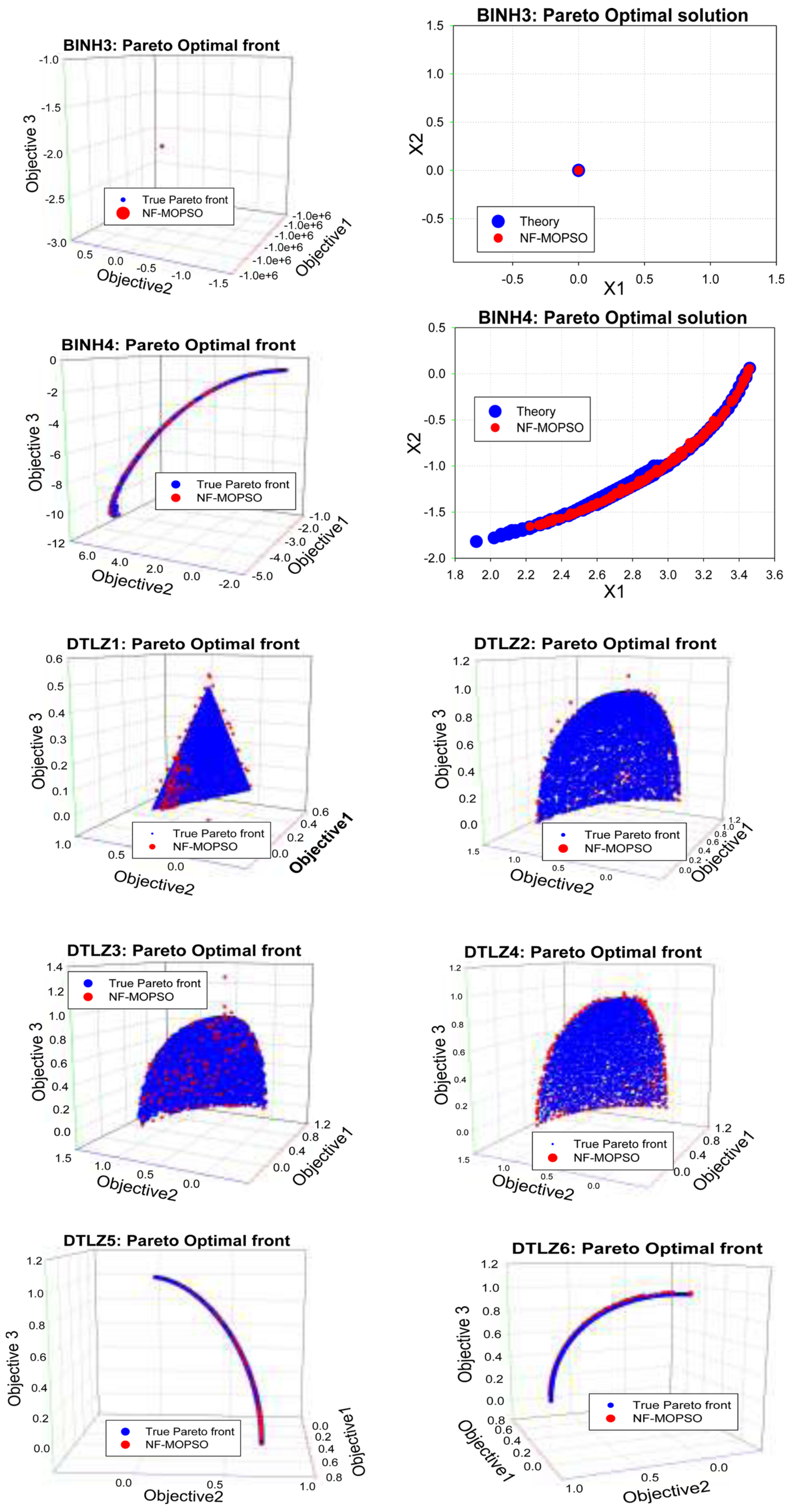


Fig. 15 Graphical performance results for three objectives test functions of NF-MOPSO vs Theory

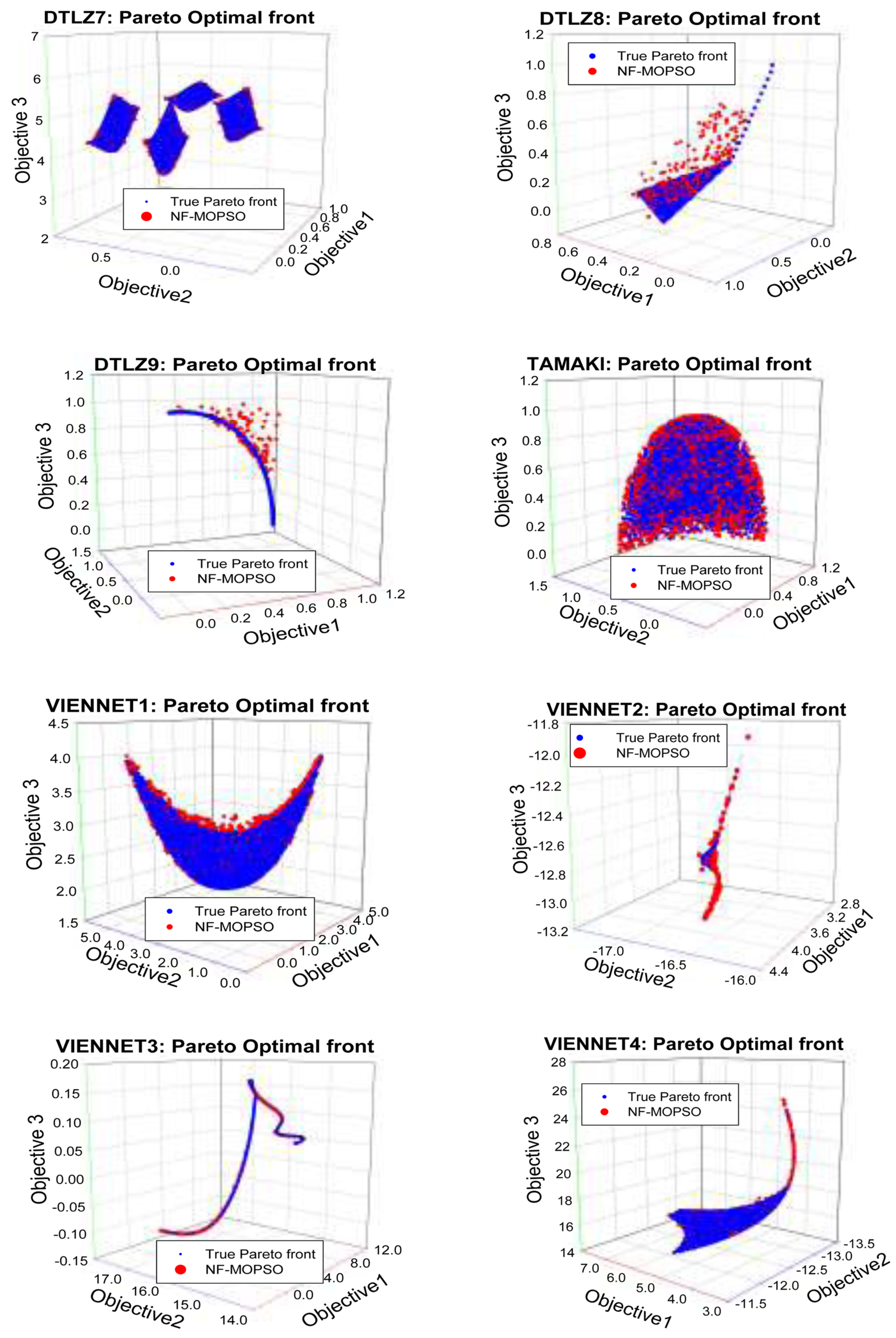


Fig. 16 Graphical performance results for three objectives test functions of NF-MOPSO vs Theory

Appendix 4: Flowchart of collimator optimization

See Fig. 17.

