Hybrid particle swarm optimization with spiral-shaped mechanism for solving high-dimension problems

Humberto M. M. Duarte[1] and Rafael Lima de Carvalho[1]

[1] *Universidade Federal do Tocantins, Curso de Ciência da Computação, Tocantins, Brazil*

Reception date of the manuscript: 11/02/2020
Acceptance date of the manuscript: 28/02/2020
Publication date: 09/03/2020

Abstract— Particle swarm optimization (PSO) is a well-known metaheuristic, whose performance for solving global optimization problems has been thoroughly explored. It has been established that without proper manipulation of the inertia weight parameter, the search for a global optima may fail. In order to handle this problem, we investigate the experimental performance of a PSO-based metaheuristic known as HPSO-SSM, which uses a logistic map sequence to control the inertia weight to enhance the diversity in the search process, a spiral-shaped mechanism as a local search operator, as well as two dynamic correction factors to the position formula. Thus, we present an application of this variant for solving high-dimensional optimization problems, and evaluate its effectiveness against 24 benchmark functions. A comparison between both methods showed that the proposed variant can escape from local optima, and demonstrates faster convergence for almost every evaluated function.

Keywords—Particle swarm optimization, High-dimensional global optimization, Optimization

I. Introduction

Global optimization is the process of selecting a specific answer, out of a space of possibilities, that better solves a given problem. For minimization problems, this is often represented in the following way [1]:

Given: $f : A \to \mathbb{R}$
Find: $X^* \in A$ such that $f(X^*) \leq f(X)$ for all $X \in A$

Where f is defined as the objective function, or fitness function, $A \subseteq \mathbb{R}^d$ denotes a decision space of d dimensions, and each vector $X^* = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d$ represents a candidate solution. A solution that minimizes, or maximizes, the objective function is called an optimal solution.

For a very broad range of applications, there are many theoretical and real-world problems that can be modeled in this framework [2]. Therefore, algorithms that can efficiently find optimal solutions for these problems become essential for the science and engineering fields. This is specially true for problems involving a high number of dimensions, which exponentially expands the size of the search space, and increases the computational cost for most optimization methods.

This paper presents a modification of the hybrid particle swarm optimization with a spiral-shaped mechanism (HPSO-SSM), as presented in [3], for solving high-dimensional global optimization problems. The effectiveness of the proposed algorithm is evaluated against a set of well-known benchmark test functions [4], with up to 5000 dimensions.

Contact data: Humberto M. M. Duarte, 109 Norte, Av. NS15, Câmpus Universitário de Palmas, hbertoduarte@gmail.com

II. HPSO-SSM Metaheuristic

The particle swarm algorithm (PSO), proposed by [5], is a metaheuristic based in the social behaviors of organisms, similar to a bird flock or school of fish. A population of particles is created, each representing a candidate solution to the chosen problem. Each of those particles has a position and velocity, whose values are iteratively adjusted based on their own best known previous position, as well as the current best position of the swarm. This way, the search-space is explored until a stop criteria is met.

Hybrid particle swarm optimization with spiral-shaped mechanism (HPSO-SSM) [3] is a modification of the traditional PSO, originally designed for feature selection in machine learning tasks. It utilizes the logistic map sequence to improve diversity, and a logarithmic spiral as a local search operator.

In the following, we describe the general procedure for this algorithm, as roughly described in Fig. 1, along with the changes made in order to adapt it for global optimization problems.

Firstly, the population of particles is created, with n individuals. During the whole procedure, the population size is constant. For a given iteration t, the current position and

