UDC 621.3 https://doi.org/10.20998/2074-272X.2025.5.07

S. Zdiri, M. Moulahi, F. Messaoudi, A. Zaafouri

# Takagi-Sugeno fuzzy model identification using improved multiswarm particle swarm optimization in solar photovoltaics

***Introduction***. *The particle swarm optimization (PSO) algorithm has proven effective across various domains due to its efficient search space exploration, ease of implementation, and capability to handle high-dimensional problems. However, it is often prone to premature convergence, which limits its performance.* ***Problem***. *This issue becomes critical in identifying Takagi-Sugeno (T-S) fuzzy models, especially in complex systems like solar photovoltaic (PV) applications, where model accuracy is vital for tasks such as maximum power point tracking (MPPT) and shading compensation.* ***Goal.*** *This manuscript introduces an improved multiswarm PSO (I-MsPSO), designed to enhance search performance and robustness in identifying T-S fuzzy systems. The method is particularly suited to nonlinear modeling challenges in renewable energy systems.* ***Methodology***. *I-MsPSO divides the swarm into 4 independent subswarms, each operating in a local region with specific inertia weights and acceleration coefficients. Periodic information sharing between subswarms allows the algorithm to converge collectively toward optimal solutions. A new modeling approach, specific Takagi-Sugeno modeling (STaSuM), is introduced, using I-MsPSO to determine both the structure and parameters of T-S fuzzy systems.* ***Results.*** *The I-MsPSO's performance was tested on benchmark optimization problems and real-world engineering cases. Results show that STaSuM produces highly accurate and generalizable fuzzy models, outperforming existing techniques.* ***Scientific novelty*** *lies in the development of I-MsPSO, which enhances the traditional PSO by using 4 interactive subswarms with customized parameters, and the creation of STaSuM for advanced T-S fuzzy system identification.* ***Practical value***. *I-MsPSO and STaSuM provide a powerful optimization and modeling framework, offering robust and accurate solutions for nonlinear and dynamic environments. Their structure makes them especially valuable for future applications in MPPT control, fault-tolerant modeling, and real-time optimization in PV energy systems.* References 39, table 5, figures 8.

*Key words:* **improved multiswarm particle swarm optimization, particle swarm optimization, specific Takagi-Sugeno modeling.**

***Вступ***. *Алгоритм оптимізації рою часток (PSO) довів свою ефективність у різних галузях завдяки ефективному дослідженню простору пошуку, простоті реалізації та здатності вирішувати завдання високої розмірності. Однак він часто схильний до передчасної збіжності, що обмежує його продуктивність. Ця* ***проблема*** *стає критично важливою при ідентифікації нечітких моделей Такагі-Сугено (T-S), особливо у складних системах, таких як сонячні фотоелектричні системи (PV), де точність моделі є критично важливою для таких завдань, як відстеження точки максимальної потужності (MPPT) та компенсація затінення.* ***Мета***. *У роботі представлений удосконалений багатороєвий PSO (I-MsPSO), розроблений для підвищення продуктивності пошуку та надійності при ідентифікації нечітких систем T-S. Цей метод особливо підходить для задач нелінійного моделювання у системах відновлюваної енергії.* ***Методологія***. *I-MsPSO ділить рій на 4 незалежні подрої, кожен з яких працює в локальній області з певними вагами інерції та коефіцієнтами прискорення Періодичний обмін інформацією між подроями дозволяє алгоритму колективно сходитися до оптимальних рішень. Наведено новий підхід до моделювання, специфічне моделювання Такагі-Сугено (STaSuM), з використанням I-MsPSO для визначення структури та параметрів нечітких систем T-S.* ***Результати***. *Продуктивність I-MsPSO протестована на еталонних задачах оптимізації та реальних інженерних прикладів. Результати показують, що STaSuM створює високоточні та узагальнені нечіткі моделі, що перевершують існуючі методи.* ***Наукова новизна*** *полягає в розробці I-MsPSO, який розширює традиційний PSO за рахунок використання 4 інтерактивних подроїв з параметрами, що настроюються, а також у створенні STaSuM для розширеної ідентифікації нечітких систем T-S.* ***Практична цінність***. *I-MsPSO та STaSuM надають потужну платформу оптимізації та моделювання, пропонуючи надійні та точні рішення для нелінійних та динамічних середовищ. Їхня структура робить їх особливо цінними для майбутніх додатків у галузі управління MPPT, відмовостійкого моделювання та оптимізації в реальному часі у PV енергетичних системах.* Бібл. 39, табл. 5, рис. 8.

*Ключові слова:* **покращена оптимізація рою часток з кількома роями, оптимізація рою часток, специфічне моделювання Такагі-Сугено.**

**Introduction.** The Takagi-Sugeno (T-S) type fuzzy model, used to model complex systems particularly in the area of fuzzy logic and control systems, was first introduced by Mamdani and Assilian [1]. It was later enhanced by Takagi and Sugeno who developed the T-S type model. In the new approach, fuzzy linguistic rules were replaced by more precise mathematical rules. Historically, a fuzzy model is described by a formalism based on fuzzy rules, providing a prolific framework to study nonlinear dynamic systems and, particularly, to analyze their stability and synthesize laws control (stabilization). From a conceptual point of view, a fuzzy system is identified by determining the structure of the model (the premise parameters) and estimating the consequent parameters [2]. The first step is performed employing identification methods based on coalescence or even fuzzy classification (fuzzy clustering algorithms). Fuzzy coalescence algorithms are also applied to identify nonlinear systems using to the T-S model. In the literature, numerous algorithms derived from the fuzzy c-mean algorithm, such as the Gustafson-Kessel algorithm [3], the fuzzy C-means algorithm [4], the Gath-Geva algorithm [5], were proposed. After determining the premise parameters of the model, the consequent parameters of the fuzzy rules are estimated. Among the identification techniques proposed in the literature we cite: the graph kernel recursive least-squares algorithms [6], weighted least squares method [7], the orthogonal least squares algorithm [8]. Several works showed that fuzzy coalescence algorithms derived from fuzzy c-means are sensitive to initialization. In fact, random initialization can generally lead to convergence towards a local minimum of the objective function. The problem of synthesizing fuzzy systems was treated by many researchers, as an optimization problem, whose resolution is reduced to the search for the optimal solutions (fuzzy models), in order to satisfy the performance criteria and the predefined constraints. In recent years, researchers have used several algorithms to optimize the structures and parameters of the T-S model. For instance, particle swarm optimization (PSO) has been utilized in many applications [9, 10] given the small number of parameters to adjust, its easy implementation, rapid convergence and

its ability to produce high-quality solutions within a shorter calculation time. The combination of T-S fuzzy systems and PSO algorithms offers a powerful and flexible approach to solve a wide range of optimization problems. By exploiting the strengths of each technique, this approach allows developing more accurate, robust and interpretable models. However, the PSO is easily trapped in a local minimum and it is difficult to guarantee that the fuzzy models obtained will have good performance and the optimized fuzzy model largely depends on the performance of this algorithm.

