Performance Comparison of Variants Based on Swarm Intelligence Algorithm of Mathematical and Structural Optimization

Phuc Tran Van¹,a, Thu Huynh Van²,b, Sawekchai Tangaramvong²,c

¹ University of Architecture Ho Chi Minh City, Pasteur str., 196, Ho Chi Minh, 700000, Vietnam
² Applied Mechanics and Structures Research Unit, Department of Civil Engineering, Chulalongkorn University, 10330 Bangkok, Thailand

apos;phuc.tranvan@uah.edu.vn, b6278306421@student.chula.ac.th, csawekchai.t@chula.ac.th

Abstract. In this paper, three different variants of the canonical particle swarm optimization (PSO) algorithm are compared in terms of their optimal solutions to engineering optimization problems. The PSO often face to premature convergence due to its weak capability in local exploitation, which leads to a low optimization precision or even failure. Moreover, the computational burden is also one of the obstacles for the PSO when solving complex problems. To overcome these drawbacks, the original PSO algorithm is integrated with the knowledge sharing and auxiliary mechanisms of comprehensive learning (CL) and Gaussian local search (GLS) strategies to form the so-called GLS-PSO, CLPSO, and GLS-CLPSO algorithms. To demonstrate the accuracy and robustness of the three variants, their performance is tested on a benchmark spatial truss structure with continuous variables. The obtained optimization results are then compared with those by other PSO variants in the literature.

1. Introduction
Optimization problems are commonly found in science and engineering applications and they are becoming more and more complicated. Some of well-known traditional approaches to solving these problems have efficiently provided theoretical concepts for the ground understanding but also solution techniques. However, some drawbacks of these techniques still exist, such as their gradient-based formulations as well as complexity in calculation of the derivatives and sensitivity of the objective and constraint functions at the starting point. Therefore, most of the traditional techniques are unfit for solving complex optimization problems. Recently, swarm intelligence (SI) algorithms belong to the stochastic search techniques which have gained popularity as an alternative solution for the traditional techniques. One of the famous SI algorithms is the particle swarm optimization (PSO) that was firstly developed by Kennedy and Eberhart [1]. The algorithm is constructed based on the intelligent behavior of animals (e.g., the colony of fish and birds) in finding food sources and avoiding enemies in nature. Due to the simplicity of implementation, and requires only a few parameters, the PSO has become the most attractive algorithm and successfully applied for many real applications [2]. However, like other stochastic search techniques, the crude PSO often suffers from trapping into local optima, with premature convergence but less accuracy. This is likely faced when solving complex problems and has
attracted the attention of many researchers to enhance the performance of the crude PSO. Many variant and hybrid versions have proposed either modifying the tuning parameters, solution update formula, and auxiliary mechanisms or combining with other existing algorithms [3].

In this study, three variants of the crude PSO algorithm are proposed by integrating with comprehensive learning (CL) and Gaussian local search (GLS) strategies as respectively termed GLS-PSO, CLPSO, and GLS-CLPSO algorithms. In particular, the CL strategy is used to improve the exploration capability of the PSO, here exemplars are constructed by utilizing personal best experiences of the entire population during the process of updated positions. Meanwhile, the GLS strategy performs a neighborhood search that uses the Gaussian distribution to generate samples through the information sharing of personal best and global best positions from the PSO. This strategy helps the PSO to strongly improve its exploitation capability. The accuracy and effectiveness of the proposed algorithm are illustrated through a spatial truss design benchmark. The obtained optimization results were then compared with those reported in the literature.

2. Formulation of the optimization problem

Generally, a design optimization problem is formulated for minimizing (or maximizing) an objective function $f(A)$, while satisfying certain design constraints, such that

Minimize $f(A)$, \hspace{1cm} $A = [x_1, x_2, \ldots, x_{nd}]$. \hspace{1cm} (1)

Subjected to $g_j(A) \leq 0$, \hspace{1cm} $j = 1, 2, \ldots, nc$. \hspace{1cm} (2)

where $A \in \mathbb{R}^{nd}$ denotes the vector of $nd$-dimensional design variables, each element of $A$ is defined by a lower bound $x_{g,\text{min}}$ and an upper bound $x_{g,\text{max}}$ ($g = 1, \ldots, nd$), and $g_j(A)$ represents the $j$-th constraint among a total of $nc$ constraints.

