We are IntechOpen,
the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter

Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems

Bruno Seixas Gomes de Almeida and Victor Coppo Leite

Abstract

This chapter will introduce the particle swarm optimization (PSO) algorithm giving an overview of it. In order to formally present the mathematical formulation of PSO algorithm, the classical version will be used, that is, the inertial version; meanwhile, PSO variants will be summarized. Besides that, hybrid methods representing a combination of heuristic and deterministic optimization methods are going to be presented as well. Before the presentation of these algorithms, the reader will be introduced to the main challenges when approaching PSO algorithm. Two study cases of diverse nature, one regarding the PSO in its classical version and another one regarding the hybrid version, are provided in this chapter showing how handful and versatile it is to work with PSO. The former case is the optimization of a mechanical structure in the nuclear fuel bundle and the last case is the optimization of the cost function of a cogeneration system using PSO in a hybrid optimization. Finally, a conclusion is presented.

Keywords: PSO algorithm, hybrid methods, nuclear fuel, cogeneration system

1. Introduction

Maximizing earns or minimizing losses has always been a concern in engineering problems. For diverse fields of knowledge, the complexity of optimization problems increases as science and technology develop. Often, examples of engineering problems that might require an optimization approach are in energy conversion and distribution, in mechanical design, in logistics, and in the reload of nuclear reactors.

To maximize or minimize a function in order to find the optimum, there are several approaches that one could perform. In spite of a wide range of optimization algorithms that could be used, there is not a main one that is considered to be the best for any case. One optimization method that is suitable for a problem might not be so for another one; it depends on several features, for example, whether the function is differentiable and its concavity (convex or concave). In order to solve a problem, one must understand different optimization methods so this person is able to select the algorithm that best fits on the features' problem.

The particle swarm optimization (PSO) algorithm, proposed by Kennedy and Eberhart [1], is a metaheuristic algorithm based on the concept of swarm intelligence capable of solving complex mathematics problems existing in engineering [2]. It is of great importance noting that dealing with PSO has some advantages

when compared with other optimization algorithms, once it has fewer parameters to adjust, and the ones that must be set are widely discussed in the literature [3].

2. Particle swarm optimization: an overview

In the early of 1990s, several studies regarding the social behavior of animal groups were developed. These studies showed that some animals belonging to a certain group, that is, birds and fishes, are able to share information among their group , and such capability confers these animals a great survival advantage [4]. Inspired by these works, Kennedy and Eberhart proposed in 1995 the PSO algorithm [1], a metaheuristic algorithm that is appropriate to optimize nonlinear continuous functions. The author derived the algorithm inspired by the concept of swarm intelligence, often seen in animal groups, such as flocks and shoals.

In order to explain how the PSO had inspired the formulation of an optimization algorithm to solve complex mathematical problems, a discussion on the behavior of a flock is presented. A swarm of birds flying over a place must find a point to land and, in this case, the definition of which point the whole swarm should land is a complex problem, since it depends on several issues, that is, maximizing the availability of food and minimizing the risk of existence of predators. In this context, one can understand the movement of the birds as a choreography; the birds synchronically move for a period until the best place to land is defined and all the flock lands at once.

In the given example, the movement of the flock only happens as described once all the swarm members are able to share information among themselves; otherwise, each animal would most likely land at a different point and at a different time. The studies regarding the social behavior of animals from the early 1990s stated before in this text pointed out that all birds of a swarm searching for a good point to land are able to know the best point until it is found by one of the swarm's members. By means of that, each member of the swarm balances its individual and its swarm knowledge experience, known as social knowledge. One may notice that the criteria to assess whether a point is good or not in this case is the survival conditions found at a possible landing point, such as those mentioned earlier in this text.

The problem to find the best point to land described features an optimization problem. The flock must identify the best point, for example, the latitude and the longitude, in order to maximize the survival conditions of its members. To do so, each bird flies searching and assessing different points using several surviving criteria at the same time. Each one of those has the advantage to know where the best location point is found until known by the whole swarm.

Kennedy and Eberhart inspired by the social behavior of birds, which grants them great surviving advantages when solving the problem of finding a safe point to land, proposed an algorithm called PSO that could mimic this behavior. The inertial version, also known as classical version, of the algorithm was proposed in 1995 [1]. Since then, other versions have been proposed as variations of the classical formulation, that is, the linear-decreasing inertia weight [5], the constriction factor weight [6], the dynamic inertia and maximum velocity reduction, also in Ref. [6], besides hybrid models [7] or even quantum inspired approach optimization techniques that can be applied to PSO [8]. This chapter will only present the inertial model of PSO, as it is the state-of-the-art algorithm, and to understand better the derivations of PSO, one should firstly understand its classical version.

The goal of an optimization problem is to determine a variable represented by a vector $X = [x_1 x_2 x_3 \dots x_n]$ that minimizes or maximizes depending on the proposed optimization formulation of the function $f(X)$. The variable vector X is known as

position vector; this vector represents a variable model and it is n dimensions vector, where n represents the number of variables that may be determined in a problem, that is, the latitude and the longitude in the problem of determining a point to land by a flock. On the other hand, the function $f(X)$ is called fitness function or objective function, which is a function that may assess how good or bad a position X is, that is, how good a certain landing point a bird thinks it is after this animal finds it, and such evaluation in this case is performed through several survival criteria.

Considering a swarm with P particles, there is a position vector $X_i^t = (x_{i1}x_{i2}x_{i3}\ldots x_{in})^T$ and a velocity vector $V_i^t = (v_{i1}v_{i2}v_{i3}\ldots v_{in})^T$ at a t iteration for each one of the i particle that composes it. These vectors are updated through the dimension j according to the following equations:

$$V_{ij}^{t+1} = wV_{ij}^t + c_1 r_1^t \left(pbest_{ij} - X_{ij}^t\right) + c_2 r_2^t \left(gbest_j - X_{ij}^t\right) \qquad (1)$$

and

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \qquad (2)$$

where $i = 1,2,\ldots,P$ and $j = 1,2,\ldots,n$.

Eq. (1) denotes that there are three different contributions to a particle's movement in an iteration, so there are three terms in it that are going to be further discussed. Meanwhile, Eq. (2) updates the particle's positions. The parameter w is the inertia weight constant, and for the classical PSO version, it is a positive constant value. This parameter is important for balancing the global search, also known as exploration (when higher values are set), and local search, known as exploitation (when lower values are set). In terms of this parameter, one may notice that it is one of the main differences between classical version of PSO and other versions derived from it.

Velocity update equation's first term is a product between parameter w and particle's previous velocity, which is the reason it denotes a particles' previous motion into the current one. Hence, for example, if $w = 1$, the particle's motion is fully influenced by its previous motion, so the particle may keep going in the same direction. On the other hand, if $0 \leq w < 1$, such influence is reduced, which means that a particle rather goes to other regions in the search domain. Therefore, as the inertia weight parameter is reduced, the swarm may explore more areas in the searching domain, which means that the chances of finding a global optimum may increase. However, there is a price when using lower w values, which is the simulations turn out to be more time consuming [1].

The individual cognition term, which is the second term of Eq. (1), is calculated by means of the difference between the particle's own best position, for example, $pbest_{ij}$, and its current position X_{ij}^t. One may notice that the idea behind this term is that as the particle gets more distant from the $pbest_{ij}$ position, the difference $\left(pbest_{ij} - X_{ij}^t\right)$ must increase; therefore, this term increases, attracting the particle to its best own position. The parameter c_1 existing as a product in this term is a positive constant and it is an individual-cognition parameter, and it weighs the importance of particle's own previous experiences. The other parameter that composes the product of second term is r_1, and this is a random value parameter with [0, 1] range. This random parameter plays an important role, as it avoids premature convergences, increasing the most likely global optima [1].

Finally, the third term is the social learning one. Because of it, all particles in the swarm are able to share the information of the best point achieved regardless of which particle had found it, for example, $gbest_j$. Its format is just like the second term, the one regarding the individual learning. Thus, the difference $\left(gbest_j - X_{ij}^t\right)$ acts as an attraction for the particles to the best point until found at some t iteration. Similarly, c_2 is a social learning parameter, and it weighs the importance of the global learning of the swarm. And r_2 plays exactly the same role as r_1.