To deal with these weaknesses, numerous improved versions of PSO and several hybrid methods were suggested [11–13]. In [14], the PSO algorithm was implemented to optimize the 5 parameters of PID controller applying El-Khazali's approach in order to minimize several error functions, satisfying some step response specifications such as the set of time domain and frequency domain constraints. In [15], the population was divided into many small swarms, different grouping strategies were used and the exchange between various small swarms improved the population diversity. Work [16] proposed a dynamic multiple swarms to solve multiobjective problems applying 2 main strategies: the swarm growth strategy and the swarm decay strategy. Besides, in [17] a methodology to automatically extract fuzzy T-S models from data using PSO was developed. In their approach, the parameters and the structures of fuzzy models were encoded in a particle and evolved together to obtain simultaneously the optimal structure and parameters. A new method, where the population was divided into 4 subswarms and heterogeneous search strategies were used to accomplish the optimization task, was applied in [18]. In this method a new strategy was applied under the so-called OptiFel to extract the structure and parameters of the T-S model. In the multiswarm PSO (MsPSO) algorithm used a homogeneous search strategy for all particles and in each subswarm, which reduced the convergence rate. In [19] authors suggested a novel cooperation strategy C-MsPSO based on the distribution of populations into 4 subswarms; each of which used inertia weight parameters and specific acceleration coefficients. This strategy allowed minimizing the risk of trapping the algorithm by the local optima.

In this article, an optimization algorithm, called improved multiswarm particle swarm optimization (I-MsPSO), is introduced to identify fuzzy T-S type models. In I-MsPSO, the population is divided into 4 subswarms; each of which ensures the internal search strategy relying on inertia weight parameters and specific acceleration coefficients. A new parameter search strategy applied by the fuzzy system T-S, called specific Takagi-Sugeno modeling (STaSuM), is also presented with the I-MsPSO algorithm to improve the search performance and ensure that the resulting fuzzy models will be highly efficient. The main contributions of this paper are:

- dividing the population into 4 subcooperative swarms in I-MsPSO algorithm.
- In this algorithm, each subswarm utilizes specific parameters (the subswarm S1 employs sigmoid inertia weights and constant acceleration coefficients, while subswarm S2 uses linear varying inertia weights and constant acceleration coefficients and subswarm S3 employs adaptive inertia weights and the coefficient of the constant accelerations).
- Determining the structure and parameters of the fuzzy models, coded in a particle, in STaSuM.

**Preliminaries. Optimization problem.** An optimization problem is defined by an objective function, a set of variables, a set of equality (or inequality) constraints and a search space formed by the set of the possible problem solutions where each dimension corresponds to a variable. Depending on the type of the problem to be solved, the best solution consists in finding an extreme value, also called extremum (i.e. the minimum or maximum of this objective function). In fact, an optimization problem corresponds to solving the following problem: min/max (function) under the constraint [20, 21]. It can be mono-objective or multiobjective (several objective functions must be optimized), static or dynamic (the objective function changes over time) and with or without constraints. In the literature many methods, such as Newton's method [22], linear programming methods [23], the simplex method and the gradient method [24] were introduced to obtain the optimal solution of the optimization problem in a reasonable time. They require that the objective function should have a minimum of characteristics such as convexity, continuity or differentiability.

**PSO algorithm**. PSO is a non-specific heuristic optimization algorithm like evolutionary algorithms, tabu search or ant colonies [25–27]. Its convergence speed also makes it efficiently used in dynamic optimization. Due to its multiple advantages, such as a rapid convergence, ease of implementation and wide search range, PSO has been employed in a variety of research fields and applications. The swarm's particles are initially dispersed randomly over the search space, where each particle has a random displacement position and speed. Thereafter, the algorithm can, at each instance, access its current position; memorize the best solution; communicate with neighboring particles; obtain, from each of them, its best performance; and modify its speed according to better solutions. The displacement of a particle between iteration $t$ and iteration $t$+1 is formulated analytically by the following velocity (1) and position relations (2):

$$v_i(t+1) = wv_i(t) + c_1r_1[x_{pbest} - x_i(t)] + c_2r_2[x_{gbest} - x_i(t)]; \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (2)$$

where $x_{pbest}$ is the best position determined by the $i^{th}$ particle; $w$ is a constant called the inertia weight; $c_1$, $c_2$ are the acceleration coefficients while $r_1$ and $r_2$ are randomly generated by a uniform distribution in [0, 1]; $x_{gbest}$ is the best overall position found by the population.

Thus, to make its next move, each particle applies the following steps:

- follow its speed;
- return to its best performance;
- move towards the best performance of its neighbor.

**T-S fuzzy model.** Although several fuzzy models were introduced in the literature and the most commonly used ones are: Mamdani type model, Takagi-Sugeno-Kong type model and T-S type model. The main difference between these models lies in their consequent part. In fact, a fuzzy model is based on the linguistic partitioning of the values of its variables. The input (premises variables) and output (consequent variables) values are described by fuzzy sets having membership functions. In the fuzzy model of the T-S, the premises of the rules are formulated symbolically and the conclusions

are expressed by linear functions [28, 29]. They are generally written in the following form:

$$R^i: \text{if } x_1 \text{ is } A_1^i, x_2 \text{ is } A_2^i, ..., x_n \text{ is } A_n^i \text{ then}$$
$$y_i = w_i^T x + b_i, i = 1,2,...,m, \quad (3)$$

where $x = x_1, x_2, \ldots, x_n$ is the input variable; $n$ is the dimension of the input variable $x$; $i = 1, 2, \ldots, m$ is the number of fuzzy rules; $w = w_1, w_2, \ldots, w_n$ are the consequent parameters; $A_i$ is the fuzzy set; $y_i$ is the output of fuzzy rule.

The output of the fuzzy model can be calculated by a weighted mean defuzzification, as shown below:

$$y = \sum_{i=1}^{m} \mu_i y_i \Big/ \sum_{i=1}^{m} \mu_i , \quad (4)$$

where the weight strength $\mu_i$ of the $i^{\text{th}}$ rule is computed as:

$$\mu_i(x) = \prod_{j=1}^{n} h_i(x), \quad (5)$$

where the $\mu_i(x)$ is the grade of membership function. Subsequently, the $c_{ij}$ and $\sigma_{ij}$ parameters of the Gaussian function specified by (6) and the parameters of the fuzzy rule of T-S model are calculated applying the fuzzy rule (3):

$$h_i(x) = \exp\left(\left(x_i - c_{ij}\right)^2 / \sigma_{ij}\right). \quad (6)$$

**Improved multiswarm particle swarm optimization (I-MsPSO). General concept of the improved multiswarm particle swarm optimization.** I-MsPSO algorithm divides the population into 4 subswarms to address the issue of premature convergence and to ensure proper exploration and exploitation of the research processes, which improves its capacity for search, communication and cooperation between subswarms. Every subswarm executes a single PSO, including updating speed and position of the particles, in accordance with the exact equations. In this study, a unique algorithm that relies on 4 subswarms and several techniques of inertia weights and acceleration coefficients is implemented to enhance the exploration and exploitation performance of the standard MsPSO. I-MsPSO algorithm is based on constant accelerations coefficients values, time-varying inertia weight, sigmoid inertia weight and adaptive inertia weight value (Fig. 1). Additionally, the $i^{\text{th}}$ particle in subswarm S3 is adjusted based on the fitness values and velocities of the particles in the base subswarms. Meanwhile, the $i^{\text{th}}$ particle in subswarm S4 updates its velocity in accordance with the velocities of the particles in subswarms S1, S2 and S3. Figure 2 describes the mechanism of exploring the new region. I-MsPSO algorithm enhances PSO by dividing the population into 4 subswarms, each with specific inertia weights and acceleration coefficients, and by implementing periodic information exchange among the subswarms.