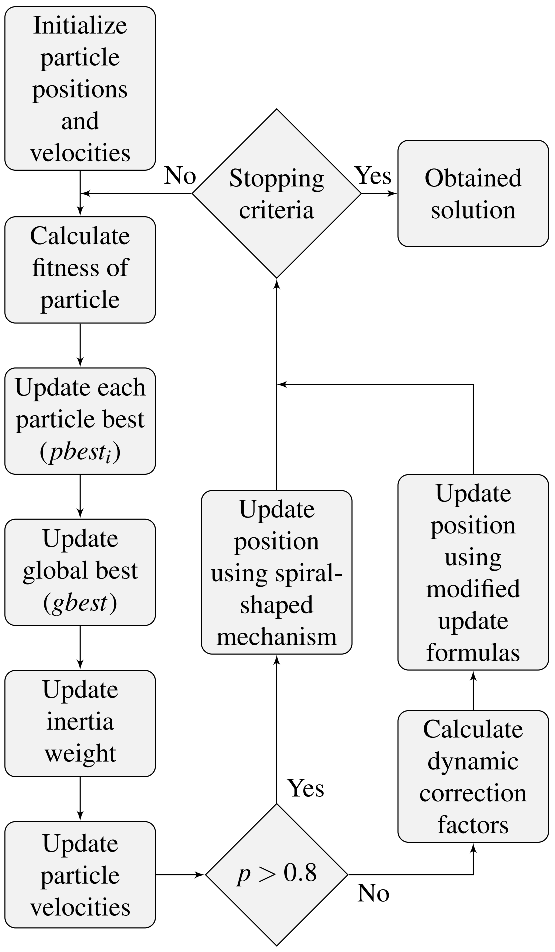
Initialize particle positions and velocities
Calculate fitness of particle
Update each particle best ($pbest_i$)
Update global best ($gbest$)
Update inertia weight
Update particle velocities
No
Stopping criteria
Yes
Obtained solution
Update position using spiral-shaped mechanism
Update position using modified update formulas
Calculate dynamic correction factors
Yes
$p > 0.8$
No

Fig. 1: HPSO-SSM algorithm.

velocity of each particle i are given by, respectively:

$$x_i^t = (x_{i1}^t, x_{i2}^t, \cdots, x_{id}^t)$$
$$v_i^t = (v_{i1}^t, v_{i2}^t, \cdots, v_{id}^t)$$

The position and velocity vectors are initialized with random values, uniformly distributed within the range of the search-space.

At every iteration, the fitness of each particle is calculated, using the position vector x_i directly as the argument of the objective function f. According the fitness value, the current best known position for each particle, and the best know position for the entire swarm, are updated, here denoted as:

$$\begin{aligned} \text{Fitness}_{\min} &= f(x_i) \\ pbest_i &= (pbest_{i1}, pbest_{i2}, \cdots, pbest_{id}) \\ gbest &= (gbest_1, gbest_2, \cdots, gbest_d) \end{aligned}$$

In order to guide the convergence of the result towards the global optima, the HPSO-SSM algorithm utilizes the logistic map to update the inertia weight w, as follows:

$$w^t = \mu \times w^{t-1} \times (1 - w^{t-1})$$

where μ is a positive constant ($\mu = 4$). The characteristics of randomness and ergodicity of the sequence enhance the ability of the algorithm to escape locally optimal solutions.

Next, the flying velocities of each particle are updated, utilizing the values of *pbest* and *gbest* to steer the swarm towards the solution, according to the following formula:

$$\begin{aligned} v_{id}^{t+1} &= w^t + v_{id}^t \\ &\quad + c_1 \times r_{1i} \times (pbest_{id} - x_{id}^t) \\ &\quad + c_2 \times r_{2i} \times (gbest_d - x_{id}^t) \end{aligned}$$

where c_1 and c_2 are position constants, usually set as $c_1 = c_2 = 2$, and the variables r_{1i} and r_{2i} are random values, uniformly distributed, between 0 and 1.

In the HPSO-SSM method, there are two different ways in which the particle positions are updated. The first makes used of the spiral-shaped mechanism, and the second utilizes a modification of the standard PSO position update formula. The method to be used is determined by the variable p, a random value between 0 and 1, chosen at every operation. Should $p > 0.8$, the spiral-shaped mechanism is used, otherwise, the modified update formula is used.

Per the spiral-shaped mechanism, the next position of a particle is calculated with the following:

$$x_{id}^{t+1} = |gbest_d - x_{id}^t| \times \exp(b \times l) \times \cos(2\pi l) + gbest_d$$

where b is a positive constant ($b = 2$), and l is a random value, between -1 and 1. Otherwise, the position is updated using the modified formula below:

$$x_{id}^{t+1} = \mathfrak{R}_1^t \times x_{id}^t + \mathfrak{R}_2^t \times v_{id}^{t+1}$$

where $\mathfrak{R}_1$ and $\mathfrak{R}_2$ are dynamic correction factors, introduced in the HPSO-SSM algorithm, in order to control the trade-off between exploration and exploitation within the search.

