111EQUATION CHAPTER 1 SECTION 1SIZE OPTIMIZATION OF THREE-DIMENSIONAL TRUSS STRUCTURES USING PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

Faisal Ahmed*[1], Jannatul Akmum[2]

[1] *Student, Khulna University of Engineering & Technology, Bangladesh, e-mail: faisalkuet1845@gmail.com*
[2] *Student, Khulna University of Engineering & Technology, Bangladesh, e-mail: jannatul0104519@gmail.com*

***Corresponding Author**

ABSTRACT

In the design phase of a construction project, enhancing the technical designs, structural configurations, and individual components is one of the key engineering activities. Structural optimization allows engineers to explore efficient design alternatives, but manual optimization process is time-consuming and becomes complicated, especially for structures with numerous variables and complex designs. However, this complexity can be mitigated through automation using programming languages and heuristic mathematical optimization procedures, whichoffer more reliable and efficient results when correctly applied. This approach still requires multiple iterations to achieveoptimal outcomes for the specified variables. Hence, researchers have shown considerable interest in algorithms that are robust, easy to implement, and exhibit rapid convergencetoward solutions in the structural optimization problem. This study aims to explore the suitability of the Particle Swarm Optimization (PSO) method, a widely used optimization algorithm, for optimizing three-dimensional truss structures. The research involved the analysis and optimization of a 72-bar spatial truss structure and a 120-bar dome-shaped truss structure, with the goal of achieving the minimum weight of the structure while adhering to permissible stress and nodal displacement constraints. OpenSeesPy, a Python interpreter for the structural analysis software framework OpenSees, was implemented to analyze both truss structures. To avoid interruptions by internal processes within the Windows operating system and other applications, the analysis and optimization process was done on the Google Colab Notebook platform.To enhance the performance of the PSO algorithm, the influences of PSO algorithm parameters, including the number of particles, cognitive coefficient c_1, social coefficient c_2, and inertia weight w, on simulation runtime, the weight of the structure, and the number of iterations required to achieve the minimum weight were observed. After numerous trials, thestudy concludedthat the best range for the values of cognitive coefficient c_1 and social coefficient c_2 to be between 0.8 and 1.0 and the inertia weight w to be equal to 0.8, and the number of particles to be between 40 to 50.Using these optimized parameter values, both truss structures were finally optimized and the outcomes were compared with the results achieved by other researchers employing alternative optimization methods. For the 72-bar spatial truss structure, the weight of the optimized structure was found to be 369.654 lb, which was consistent with the findings of other researchers utilizing other algorithms. The cross-sectional area of each member in the optimized structure also showed similar patterns to the findings of other researchers. For the 120-bar dome-shaped truss structure, the weight of the optimized structure was found to be 31974.9 lb, which was slightly better than the results found by other researchers. These findings finally conclude that the PSO algorithm can effectively be used for the optimization of truss structures.

Keywords:*Particle Swarm Optimization, Structural Optimization, Truss Optimization, OpenSeesPy, Python Structural Analysis*

1. INTRODUCTION

Truss structures are widely used in the construction of bridges, roofs, towers, cranes, space frames, etc. While designing a structure, an engineer must ensure both structural integrity and functionality while maintaining cost-effectiveness as much as possible. Truss optimization is the process of determining the most efficient design of the structure. The primary objective of the optimization process is to achieve the most lightweight structure without compromising its strength and stability. Reducing the weight of the structure not only minimizes material costs but also facilitates transportation and the installation process, leading to an overall enhancement in structural efficiency.

Tejani et al. (2018)have stated that truss optimization can be classified into three types: size optimization, shape optimization, and topology optimization. Size optimization involves finding the optimum cross-sectional areas of the elements of the structure. Shape optimization refers to the process of changing node coordinates to determine the optimum shape of the structure. Topology optimization deals with the process ofadding and removing elements and nodes to find the most efficient shape of the structure (Tejani et al., 2018).According to Kochenderfer & Wheeler (2019), various optimization methods are available, for example, first-order methods, second-order methods, direct methods, stochastic methods, and population methods. Within the category of population methods, there are several widely used algorithms, including genetic algorithms, particle swarm optimization, cuckoo search, and ant colony optimization, etc.(Kochenderfer & Wheeler, 2019). Different population methods, including the Chaotic Coyote Algorithm, Harmony Search Algorithm (HSA), Charged System Search Algorithm, Colliding Bodies Optimization, and Vibrating Particles System, have been developed by various researchers, includingKaveh (2017), Kaveh et al. (2017), Kaveh & Mahdavi (2014), and Pierezan et al. (2021).