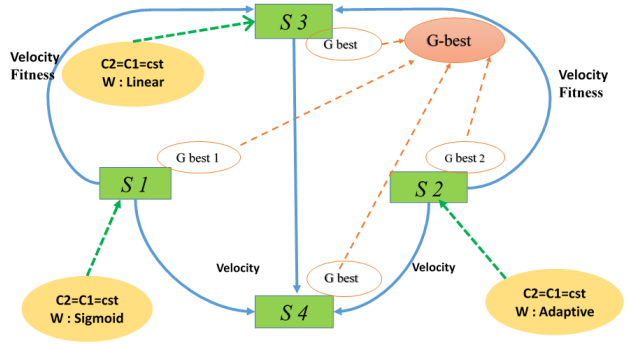


Fig. 1. Communication model in I-MsPSO

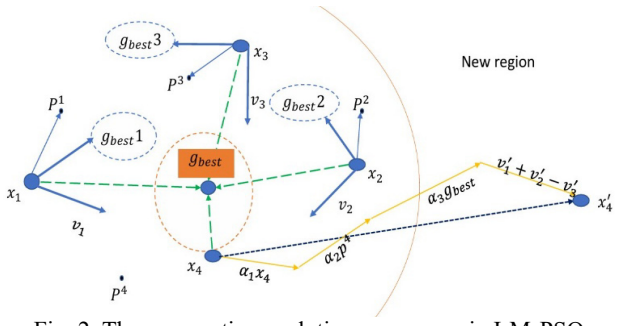


Fig. 2. The cooperative evolutionary process in I-MsPSO

I-MsPSO algorithm can be summarized as follows:

| Algorithm 1. Pseudo-code for I-MsPSO algorithm |
|---|
| Begin. |
| Initialize the particle size of each subswarm. |
| Initialize the positions and velocities of all particles in the search space. |
| Initialize the global best position of each swarm. |
| Find the best local position ($P_{best}$) in each subswarm and the best global position ($G_{best}$). |
| Do in parallel until the maximum number iteration reached. |
| Calculate the velocity of each particle in subswarm S1, S2, S3 and S4. |
| Update the position of each particle in subswarm S1, S2, S3 and S4. |
| Evaluate the fitness of the $i^{\text{th}}$ particle. |
| Update the global best of the swarms. |
| If the guide condition is satisfied. |
| Apply diversity guided convergence strategy to the current particle in each subswarm. |
| End Do. |
| Return the best solution of the algorithm. |
| End. |

**Convergence of the I-MsPSO**. The particle paths, the convergence of I-MsPSO algorithm and the particle velocity of each subswarm are theoretically investigated. According to the following limit, convergence is defined as:

$$\lim_{t \to +\infty} x_i(t) = p_i , \quad (7)$$

where $p_i$ is the local or global optimum; $x_i$ is the location of the $i^{\text{th}}$ particle at time $t$. If I-MsPSO is applied with the adaptive inertia weight, the sigmoid inertia weight and the linear varying inertia weight, the velocity and the position update their equations using (1) and (2). Therefore, the following I-MsPSO system is obtained:

$$x(t+1) = F \cdot x(t) + R \cdot E, \quad (8)$$

where $R$ is the constant matrix; $F$ is the system matrix:

$$x(t) = [x_t^1, x_t^2, x_t^3, x_t^4, x_{t-1}^1, x_{t-1}^2, x_{t-1}^3], \quad (9)$$

and

$$E = [p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)}, g]^T , \quad (10)$$

designates the vector containing 4 local solutions and a single global solution. The used symbols are listed below:

$$\begin{cases} \varphi_1 = v_{11} + v_{12} = c_1 r_{11} + c_2 r_{12} \\ \varphi_2 = v_{21} + v_{22} = c_1 r_{21} + c_2 r_{22} \\ \varphi_3 = v_{31} + v_{32} = c_1 r_{31} + c_2 r_{32} \end{cases} , \quad (11)$$

where $r_{ij}$ is the random number in [0; 1]; $j = 1; 2$ and $i = 1; 2; 3$. The equation applied to obtain the position of a particle in S1 is:

$$x_{t+1}^{(1)} = x_t^{(1)} + v_{t+1}^{(1)} = x_t^{(1)}(1 + w_s - \varphi_1) - w_s x_{t-1}^{(1)} + v_{11} p_t^{(1)} + v_{22} g_t , \quad (12)$$

where $w_s$ is the sigmoid inertia weight:

$$w_s = \frac{0.9 \cdot numiter - 0.5i}{numiter}, \quad (13)$$

where *numiter* is the maximum number of the iterations; *i* is the current iteration.

The equation used to calculate the position of a particle in S2 is:

$$x_{t+1}^{(2)} = x_t^{(2)} + v_{t+1}^{(2)} = x_t^{(2)}(1 + w_a - \varphi_2) - w_a x_{t-1}^{(2)} + v_{11} p_t^{(2)} + v_{22} g_t, \quad (14)$$

where $w_a$ is the adaptive inertia weight:

$$w_a(i) = w_{\max} + (w_{\max} - w_{\min}) \cdot \left( \frac{e^{x(i)} - 1}{e^{x(i)} + 1} \right), \quad (15)$$

where $x_i = (g_{best} - i) / (g_{best} + i)$; $w_{\min}$, $w_{\max}$ are the initial and the final values of inertia weight; *i* is the current iteration.

In the subswarm S3, the equation applied to obtain the position of the particles is:

$$x_{t+1}^{(3)} = x_t^3 + w_L \left[ \frac{\gamma}{\gamma_1} v_{t+1}^{(1)} + \frac{\gamma}{\gamma_2} v_{t+1}^{(2)} + v_t^3 \right] + \\ + v_{31} \left[ p_t^{(3)} - x_t^3 \right] + v_{32} \left[ g_t - x_t^{(3)} \right] \quad (16)$$

where $w_L$ is the linear varying inertia weight represented by the following equation; $\gamma = \gamma_1 + \gamma_2$; $\gamma_1$, $\gamma_2$ are the fitness values of the particles in the subswarm S1 and S2. As $r_1$ and $r_2$, $r_3$ and $r_4$ are vectors of the random numbers.

$$w_L(i) = \left( \frac{numiter - 1}{numiter} \right) (w_{\min} - w_{\max}) + w_{\max}. \quad (17)$$

The equation employed to compute the location of a particle in S4 is:

$$x_{(t+1)}^{(4)} = \alpha_1 x_t^{(4)} + \alpha_2 p_t^{(4)} + \alpha_3 g_t + v_{(t+1)}^{(4)}. \quad (18)$$

In this study, 3 impact factors ($\alpha_1$–$\alpha_3$) are used to determine the influence of the past information on the current position of the particles within subswarm S4. They are constrained by this expression $\alpha_1 + \alpha_2 + \alpha_3 = 1$. In the performed analysis, the following values were assigned to the impact factors: $\alpha_1 = 1/6$, $\alpha_2 = 1/3$, $\alpha_3 = 1/2$. The bigger $\alpha_i$ ($i$ = 1, 2, 3) was the larger the effect of the previous information on the current search would be, and vice versa. The larger effect of the previous information on the current search will be, and vice versa. As shown in (18), the different impact factors regulate the effect of the historical data on the particle's location within S4.