Since that, in the HPSO-SSM algorithm, the position values are limited to the range between 0 and 1, the formulas for the dynamic correction factors were slightly modified, in order to preserve their functionality in the domain of the chosen objective function. Therefore:

$$\begin{aligned} \mathfrak{R}_1^t &= \frac{1}{(1 + \exp(a \times (-\frac{\min(SP)}{\max(SP)})))^t} \\ \mathfrak{R}_2^t &= 1 - \mathfrak{R}_1^t \\ SP_i^t &= \frac{x_i^{t-1}}{M} \times \left[\frac{x_i^{t-1}}{M}\right]^T \\ M &= \max(|A_{min}|, |A_{max}|) \end{aligned}$$

where a is a constant ($a = 2$), SP is the modified position magnitude formula, and M is the normalization factor, in which A_{min} and A_{max} denote the boundaries of the search-space.

The search loop is executed either until a set fitness value is reached, or for a fixed number of iterations. At the end, the particle with the best known position yields the obtained solution. However, the HPSO-SSM algorithm, as with the traditional PSO, does not guarantee that an optimal solution will be found.

III. Experimental Results

In order to evaluate the effectiveness of the HPSO-SSM algorithm over global optimization problems with high dimensions, the following experimentation was conducted. A set

TABLE 1: TEXT FUNCTIONS UTILIZED IN THE EXPERIMENTATION.

Function Name	Function Formula	Search Range
Sphere	$f_1(x) = \sum_{i=1}^{d} x_i^2$	[-100,100]
Sum-Square	$f_2(x) = \sum_{i=1}^{d} ix_i^2$	[-10,10]
Schwefel's 2.22	$f_3(x) = \sum_{i=1}^{d} \lvert x_i\rvert + \prod_{i=1}^{d} \lvert x_i\rvert$	[-10,10]
Rotated hyper-ellipsoid	$f_4(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} x_j)^2$	[-100,100]
Schwefel's 2.21	$f_5(x) = \max\{\lvert x_i\rvert, 1 \le x_i \le d\}$	[-100,100]
Rosenbrock	$f_6(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	[-30,30]
Step	$f_7(x) = \sum_{i=1}^{d} (\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]
Quartic	$f_8(x) = \sum_{i=1}^{d} ix_i^4$	[-1.28,1.28]
Noise	$f_9(x) = \sum_{i=1}^{d} ix_i^4 + \text{random}[0,1)$	[-1.28,1.28]
Sum-Power	$f_{10}(x) = \sum_{i=1}^{d} \lvert x_i\rvert^{i+1}$	[-1,1]
Rastrigin	$f_{11}(x) = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]
Ackley	$f_{12}(x) = -20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - \exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)) + 20 + e$	[-32,32]
Griewank	$f_{13}(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]
Levy	$f_{14}(x) = \sum_{i=1}^{d-1} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + \lvert x_d - 1\rvert[1 + \sin^2(3\pi x_n)]$	[-10,10]
Alpine	$f_{15}(x) = \sum_{i=1}^{d} \lvert x_i \sin(x_i) + 0.1x_i\rvert$	[-10,10]
Inverted Cosine mixture	$f_{16}(x) = 0.1d - (0.1\sum_{i=1}^{d}\cos(5\pi x_i) - \sum_{i=1}^{d} x_i^2)$	[-1,1]
Zakharov	$f_{17}(x) = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + (\sum_{i=1}^{d} 0.5ix_i)^4$	[-5,10]
Pathological	$f_{18}(x) = \sum_{i=2}^{d} 0.5 + \frac{\sin^2(\sqrt{100x_{i-1}^2 + x_i^2}) - 0.5}{1 + 0.001(x_{i-1}^2 - 2x_{i-1}x_i + x_i^2)^2}$	[-100,100]
Levy and Montalvo	$f_{19}(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{d-1} (x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_d - 1)^2(1 + \sin^2(2\pi x_d)))$	[-5,5]
Elliptic	$f_{20}(x) = \sum_{i=1}^{d} (10^6)^{\frac{i-1}{d-1}} x_i^2$	[-100,100]
Easom	$f_{21}(x) = (-1)^{d+1}\prod_{i=1}^{d}\cos(x_i) \cdot \exp(-\sum_{i=1}^{d}(x_i - \pi)^2)$	[-100,100]
Salomon	$f_{22}(x) = 1 - \cos(2\pi\sqrt{\sum_{i=1}^{d} x_i^2}) + 0.1\sqrt{\sum_{i=1}^{d} x_i^2}$	[-100,100]
Schaffer	$f_{23}(x) = 0.5 + \frac{\sin^2(\sqrt{\sum_{i=1}^{d} x_i^2}) - 0.5}{(1 + 0.001(\sum_{i=1}^{d} x_i^2))^2}$	[-100,100]
Stretched V-sine	$f_{24}(x) = \sum_{i=1}^{d-1} (x_i^2 + 2x_{i+1}^2)^{0.25} \cdot ((\sin 50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1)$	[-10,10]