Lastly, **Figure 1** shows the PSO algorithm flowchart, and one may notice that the optimization logic in it searches for minimums and all position vectors are assessed by the function $f(X)$, known as fitness function. Besides that, **Figures 2** and **3** present the update in a particle's velocity and in its position at a t iteration, regarding a bi-dimensional problem with variables x_1 and x_2.

1. Initialization

    1.1. For each particle i in a swarm population size P:

        1.1.1. Initialize X_i randomly

        1.1.2. Initialize V_i randomly

        1.1.3. Evaluete the fitness $f(X_i)$

        1.1.4. Initialize $pbest_i$ with a copy of X_i

    1.2. Initialize $gbest$ with a copy of X_i with the best fitness

2. Repeat until a stopping criterion is satisfied:

    2.1. For each particle i:

        2.1.1. Update V_i^t and X_i^t according to Eqs. (1) and (2)

        2.1.2. Evaluete the fitness $f(X_i^t)$

        2.1.3. $pbest_i \longleftarrow X_i^t$ if $f(pbest_i) < f(X_i^t)$

        2.1.4. $gbest \longleftarrow X_i^t$ if $f(gbest) < f(X_i^t)$
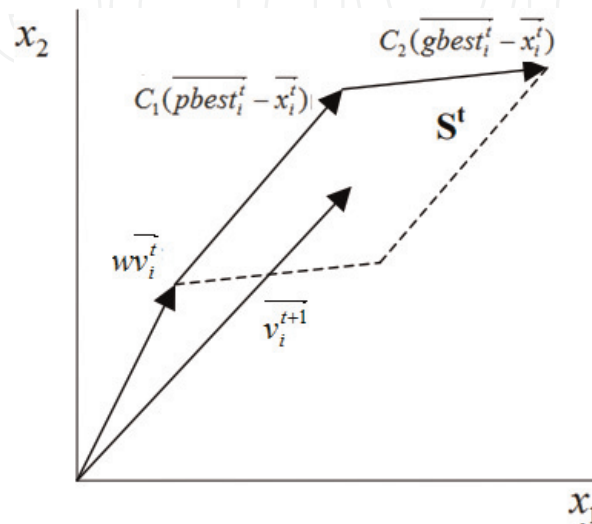
Figure 1.
The PSO algorithm.

Figure 2.
The velocity vector at a t iteration as being composed by two components regarding a bi-dimensional problem.
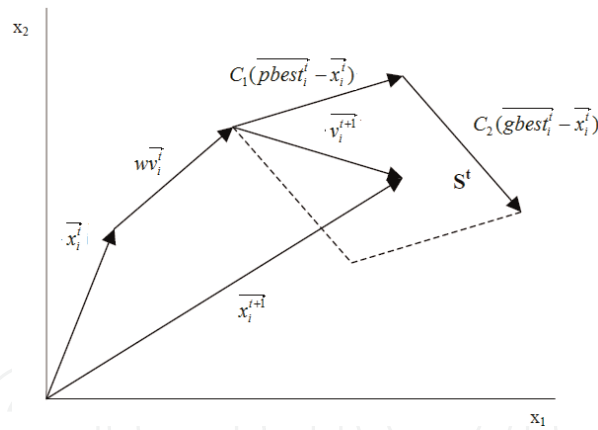
Figure 3.
The position vector being updated at a t iteration as being composed by two components regarding a bi-dimensional problem.

3. Hybrid methods: coupling PSO with deterministic methods

In general, optimization methods are divided into deterministic and heuristic. Deterministic methods aim to establish an iterative process involving a gradient, which, after a certain number of iterations, will converge to the minimum of the objective function. The iterative procedure of this type of method can be written as follows:

$$x^{k+1} = x^k + \alpha^k d^k \tag{3}$$

where x is the variable vector, α is the step size, d is the descent direction, and k is the iteration number. The best that can be expected from any deterministic gradient method is its convergence to a stationary point, usually a local minimum.

Heuristic methods, in contrast to deterministic methods, do not use the objective function gradient as a downward direction. Its goal is to mimic nature in order to find the minimum or maximum of the objective function by selecting, in an elegant and organized manner, the points where such a function will be calculated [9].

Hybrid methods represent a combination of deterministic and heuristic methods in order to take advantage of both approaches. Hybrid methods typically use a heuristic method to locate the most likely region where the global minimum is. Once this region is determined, the hybrid formulation algorithm switches to a deterministic method to get closer and faster to the minimum point. Usually, the most common approach used for this formulation is using the heuristic method to generate good candidates for an optimal solution and then using the best point found as a start point for the deterministic methods in order to converge to local minimums.

Numerous papers have been published over the last few years showing the efficiency and effectiveness of hybrid formulations [10–12]. There are also a growing number of publications over the last decade regarding hybrid formulations for optimization [13].

In this context, PSO algorithm can be combined with deterministic methods, increasing the chance of finding the function's most likely global optimal. This chapter presents the three deterministic methods in which the PSO was coupled: conjugate gradient method, Newton's method, and quasi-Newton method (BFGS). The formulation of each one of those is briefly presented in the following sections.

3.1 Conjugate gradient

The conjugate gradient method improves the convergence rate of the steepest descent method by choosing descending directions that are a linear combination of

the gradient direction with the descending directions of previous iterations. Therefore, their equations are:

$$x^{k+1} = x^k + \alpha^k d^k \tag{4}$$

$$d^k = -\nabla(x^k) + \gamma^k d^{k-1} \tag{5}$$

where γ is the conjugation coefficient that acts by adjusting the size of the vectors. In the Fletcher-Reeves version, the conjugation coefficient is given by:

$$\gamma^k = \frac{\|-\nabla(x^k)\|^2}{\|-\nabla(x^{k-1})\|^2} \tag{6}$$

3.2 Newton's method

While the steepest descent and conjugate gradient methods use first derivative information, Newton's method also uses second derivative information to accelerate the convergence of the iterative process. The algorithm used in this method is presented below:

$$x^{k+1} = x^k + \alpha^k d^k \tag{7}$$

$$d^k = -[H(x)]^{-1} \nabla U(x^k) \tag{8}$$

where $H(x)$ is the Hessian of the function. In general, this method requires few iterations to converge; however, it requires a matrix that grows with the size of the problem. If the estimate is far from the minimum, the Hessian matrix may be poorly conditioned. In addition, it involves inverting a matrix, which makes the method even more computationally expensive.

3.3 Quasi-Newton (BFGS)

BFGS is a type of quasi-Newton method. It seeks to approximate the inverse of the Hessian using the function's gradient information. This approximation is such that it does not involve second derivatives. Thus, this method has a slower convergence rate than Newton's methods, although it is computationally faster. The algorithm is presented below:

$$x^{k+1} = x^k + \alpha^k d^k \tag{9}$$

$$d^k = -H^k \nabla U(x^k) \tag{10}$$

$$H^k = H^{k-1} + M^{k-1} + N^{k-1} \tag{11}$$

$$M^{k-1} = \left[1 + \frac{\left(Y^{k-1}\right)^T \cdot H^{k-1} \cdot Y^{k-1}}{\left(Y^{k-1}\right)^T \cdot d^{k-1}}\right] \frac{d^{k-1} \cdot \left(d^{k-1}\right)^T}{\left(d^{k-1}\right)^T \cdot Y^{k-1}} \tag{12}$$

$$N^{k-1} = -\left[\frac{d^{k-1} \cdot \left(Y^{k-1}\right)^T \cdot H^{k-1} + H^{k-1} \cdot Y^{k-1} \left(d^{k-1}\right)^T}{\left(d^{k-1}\right)^T}\right] \tag{13}$$

$$Y^{k-1} = \nabla U(x^k) - \nabla U(x^{k-1}) \tag{14}$$

4. Recent applications and challenges

PSO can be applied to many types of problems in the most diverse areas of science. As an example, PSO has been used in healthcare in diagnosing problems of a type of leukemia through microscopic imaging [14]. In the economic sciences, PSO has been used to test restricted and unrestricted risk investment portfolios to achieve optimal risk portfolios [15].