In a convergence analysis, researchers observed that particles within each subswarm converge towards stable positions defined by the limits presented in (19) – (22):

$$\lim_{t \to +\infty} x^{(1)} = \frac{c_1}{2} p^{(1)} + \frac{c_2}{2} g_{best}, \quad (19)$$

$$\lim_{t \to +\infty} x^{(2)} = \frac{c_1}{2} p^{(2)} + \frac{c_2}{2} g_{best}, \quad (20)$$

$$\lim_{t \to +\infty} x^{(3)} = wc_1 (p^{(1)} + p^{(2)}) + \frac{c_1}{2} p^{(3)} + g_{best} (2wc_2 + \frac{c_2}{2}), \quad (21)$$

$$\lim_{t \to +\infty} x^{(4)} = \frac{(1-w)c_1}{2} (p^{(1)} + p^{(2)}) - \frac{c_1}{2} p^{(3)} + \\ + \alpha_2 p^{(4)} + g_{best} ((1-2w)c_2 - \frac{c_2}{2} + \alpha_3). \quad (22)$$

**Methodology of STaSuM based on I-MsPSO.** A new parameter search strategy applied by the fuzzy system T-S, called STaSuM, is also presented with the I-MsPSO algorithm to improve the search performance. STaSuM framework for T-S fuzzy model identification is presented in this section. The structure and parameters of T-S fuzzy model are all encoded in a particle. The following sections provide the details.

**Particle mapping and objective function.** In the identification process, the structure and parameters of the fuzzy model were all coded in a particle of the I-MsPSO algorithm, and the mean square error (MSE) value was used to choose the best local position in a swarm and the global optimum in a population. A single nest in the I-MsPSO algorithm is shown in Fig. 3. Each particle is specified as a vector corresponding to a particular fuzzy model and each vector corresponds to a fuzzy rule made up of the premise parameters (structure) and the consequent parameters.

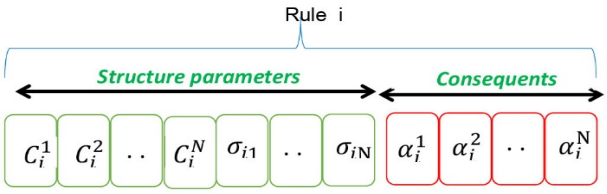


Fig. 3. $i^{th}$ rule encoded in a particle. The code consists of 2 necessary items: structure and consequent parameters

To create an accurate mathematical model, an objective function was applied to measure the difference between the output of the model and that of the actual process. MSE was utilized to measure the difference between the output of the model and the real value. MSE was mathematically formulated as:

$$MSE = \frac{1}{N} \sum_{k=1}^{n} (y_k - \hat{y}_k)^2, \quad (23)$$

where $N$ is the number of observations; $y_k$ is the actual output; $\hat{y}_k$ is the output estimated by the model.

**Implementation of STaSuM.** Algorithm 2 describes the implementation of STaSuM. The rules of the T-S fuzzy model were encoded in the particle $x^s$ and the MSE value was utilized to select the best solution in the subswarm. The best overall structure was obtained from the 4 optimal $g_{best}$ in the subswarms $s$ = 1, 2, 3, 4.

| Algorithm 2. The STaSuM algorithm |
|---|
| 1. Initialization. |
| (a) Set the number of iterations and rules s. |
| (b) Specify the size of each subswarm. |
| (c) Initialize the position of particle. |
| (d) Initialize the position of particle. |
| (e) Determine the global best nest g. |
| 2. Set the number of rules as constant. |
| 3. Termination check. |
| (a) If the termination criterion holds stop. |
| (b) Else go to Step 4. |
| 4. For do. |
| (a) Update the position $x_k^s$ according to Equations 12, 14, 16, 18, respectively. |
| (b) Update the velocity $v_k^s$ according to Equation 1. |
| (c) Evaluate the fitness of the $i^{th}$ particle $f(x_k^s)$ |
| (d) If the $f(x_k^s)$ is better than $f(p_k^s)$, then $p_k^s = s_k^s$ |
| End For |
| Update $gbest = \arg\{\min f\{p_{best}\}\}$. |
| End For |
| Update $gbest = \arg\{\min f\{g_{best}\}\}$. |
| Set $t = t + 1$ |
| 5. Go to step 3. |

**Simulation result and discussion.** This section is composed of 2 parts. In the initial part, the performance of the I-MsPSO algorithm is evaluated through numerical experiments using benchmark functions and engineering problems. On the other hand, the subsequent part shows the effectiveness of the STaSuM method applied in nonlinear systems.

**Convergence analysis of I-MsPSO.** To validate the performance and efficiency of the proposed algorithm, 12 benchmark functions from the CEC-2017 test set were selected to test the I-MsPSO algorithm [30]. The upper limit for optimization processes was set at 1000, and each test run a maximum of 30 times. The experimental machine in use is equipped with a 3rd generation i3 processor running at 2.5 GHz with a storage capacity of 128 GB. The utilized programming language is MATLAB. The maximum number of iterations in the numerical experiences was set at 1000 on each of the 12 reference functions for each algorithm. Each experiment was carried out independently of the reference functions. The mean values and standard deviations are presented in Table 1. Figures 4–6 show the average convergence characteristics of each approach on the reference functions. A comparison of the proposed algorithm to common strategies is presented in this section. The performances of different strategies, including adaptive MsPSO (AMsPSO), linear time-varying (L-MsPSO) and standard MsPSO, were analyzed in the experiments carried out on 12 static problems.

Table 1

The results on the 12 benchmark functions of each algorithm

| Function | F1 | F2 |
|---|---|---|
| L-MsPSO | $4.97\cdot10^{-121}\pm9.90\cdot10^{-121}$ | $4.26\cdot10^{-60}\pm1.24\cdot10^{-59}$ |
| A-MsPSO | $5.23\cdot10^{-153}\pm5.23\cdot10^{-150}$ | $2.94\cdot10^{-79}\pm2.40\cdot10^{-74}$ |
| MsPSO | $4.01\cdot10^{-108}\pm7.80\cdot10^{-108}$ | $3.82\cdot10^{-54}\pm6.18\cdot10^{-54}$ |
| **I-MsPSO** | $\mathbf{2.83\cdot10^{-186}\pm2.52\cdot10^{-183}}$ | $\mathbf{7.14\cdot10^{-92}\pm6.12\cdot10^{-91}}$ |
| Function | F3 | F4 |
| L-MsPSO | $2.97\cdot10^{-119}\pm9.22\cdot10^{-118}$ | $3.02.10^{-60}\pm8.21\cdot10^{-60}$ |
| A-MsPSO | $1.60\cdot10^{-150}\pm1.60\cdot10^{-149}$ | $1.121\cdot10^{-76}\pm1.21\cdot10^{-73}$ |
| MsPSO | $9.30\cdot10^{-105}\pm2.92\cdot10^{-104}$ | $7.37\cdot10^{-53}\pm2.24\cdot10^{-52}$ |
| **I-MsPSO** | $\mathbf{1.03\cdot10^{-178}\pm1.051\cdot10^{-178}}$ | $\mathbf{1.33\cdot10^{-94}\pm1.01\cdot10^{-93}}$ |
| Function | F5 | F6 |
| L-MsPSO | $2.89\cdot10^{-01}\pm2.71\cdot10^{-01}$ | 0 |
| A-MsPSO | $2.89\cdot10^{-01}\pm2.00\cdot10^{-01}$ | 0 |
| MsPSO | $2.89\cdot10^{-01}\pm3.60\cdot10^{-02}$ | 0 |
| **I-MsPSO** | $\mathbf{2.89\cdot10^{-01}\pm0}$ | **0** |
| Function | F7 | F8 |
| L-MsPSO | $1.001\cdot10^{-81}\pm3.966\cdot10^{-81}$ | $3.995\cdot10^{-43}\pm7.59\cdot10^{-43}$ |
| A-MsPSO | $1.401\cdot10^{-81}\pm3.966\cdot10^{-81}$ | $5.601\cdot10^{-43}\pm8.00\cdot10^{-43}$ |
| MsPSO | $1.451\cdot10^{-81}\pm3.966\cdot10^{-81}$ | $3.005\cdot10^{-43}\pm8.78\cdot10^{-43}$ |
| **I-MsPSO** | $\mathbf{1.321\cdot10^{-81}\pm3.966\cdot10^{-81}}$ | $\mathbf{4.99\cdot10^{-43}\pm8.111\cdot10^{-44}}$ |
| Function | F9 | F10 |
| L-MsPSO | 0 | $4.665\cdot10^{-43}\pm8.52\cdot10^{-43}$ |
| A-MsPSO | 0 | $4.665\cdot10^{-43}\pm8.52\cdot10^{-43}$ |
| MsPSO | 0 | $4.665\cdot10^{-43}\pm8.52\cdot10^{-43}$ |
| **I-MsPSO** | **0** | $\mathbf{4.665\cdot10^{-43}\pm8.52\cdot10^{-43}}$ |
| Function | F11 | F12 |
| L-MsPSO | 0 | $3.65\cdot10^{-43}\pm8.3\cdot10^{-44}$ |
| A-MsPSO | 0 | $4.65\cdot10^{-40}\pm7.52\cdot10^{-44}$ |
| MsPSO | 0 | $4.115\cdot10^{-43}\pm7.51\cdot10^{-44}$ |
| **I-MsPSO** | **0** | $\mathbf{3.52\cdot10^{-43}\pm8.114\cdot10^{-42}}$ |