of 24 well-known benchmark functions was assembled, from existing literature involving optimization algorithms [4], that are delineated in Table 1. For all functions, the theoretical optima is $f_{min} = 0$.

For all tests, each function was executed 30 times, for dimension sizes of $d = \{30, 100, 500, 1000, 5000\}$. The population size n was fixed to 30, and the maximum number of iterations t_{max} was 500, for all simulations, resulting in a maximum of 15000 fitness function evaluations per test. In addition, the same control parameters were used for both PSO and HPSO-SSM algorithms, as follows: the position constants c_1 and c_2 were both set as 2, and the initial value of the inertia weight w was set as 0.5. The HPSO-SSM constants μ, a, and b were set to 4, 2, and 2, respectively.

Both algorithms, as well as the benchmark functions, were coded in Python 3.6, and all experiments were performed on the cloud-based environment Google Colaboratory [6].

a. Comparison with traditional PSO

First, a comparison is made between the traditional PSO and the proposed HPSO-SSM method, for a low number of dimensions ($d = 30$). The mean of the fitness values reached for each function, and their standard deviations, are arranged in Table 2. It can be seen in Table 2 that for almost every benchmark function, the HPSO-SSM method reaches the global optima, with exception of six functions (i.e., f_6, f_9, f_{14}, f_{19}, f_{22}, and f_{23}).

The PSO method, however, demonstrated results orders of magnitude worse, only reaching global optima on two

TABLE 2: COMPARISON RESULTS BETWEEN PSO AND HPSO-SSM, WITH 30 DIMENSIONS.

F	PSO (Mean ± Std Dev)	HPSO-SSM (Mean ± Std Dev)
f_1	$1.67 \times 10^3 \pm 3.73 \times 10^3$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_2	$4.47 \times 10^2 \pm 4.45 \times 10^2$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_3	$3.04 \times 10^1 \pm 1.49 \times 10^1$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_4	$2.04 \times 10^4 \pm 1.01 \times 10^4$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_5	$2.92 \times 10^1 \pm 8.04 \times 10^0$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_6	$1.87 \times 10^4 \pm 3.57 \times 10^4$	$\mathbf{2.90 \times 10^1 \pm 1.53 \times 10^{-2}}$
f_7	$1.67 \times 10^3 \pm 3.73 \times 10^3$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_8	$2.24 \times 10^0 \pm 5.14 \times 10^0$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_9	$1.49 \times 10^1 \pm 5.26 \times 10^0$	$\mathbf{1.24 \times 10^1 \pm 9.40 \times 10^{-1}}$
f_{10}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{11}	$2.67 \times 10^1 \pm 2.16 \times 10^0$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{12}	$1.00 \times 10^0 \pm 1.52 \times 10^0$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{13}	$1.21 \times 10^1 \pm 3.07 \times 10^1$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{14}	$\mathbf{3.66 \times 10^1 \pm 5.19 \times 10^1}$	$5.77 \times 10^1 \pm 1.99 \times 10^0$
f_{15}	$3.02 \times 10^0 \pm 3.27 \times 10^0$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{16}	$2.91 \times 10^{-1} \pm 2.14 \times 10^{-1}$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{17}	$3.43 \times 10^2 \pm 1.06 \times 10^2$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{18}	$2.99 \times 10^0 \pm 8.96 \times 10^{-1}$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{19}	$\mathbf{1.59 \times 10^0 \pm 1.51 \times 10^0}$	$2.99 \times 10^0 \pm 2.03 \times 10^{-2}$
f_{20}	$9.69 \times 10^7 \pm 1.05 \times 10^8$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$
f_{21}	$0.00 \times 10^0 \pm 0.00 \times 10^0$	$0.00 \times 10^0 \pm 0.00 \times 10^0$
f_{22}	$1.48 \times 10^0 \pm 1.63 \times 10^0$	$\mathbf{9.30 \times 10^{-2} \pm 2.84 \times 10^{-1}}$
f_{23}	$3.63 \times 10^{-1} \pm 7.88 \times 10^{-2}$	$\mathbf{2.09 \times 10^{-1} \pm 1.97 \times 10^{-1}}$
f_{24}	$3.70 \times 10^1 \pm 9.96 \times 10^0$	$\mathbf{0.00 \times 10^0 \pm 0.00 \times 10^0}$