In the engineering field, the applications are as diverse as possible. Optimization problems involving PSO can be found in the literature in order to increase the heat transfer of systems [16] or even in algorithms to predict the heat transfer coefficient [17]. In the field of thermodynamics, one can find papers involving the optimization of thermal systems such as diesel engine–organic Rankine cycle [18], hybrid diesel-ORC/photovoltaic system [19], and integrated solar combined cycle power plants (ISCCs) [20].

PSO has also been used for geometric optimization problems in order to find the best system configurations that best fit the design constraints. In this context, we can mention studies involving optical-geometric optimization of solar concentrators [21] and geometric optimization of radiative enclosures that satisfy temperature distribution and heat flow [22].

After having numerous versions of PSO algorithm such as those mentioned in the first section, PSO is able to deal with a broad range of problems, from problems with a few numbers of goals and continuum variables to others with challenging multipurpose problems with many discreet and/or continuum variables. Besides its potential, the user must be aware that the PSO will only achieve appreciated results if one implements an objective function capable of reflecting all goals at once. To derive such a function may be a challenging task that should require a good understanding of the physical problem to be solved and the ability to abstract ideas into a mathematical equation as well. The problems presented in the fourth section of this work provide examples of objective functions capable of playing this role.

Another challenge for one using PSO is how to handle the bounds of the search domain whenever a particle moves beyond it. Many popular strategies that had already been proposed are reviewed and compared for PSO classical version in [23]. Those strategies may be reviewed and understood by PSO users so this person can pick up the one that best fits the optimization problem features.

5. Engineering problems

In this chapter, two engineering problems will be described, one involving the fuel element of a nuclear power plant and the other involving a thermal cogeneration system. In the first problem, the traditional PSO formulation is used to find the optimal fuel element spacing. In the second problem, hybrid optimization algorithms are used to find the operating condition that minimizes total cost of operation of a cogeneration system.

5.1 Springs and dimples of a nuclear fuel bundle spacer grid

In [24], the authors perform the optimization of dimples and spring geometries existing in the nuclear fuel bundle (FB) spacer grid (SG). An FB is a structured group of fuel rods (FRs), and it is also known as fuel assembly, and on the other hand, an FR is a long, slender, zirconium metal tube containing pellets of fissionable material, which provide fuel for nuclear reactors [25]. An SG is a part of the nuclear

fuel bundle and, **Figure 4** shows a schematic view of a nuclear FB; it is possible to see in this illustration how the FRs and the SGs are assembled together. In addition, **Figure 5** gives more details on how an SG's springs and dimples grip an FR, and **Figure 6** shows exactly what parts in the SG are the springs and the dimples that may be in contact with an FR. For this work, the PSO algorithm had been developed in MATLAB® (MathWorks Inc.); meanwhile, the mechanical calculations were performed with finite element analysis (FEA), using ANSYS 15.0 software.

The springs and the dimples act as supports required having special features once an FR releases a great amount of energy, caused by the nuclear reactions occurring within it. Hence, the material of an FR must face a broad range of temperatures when in operation; for example, around a variation of 300°C, this fact is an important matter for the springs and the dimples as those must not impose an
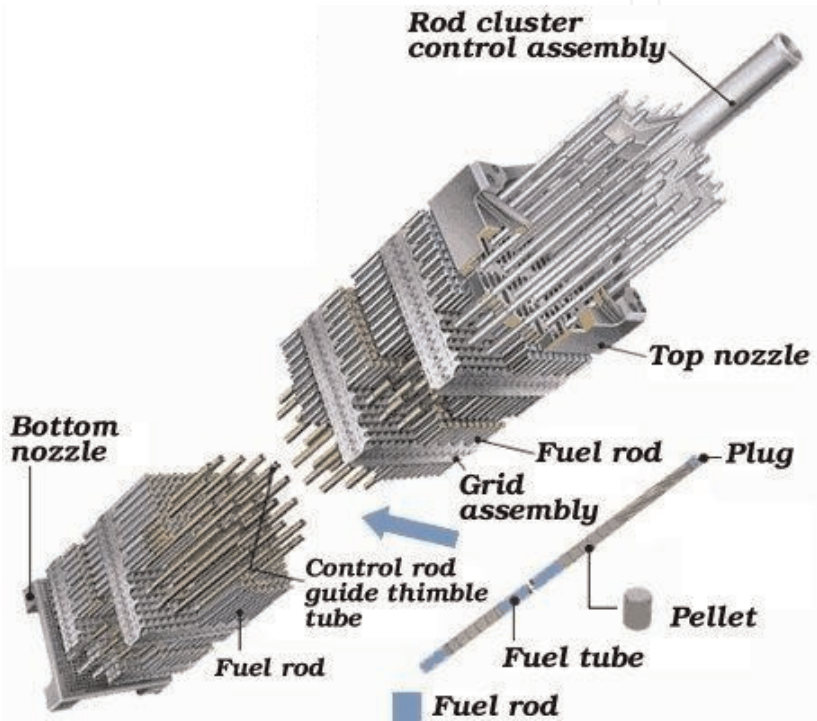

Figure 4.
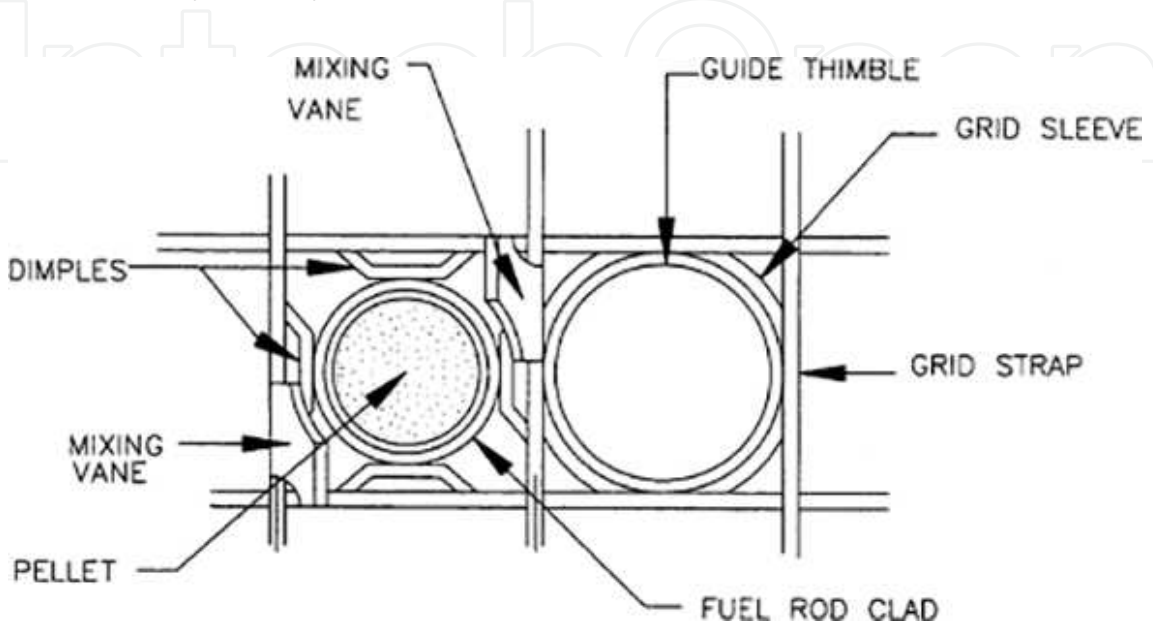*A schematic view of a nuclear fuel bundle.*