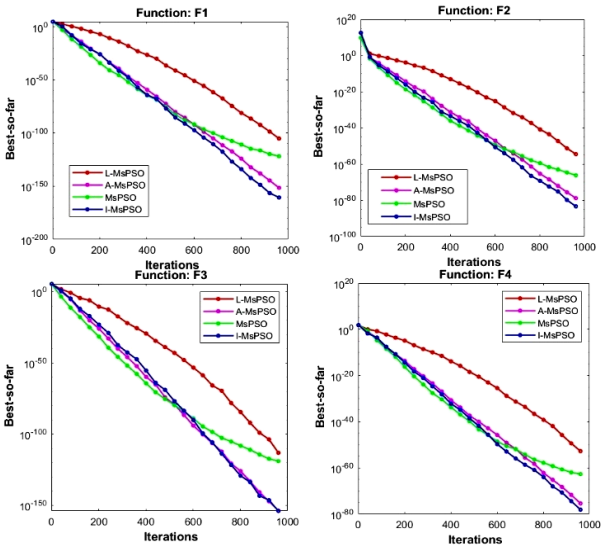


Fig. 4. Convergence characteristics on 4 reference functions (F1–F4) with 30 dimensions

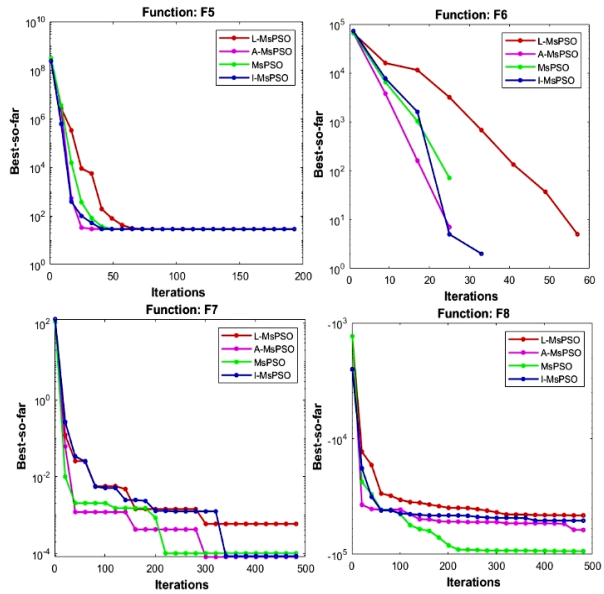


Fig. 5. Convergence characteristics on 4 reference functions (F5–F8) with 30 dimensions

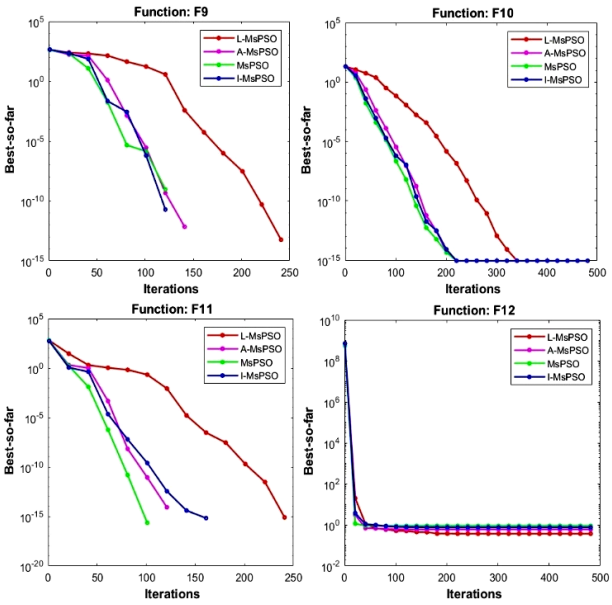


Fig. 6. Convergence characteristics on 4 reference functions (F9–F12)

The mean values and the standard deviation throughout the optimization runs are shown in Table 1. On the other hand, Fig. 4–6 outline the average convergence characteristics to each approach on the reference functions. Figures 4–6 demonstrate that I-MsPSO performs better than the other PSO variants on the benchmark functions. As exposed in Fig. 4–6 I-MsPSO reaches the target optima in the majority of benchmark functions (Table 1). The results obtained by I-MsPSO are superior to those of the other PSO versions on each benchmark function, with the exception of the Function F8 and F11.

**Application of I-MsPSO to engineering problems.** This part examines the I-MsPSO's effectiveness by extending its application to solve real-world engineering optimization problems. Specifically, the next section delves into the optimization of the tension/compression spring, while the after section summarizes the findings provided by pressure vessel design. The performance of I-MsPSO is, then, benchmarked against those of the existing algorithms such as PSO [31], genetic algorithm (GA) [32], velocity pausing particle swarm optimization (VPPSO) [33] and grey wolf optimizer (GWO) [34].

**Tension/compression spring design (TCSD).** TCSD problem is a classic engineering problem whose primary objective consists in minimizing the spring's weight. This optimization task requires finding the lightest possible spring while meeting specific design constraints, including limitations on shear stress, surge frequency and deflection. It ultimately translates into a minimization problem where the weight of the spring is minimized while adhering to all boundary and constraints conditions. The design variables include the wire diameter $d(X_1)$, the mean coil diameter $h(X_2)$, and the number of turns of the spring $P(X_3)$. The following subsection outlines the objective functions and the constraints associated with these three optimization variables. Consider:

$$X = [d, h, P] = [X_1, X_2, X_3]. \quad (24)$$

Minimize:

$$F(x) = X_1^2 X_2 (X_3 + 2). \quad (25)$$

Subject to:

$$g_1(X) = 1 - \frac{X_2^3 X_3}{71785 X_1^4} \le 0; \quad (26)$$

$$g_2(X) = \frac{4X_2^2 - X_1 X_2}{12566(X_1^3 X_2 - X_1^4)} + \frac{1}{5108 X_1^2} - 1 \le 0; \quad (27)$$

$$g_3(x) = 1 - \frac{140.45 X_1}{X_2^2 X_3} \le 0; \quad (28)$$

$$g_4(x) = \frac{X_1 + X_2}{1.5} \le 0. \quad (29)$$

Variable range: $0.05 \le X_1 \le 2$, $0.25 \le X_2 \le 1.3$, $2 \le X_3 \le 15$.