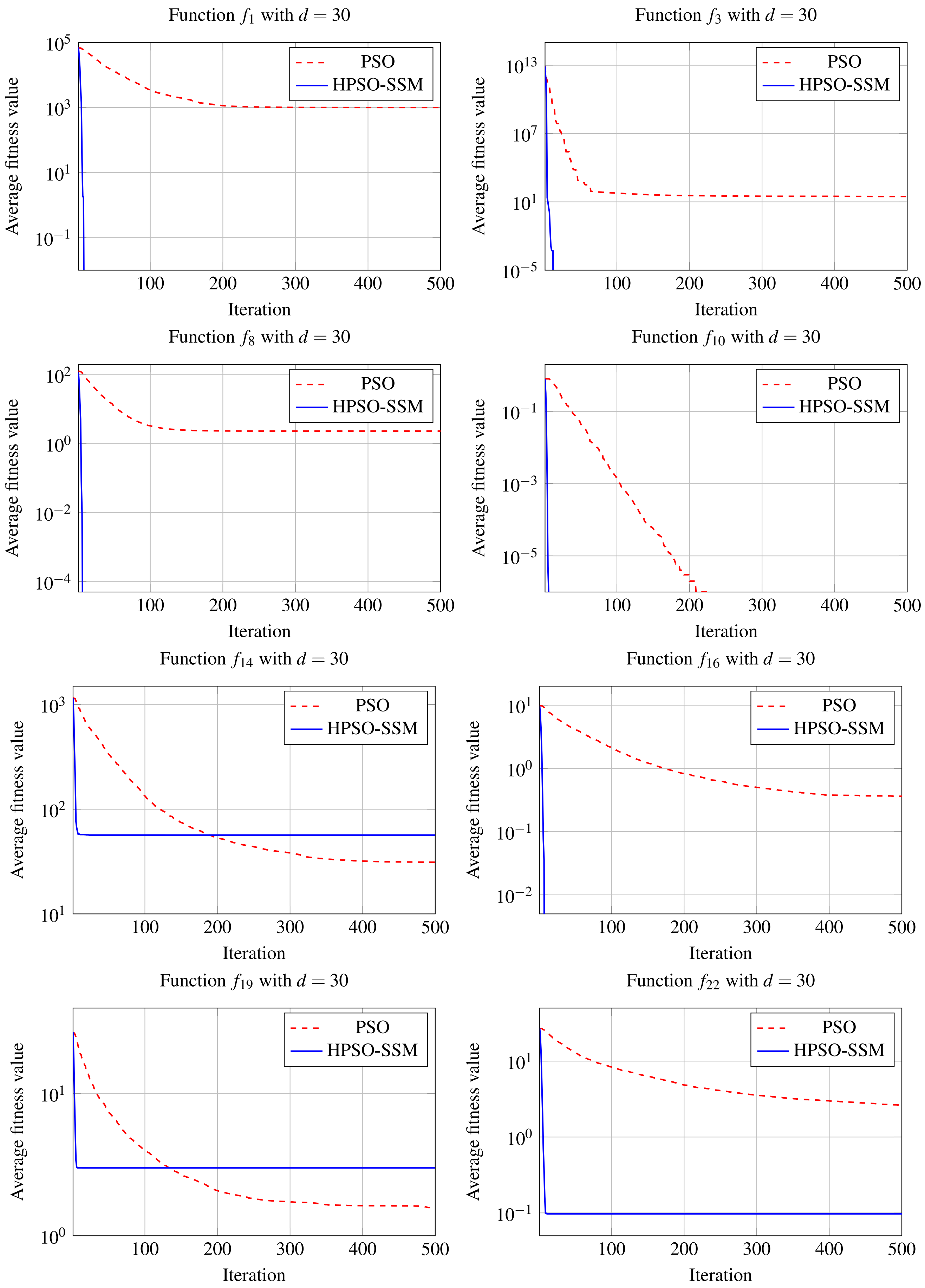

Fig. 2: Convergence curves of PSO and HPSO-SSM on eight benchmark functions, with 30 dimensions.

TABLE 3: RESULTS OBTAINED OVER 30 INDEPENDENT RUNS ON 100, 500, 1000, AND 5000 DIMENSIONS.

Function	Mean ± Std Dev ($d = 100$)	Mean ± Std Dev ($d = 500$)	Mean ± Std Dev ($d = 1000$)	Mean ± Std Dev ($d = 5000$)
f_1	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_2	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_3	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_4	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_5	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	**$6.65 \times 10^{0} \pm 2.49 \times 10^{1}$**	**$1.33 \times 10^{1} \pm 3.40 \times 10^{1}$**
f_6	**$9.90 \times 10^{1} \pm 2.56 \times 10^{-2}$**	**$4.99 \times 10^{2} \pm 2.99 \times 10^{-2}$**	**$9.99 \times 10^{2} \pm 3.07 \times 10^{-2}$**	**$5.00 \times 10^{3} \pm 1.50 \times 10^{-1}$**
f_7	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_8	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_9	**$4.60 \times 10^{1} \pm 2.16 \times 10^{0}$**	**$2.40 \times 10^{2} \pm 3.97 \times 10^{0}$**	**$4.90 \times 10^{2} \pm 7.18 \times 10^{0}$**	**$2.47 \times 10^{3} \pm 1.32 \times 10^{1}$**
f_{10}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	**$3.44 \times 10^{-1} \pm 8.77 \times 10^{-1}$**	**$8.90 \times 10^{-1} \pm 1.78 \times 10^{0}$**