Figure 5.
The top view of a spacer grid gripping an FR through its dimples and springs.
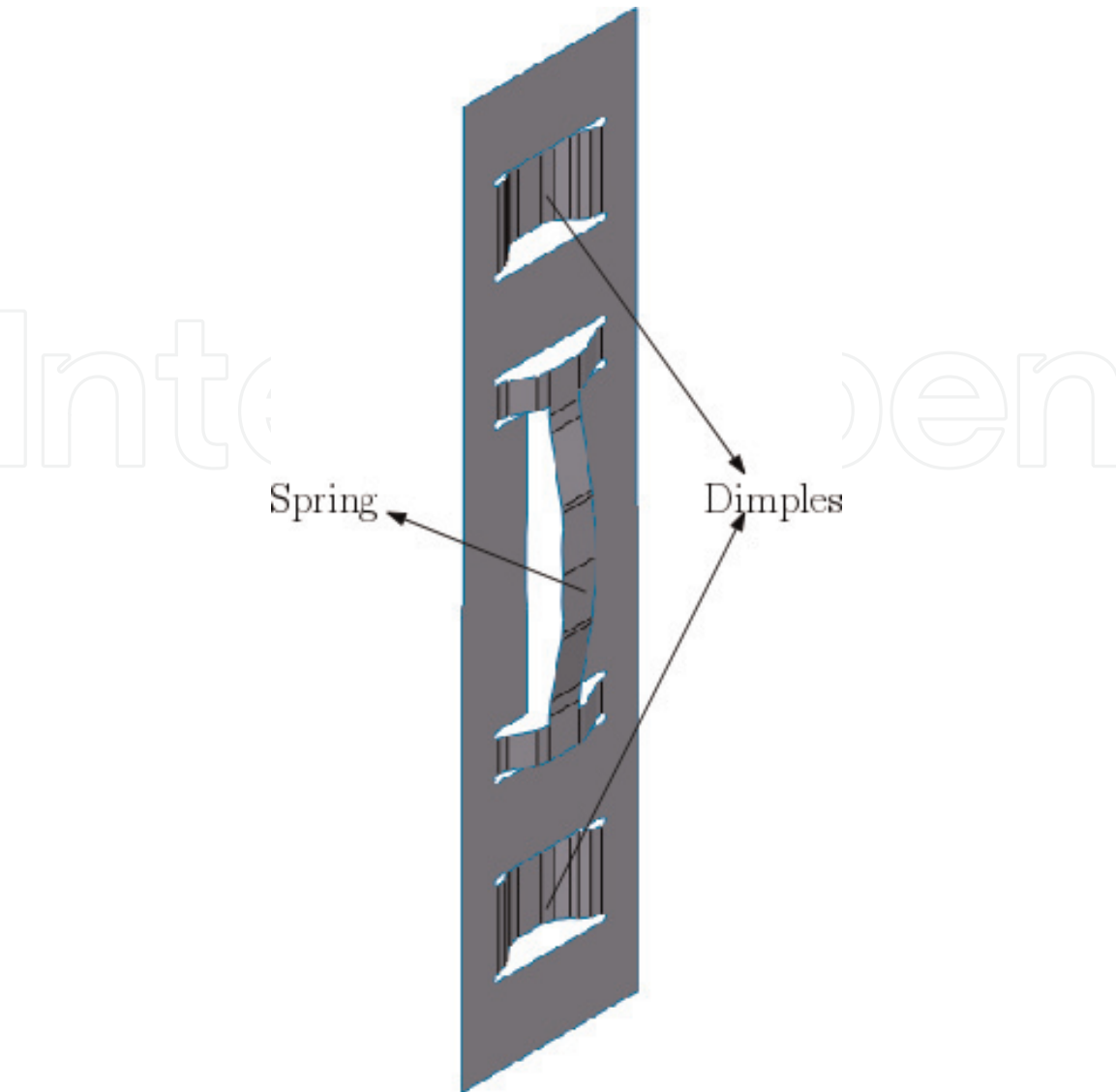
Figure 6.
A part of an SG strip with one spring and two dimples.

excessive gripping force on the rod, allowing it some axial thermal expansion. On the other hand, the upward water flow cooling the great amount of heat released by fission occurring within the rod creates a flow-induced vibration, so the springs and dimples must also limit the lateral displacement of the fuel rods. Besides that, the SG may also support the FRs through its dimples and springs at many loading conditions, that is, earthquakes and shipping and handling. To support safely the fuel in a nuclear reactor is an important matter during operation, and consequences such as the release of fission products from a fuel rod and a reactor safety shutdown could happen because of a poor design.

Finally, one can understand that as the springs and the dimples of an FB must have a geometry able to comply with conflicting requirements so the FRs remain laterally restrained, avoiding it from bowing and vibrating [26], using an optimization algorithm could be useful.

Jourdan et al. [13] had performed the optimization of the dimples and springs of an FB's SG using PSO classical version algorithm. The authors chose some geometry variables that should be important to features such as the gripping stiffness and the stress distribution in the spacer grid, which are the optimization goals in their work. Thus, the position vector is written as $X_i^t = (d_{i1}, d_{i2}, d_{i3}, d_{i4}, d_{i5}, d_{i6})^T$, and these lengths are those in **Figure 7**, while **Table 1** shows the range of such variables, that is, the search domain of the problem.

In PSO simulations from Ref. [24], for each position vector X_i^t, there is an FEA model with the geometry variable values of its related vector. In such FEA model, there are boundary conditions of an elastic static analysis. The boundary conditions considered in these simulations regard one spring and two dimples gripping two FRs, one in contact with the spring and the other one in contact with two dimples. Contacts were not modeled actually in order to simplify the model, and those were replaced by displacements similar to the condition of an FR with the diameter of 9.7 mm being gripped in the available space considering the X_i^t geometry. Other boundary conditions are also the restriction of translations and rotations on the welding nodes. **Figure 8** presents these boundary conditions regarding any position vector. All simulations were built using SHELL181 finite element [27], considering the material to be the Inconel 718.
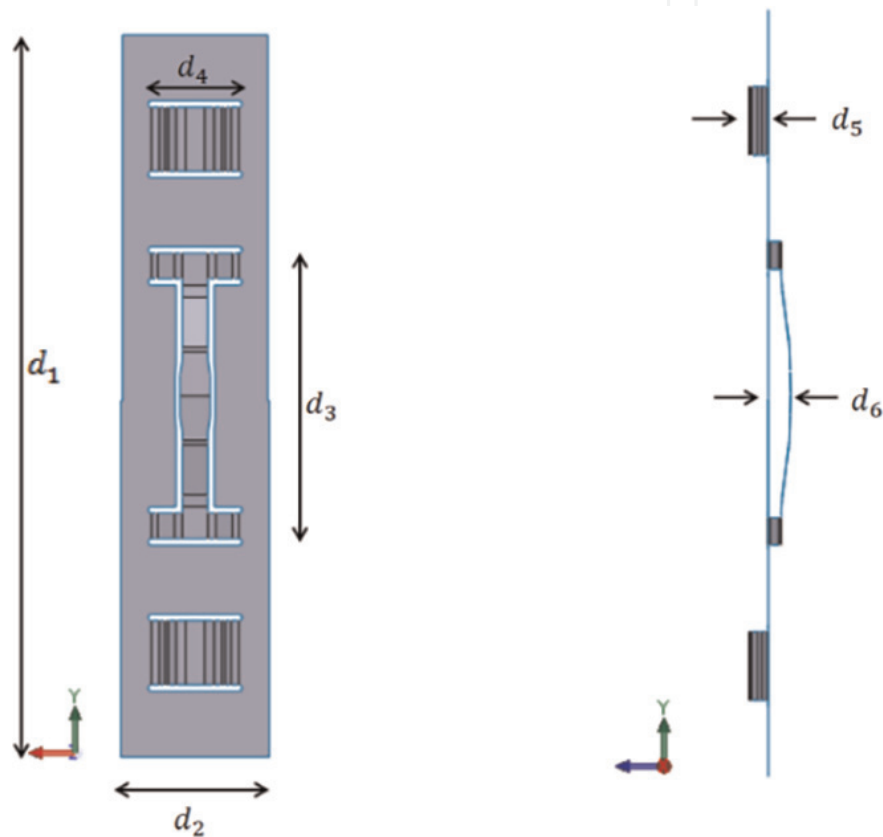

Figure 7.
Variable lengths that should feature the goals of the optimization.

Variable	Lower bound (mm)	Upper bound (mm)
d_1	50	70
d_2	10	15
d_3	5	30
d_4	5	10
d_5	1	5
d_6	1	5

Table 1.
Variable boundaries for the SG optimization.

The goals of the optimization performed in [24] are three: first, to minimize the stress intensity (SI) within the structure; second, to create an SG geometry featuring a gripping stiffness value as close as possible to some $K_{reference}$; and finally, to find a geometry that allows some axial thermal expiation by the FR. These three features are the main mechanical design requirements for an SG [26].