The structural optimization process involves a continuous iteration where the values of variables to be optimized are altered, structural analysis is conducted, and violations of constraints are checked until an optimized result is achieved. Real-world structures often involve numerous variables and complex designs, leading to significant runtime requirements for structural analysis. As a result, a high number of iterations in the optimization process can result in the loss of valuable time, computational power, and other associated costs, such as electricity expenses. So, exploring algorithms that are robust, memory-efficient, easy to implement, and exhibit a high rate of convergence toward solutions to structural optimization problems has been of great interest to many researchers. Thus, this article aims to explore the applicability of the Particle Swarm Optimization (PSO) method in optimizing three-dimensional truss structures.

Kennedy & Eberhart (1995)have mentioned that the concept behind the particle swarm optimization method is easy to understand and can readily be implemented through computer programming. It does not require complex mathematical operations, such as derivatives, differentials, or continuity equations of the function that needs to be optimized. This optimization approach is also highly efficient in terms of memory usage while exhibiting rapid convergence towards the solution (Kennedy & Eberhart, 1995). According to Gad (2022), having a few parameters to deal with within the algorithm enables it to find optimal solutions quickly. However,in high-dimensional search spaces, the algorithm exhibits slow convergence towards the global optimum and shows poor-quality results in large and complex datasets. Numerous researchers have addressed this issue by introducing various techniques into the algorithm to enhance its convergence rate, for example, improved learning strategies, mutation techniques, fuzzy logic, Lévy Flight, opposition-based learning, and surrogate methods, leading to the development of various modified PSO algorithms(Gad, 2022).

Throughout the course of this research, the author was unable to find any existing literature mentioning the impact of varying values of PSO algorithm parameters on optimized results, convergence rate, and simulation runtime while optimizing a truss structure. Rather than modifying the PSO algorithm, this article emphasizesfine-tuning the algorithm parameters and implementing a simple penalty function for constraint violations to enhance the convergence rate towards the optimal outcome of the PSO algorithm.

This article presents the implementation of the Particle Swarm Optimization (PSO) method for optimizing three-dimensional truss structures with the aim of minimizing the structure's weight as much as possible while maintaining its allowable stress limits and deflection limits. The suitability of the algorithm is assessed by comparing the obtained results with those derived from alternative optimization techniques employed by other researchers in optimizing truss structures. In order to enhance the rate of convergence towards the optimal solution, the paper investigates different values of the parameters of the algorithm.

2. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

The Particle Swarm Optimization (PSO) algorithm was initially presented by Eberhart and Kennedy in 1995. This algorithm is a stochastic optimization method that emulates the collective behavior of various animals, such as insect swarms, flocks of birds, shoals of fish, and herds. It is developed based on the fundamental principles governing the cooperative food-finding behaviors of these animal groups and the adaptation of search patterns by individual members of the group drawing from their self-learning experiences and those of their peers (Wang et al., 2018).

The PSO algorithm consists of a set of particles. Initially, these particles are defined randomly within a predefined range of domain of a required objective function. These groups of particles are also often referred to as swarms. Each particle in these swarms represents the potential solution to the objective function. As the particles traverse through the search space of the objective function, their positions are updated in each iteration based on the best position among all particles. Thedetermination of the optimal position within the search space relies on the fitness values of the particles, which are derived from the evaluation of the objective function for each particle (Li et al., 2007).illustrates a flowchart of the PSO algorithm, which is well described by Kennedy & Eberhart (1995), Gad (2022), andWang et al. (2018).

The first task is to define N number of particles, each with a position vector in a D-dimensional space as denoted in equation (1), along with a velocity vector mentioned in equation (2). Furthermore, the initial optimum positions of individual particles are represented in equation (3) and the optimum position of the swarm i.e., the best position among all particles' positions, is denoted in equation (4).

$$x^{(i)} = \left(x^{(i_1)}, x^{(i_2)}, x^{(i_3)}, \ldots\ldots\ldots\ldots, x^{(i_D)} \right) \quad 22\backslash* \text{ MERGEFORMAT ()}$$

$$v^{(i)} = \left(v^{(i_1)}, v^{(i_2)}, v^{(i_3)}, \ldots\ldots\ldots\ldots, v^{(i_D)} \right) \quad 33\backslash* \text{ MERGEFORMAT ()}$$

$$P^{(i)} = \left(P^{(i_1)}, P^{(i_2)}, P^{(i_3)}, \ldots\ldots\ldots\ldots, P^{(i_D)} \right) \quad 44\backslash* \text{ MERGEFORMAT ()}$$

$$P_g^{(i)} = \left(P_g^{(i_1)}, P_g^{(i_2)}, P_g^{(i_3)}, \ldots\ldots\ldots\ldots, P_g^{(i_D)} \right) \quad 55\backslash* \text{ MERGEFORMAT ()}$$