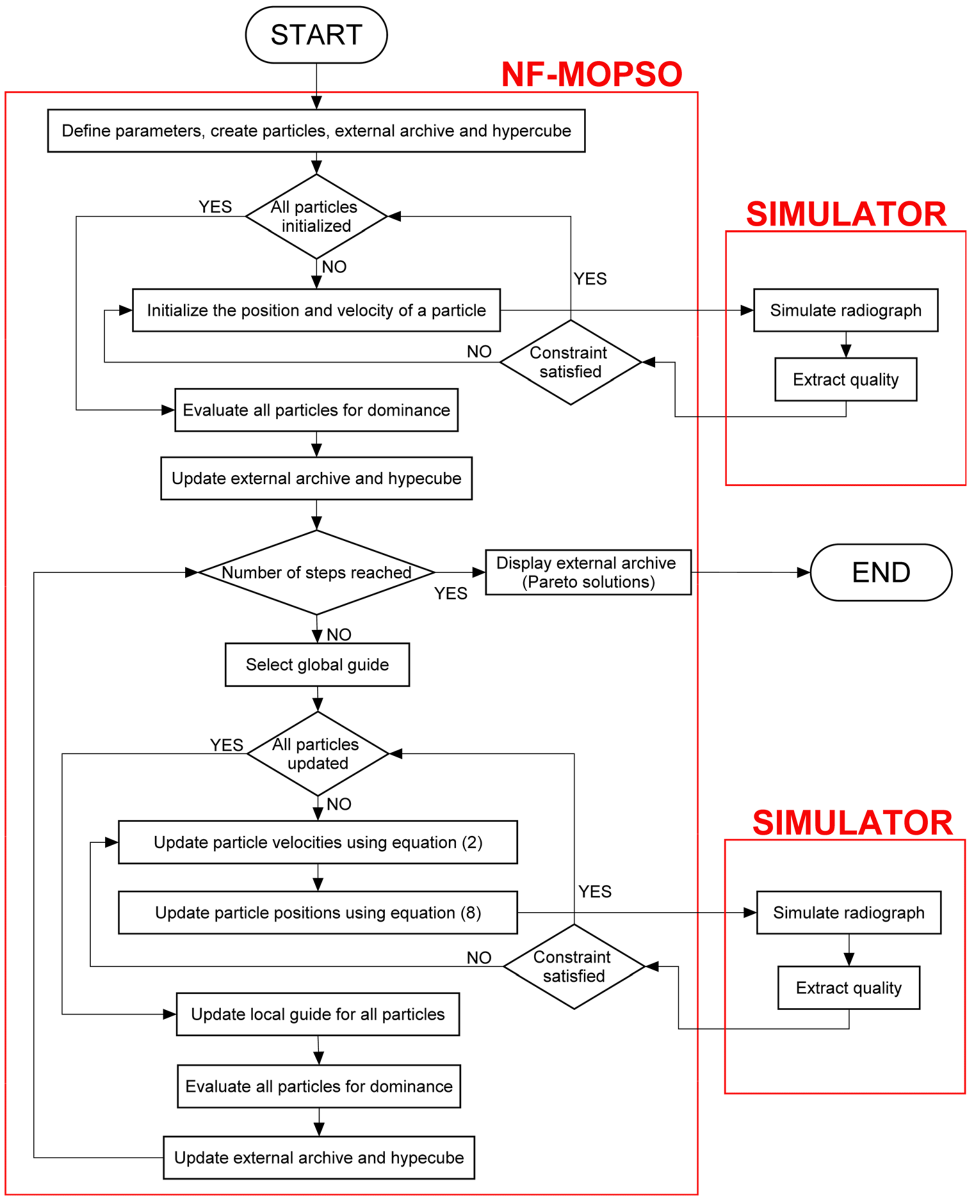

START
NF-MOPSO
Define parameters, create particles, external archive and hypercube
YES
All particles initialized
NO
YES
Initialize the position and velocity of a particle
NO
Constraint satisfied
Evaluate all particles for dominance
Update external archive and hypecube
SIMULATOR
Simulate radiograph
Extract quality
Number of steps reached
YES
Display external archive (Pareto solutions)
END
NO
Select global guide
YES
All particles updated
NO
Update particle velocities using equation (2)
Update particle positions using equation (8)
YES
NO
Constraint satisfied
Update local guide for all particles
Evaluate all particles for dominance
Update external archive and hypecube
SIMULATOR
Simulate radiograph
Extract quality

Fig. 17 Flowchart of collimator optimization

References

1. Osyczka A (1985) Multicriteria optimization for engineering design. In: Gero J (ed) Design optimization. Academic Press, London, UK, pp 193–227. https://doi.org/10.1016/b978-0-12-280910-1.50012-x
2. Kumar V, Minz S (2014) Multi-objective particle swarm optimization: an Introduction. Smart Comput Rev 4(5):335–353
3. Deb K (2001) Multi-objective optimization. Multi-objective optimization using evolutionary algorithms. Wiley, West Sussex, pp 13–45
4. Lalwani S, Singhal S, Kumar R, Gupta N (2013) A comprehensive survey: applications of multi-objective particle swarm

optimization (MOPSO) algorithm. Trans Combin 2(1):39–101. https://doi.org/10.22108/TOC.2013.2834
5. Kulkarni MNK, Patekar MS, Bhoskar MT, Kulkarni MO, Kakandikar GM, Nandedkar VM (2015) Particle swarm optimization applications to mechanical engineering—a review. Mater Today Proc 2(4–5):2631–2639. https://doi.org/10.1016/j.matpr.2015.07.223
6. Zhang Y, Wang S, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. Math Probl Eng 2015:1–38. https://doi.org/10.1155/2015/931256
7. Vandenbergh F, Engelbrecht A (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176(8):937–971. https://doi.org/10.1016/j.ins.2005.02.003
8. Zhang C, Sun J (2009) An alternate two phases particle swarm optimization algorithm for flow shop scheduling problem. Expert Syst Appl 36(3):5162–5167. https://doi.org/10.1016/j.eswa.2008.06.036
9. NAKISA (2014) A survey: particle swarm optimization based algorithms to solve premature convergence problem. J Comput Sci 10(9):1758–1765. https://doi.org/10.3844/jcssp.2014.1758.1765
10. Rezaee Jordehi A, Jasni J (2013) Parameter selection in particle swarm optimisation: a survey. J Exp Theor Artif Intell 25(4):527–542. https://doi.org/10.1080/0952813x.2013.782348
11. Coello Coello CA, Reyes-Sierra M (2006) Multi-objective particle swarm optimizers: a survey of the state-of-the-art. Int J Comput Intell Res 2(3):287–308. https://doi.org/10.5019/j.ijcir.2006.68
12. Atyabi A, Samadzadegan S (2011) Particle swarm optimization: a survey. In: Walters LP (ed) Applications of swarm intelligence. Nova Science Publishers, New York, UK, pp 167–179
13. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. IEEE Trans Evol Comput 8(3):256–279. https://doi.org/10.1109/tevc.2004.826067
14. Banks A, Vincent J, Anyakoha C (2007) A review of particle swarm optimization. Part I: background and development. Nat Comput 6(4):467–484. https://doi.org/10.1007/s11047-007-9049-5