A simulation considering a population of $P = 100$ particles in a swarm and an inertial weight of $w = 0.3$ was performed in [26]. In order to obtain good results from PSO simulations, in other words, to determine the variable values that might fit on actual desired features, one must derive a fitness function able to properly grade all the optimization goals at once, without privileging none of the goals comparing to all others.

It should be noted that the grades assessed by the fitness function could be in an increasing scale or in a decreasing one, depending on the conception of the PSO algorithm. In [26], the authors chose to perform the search at a decreasing scale, and then the fitness function, Eq. (15), was designed to be minimized.

$$f(X) = \sigma + c_k\left(k_{calculated} - k_{reference}\right) \quad \text{if } displacement \geq 0.4mm$$
$$f(X) = 1,000,000 \quad \quad \quad (\text{otherwise}) \quad\quad (15)$$

The fitness function implemented assesses three different terms through two conditions. The two conditions regard the fact that the SG must allow some axial thermal expansion by the FR. To do so, a parameter *displacement* is created, and it measures the space that an FR with 9.7 mm diameter will use when gripped by an SG with some position vector geometry. Thus, a geometry producing a *displacement*
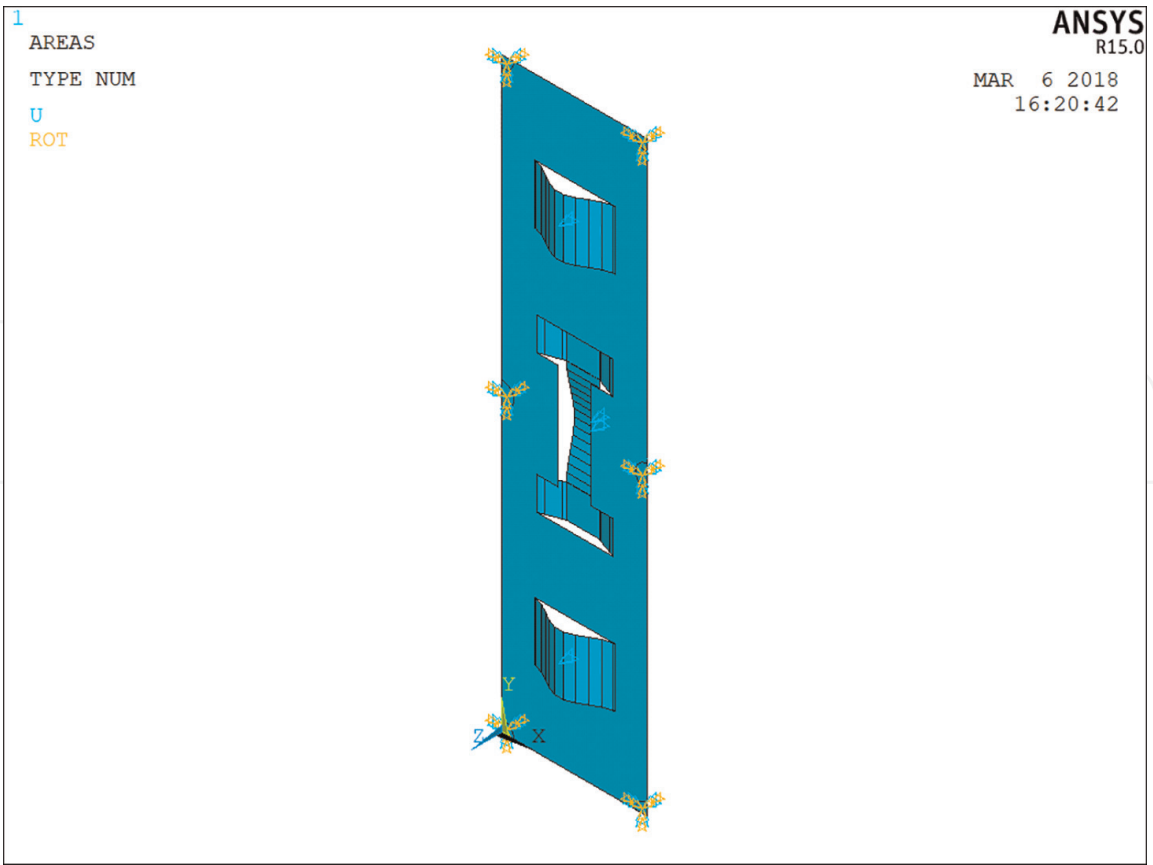

Figure 8.
Model's boundary condition considering any position vector.

over 0.4 mm will receive a high grade, meaning that this is an undesired feature, as the algorithm performs its optimization at a decreasing scale. The value of 0.4 mm is considered to be a good value for the design of an SG [28–31].

The σ parameter represents the SI, and then it is easy to understand that as the SI gets lower this term also does, which is desirable. Finally, the term $c_k\left(k_{calculated} - k_{reference}\right)$ plays the role of finding a geometry that its stiffness, that is, $k_{calculated}$, gets as close as possible to a reference stiffness $k_{reference}$, where this last parameter is set to be 27.2 N/mm [31]. Meanwhile, the parameter c_k is a coefficient that must be set in order to fit the order of magnitude between the fitness function's terms, so none of them gets greater importance. In [24], c_k parameter had been calibrated by performing several PSO simulations, and then, this value was set to be 60. One should notice that the fitness function does not require a unit consistency, as its value is only a mathematical abstraction.

Figure 9 shows the fitness improvement performed to optimize the geometry of an SG's dimples and spring. This simulation resulted in an optimized geometry with an SI of 196 MPa and a gripping stiffness of 27.2 N/mm.

In [31], the authors performed an FEA and a real experiment to measure the SI and the gripping stiffness of the Chashma Nuclear Power Plant Unit 1's (CHASNUPP-1's) SG spring under the same conditions as considered in [24]. The results from [31] regarding a real SG that is in operation at CHASNUPP-1, which might not have been optimized, are 27.2 N/mm for the gripping stiffness and 816 MPa for the SI; meanwhile, the optimized result found in [24] has the same gripping stiffness although with an SI over 75% lower than CHASNUPP-1's SG. Thus, when comparing the results of the most likely optimal found using the PSO algorithm with those from a real SG [31], one can conclude that PSO had played its role well to design the component under study.

5.2 Cost of a cogeneration system

The second problem involves minimizing the function that represents the total cost of operation of a cogeneration system called CGAM. It is named after its creators (C. Frangopoulos, G. Tsatsaronis, A. Valero, and M. von Spakovsky) who
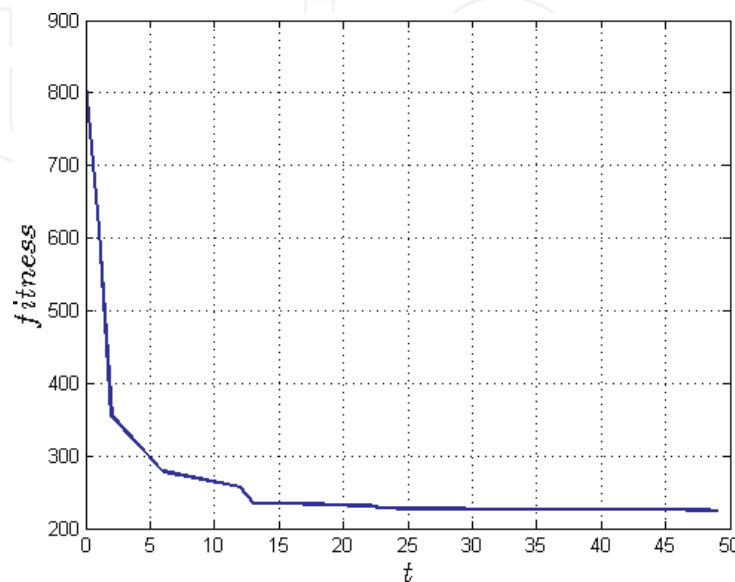
Figure 9.
Fitness improvements from simulation performed in [24].

decided to use the same system to compare the solution of the optimization problem with different methodologies [13]. **Figure 10** indicates the system.

The CGAM system is a cogeneration system consisting of an air compressor (AC), a combustion chamber (CC), a gas turbine (GT), an air preheater (APH), and a heat recovery steam generator (HRSG), which consists of an economizer for preheating water and an evaporator. The purpose of the cycle is the generation of 30 MW of electricity and 14 kg/s of saturated steam at a pressure of 20 bar.