f_{11}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{12}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{13}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{14}	**$1.99 \times 10^{2} \pm 3.53 \times 10^{0}$**	**$9.98 \times 10^{2} \pm 1.91 \times 10^{-1}$**	**$1.99 \times 10^{3} \pm 3.08 \times 10^{1}$**	**$1.04 \times 10^{4} \pm 1.44 \times 10^{3}$**
f_{15}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{16}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{17}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{18}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{19}	**$9.99 \times 10^{0} \pm 1.41 \times 10^{-2}$**	**$5.00 \times 10^{1} \pm 2.80 \times 10^{-3}$**	**$1.00 \times 10^{2} \pm 5.22 \times 10^{-3}$**	**$5.00 \times 10^{2} \pm 8.51 \times 10^{-3}$**
f_{20}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{21}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{22}	**$7.74 \times 10^{-2} \pm 2.92 \times 10^{-1}$**	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$
f_{23}	**$1.36 \times 10^{-1} \pm 2.10 \times 10^{-1}$**	**$1.86 \times 10^{-1} \pm 2.30 \times 10^{-1}$**	**$3.14 \times 10^{-1} \pm 2.24 \times 10^{-1}$**	**$3.30 \times 10^{-1} \pm 2.33 \times 10^{-1}$**
f_{24}	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$

occasions (i.e., f_{10}, and f_{21}). Nevertheless, it did display marginally better results than the HPSO-SSM method for only two functions, for which neither the PSO nor the HPSO-SSM were able to reach the theoretical optima (i.e., f_{14} and f_{19}).