3. Comprehensive learning PSO

Consider a swarm of a finite number of particles. Let $pbest$ and $gbest$ denote the best position each particle has seen and the best position found by any particle, respectively. In the crude PSO, each particle mimics the moving trajectory of $pbest$ and $gbest$ simultaneously. The $gbest$ is a piece of important information to navigate the direction of the whole swarm. However, when particles in the swarm learn from $gbest$ even if the current $gbest$ is far from the global solution. This results in easily attracting to the region around $gbest$ and trapping into a local minimum. In order to improve the sharing of information, Liang and Huang [4] developed the CLPSO algorithm using the CL strategy. The CL uses $pbest$ from all particles to maintain the population diversity as well as enhancing the exploration capability. Each particle learns from different exemplars on each dimensions. The next velocity and position of CLPSO are updated according to Eqs. (3) and (4).

\begin{align*}
V_i^{\text{next}} &= w \times V_i + c_1 \times r_1 \times \left( pbest_{f(i)} - X_i \right) + c_2 \times r_2 \times \left( gbest - X_i \right). \hspace{1cm} (3) \\
X_i^{\text{next}} &= X_i + V_i^{\text{next}}. \hspace{1cm} (4)
\end{align*}

where $fi = [fi(1), fi(2), \ldots, fi(nd)]$ denotes if the $i$-th particle follows its own or other’s $pbest$ for each dimension $nd$, $X_i$ the position of particle $i$, $V_i$ the velocity of particle $i$, $w$ an inertial weight scaling factor, $c_1$ and $c_2$ two acceleration coefficients scaling respectively the influences of the $pbest$ and $gbest$, and $r_1$ and $r_2$ independent random numbers within the interval of $[0, 1]$, $pbest_{f(i)}$ denotes the exemplar of the $i$-th particle on the $nd$-dimension, $gbest$ the best position found by the entire swarm as follows in two Eqs. (5) and (6), respectively.

\begin{align*}
pbest_{f(i)}^{\text{next}} &= \begin{cases} 
pbest_{f(i)} \hspace{1cm} \text{if } f\left( pbest_{f(i)} \right) \leq f\left( X_i \right) \\
X_i \hspace{1cm} \text{if } f\left( pbest_{f(i)} \right) > f\left( X_i \right) \end{cases} \hspace{1cm} \text{for } \forall i \in \{1, \ldots, ps\}. \hspace{1cm} (5)
\end{align*}
The CL strategy is illustrated as follows [5]:

**Step 1:** If the \( i \)-th particle’s historical \( pbest \) does not improve consecutively and over a refreshing gap \( m \) (set \( m = 7 \)), then two particles are randomly chosen from the population (population size \( ps \)).

**Step 2:** The objective function values of these two particles are compared together, and the better one is selected as the exemplar based on the learning probability \( P_{c_i} \), that is determined by Eq. (7).

**Step 3:** If the exemplar located the same position with \( pbest \), then a dimension is randomly selected from another particle’s \( pbest \) in the corresponding dimension.

\[
gbest = \min \left\{ f\left( pbest_{j(td)} \right), f\left( gbest \right) \right\} \quad \text{for } \forall i \in \{1, \ldots, ps\} \tag{6}
\]

4. Gaussian local search

Unlike other distributions, the Gaussian distribution describes a bell-curved shape. It exploits over a central field, and improves the optimization accuracy. The distribution typically denoted by \( N(\mu, \sigma^2) \) is characterized a mean value \( \mu \) and variance \( \sigma^2 \). The Gaussian probability density function is defined by:

\[
f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( \frac{-(x-\mu)^2}{2\sigma^2} \right), \quad \sigma > 0. \tag{8}
\]

The GLS strategy improves the exploitation ability using the particle \( gbest \) and \( pbest \) from the PSO procedure. The random numbers around \( gbest \) are generated by Gaussian distribution [6]. To characterize Gaussian distribution and obtain the random values, the mean and standard deviation are determined by Eq. (9):

\[
x_i \sim \mathcal{N}(\mu, \sigma^2) = \mathcal{N}\left(\text{gbest}, |\text{gbest} - pbest|\right). \tag{9}
\]

The number of GLS iterations is reset such that the \( gbest \) exploits sufficiently the current region and jumps out of the local optima.