The economic description of the system used in the present work is the same as the one adopted in the original work and considers the annual fuel cost and the annual cost associated with the acquisition and operation of each equipment. More details can be found in [32]. The equations for each component are presented below:

Air compressor:

$$Z_{AC} = \left(\frac{C_{11}\dot{m}_a}{C_{12} - \eta_{AC}}\right)\left(\frac{P_2}{P_1}\right)\ln\left(\frac{P_2}{P_1}\right) \tag{16}$$

Combustion chamber:

$$Z_{cc} = \left(\frac{C_{21}\dot{m}_a}{C_{22} - \frac{P}{P_3}}\right)[1 + \exp(C_{23}T_4 - C_{24})] \tag{17}$$

Turbine:

$$Z_{GT} = \left(\frac{C_{31}\dot{m}_g}{C_{32} - \eta_{GT}}\right)\ln\left(\frac{P_4}{P_5}\right)[1 + \exp(C_{33}T_4 - C_{34})] \tag{18}$$

Preheater:

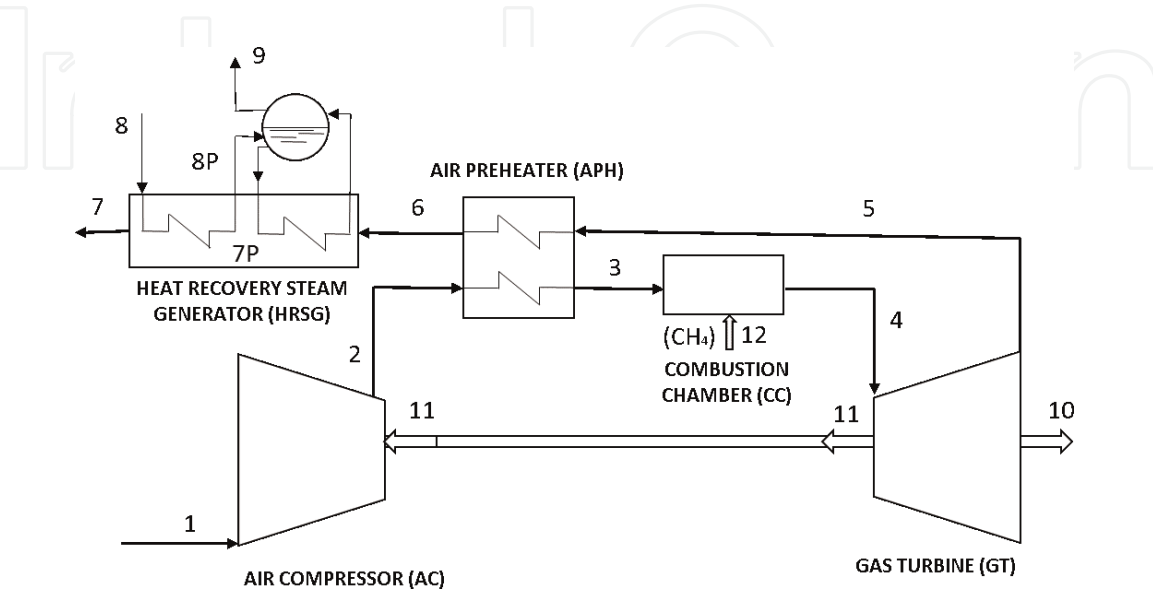$$Z_{APH} = C_{41}\left(\frac{\dot{m}_g(h_5 - h_6)}{(U)(\Delta TLM)}\right)^{0.6} \tag{19}$$

Figure 10.
CGAM system.

Heat recovery steam generator:

$$Z_{HRSG} = C_{51}\left(\left(\frac{Q_{PH}}{(\Delta TLM)_{PH}}\right)^{0.8} + \left(\frac{Q_{PH}}{(\Delta TLM)_{PH}}\right)^{0.8}\right) + C_{52}\dot{m}_{st} + C_{53}\dot{m}_g^{1.2} \quad (20)$$

The general expression for the investment-related cost rate (\$/s) of each component is given by the following equation:

$$\dot{Z}_{i,invest} = \frac{Z_i \varphi CRF}{N.3600} \quad (21)$$

CRF is the capital recovery factor (18.2%), N is the number of annual plant-operating hours (8000 h), and φ is a maintenance factor (1.06). In addition, c_f is the fuel cost per unit of energy (0.004 \$/MJ). **Table 2** indicates the cost constants adopted for each component. The following equation represents the total cost of operation rate:

$$F = c_f \dot{m}_f PCI + \dot{Z}_{AC} + \dot{Z}_{APH} + \dot{Z}_{CC} + \dot{Z}_{GT} + \dot{Z}_{HRSG} \quad (22)$$

In order to perform the optimization of Eq. (22), the five decision variables adopted in the definition of the original problem are considered: the compression ratio (P_2/P_1), the isentropic efficiency of the compressor (η_{CA}), the isentropic efficiency of the turbine (η_{GT}), the air temperature at the preheater outlet (T_3), and the fuel gas temperature at the turbine inlet (T_4). To optimize the objective function, three optimization routines coupling PSO with different deterministic methods were used as indicated in **Table 3**.

Air compressor	$C_{11} = 39.5\ \$/\left(\frac{\text{kg}}{\text{s}}\right) C_{12} = 0.9$
Combustion chamber	$C_{21} = 25.6\ \$/\left(\frac{\text{kg}}{\text{s}}\right) C_{22} = 0.995$
	$C_{23} = 0.018\ (\text{K}^{-1}) C_{24} = 26.4$
Gas turbine	$C_{31} = 266.3\ \$/\left(\frac{\text{kg}}{\text{s}}\right) C_{32} = 0.92$
	$C_{33} = 0.036\ (\text{K}^{-1}) C_{34} = 54.4$
Preheater	$C_{41} = 39.5\ \$/(\text{m}^{1,2})\ U = 0.018 k\text{W}/(\text{m}^2\text{K})$
HRSG	$C_{51} = 3650\ \$/\left(\frac{\text{kW}}{\text{K}}\right)^{0,8} C_{52} = 11,820\ \$/\left(\frac{\text{kg}}{\text{s}}\right)$
	$C_{53} = 658\ \$/\left(\frac{\text{kg}}{\text{s}}\right)^{1,2}$

Table 2.
Cost constants.

	Heuristic	Deterministic
Hybrid 1	Particle swarm	Conjugate gradient
Hybrid 2	Particle swarm	Quasi-Newton
Hybrid 3	Particle swarm	Newton

Table 3.
Hybrid methods.

To solve the thermodynamic equations of the problem, the professional process simulator IPSEpro® version 6.0 was adopted. IPSEpro® is a process simulator used to model and simulate different thermal systems through their thermodynamic equations. This program was developed by SimTech and has a user-friendly interface, as well as a library with a wide variety of components, allowing the user to model and simulate conventional plants, cogeneration systems, cooling cycles, combined cycles, and more. The optimization method routines were written in MATLAB® (MathWorks Inc.), and the algorithm was integrated with IPSEpro® in order to solve the thermodynamic problem and perform the optimization.

To perform the optimization, the limits for the problem variables were established, as indicated in **Table 4** [33].

Table 5 presents the results found for the variables in each method and the value of the objective function. **Figures 11–13** present the graphs of the evolution of the cost function in relation to the function call for the performed optimizations.

In order to evaluate the algorithm's efficiency, a comparison was made between the results obtained in the present work and those obtained by [32, 33]. It is worth mentioning that the thermodynamic formulation used by [32] is slightly different from that constructed in the simulator; therefore, some differences in the final value of the objective function were already expected. In [33], the CGAM system was also built in IPSEpro® and the optimization was performed in MATLAB® using the following optimization methods: differential evolution (DE), particle swarm (PSO), simulated annealing (SA), genetic algorithm (GA), and direct pattern search (DPS). A comparison between the results is presented in **Figure 14**.

It is possible to verify that the hybrid methods used in this work have excellent performance, and the values found are compatible with the other references. This result consolidated the use of hybrid formulations used to optimize the objective function of the problem.