15. Pedersen MEH, Chipperfield AJ (2010) Simplifying particle swarm optimization. Appl Soft Comput 10(2):618–628. https://doi.org/10.1016/j.asoc.2009.08.029
16. Cheng T, Chen M, Fleming PJ, Yang Z, Gan S (2017) A novel hybrid teaching learning based multi-objective particle swarm optimization. Neurocomputing 222:11–25. https://doi.org/10.1016/j.neucom.2016.10.001
17. Cheng S, Zhan H, Shu Z (2016) An innovative hybrid multi-objective particle swarm optimization with or without constraints handling. Appl Soft Comput 47:370–388. https://doi.org/10.1016/j.asoc.2016.06.012
18. Dai C, Wang Y, Ye M (2015) A new multi-objective particle swarm optimization algorithm based on decomposition. Inf Sci 325:541–557. https://doi.org/10.1016/j.ins.2015.07.018
19. Lin Q, Li J, Du Z, Chen J, Ming Z (2015) A novel multi-objective particle swarm optimization with multiple search strategies. Eur J Oper Res 247(3):732–744. https://doi.org/10.1016/j.ejor.2015.06.071
20. Zhu Q, Lin Q, Chen W, Wong KC, Coello Coello CA, Li J, Chen J, Zhang J (2017) An external archive-guided multiobjective particle swarm optimization algorithm. IEEE Trans Cybern 47(9):2794–2808. https://doi.org/10.1109/TCYB.2017.2710133
21. Fan J (2010) An improving multi-objective particle swarm optimization. Web Inf Syst Min Sanya. https://doi.org/10.1007/978-3-642-16515-3_1
22. Beheshti Z, Shamsuddin SM (2015) Non-parametric particle swarm optimization for global optimization. Appl Soft Comput 28:345–359. https://doi.org/10.1016/j.asoc.2014.12.015
23. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: The IEEE international joint conference on neural networks, 1995. IEEE, New Jersey, pp 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
24. Abraham A, Guo H, Liu H (2006) Swarm intelligence: foundations, perspectives and applications. Stud Comput Intell (SCI) 26:3–25. https://doi.org/10.1007/978-3-540-33869-7_1
25. Engelbrecht AP (2007) Computational swarm intelligence. Computational intelligence: an introduction. Wiley, New Jersey, pp 285–411
26. Bergh FVD, Engelbrecht AP (2001) Effects of swarm size on cooperative particle swarm optimizers. In: Proceedings of GECCO-2001, San Francisco, pp 892–899
27. Yassin IM, Taib MN, Adnan R, Salleh MKM, Hamzah MK (2012) Effect of swarm size parameter on binary particle swarm optimization-based NARX structure selection. In: IEEE symposium on industrial electronics and applications, Bandung, 2012. IEEE, pp 219–223. https://doi.org/10.1109/ISIEA.2012.6496632
28. Carlisle A, Dozier G (2001) An off-the-shelf PSO. In: Proceedings of the workshop on particle swarm optimization, Indianapolis, pp 1–6
29. Lin Y-T, Huang Y-M, Cheng S-C (2010) An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. Comput Educ 55(4):1483–1493. https://doi.org/10.1016/j.compedu.2010.06.014
30. Liu Q, Wei W, Yuan H, Zhan Z-H, Li Y (2016) Topology selection for particle swarm optimization. Inf Sci 363(1):154–173. https://doi.org/10.1016/j.ins.2016.04.050
31. Higashi N, Iba H (2003) Particle swarm optimization with Gaussian mutation In: 2003 IEEE swarm intelligence symposium, Indianapolis, 2003. IEEE, pp 72–79. https://doi.org/10.1109/SIS.2003.1202250
32. Stacey A, Jancic M, Grundy I (2003) Particle swarm optimization with mutation. In: The 2003 congress on evolutionary computation, Canberra, pp 1425–1430. https://doi.org/10.1109/CEC.2003.1299838