5. Illustrative Example

Consider a space 120-bar dome truss as shown in Fig 1 [7, 8]. Due to symmetry of the dome structure geometry, all 120 members are divided into 7 different groups (\( nd = 7 \)). The members of each group have the same length and cross-sectional area. In addition, the allowable tensile and compressive stress are defined complying with the AISC-ASD (1989) [9] specification, as follows:

\[
\begin{cases}
\sigma_m^+ = 0.6F_y & \text{for } \sigma_m \geq 0 \\
\sigma_m^- & \text{for } \sigma_m < 0
\end{cases} \tag{10}
\]

Where \( \sigma_m^- \) is calculated based on the slenderness ratio

\[
\sigma_m = \begin{cases}
\left[1 - \frac{\lambda_m^2}{2C_c^2}\right]F_y \left(\frac{5}{3} + \frac{3\lambda_m^2}{8C_c^2} - \frac{\lambda_m^4}{8C_c^4}\right) & \text{if } \lambda_m < C_c \\
\frac{12\pi^2E}{23\lambda_m^2} & \text{if } \lambda_m \geq C_c
\end{cases} \tag{11}
\]
where yield stress of $F_y = 58.0$ ksi (400 Mpa), $C_c$ is the member slenderness limit ($C_c = \sqrt{2\pi^2 E / F_y}$), $\lambda_m$ is the slenderness ratio of member $m$ ($\lambda_m = k_m l_m / r_m$), $k_m$ is the effective length factor ($k_m = 1$ for all members), $r_m$ is the radius of gyration that can be expressed based on the cross-sectional areas as $r_m = k A_m^p$, in which $k$ and $p$ are constants related to the types of cross sections for truss members (e.g., pipes, angles, and tees). In this example, pipe sections ($k = 0.4993$ and $p = 0.6777$) are considered for bars [7].

Figure 1. Space 120-bar dome truss structure

Young’s modulus and density of material are 30,450 ksi (210,000 Mpa) and 0.288 lb/in$^3$ (7971.81 kg/m$^3$), respectively. The displacement limitations of $\pm 0.1969$ (5 mm) are imposed at all nodes in x, y, and z directions. The dome truss is subjected to vertical forces at all the unsupported nodes. These forces involve -13.49 kips (-60 kN) at node 1, -6.744 kips (-30 kN) at nodes 2 through 14, and -2.248 kips (-10 kN) at the rest of the nodes. The minimum and maximum available areas of all truss members are specified as 0.775 in$^2$ (5 cm$^2$) and 20.0 in$^2$ (129.032 cm$^2$), respectively. Three proposed PSO variants are encoded in Python, and run by Intel Core i5-9400 CPU @ 2.9 GHz.
For the three proposed PSO variants, a population of 30 particles is used, \( c_1 \) and \( c_2 \) are set equal to 2.0, and the inertia weight \( w \) is calculated by linearly decreasing from 0.9 to 0.2. The stopping criterion is assigned as the maximum number of iterations, set equal to 1500. The number of GLS iterations is 1000.

As a result, Fig. 2 reports the weight convergence of the truss structure by the PSO and the three proposed variants. Table 1 provides the performance comparison of three proposed PSO variants with the original PSO and other PSO variants in the literature. Noticeably, the GLS-CLPSO provides outstanding results without violating the limitations of the maximum displacement and stresses. It is also seen that the minimum weight of 120-bar dome by the GLS-CLPSO is 33,250.66 lb, which is lighter than most designs by CLPSO, GLS-PSO, PSO, HPSSO, and MSPSO.