This results can be further illustrated by the convergence curves of both algorithms, for eight selected benchmark functions, displayed in Fig. 2. It can be seen that, for the HPSO-SSM method, the convergence is much faster, and there is no apparent tendency towards local optima for most functions. For this benchmark, these partial results display the advantages of proper control over the inertia weight parameter, and that the modified position update strategies can yield a faster search.

b. Scalability over high dimensions

In order to verify the scalability of the HPSO-SSM method, we ran tests over a higher number of dimensions ($d = \{100, 500, 1000, 5000\}$). The mean and standard deviation of such experiment is shown in Table 3. It could be observed that, for most of the benchmark functions, the HPSO-SSM method was able to reach the global optima for any number of evaluated dimensions. Although for five of the benchmark functions, the method was not able to reach global optima (i.e, f_6, f_9, f_{14}, f_{19}, and f_{23}). Besides that, for two of the benchmark functions (i.e. f_5 and f_{10}), the algorithm was only able to reach the theoretical optima for dimension 100 and 500, with decreasing effectiveness on higher dimensions.

It should also be noted that, only in the case of function f_{22}, the method was only unable to reach the global optima for dimension 100, but performed better on the ensuing numbers of dimensions. To better illustrate the changes in convergence speed and robustness over increasing dimensions, the convergence values of four selected benchmark functions, for the first 50 iterations, are displayed in Fig. 3.

We can observe that, in Function f_{16}, for which the algorithm was always able to reach global optima, the convergence speed is about the same for all dimensions. As for the functions, for which the algorithm was unable to ever reach global optima (i.e. f_{14} and f_{19}), the convergence speed is significant at first, however, once an apparent local optima is reached, the search is unable to escape from it, in the remaining iterations. On a function for which the algorithm was only able to reach the theoretical optima for lower dimensional sizes (i.e. f_{10}), the convergence becomes more sensitive to local optima as the dimension increases.

We also noted that, for most functions, the convergence behavior of the algorithm remains the same, given a set of control parameters, regardless of the number of dimensions. Nonetheless, some changes in convergence behavior was only observed for two functions (i.e. f_5 and f_{10}). This suggests that, once the control parameters have been fine-tuned for a specific function, the algorithm could be able to reach a similar convergence speed for any number of dimensions.

The observed changes in the convergence behavior over higher dimensional functions suggests different approaches for control parameter tuning. For example, one could perform partial executions of the algorithm, with gradual increases in dimension, in order to find the tuned parameters. A procedure like this could probably minimize the number of iterations in order to find optimized control parameters.