33. Iwasaki N, Yasuda K, Ueno G (2006) Dynamic parameter tuning of particle swarm optimization. Trans Electr Electron Eng 1(4):353–363. https://doi.org/10.1002/tee.20078
34. Shi Y, Eberhart RA (1998) Modified particle swarm optimizer. In: IEEE international conference on evolutionary computation, Anchorage, 1988. IEEE, pp 69–73. https://doi.org/10.1109/ICEC.1998.699146
35. Zhang Y, Zhao Y, Fu X, Xu J (2016) A feature extraction method of the particle swarm optimization algorithm based on adaptive inertia weight and chaos optimization for Brillouin scattering spectra. Opt Commun 376:56–66. https://doi.org/10.1016/j.optcom.2016.04.049
36. Lin W-C, Yin Y, Cheng S-R, Cheng TCE, Wu C-H, Wu C-C (2017) Particle swarm optimization and opposite-based particle swarm optimization for two-agent multi-facility customer order scheduling with ready times. Appl Soft Comput 52:877–884. https://doi.org/10.1016/j.asoc.2016.09.038
37. Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: The IEEE congress on evolutionary computation, La Jolla, 2000. IEEE, pp 84–88. https://doi.org/10.1109/CEC.2000.870279
38. Cleghorn CW, Engelbrecht AP (2017) Particle swarm stability: a theoretical extension using the non-stagnate distribution assumption. Swarm Intell 12(1):1–22. https://doi.org/10.1007/s11721-017-0141-x
39. Engelbrecht AP (2007) Particle swarm optimization. Computational intelligence: an introduction. Wiley, West Sussex, pp 289–357
40. Peng J, Chen Y, Eberhart R (2000) Battery pack state of charge estimator design using computational intelligence approaches. In:

Fifteenth annual battery conference on applications and advances, Long Beach, 2000. IEEE, pp 173–177. https://doi.org/10.1109/BCAA.2000.838400

41. Cooren Y, Clerc M, Siarry P (2009) MO-TRIBES, an adaptive multiobjective particle swarm optimization algorithm. Comput Optim Appl 49(2):379–400. https://doi.org/10.1007/s10589-009-9284-z
42. Cagnina L, Esquivel S, Coello CAC (2005) A particle swarm optimizer for multi-objective optimization. J Comput Sci Technol 5(4):204–210
43. de Miranda PeBC, de Carvalho ACPLF, Soares C (2012) Combining a multi-objective optimization approach with meta-learning for SVM parameter selection. In: IEEE international conference on systems, man, and cybernetics (SMC), Seoul, South Korea, 2012. IEEE, pp 2909–2914. https://doi.org/10.1109/ICSMC.2012.6378235
44. Dupont G, Adam S, Lecourtier Y, Grilheres B (2008) Multi objective particle swarm optimization using enhanced dominance and guide selection. Int J Comput Intell Res 4(2):145–158. https://doi.org/10.5019/j.ijcir.2008.134
45. Fan Z, Wang T, Cheng Z, Li G, Gu F (2017) An improved multiobjective particle swarm optimization algorithm using minimum distance of point to line. Shock Vib 2017:1–16. https://doi.org/10.1155/2017/8204867
46. López J, Lanzarini L, De Giusti A (2010) VarMOPSO: multi-objective particle swarm optimization with variable population size. In: Kuri-Morales (ed) Advances in artificial intelligence—IBERAMIA 2010, vol 6433 (Lecture notes in computer science). Springer, Berlin, pp 60–69. https://doi.org/10.1007/978-3-642-16952-6_7
47. Pellegrini R, Serani A, Leotardi C, Iemma U, Campana EF, Diez M (2017) Formulation and parameter selection of multi-objective deterministic particle swarm for simulation-based optimization. Appl Soft Comput 58:714–731. https://doi.org/10.1016/j.asoc.2017.05.013