Table 2 illustrates the statistical results of the TCSD problem. Each algorithm was independently run 50 times, the maximum number of iterations and the population size were set to 1000 and 30, respectively. Overall, the I-MsPSO algorithm ranks first since it explores a solution to make the spring weight smaller for the TCSD problem. VPPSO offer similar solution, ranking second.

Table 2

Optimal solutions of tension/ compression spring design problem optimized by different algorithms

| Algorithm | *d* | *H* | *p* | Value |
|---|---|---|---|---|
| GA | 0.0598 | 0,3521 | 11,5980 | 0,032 |
| GWO | 0.0513 | 0.3474 | 11.8763 | 0.0127 |
| PSO | 0,0500 | 0,3104 | 14,998 | 0,0131 |
| VPPSO | 0.0525 | 0.3756 | 10.2659 | 0.0127 |
| **I-MsPSO** | **0.0516** | **0.356** | **11.3186** | **0.01266** |

**Pressure vessel design (PVD).** Pressure vessels typically comprise a cylindrical shell and 2 hemispherical heads, fabricated through the welding processes. The design objective is to minimize the overall cost, encompass material acquisition, form operations, and weld expenses. This optimization problem involves 4 design variables: cylinder wall thickness $E_s(X_1)$, the thickness of the spherical cover $E_h(X_2)$, cylinder inner diameter $D(X_3)$, and cylinder length $L(X_4)$. A description of the objective functions and constraints relevant to these 3 optimization variables is presented. Consider:

$$X = [E_s, E_h, D, L] = [X_1, X_2, X_3, X_4]. \quad (30)$$

Minimize:

$$F(x) = 0.6224 X_1 X_3 X_4 + 1.7781 X_2 X_3^2 + 3.1661 X_1^2 X_4 + 19.84 X_1^2 X_3. \quad (31)$$

Subject to:

$$g_1(X) = -X_1 + 0.0193 X_3 \le 0; \quad (32)$$

$$g_2(X) = -X_2 + 0.00954 X_3 \le 0; \quad (33)$$

$$g_3(X) = -\pi X_3^2 X_4 - \frac{4}{3}\pi X_3^3 + 1296000 \le 0; \quad (34)$$

$$g_4(X) = X_4 - 240 \le 0. \quad (35)$$

Variable range:

$$X_1, X_2 \in \{1 \cdot 0.0625, 2 \cdot 0.0625, ..., 99 \cdot 0.0625\},$$

$$10 \le X_3 \text{ and } X_4 \le 200.$$

Table 3 presents the best solutions of all algorithms. It is evident that I-MsPSO achieved the best result.

Table 3

Optimal solutions of PVD problem optimized by different algorithms

| Algorithm | $X_1$ | $X_2$ | $X_3$ | $X_4$ | Optimal cost |
|---|---|---|---|---|---|
| GA | 0.810 | 0.436 | 42.096 | 176.655 | 6059.945 |
| GWO | 0.812 | 0.4375 | 42.098 | 176.636 | 6059.719 |
| PSO | 0.875 | 0.4375 | 45.288 | 140.743 | 6096.830 |
| VPPSO | 0.8125 | 0.4375 | 42.0979 | 176.646 | 6059.850 |
| **I-MsPSO** | **0.8125** | **0.4375** | **42.0973** | **176.654** | **6059.714** |

**Application of STaSuM to Box-Jenkins gas furnace data.** Due to its non-linear nature, the Box-Jenkins system [29] has become widely adopted to validate the performance of the recently developed modeling methods. The used dataset contained 296 paired input-output observations ($y(t)$, $u(t)$) for a gas furnace process, where $t$ ranged from 1 to 296. At each sampling time $t$, $u(t)$ is the input gas flow rate and $y(t)$ is the output $CO_2$ concentration. The simulation was conducted to predict $y(t)$ based on $y(t_1)$, $y(t_2)$, $u(t_1)$, $u(t_2)$. The first 148 input-output data were employed as training data and the final 148 were utilized as testing data in order to validate the efficiency of the suggested method. The population size in the 4 subswarms was set to 6, the number of rules was 3, the number of iterations was 50, the acceleration coefficients were set according to the equations and the

inertia weight was chosen as shown in (13), (15), (17). In prediction case, compared to the A-MsPSO algorithm and other algorithms mentioned in Table 4 the I-MsPSO algorithm had the best performance index of 0.104.

Table 4

Identification results obtained by the different methods on the Box-Jenkins system

| Reference | Number of inputs | Number of rules | MSE |
|---|---|---|---|
| [35] | 6 | – | 0.202 |
| [36] | 2 | 3 | 0.110 |
| [37] | 2 | 4 | 0.148 |
| [38] | 2 | 2 | 0.161 |
| [18] | 2 | 3 | 0.106 |
| **I-MsPSO** | **2** | **3** | **0.104** |

Figure 7 shows the STaSuM model's prediction, the actual outputs and the errors between them for the training set of data. The other 148 data points were used to validate the generalization performance of the obtained fuzzy mode – Fig. 8 reveals the test results. Their respective related MSEs are 0.057 and 0.145.

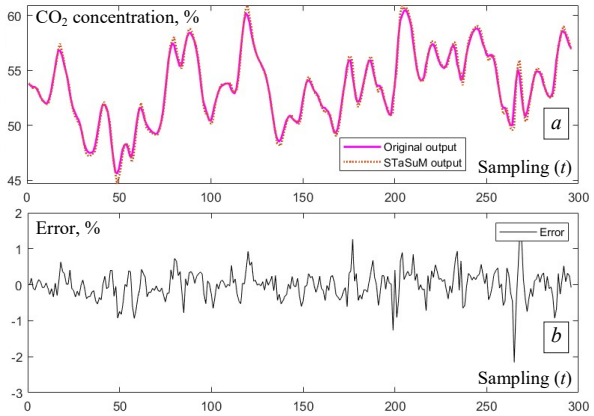


Fig. 7. Modeling with 296 pairs of observations: *a* – the real output and the output of the fuzzy model; *b* – the estimation error

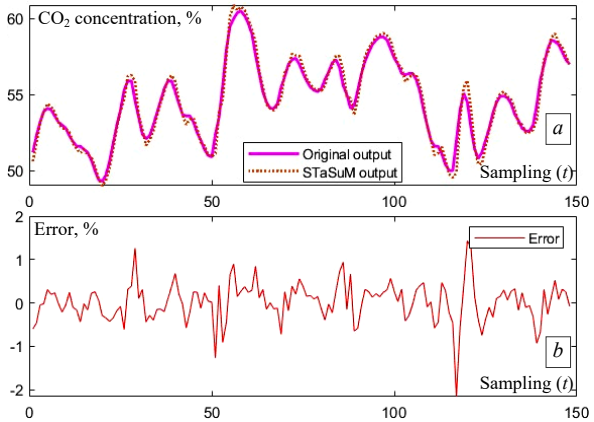


Fig. 8. Testing with 148 pairs of observations: *a* – the real output and the output of the fuzzy model; *b* – the estimation error

Table 5 presents a comparison with various models and demonstrates that our method's generalization ability outperforms that reported in the literature. Therefore, it can be noticed that the real output and the estimated output were within a negligible error. These results were justified by observing the values of the MSE performance index given in Table 5. The latter reveals that the performance indices obtained by the developed model during the identification and validation phase are the best, compared to those provided by other models mentioned in this table, even in the case of reduced number of input variables and minimized number of rules, which guarantees better quality of approximation. The results show that the proposed model has more powerful generalization ability with a good accuracy in modeling the system of the Box-Jenkins gas furnace dataset.