Limits
$7 \leq P_2/P_1 \leq 27$
$0.7 \leq \eta_{CA} \leq 0.9$
$0.7 \leq \eta_{GT} \leq 0.9$
$700 \text{ K} \leq T_3 \leq 1100 \text{ K}$
$1100 \text{ K} \leq T_4 \leq 1500 \text{ K}$

Table 4.
Variable limits.

	Hybrid 1	Hybrid 2	Hybrid 3
P_2/P_1	9.46	9.04	8.29
η_{CA}	0.83	0.83	0.85
T_3	600.43	612.53	606.47
η_{GT}	0.88	0.88	0.88
T_4	1210.95	1212.67	1214.65
Cost function ($/s)	0.33948	0.33953	0.33949

Table 5.
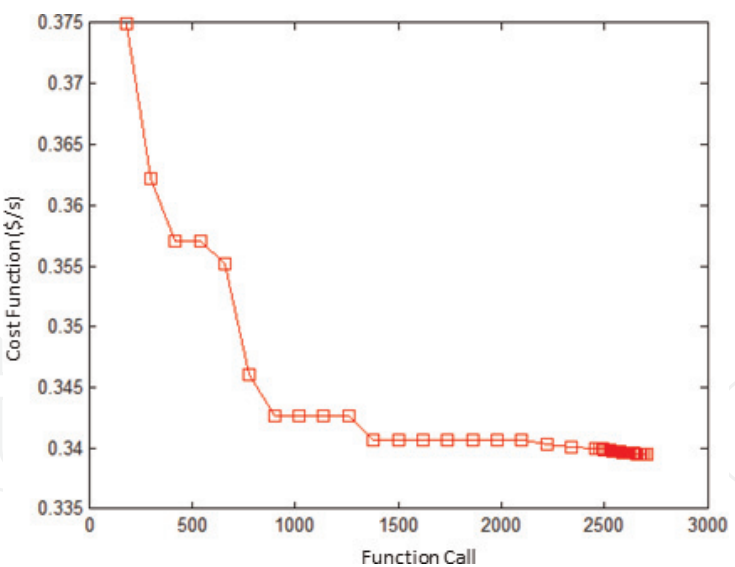*Optimization results.*

15

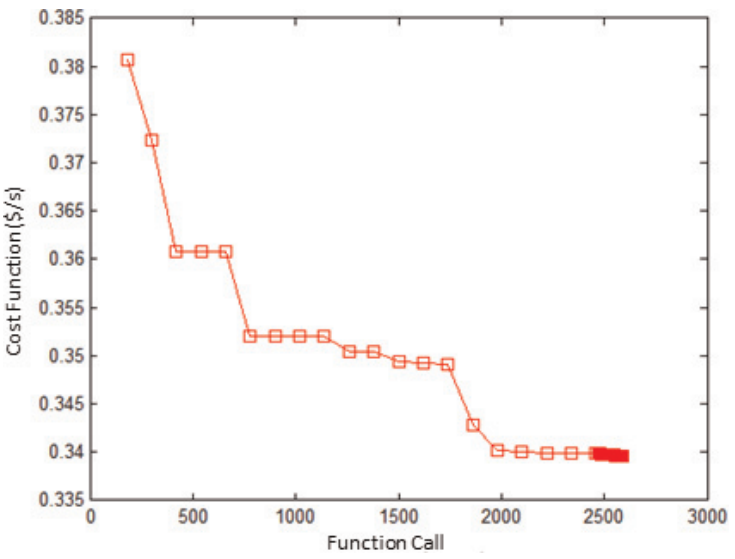**Figure 11.**
Hybrid 1 optimization.

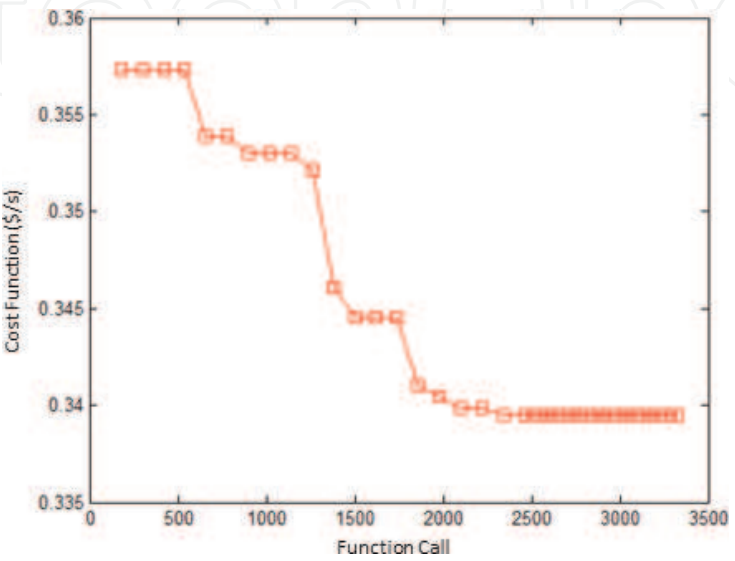**Figure 12.**
Hybrid 2 optimization.

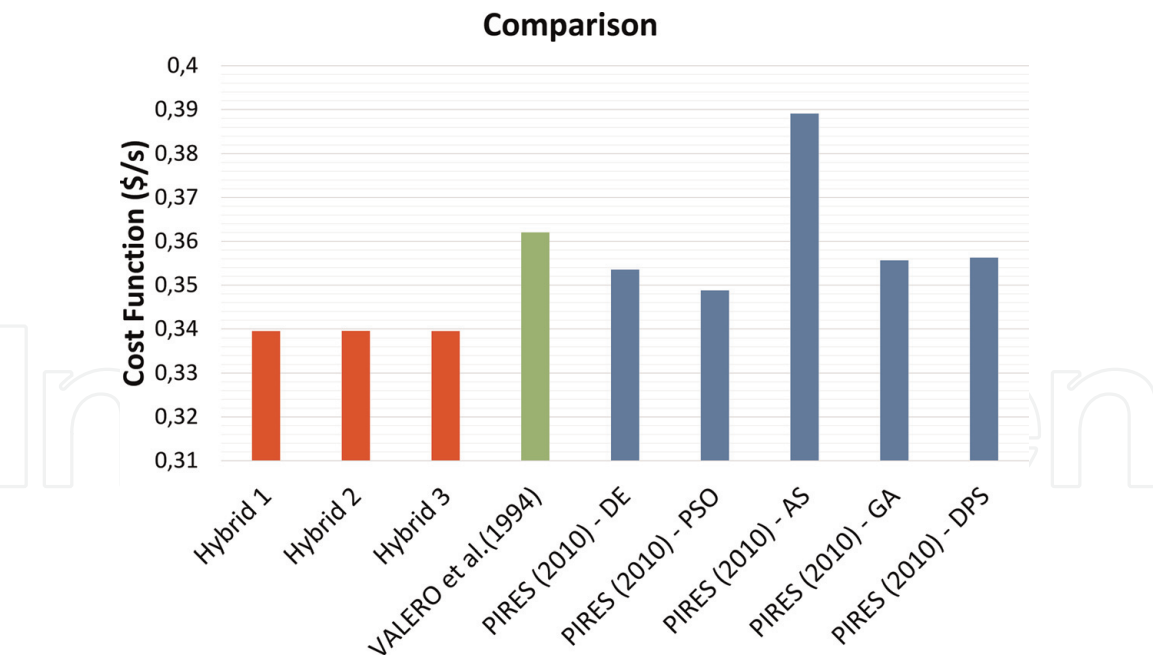**Figure 13.**
Hybrid 3 optimization.

Figure 14.
Comparison between the results obtained and bibliographic references.

6. Conclusions

In the present work, it was possible to present the basic fundamentals involving the PSO method. The advantages and disadvantages of the method were discussed, as well as interpretations were provided to its algorithm. It was also possible to discuss about hybrid methods that combine deterministic and heuristic methods in order to extract the advantages of each one.

As discussed earlier, it is impracticable to say that the result obtained by an optimization method such as PSO is the global maximum or minimum, so some authors call the results as the most likely optimal global. Thus, some strategies can be employed in order to verify the validity of the optimal results obtained. One of the strategies is to compare with the results obtained by other optimization algorithms, as used in the present work. In the absence of optimal data available, due to either computational limitations or even lack of results of the subject, it is possible to use as strategy the comparison of information from real physical models, that is, that were not obtained through optimization algorithms, but instead good engineering practice and judgment gained through technical experience.