48. Santana RA, Pontes MR, Bastos-Filho CJA (2009) A multiple objective particle swarm optimization approach using crowding distance and roulette wheel. In: Ninth international conference on intelligent systems design and applications, Pisa, Italy, 2009. IEEE, pp 237–242. https://doi.org/10.1109/ISDA.2009.73
49. Santana-Quintero LV, Ramírez-Santiago N, Coello Coello CA (2008) Towards a more efficient multi-objective particle swarm optimizer. In: Bui LT (ed) Multi-objective optimization in computational intelligence, 1st edn. IGI Global, London, pp 76–105. https://doi.org/10.4018/978-1-59904-498-9.ch004
50. Sun Y, Gao Y, Shi X (2019) Chaotic multi-objective particle swarm optimization algorithm incorporating clone immunity. Mathematics 7(2):1–16. https://doi.org/10.3390/math7020146
51. Toscano-Pulido G, Coello CAC, Santana-Quintero LV (2007) EMOPSO: a multi-objective particle swarm optimizer with emphasis on efficiency. In: 4th international conference on evolutionary multi-criterion optimization, Matshushima (Lecture notes in computer science), 2007. Springer, pp 272–285. https://doi.org/10.1007/978-3-540-70928-2_23
52. Tripathi PK (2007) Adaptive mufti-objective particle swarm optimization algorithm. In: IEEE congress on evolutionary computation, Singapore, 2007. IEEE, pp 2281–2288. https://doi.org/10.1109/CEC.2007.4424755
53. Parsopoulos KE, Vrahatis MN (2008) Multi-objective particles swarm optimization approaches. In: Bui LT, Alam S (eds) Multi-objective optimization in computational intelligence: theory and practice. IGI Global, Hershey, pp 20–42. https://doi.org/10.13140/2.1.5189.4721
54. Tripathi PK, Bandyopadhyay S, Pal SK (2007) Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Inf Sci 177(22):5033–5049. https://doi.org/10.1016/j.ins.2007.06.018
55. Wang H, Yen GG (2015) Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. IEEE Trans Evol Comput 19(1):1–18. https://doi.org/10.1109/tevc.2013.2296151
56. Han H, Lu W, Qiao J (2017) An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. IEEE Trans Cybern 47(9):2754–2767. https://doi.org/10.1109/TCYB.2017.2692385
57. Coello CAC, Lamont GB, Veldhuizen DAV (2007) Basic concepts. Evolutionary algorithms for solving multi-objective problems. Springer, Berlin, pp 1–57
58. Jordehi AR (2015) A review on constraint handling strategies in particle swarm optimisation. Neural Comput Appl 26(2015):1265–1275. https://doi.org/10.1007/s00521-014-1808-5
59. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5):477–506. https://doi.org/10.1109/TEVC.2005.861417
60. Coello CAC, Lamont GB, Veldhuizen DAV (2007) MOEA testing and analysis. In: Goldberg DE, Koza JR (eds) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, New York, pp 233–276. https://doi.org/10.1007/978-0-387-36797-2
61. Deb K (2001) Salient issues of multi-objective evolutionary algorithms. Multi-objective optimization using evolutionary algorithm. Wiley, West Sussex, pp 301–424
62. Coello CAC, Cortes NC (2005) Solving multiobjective optimization problems using an artificial immune system. Genet Program Evol Mach 6(2):163–190. https://doi.org/10.1007/s10710-005-6164-x
63. Van Veldhuizen DA, Lamont GB (1999) Multi objective evolutionary algorithm test suites. In: ACM symposium on applied computing, San Antonio, 1999. ACM, pp 351–357. https://doi.org/10.1145/298151.298382
64. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195. https://doi.org/10.1162/106365600568202
65. Deb K, Thiele L, Laumanns M, Zitzler E (2002) Scalable test problems for evolutionary multiobjective optimization. In: Proceedings of the 2002 congress on evolutionary computation, Honolulu, 2002. IEEE, pp 825–830. https://doi.org/10.1109/CEC.2002.1007032
66. Li H, Zhang Q (2009) Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Trans Evol Comput 13(2):284–302. https://doi.org/10.1109/TEVC.2008.925798
67. Garcia S, Trinh CT (2019) Comparison of multi-objective evolutionary algorithms to solve the modular cell design problem for novel biocatalysis. Processes 7(6):1–13. https://doi.org/10.3390/pr7060361
68. Van Veldhuizen DA, Lamont GB (1998) Evolutionary computation and convergence to a Pareto front. In: Late breaking papers at the genetic programming, Stanford, 1998. Stanford University Bookstore, pp 221–228
69. Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VGd (2003) Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evol Comput 7(2):117–132. https://doi.org/10.1109/TEVC.2003.810758
70. Schott JR (1995) MCGA performance parameters. In: Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, Massachusetts, USA, pp 135–138

71. Arnold K, Gosling J, Holmes D (2000) The java programming language. Addison-Wesley Longman Publishing Co, Boston
72. Martínez SZ, Coello CAC (2011) A multi-objective particle swarm optimizer based on decomposition. In: 13th annual conference on genetic and evolutionary computation, Dublin, 2011. pp 69–76. https://doi.org/10.1145/2001576.2001587
73. Sierra MR, Coello CAC (2005) Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance. In: Coello CAC, Hernández Aguirre A, Zitzler E (eds) Evolutionary multi-criterion optimization, Marzo de. Springer, pp 505–519. https://doi.org/10.1007/978-3-540-31880-4_35
74. Durillo JJ, García-Nieto J, Nebro AJ, Coello CAC, Luna F, Alba E (2009) Multi-objective particle swarm optimizers: an experimental comparison. In: The 5th international conference on evolutionary multi-criterion optimization, Nantes, 2009. Springer, pp 495–509. https://doi.org/10.1007/978-3-642-01020-0_39
75. Pulido GT, Coello CAC (2004) A constraint-handling mechanism for particle swarm optimization. In: The 2004 congress on evolutionary computation, Portland. IEEE, pp 1396–1403. https://doi.org/10.1109/CEC.2004.1331060
76. Domínguez JSH, Pulido GT (2011) A comparison on the search of particle swarm optimization and differential evolution on multi-objective optimization. In: IEEE congress of evolutionary computation, Ritz-Carlton, New Orleans, LA, USA, 2011. IEEE. https://doi.org/10.1109/CEC.2011.5949858
77. Godinez AC, Espinosa LEM, Montes EM (2010) An experimental comparison of multiobjective algorithms: NSGA-II and OMOPSO. In: IEEE electronics, robotics and automotive mechanics conference, Morelos, 2010. IEEE, pp 28–33. https://doi.org/10.1109/CERMA.2010.13
78. Mishra BSP, Dehuri S, Cho S-B (2015) Swarm intelligence in multiple and many objectives optimization: a survey and topical study on EEG signal analysis. Stud Comput Intell 592:27–73. https://doi.org/10.1007/978-3-662-46309-3_2
79. de Carvalho AB, Pozo A (2012) Measuring the convergence and diversity of CDAS multi-objective particle swarm optimization algorithms: a study of many-objective problems. Neurocomputing 75(1):43–51. https://doi.org/10.1016/j.neucom.2011.03.053