Table 5

Prediction results provided the different methods on the Box-Jenkins system

| Reference | Number of rules | MSE identification | MSE validation |
|---|---|---|---|
| [36] | 3 | 0.059 | 0.152 |
| [37] | 6 | 0.022 | 0.236 |
| [39] | 3 | 0.0159 | 0.126 |
| [18] | 3 | 0.058 | 0.146 |
| **I-MsPSO** | **3** | **0.057** | **0.145** |

**Conclusions.** In this paper, improved multiswarm particle swarm optimization (I-MsPSO) algorithm was used to optimize and estimate the parameters of Takagi-Sugeno (T-S) fuzzy systems. In the proposed specific Takagi-Sugeno modeling (STaSuM), the structure and the parameters of T-S fuzzy model were encoded into a nest vector to find the optimal solution simultaneously. The main advantage of STaSuM is that it can keep the inner-correlation between the system structure and the parameters, and more highly-accurate model than the traditional 2-stage identification process method. I-MsPSO divided the population into 4 subswarms; each of which utilized a search strategy independent of the other. The exchange of information between the 4 subswarms allowed collecting useful messages from the subswarms, maintaining particle diversity and improving the search capability. The best personal interactive learning strategy increased the convergence speed. The experimental results on 12 benchmark functions proved that the proposed algorithm had good comprehensive performance in the optimization of unimodal and multimodal functions and kept a good balance between exploration and exploitation. Additionally, the developed method was applied to estimate blur T-S system models using a fuzzy model STaSuM. The obtained finding showed experimental results proved that the suggested method can produce robust, reliable and effective fuzzy T-S models. The obtained finding showed experimental results proved that the suggested method can produce robust, reliable and effective fuzzy T-S models. In our upcoming work, we will: apply I-MsPSO to solve real industry problems; use I-MsPSO in solar PV; analyze the influence of different levels of noise on the accuracy of this algorithm.

**Conflict of interest**. The authors declare that they have no conflicts of interest.

REFERENCES


***1.*** Mamdani E.H., Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 1975, vol. 7, no. 1, pp. 1-13. doi: https://doi.org/10.1016/S0020-7373(75)80002-2.

***2.*** Alonso Moral J.M., Castiello C., Magdalena L., Mencar C. An Overview of Fuzzy Systems. In: Explainable Fuzzy Systems. *Studies in Computational Intelligence*, 2021, vol. 970, pp. 25-47. doi: https://doi.org/10.1007/978-3-030-71098-9_2.

***3.*** Bas E., Egrioglu E. A fuzzy regression functions approach based on Gustafson-Kessel clustering algorithm. *Information Sciences*, 2022, vol. 592, pp. 206-214. doi: https://doi.org/10.1016/j.ins.2022.01.057.

***4.*** Krasnov D., Davis D., Malott K., Chen Y., Shi X., Wong A. Fuzzy C-Means Clustering: A Review of Applications in Breast Cancer Detection. *Entropy*, 2023, vol. 25, no. 7, art. no. 1021. doi: https://doi.org/10.3390/e25071021.

***5.*** Wu X., Zhou H., Wu B., Zhang T. A possibilistic fuzzy Gath-Geva clustering algorithm using the exponential distance. *Expert Systems with Applications*, 2021, vol. 184, art. no. 115550. doi: https://doi.org/10.1016/j.eswa.2021.115550.