**Table 1.** Performance comparison of optimal solutions of 120-bar truss design.

| Group | MSPSO [10] | HPSSO [7] | PSO | GLS-PSO | CLPSO | GLS-CLPSO |
|-------|-------------|------------|-----|---------|-------|-----------|
| \( x_i \) (in\(^2\)) | \( x_i \) (in\(^2\)) | \( x_i \) (in\(^2\)) | \( x_i \) (in\(^2\)) | \( x_i \) (in\(^2\)) | \( x_i \) (in\(^2\)) | \( x_i \) (in\(^2\)) |
| \( x_1 \) | 3.0244 | 3.0241 | 3.0277 | 3.0287 | 3.0309 | 3.0274 |
| \( x_2 \) | 14.7804 | 14.7808 | 14.0977 | 14.4397 | 14.6184 | 15.2985 |
| \( x_3 \) | 5.0567 | 5.0522 | 5.1552 | 5.2001 | 5.3269 | 5.0303 |
| \( x_4 \) | 3.1359 | 3.1369 | 3.1066 | 3.0919 | 3.0880 | 3.0733 |
| \( x_5 \) | 8.4830 | 8.5004 | 8.7257 | 8.5107 | 8.3481 | 8.2674 |
| \( x_6 \) | 3.3104 | 3.2888 | 3.4711 | 3.4527 | 3.4091 | 3.4077 |
| \( x_7 \) | 2.4977 | 2.4969 | 2.4963 | 2.5057 | 2.4969 | 2.5010 |
| Best weight (lb) | 33,251.22 | 33,250.05 | **33,260.82** | **33,252.31** | **33,251.78** | **33,250.66** |

In summary, the CLPSO is apparently good at preserving the particles’ diversity and is thus excellent in exploration. However, it is weak in the exploitation ability. Meanwhile, the GPSO is efficient in performing a local search, but weak in the exploration ability. The GLS-CLPSO may be a remedy that significantly improves the exploitation ability of the CLPSO and the exploration ability of the GPSO.
6. Conclusions
This paper proposed three variants of the PSO algorithm by integrating it with CL and GLS strategies, namely, GLS-PSO, CLPSO, and GLS-CLPSO. To demonstrate the effectiveness of these variants, their performance was tested on a space truss structure. The optimal results showed that the GLS-CLPSO algorithm outperforms the PSO, CLPSO, GLS-CLPSO, and other PSO variants in the literature. Finally, it can be concluded that the PSO variants provide suitable ways to achieve not only a good balance between exploration and exploitation but also a competitive solution to solve complicated optimization problems in reality.

References
[1] Eberhart R, Kennedy J. (1995) Particle swarm optimization, proceeding of IEEE International Conference on Neural Network. Perth, Australia. pp. 1942-8.
[2] Mavrovouniotis M, Li C, Yang S. (2017) A survey of swarm intelligence for dynamic optimization: Algorithms and applications. Swarm and Evolutionary Computation. 33 pp. 1-17.
[3] Wang D, Tan D, Liu L. (2018) Particle swarm optimization algorithm: an overview. Soft Computing. 22 pp. 387-408.
[4] Liang JJ, Qin AK, Suganthan PN, Baskar S. 2006 Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation. 10 pp. 281-95.
[5] Zhang K, Huang Q, Zhang Y. (2019) Enhancing comprehensive learning particle swarm optimization with local optima topology. Information Sciences. 471 pp. 1-18.
[6] Jia D, Zheng G, Qu B, Khan MK. (2011) A hybrid particle swarm optimization algorithm for high-dimensional problems. Computers & Industrial Engineering. 61 pp. 1117-22.
[7] Kaveh A, Bakhshpoori T, Afshari E. (2014) An efficient hybrid particle swarm and swallow swarm optimization algorithm. Computers and Structures. 143 pp. 40-59.
[8] Jafari M, Salajegheh E, Salajegheh J. (2019) An efficient hybrid of elephant herding optimization and cultural algorithm for optimal design of trusses. Engineering with Computers. 35 pp. 781-801.
[9] AISC-Manual of steel construction–allowable stress design. American Institute of Steel Construction, Chicago, 1989.
[10] Talatahari S, Kheirollahi M, Farahmandpour C, Gandomi AH. (2013) A multi-stage particle swarm for optimum design of truss structures. Neural Computing and Applications. 23 pp. 1297-309.