80. Wang J, Wang D (2008) Particle swarm optimization with a leader and followers. Prog Natl Sci 18(11):1437–1443. https://doi.org/10.1016/j.pnsc.2008.03.029
81. Geetika SJ (2015) Hybridization of particle swarm optimization—a survey. Int J Sci Res 4(1):2417–2420
82. Brits R, Engelbrecht AP, Fvd B (2007) Locating multiple optima using particle swarm optimization. Appl Math Comput 189(2):1859–1883. https://doi.org/10.1016/j.amc.2006.12.066
83. Barton JP (1976) Neutron radiography—an overview. In: Practical application of neutron radiography and gaging. American Society for Testing and Materials STP 586, Philadelphia, pp 5–19
84. Domanus JC, Greim L (1992) Collimators. Practical neutron radiography. Kluwer Academic Publishers, Brussels, pp 96–126
85. Domanus JC, Markgref JFW (1987) Introduction. In: Markgref JFW (ed) Collimators for thermal neutron radiography an overview, 1st edn. Springer, Netherlands, p 5
86. Kobayashi H (1999) Design and basic character of neutron collimator on radiography. In: The sixth Asian symposium on research reactors, Mito, 1999, vol 9. Japan Atomic Energy Research Institute, pp 367–372
87. Amalia AF, Budhi W, Prabowo UN, Suparta GB (2018) The image quality analysis of neutron digital radiography through the variation of multiple image capturing. In: International conference on science and applied science, Surakarta, 2018. AIP Conference Proceedings, pp 1–8. https://doi.org/10.1063/1.5054544
88. Guo Z, Zou Y, Lu Y, Yan X, Peng S, Zhu K, Tang G, Mo D, Chen J (2012) Neutron radiography with compact accelerator at Peking University: problems and solutions. Phys Proc 26:70–78. https://doi.org/10.1016/j.phpro.2012.03.011
89. Jamro R, Kardjilov N, HairieRabir M, Zain MRM, Mohamed AA, Ali NM, Idris F, Ahmad MHARM, Yazid K, Yazid H, Azman A, Mamat MR (2016) Monte Carlo simulation for designing collimator of the neutron radiography facility in Malaysia. In: 8th international topical meeting on neutron radiography, Beijing, vol 361–368. Physics Procedia. https://doi.org/10.1016/j.phpro.2017.06.049
90. Mishra KK, Hawari AI, Gillette VH (2006) Design and performance of a thermal neutron imaging facility at the North Carolina State University PULSTAR reactor. IEEE Trans Nucl Sci 53(6):3904–3911. https://doi.org/10.1109/tns.2006.884323
91. da Silva AX, Crispim VR (2001) Moderator–collimator-shielding design for neutron radiography systems using 252Cf. Appl Radiat Isot 54(2):217–225. https://doi.org/10.1016/s0969-8043(00)00291-8
92. Jafari H, Feghhi SAH (2012) Design and simulation of neutron radiography system based on 241Am–Be source. Radiat Phys Chem 81(5):506–511. https://doi.org/10.1016/j.radphyschem.2011.12.027
93. de Beer FC (2005) Characteristics of the neutron/X-ray tomography system at the SANRAD facility in South Africa. Nucl Instrum Methods Phys Res A 542:1–8. https://doi.org/10.1016/j.nima.2005.01.003
94. Nshimirimana R, Abraham A, Nothnagel G, Engelbrecht A (2020) X-Ray and neutron radiography system optimization by means of a multiobjective approach and a simplified ray-tracing method. Nucl Technol. https://doi.org/10.1080/00295450.2020.1740562
95. Grünauer F (2009) Monte Carlo simulations for the SAFARI reactor and its instruments: neutron radiography facility. NECSA, Pelindaba