**6.** Gogineni V.C., Naumova V., Werner S., Huang Y.-F. Graph Kernel Recursive Least-Squares Algorithms. *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Tokyo, Japan, 2021, pp. 2072-2076.
**7.** Gholinejad S., Amiri-Simkooei A. Multivariate Weighted Total Least Squares Based on the Standard Least-Squares Theory. *Journal of Surveying Engineering*, 2023, vol. 149, no. 4, art. no. 04023008. doi: https://doi.org/10.1061/JSUED2.SUENG-1424.
**8.** Björck Å. *Numerical methods for least squares problems. 2nd edition*. SIAM Publ., 2024. 494 p.
**9.** Saeed H., Mehmood T., Khan F.A., Shah M.S., Ullah M.F., Ali H. An improved search ability of particle swarm optimization algorithm for tracking maximum power point under shading conditions. *Electrical Engineering & Electromechanics*, 2022, no. 2, pp. 23-28. doi: https://doi.org/10.20998/2074-272X.2022.2.04.
**10.** Labed M.A., Zellagui M., Benidir M., Sekhane H., Tebbakh N. Optimal hybrid photovoltaic distributed generation and distribution static synchronous compensators planning to minimize active power losses using adaptive acceleration coefficients particle swarm optimization algorithms. *Electrical Engineering & Electromechanics*, 2023, no. 6, pp. 84-90. doi: https://doi.org/10.20998/2074-272X.2023.6.15.
**11.** Jain M., Saihjpal V., Singh N., Singh S.B. An Overview of Variants and Advancements of PSO Algorithm. *Applied Sciences*, 2022, vol. 12, no. 17, art. no. 8392. doi: https://doi.org/10.3390/app12178392.
**12.** Gomathi S., Mayurappriyan P.S., BabyPriya B. Performance Improvement of Grid-Tied PV System With Boost and Quadratic Boost Converters Using Innovatory Hybridized HPO-PO MPPT. *Electric Power Components and Systems*, 2024, pp. 1-13. doi: https://doi.org/10.1080/15325008.2024.2318399.
**13.** Zanganeh M., Chaji A. A new aspect of the ApEn application to improve the PSO-ANFIS model to forecast Caspian Sea levels. *Regional Studies in Marine Science*, 2024, vol. 69, art. no. 103347. doi: https://doi.org/10.1016/j.rsma.2023.103347.
**14.** Momani S., Batiha I.M Tuning of the Fractional-order PID Controller for some Real-life Industrial Processes Using Particle Swarm Optimization. *Progress in Fractional Differentiation and Applications*, 2022, vol. 8, no. 3, pp. 377-391. doi: https://doi.org/10.18576/pfda/080303.
**15.** Lin G., Zhao K., Wan Q. Takagi-Sugeno fuzzy model identification using coevolution particle swarm optimization with multi-strategy. *Applied Intelligence*, 2016, vol. 45, no. 1, pp. 187-197. doi: https://doi.org/10.1007/s10489-015-0752-0.
**16.** Yen G.G., Leong W.F. Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 2009, vol. 39, no. 4, pp. 890-911. doi: https://doi.org/10.1109/TSMCA.2009.2013915.
**17.** Zhao L., Qian F., Yang Y., Zeng Y., Su H. Automatically extracting T–S fuzzy models using cooperative random learning particle swarm optimization. *Applied Soft Computing*, 2010, vol. 10, no. 3, pp. 938-944. doi: https://doi.org/10.1016/j.asoc.2009.10.012.
**18.** Zdiri S., Chrouta J., Zaafouri A. An Expanded Heterogeneous Particle Swarm Optimization Based on Adaptive Inertia Weight. *Mathematical Problems in Engineering*, 2021, vol. 2021, art. no. 4194263. doi: https://doi.org/10.1155/2021/4194263.
**19.** Zdiri S., Chrouta J., Zaafouri A. Cooperative multi-swarm particle swarm optimization based on adaptive and time-varying inertia weights. *2021 IEEE 2nd International Conference on Signal, Control and Communication (SCC)*, 2021, pp. 200-207. doi: https://doi.org/10.1109/SCC53769.2021.9768349.
**20.** Tamura K. Evaluation-Number Constrained Optimization Problem and its Solution Strategy. *IEEJ Transactions on Electrical and Electronic Engineering*, 2024, vol. 19, no. 4, pp. 490-506. doi: https://doi.org/10.1002/tee.24000.
**21.** Mezhoud N., Ayachi B., Amarouayache M. Multi-objective optimal power flow based gray wolf optimization method. *Electrical Engineering & Electromechanics*, 2022, no. 4, pp. 57-62. doi: https://doi.org/10.20998/2074-272X.2022.4.08.
**22.** Fornæss J.E., Hu M., Truong T.T., Watanabe T. Backtracking New Q-Newton's Method, Newton's Flow, Voronoi's Diagram and Stochastic Root Finding. *Complex Analysis and Operator Theory*, 2024, vol. 18, no. 5, art. no. 112. doi: https://doi.org/10.1007/s11785-024-01558-6.
**23.** Mayer J. *Stochastic Linear Programming Algorithms. A Comparison Based on a Model Management System*. Routledge Publ., 2022. 164 p. doi: https://doi.org/10.1201/9780203738269.
**24.** Braun G., Carderera A., Combettes C.W., Hassani H., Karbasi A., Mokhtari A., Pokutta S. Conditional Gradient Methods. *arXiv preprint*, 2025. doi: https://doi.org/10.48550/arXiv.2211.14103.
**25.** Gad A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, 2022, vol. 29, no. 5, pp. 2531-2561. doi: https://doi.org/10.1007/s11831-021-09694-4.
**26.** Labed M.A., Zellagui M., Benidir M., Sekhane H., Tebbakh N. Optimal hybrid photovoltaic distributed generation and distribution static synchronous compensators planning to minimize active power losses using adaptive acceleration coefficients particle swarm optimization algorithms. *Electrical Engineering & Electromechanics*, 2023, no. 6, pp. 84-90. doi: https://doi.org/10.20998/2074-272X.2023.6.15.
**27.** Anwar N., Hanif A., Ali M.U., Zafar A. Chaotic-based particle swarm optimization algorithm for optimal PID tuning in automatic voltage regulator systems. *Electrical Engineering & Electromechanics*, 2021, no. 1, pp. 50-59. doi: https://doi.org/10.20998/2074-272X.2021.1.08.
**28.** Song R., Huang S., Xiong L., Zhou Y., Li T., Tan P., Sun Z. Takagi–Sugeno Fuzzy Parallel Distributed Compensation Control for Low-Frequency Oscillation Suppression in Wind Energy-Penetrated Power Systems. *Electronics*, 2024, vol. 13, no. 19, art. no. 3795. doi: https://doi.org/10.3390/electronics13193795.
**29.** Hadjidj N., Benbrahim M., Ounnas D., Mouss L.H. Global maximum power point tracking method for photovoltaic systems using Takagi-Sugeno fuzzy models and ANFIS approach. *Electrical Engineering & Electromechanics*, 2025, no. 2, pp. 31-38. doi: https://doi.org/10.20998/2074-272X.2025.2.05.
**30.** Amézquita L., Castillo O., Soria J., Cortes-Antonio P. A New Variant of the Multiverse Optimizer Using Multiple Chaotic Maps and Fuzzy Logic for Optimization in CEC-2017 Benchmark Suite. *Studies in Computational Intelligence*, 2024, vol. 1149, pp. 267-283. doi: https://doi.org/10.1007/978-3-031-55684-5_18.
**31.** Kennedy J., Eberhart R. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942-1948. doi: https://doi.org/10.1109/ICNN.1995.488968.
**32.** Holland J.H. Genetic Algorithms. *Scientific American*, 1992, vol. 267, no. 1, pp. 66-72. doi: https://doi.org/10.1038/scientificamerican0792-66.
**33.** Shami T.M., Mirjalili S., Al-Eryani Y., Daoudi K., Izadi S., Abualigah L. Velocity pausing particle swarm optimization: a novel variant for global optimization. *Neural Computing and Applications*. 2023, vol. 35, no. 12, pp. 9193-9223. doi: https://doi.org/10.1007/s00521-022-08179-0.
**34.** Mirjalili S., Mirjalili S.M., Lewis A. Grey Wolf Optimizer. *Advances in Engineering Software*, 2014, vol. 69, pp. 46-61. doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.
**35.** Jenkins G.M., Box G.E.P. *Time Series Analysis: Forecasting and Control*. Holden-Day Publ., 1976. 575 p.
**36.** Cheung N.J., Ding X.-M., Shen H.-B. OptiFel: A Convergent Heterogeneous Particle Swarm Optimization Algorithm for Takagi–Sugeno Fuzzy Modeling. *IEEE Transactions on Fuzzy Systems*, 2014, vol. 22, no. 4, pp. 919-933. doi: https://doi.org/10.1109/TFUZZ.2013.2278972.
**37.** Tsekouras G.E. On the use of the weighted fuzzy c-means in fuzzy modeling. *Advances in Engineering Software*, 2005, vol. 36, no. 5, pp. 287-300. doi: https://doi.org/10.1016/j.advengsoft.2004.12.001.
**38.** Bagis A. Fuzzy rule base design using tabu search algorithm for nonlinear system modeling. *ISA Transactions*, 2008, vol. 47, no. 1, pp. 32-44. doi: https://doi.org/10.1016/j.isatra.2007.09.001.
**39.** Li C., Zhou J., Fu B., Kou P., Xiao J. T–S Fuzzy Model Identification With a Gravitational Search-Based Hyperplane Clustering Algorithm. *IEEE Transactions on Fuzzy Systems*, 2012, vol. 20, no. 2, pp. 305-317. doi: https://doi.org/10.1109/TFUZZ.2011.2173693.

*Received 19.03.2025*
*Accepted 09.06.2025*
*Published 02.09.2025*

*S. Zdiri*[1]*, Doctor of Technical Science,*
*M. Moulahi*[1]*, Professor,*
*F. Messaoudi*[1]*, Doctor of Technical Science,*
*A. Zaafouri*[1]*, Professor,*
[1] National Higher Engineering School of Tunis (ENSIT), Laboratory of Engineering of Industrial Systems and Renewable Energy (LISIER), University of Tunis, Tunisia,
e-mail: zdiri_sami@yahoo.fr (Corresponding Author)

*How to cite this article:*
Zdiri S., Moulahi M., Messaoudi F., Zaafouri A. Takagi-Sugeno fuzzy model identification using improved multiswarm particle swarm optimization in solar photovoltaics. *Electrical Engineering & Electromechanics*, 2025, no. 5, pp. 49-56. doi: https://doi.org/10.20998/2074-272X.2025.5.07