Nuclear Instruments and Methods in Physics Research A 835 (2016) 157–162

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Population-based metaheuristic optimization in neutron optics and shielding design

D.D. DiJulio [a,b,*], H. Björgvinsdóttir [a,c], C. Zendler [a], P.M. Bentley [a,c]

[a] European Spallation Source ERIC, P.O. Box 176, SE-221 00 Lund, Sweden
[b] Division of Nuclear Physics, Lund University, SE-221 00 Lund, Sweden
[c] Department of Physics and Astronomy, Uppsala University, SE-751 20 Uppsala, Sweden

ARTICLE INFO

Article history:
Received 2 May 2016
Received in revised form
19 July 2016
Accepted 17 August 2016
Available online 18 August 2016

Keywords:
Metaheuristic
Optimization
Neutron optics
Shielding
Monte-Carlo

ABSTRACT

Population-based metaheuristic algorithms are powerful tools in the design of neutron scattering instruments and the use of these types of algorithms for this purpose is becoming more and more commonplace. Today there exists a wide range of algorithms to choose from when designing an instrument and it is not always initially clear which may provide the best performance. Furthermore, due to the nature of these types of algorithms, the final solution found for a specific design scenario cannot always be guaranteed to be the global optimum. Therefore, to explore the potential benefits and differences between the varieties of these algorithms available, when applied to such design scenarios, we have carried out a detailed study of some commonly used algorithms. For this purpose, we have developed a new general optimization software package which combines a number of common metaheuristic algorithms within a single user interface and is designed specifically with neutronic calculations in mind. The algorithms included in the software are implementations of Particle-Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), and a Genetic Algorithm (GA). The software has been used to optimize the design of several problems in neutron optics and shielding, coupled with Monte-Carlo simulations, in order to evaluate the performance of the various algorithms. Generally, the performance of the algorithms depended on the specific scenarios, however it was found that DE provided the best average solutions in all scenarios investigated in this work.

1. Introduction

The use of population-based metaheuristic algorithms in the design of neutron scattering instruments is becoming more and more commonplace. These types of algorithms are well suited for the task, due in part to the large number of parameters involved in a typical design scenario and the resulting noisy parameter spaces. These two points make it often difficult to apply traditional optimization algorithms for the design of such systems [1]. Furthermore, the optimization of such an instrument is frequently a tedious and complex procedure. Automated algorithms, which can efficiently search the parameter space with little user input, provide a great advantage in this process.

In a number of previous studies, population-based algorithms have been successfully applied to neutron optics and shielding design. For example, a Genetic Algorithm (GA) [2] was used for the design of specific instruments at the Hahn-Meitner Institute, Berlin and the Institut Laue-Langevin, France [3,4]. A GA was also used to optimize the composition of shielding material for mixed neutron and photon fields [5]. In some additional studies, Particle-Swarm Optimization (PSO) [6] was applied to the design of an entire neutron guide hall [7] and to multi-channel focusing guides [8], where it showed improved performance over GA, and it was also used in the design of elliptic focusing guides [9]. Artificial Bee Colony (ABC) [10] and Differential Evolution (DE) [11] were also used in the design of multi-channel focusing guides for extreme sample environments [12], where it was found that these algorithms demonstrated improved performance compared to PSO for the investigated design scenarios. The benefits of population-based algorithms were also noted in Ref. [13], where a number of algorithms were tested on a set of multi-dimensional objective functions where the global minima were known.

While metaheuristic algorithms have exhibited exceptional performance for the design of neutron instruments, it can be expected that there may be noticeable differences between the results of individual algorithms when applied to the same

* Corresponding author at: European Spallation Source ERIC, P.O. Box 176, SE-221 00 Lund, Sweden.
E-mail address: Douglas.DiJulio@esss.se (D.D. DiJulio).

http://dx.doi.org/10.1016/j.nima.2016.08.035
0168-9002/

design scenario. In practice, often only one type of algorithm is used for the design of a particular system. As the usage of metaheuristic algorithms cannot guarantee that the global optimum will be found, it can be of great benefit to repeat the optimization with a number of different algorithms and starting conditions. To explore this further, we have developed software containing a suite of algorithms combined under a single user interface and with neutron scattering applications in mind. The software is general enough that it can be easily coupled to an external simulation package. In this work, we have coupled the software to VITESS [14] for neutron optics calculations and Geant4 [15,16] for neutron shielding calculations and applied it to several different design scenarios in order to evaluate the performance of each algorithm. In the following, we first provide a description of the software developed, followed by the results of the applications of the software to the above mentioned type of calculations. Lastly, we discuss and present the conclusions of our work.

2. Description of the software

The newly developed software contains implementations of PSO, GA, ABC, and DE. The ABC and DE packages are based on the freely downloadable codes from Ref. [17,18]. The algorithms are described in detail below and an outline of the of the structure of the software is indicated in Fig. 1. The parameters to be optimized, here refereed to as ω, are specified in a parameter file which is read by the software. These parameters can be either discreet or continuous, and have varying boundaries. An individual is associated with a single set of parameters and it's corresponding figure of merit (FoM). It is also possible to define simulation parameters. The simulation parameters are not optimized themselves, but depend on one or more optimization parameters. The dependency is described by a mathematical expression, parsed and evaluated by an implementation of the Shunting-yard algorithm [19].