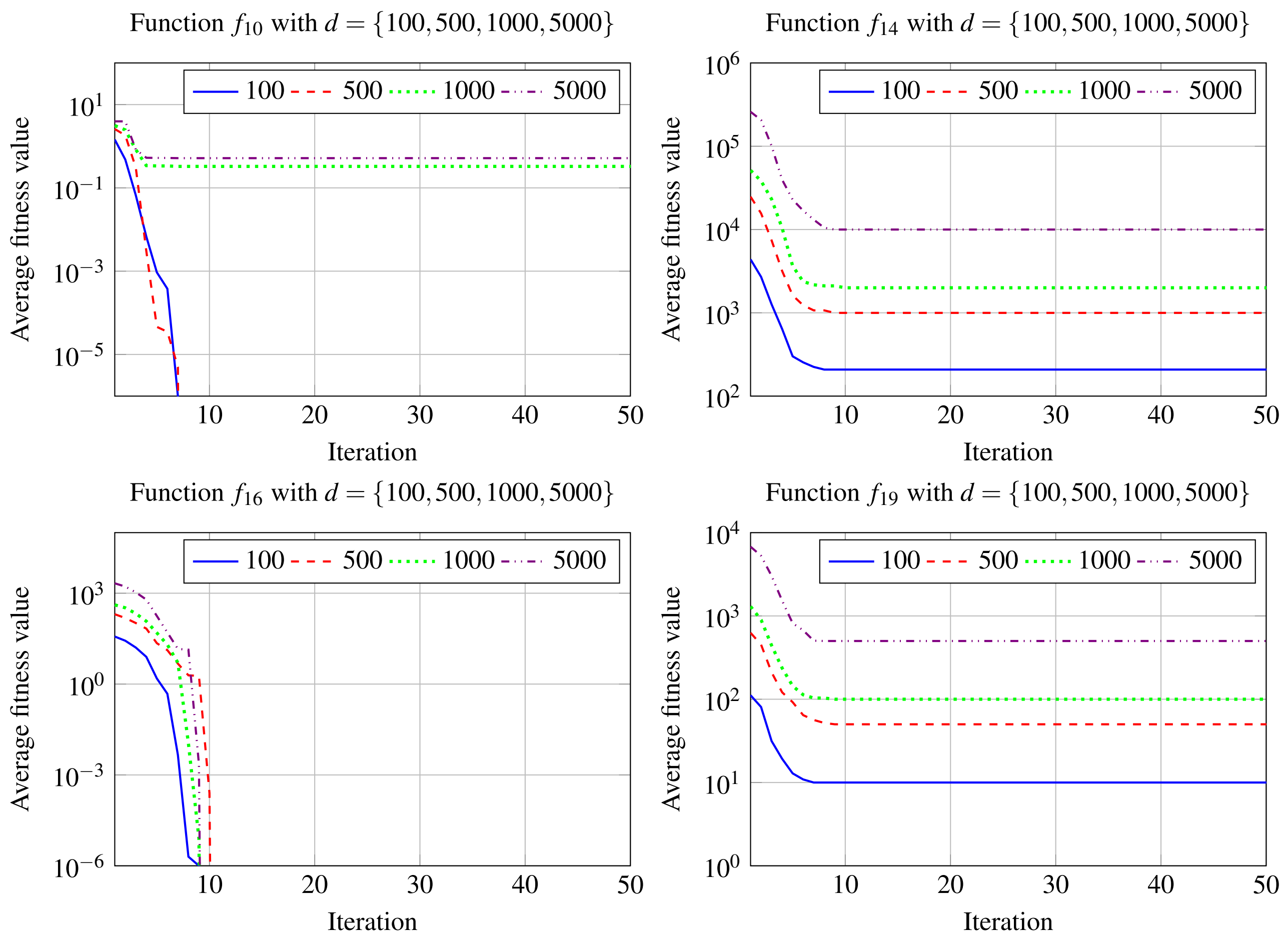

Fig. 3: Convergence curves of HPSO-SSM on four benchmark functions for the first 50 iterations, with 100, 500, 1000, and 5000 dimensions.

IV. Final Considerations

The hybrid particle swarm optimization with a spiral-shaped mechanism (HPSO-SSM), a variant of the particle swarm optimization (PSO), has been introduced, and its use on solving global optimization problems with a high number of dimensions was demonstrated. The efficiency of the considered method was tested against a set of 24 benchmark minimization functions, and the results were compared against those of the traditional PSO algorithm.

We have determined that the effectiveness of the proposed method surpassed that of the traditional algorithm by orders of magnitude, for most of the analyzed functions, both in the achieved fitness values and in the speed of convergence. The experiment results demonstrated that, for a considerable number of functions, the HPSO-SSM algorithm displays similar convergence behavior and speed, even after considerable increments in the number of dimensions. As future works, a meta-optimization process of control parameters could be investigated in order to get the best performance of HPSO-SSM.

References

[1] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, ser. Lecture Notes in Computer Science. Berlin, etc.: Springer-Verlag, 1987, vol. 268.

[2] R. Eberhart and Y. Shi, “Particle swarm optimization: developments, applications and resources,” in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 1, 2001, pp. 81–86.

[3] K. Chen, F.-Y. Zhou, and X.-F. Yuan, “Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection,” *Expert Systems with Applications*, vol. 128, pp. 140–156, 2019.

[4] W. Long, T. Wu, X. Liang, and S. Xu, “Solving high-dimensional global optimization problems using an improved sine cosine algorithm,” *Expert Systems with Applications*, vol. 123, pp. 108–126, 2019.

[5] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[6] Google. (2019) Colaboratory: Frequently asked questions. Retrieved from https://research.google.com/colaboratory/faq.html (12/02/2019).