In addition, it was possible to apply the PSO algorithm to different engineering problems. The first involves the spacer grid of the fuel element and the second involves the optimization of the cost function of a cogeneration system. In both problems, satisfactory results were obtained demonstrating the efficiency of the PSO method.

Author details

Bruno Seixas Gomes de Almeida* and Victor Coppo Leite
Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

*Address all correspondence to: brunoseixas@poli.ufrj.br

IntechOpen

References

[1] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of the International Conference on Neural Networks; Institute of Electrical and Electronics Engineers. Vol. 4. 1995. pp. 1942-1948. DOI: 10.1109/ICNN.1995.488968

[2] Meneses AAM, Machado MD, Schirru R. Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor. Progress in Nuclear Energy. 2009;**51**:319-326. DOI: 10.1016/j.pnucene.2008.07.002

[3] Sarkar S, Roy A, Purkayastha BS. Application of particle swarm optimization in data clustering: A survey. International Journal of Computers and Applications. 2013;**65**: 38-46. DOI: 10.5120/11276-6010

[4] Kennedy J, Eberhart RC. A new optimizer using particles swarm theory. In: Proceedings of Sixth International Symposium on Micro Machine and Human Science IEEE. 1995. pp. 39-43. DOI: 10.1109/MHS.1995. 494215

[5] Shi Y, Eberhart RC. Empirical study of particle swarm optimization. In: Proceedings of the 1999 IEEE Congress on Evolutionary Computation (CEC'99). Vol. 3. 1999. pp. 1945-1950. DOI: 10.1109/CEC.1999.785511

[6] Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 Congress on Evolutionary Computation (CEC '00). Vol. 1. 2000. pp. 84-88. DOI: 10.1109/CEC.2000.870279

[7] Coelho L, Mariani VC. Particle swarm optimization with quasi-Newton local search for solving economic dispatch problem. In: IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06. Vol. 4.

2006. pp. 3109-3113. DOI: 10.1109/ICSMC.2006.384593

[8] Han KH, Kim JH. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Transactions on Evolutionary Computation. 2002;**6**:580-593. DOI: 10.1109/TEVC.2002.804320

[9] Colaço MJ, Orlande HRB, Dulikravich GS. Inverse and optimization problems in heat transfer. Journal of the Brazilian Society of Mechanical Sciences and Engineering. 2006;**28**:1-24. DOI: 10.1590/S1678-58782006000100001

[10] Nery RS, Rolnik V. Métodos Híbridos para Otimização global não-linear. In: Congresso Nacional de Matemática Aplicada e Computacional; Florianópolis, SC, Brasil. 2007

[11] Zadeh PM, Sokhansefat T, Kasaeian AB, et al. Hybrid optimization algorithm for thermal analysis in a solar parabolic trough collector based on nanfluid. Energy. 2015;**82**:857-864. DOI: 10.1016/j.energy.2015.01.096

[12] Dominkovic DF, Cosic B, Medic B, et al. A hybrid optimization model of biomass trigeneration system combined with pit thermal energy storage. Energy Conversion and Management. 2015;**104**: 90-99. DOI: 10.1016/j.enconman.2015. 03.056

[13] Jourdan L, Basseur M, Talbi EG. Hybridizing exact methods and metaheuristics: A taxonomy. European Journal of Operational Research. 2009; **199**:620-629. DOI: 10.1016/j.ejor.2007. 07.035

[14] Srisukkham W, Zhang L, Neoh SC, Todryk S, Lim CP. Intelligent leukaemia diagnosis with bare-bones PSO based feature optimization. Applied Soft

Computing. 2017;**56**:405-419. DOI: 10.1016/j.asoc.2017.03.024

[15] Zhu H, Wang Y, Wang K, Chen Y. Particle swarm optimization (PSO) for the constrained portfolio. Expert System with Applications. 2011;**38**:10161-10169. DOI: 10.1016/j.eswa.2011.02.075

[16] Payan S, Azimifar A. Enhancement of heat transfer of confined enclosures with free convection using blocks with PSO algorithm. Applied Thermal Engineering. 2016;**101**:79-91. DOI: 10.1016/j.applthermaleng.2015.11.122

[17] Malekan M, Khosravi A. Investigation of convective heat transfer of ferrofluid using CFD simulation and adaptive neuro-fuzzy inference system optimized with particle swarm optimization algorithm. Powder Technology. 2018;**333**:364-376. DOI: 10.1016/j.powtec.2018.04.044

[18] Zhao R, Zhang H, Song S, Yang F, Hou X, Yan Y. Global optimization of the diesel engine–organic Rankine cycle (ORC) combined system based on particle swarm optimizer (PSO). Energy Convesion and Management. 2018;**174**: 248-259. DOI: 10.1016/j.enconman. 2018.08.040

[19] Nogueira ALN, Castellanos LSM, Lora EES, Cobas VRM. Optimum design of a hybrid diesel-ORC/photovoltaic system using PSO: Case study for the city of Cujubim, Brazil. Energy. 2018; **142**:33-45. DOI: 10.1016/j.energy.2017. 10.012

[20] Mabrouk MT, Kheiri A, Feidt M. A systematic procedure to optimize integrated solar combined cycle power plants (ISCCs). Applied Thermal Engineering. 2018;**136**:97-107. DOI: 10.1016/j.applthermaleng.2018.02.098

[21] Ajdad H, Baba YF, Mers AA, Merron O, Bouatem A, Boutmmachte N. Particle swarm optimization algorithm for optical-geometric optimization of linear fresnel solar concentrators. Renewable Energy. 2019;**130**:992-1001. DOI: 10.1016/j.renene.2018.07.001

[22] Farahmand A, Payan S, Sarvari SMH. Geometric optimization of radiative enclosures using PSO algorithm. International Jounal of Thermal Sciences. 2012;**60**:61-69. DOI: 10.1016/j.ijthermalsci.2012.04.024

[23] Padhye N, Mittal P, Deb K. Boundary handling approaches in particle swarm optimization. Advances in Intelligents Systems and Computing. 2013;**201**:287-298. DOI: 10.1007/978-81-322-1038-2_25

[24] Leite VC, Schirru R, Neto MM. Particle swarm optimization applied to the nuclear fuel bundle spacer grid spring design. Nuclear Technology. 2018;**205**:637-645. DOI: 10.1080/00295450.2018.1516056

[25] United States Nuclear Regulatory Commission (U.S.NRC). Glossary. Available from: https://www.nrc.gov/reading-rm/basic-ref/glossary [Accessed: 2019-06-14]

[26] United States Nuclear Regulatory Commission (U.S.NRC). Westinghouse AP1000 Design Control Document. Final safety analysis report. Westinghouse Electric Company; 2011

[27] ANSYS User's Manual for Revision 5.0. Swanson Analysis System, Inc.; 2013

[28] Shin MK et al. Optimization of a nuclear fuel spacer grid spring using homology constraints. Nuclear Engineering and Design. 2008;**238**: 2624-2634. DOI: 10.1016/j. nucengdes.2008.04.003

[29] Lee S, Kim Y, Song K. Parameter study for a dimple location in a space grid under the critical impact load. Journal of Mechanical Science and

Technology. 2008;**22**:2024-2029. DOI: 10.1007/s12206-008-0620-5

[30] Park KJ et al. Design of a spacer grid using axiomatic design. Journal of Nuclear Science and Technology. 2003; (12):989-997. DOI: 10.3327/jnst.40.989

[31] Wassen W et al. Fuel rod-to-support contact pressure and stress measurement for CHASNUPP-1 (PWR) fuel. Nuclear Engineering and Design. 2011;**241**:32-38. DOI: 10.1016/j. nucengdes.2010.11.004

[32] Frangopoulos C, Tsatsaronis G, Valero A, et al. CGAM problem: Definition and conventional solution. Energy. 1994;**19**:279-286. DOI: 10.1016/ 0360-5442(94)90112-0

[33] Pires TS. Método de Superfície de Resposta Aplicado à Otimização Termoeconômica de Sistemas de Cogeração Modelados em um Simulador de Processos [thesis]. Rio de Janeiro, Brasil: COPPE-UFRJ; 2010