In the main optimization loop of all algorithms, the optimization progress of each iteration is logged. When the optimization is completed, the best FoM and corresponding parameter values are put in a result file, along with the total number of evaluations, and information regarding the optimization mode. The result file also contains information about each iteration, including the number of evaluations required, best FoM found so far, and corresponding parameter values. Optionally, a file containing the best FoM of each iteration (as opposed to overall best FoM) and corresponding parameter values can be created. Additionally, the user can choose to trace any number of individuals and save all parameter values assumed by those individuals.

In all of the implemented optimization algorithms, the optimization parameters were initialized randomly within their individual boundaries. If an updated ω was outside of its boundaries at any point of the optimization process, it was shifted to the closest boundary.

Discreet optimization parameters are treated as continuous throughout the optimization, and truncated before simulation, as suggested in Ref. [20]. After each simulation, the simulation result is used to calculate the FoM.

2.1. Particle swarm implementation

PSO is a type of machine-learning algorithm which has its origins in the swarming of social animals, such as the schooling of fish or flocking of birds. A swarm consists of a set of candidate solutions called particles (refereed to as individuals here), which are characterized by their positions and velocities in time. Each individual is aware of the best position it has seen and also the best position seen by any member of the swarm. When an individual i moves in PSO, its new position is calculated by adding its velocity, v_i, to its position, ω_i and the velocity in a certain direction, $v_{i,j}$, is calculated as

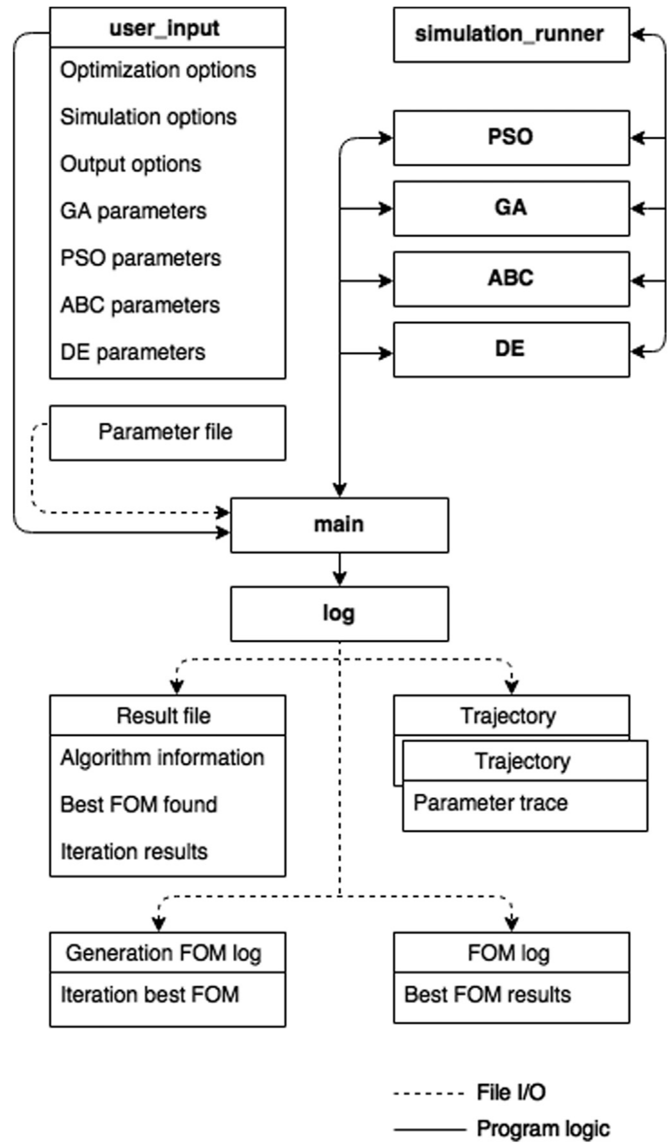

Fig. 1. Simplified visualization of software relations, focusing on input and output.

$$v_{i,j} = c_I \cdot v_{i,j} + w_L \cdot \text{rand}(0, 1) \cdot (\omega_{ibest,j} - \omega_{i,j}) + w_C \cdot \text{rand}(0, 1) \cdot (\omega_{best,j} - \omega_{i,j})$$

where $\omega_{i,j}$ is the individual's position in direction j, and $\omega_{ibest,j}$ is the position in direction j which has produced the best FoM so far for individual i. $\omega_{best,j}$ is the position in direction j that has produced the overall best FoM for any individual. The inertial constant c_I, the local search weight w_L, and the collective search weight w_C, are all specified by the user and decide how greatly an individual's movement is influenced by it's own independent movement and that of the swarm. The random values of the equation are used in order to make the movement less predictable.

2.2. Genetic algorithm implementation

GAs are evolutionary algorithms which work on a population of individuals, which are selectively bred, to improve the quality of the population. When selecting individuals for reproduction in GA, two selection methods are available. Rank selection sorts the individuals from best to worst FoM and selects parents for