The notation $x^{(i)}(t)$ represents the position of particle P_i in hyperspace at a time step t. To update the position of the particle P_i, the velocity $v^{(i)}(t+1)$is added to the current position, as mentioned in equation (5). The adjustment of the velocity vector is presented in equation (6) and involves updates based on both the individual best position (the local best position) and the global best position (the best position among all particles).

$$x^{(i)}(t+1) = x^{(i)}(t) + v^{(i)}(t+1) \quad 66\backslash* \text{ MERGEFORMAT ()}$$

$$v^{(i)}(t+1) = wv^{(i)}(t) + c_1 r_1 \left\{ x_{best}^{(i)}(t) - x^{(i)}(t) \right\} + c_2 r_2 \left\{ x_{best}(t) - x^{(i)}(t) \right\} \quad 77\backslash* \text{ MERGEFORMAT ()}$$

Where x_{best} denotes the best position ever found among all particles; w, c_1, and $c_2$are particle parameters known as inertia weight, cognitive coefficient and social coefficient respectively; and r_1 and r_2 are random numbers ranging from 0 to 1.

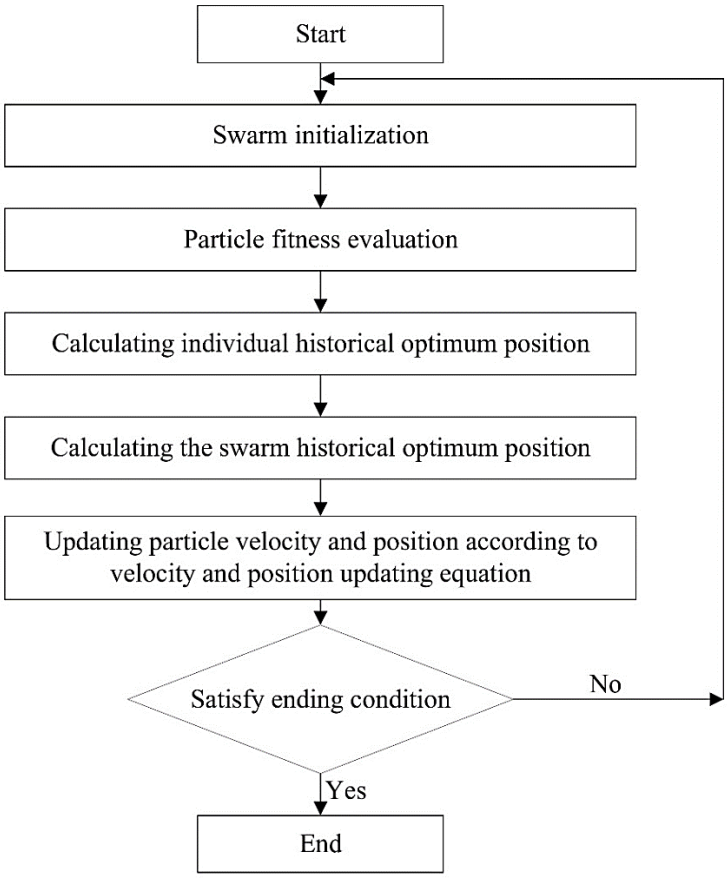

Figure 1: Flowchart of PSO algorithm. Reproduced from Wang et al. (2018)

3. METHODOLOGY

The methodology of this research started with the validation of analysis results obtained from the OpenSeesPy structural analysis framework, such as axial forces and nodal displacements of a 25-bar spatial truss structure, with respect to one of the widely used structural analysis software named STAAD Pro. Subsequently, analysis and optimization of two truss structures, such as a 72-bar spatial truss structure and a 120-bar dome truss structure, were done. Initially, the values of PSO algorithm parameters such asc_1, c_2, w, and the number of particleswere varied and corresponding changes in simulation runtime, the best-obtained results, and the number of iterations needed to achieve optimal outcomes were observed. By selecting the optimal values for these parameters, both structures were analyzed and optimized again and the results were compared with outcomes obtained by other researchers using alternative algorithmsto assess the applicability of the PSO algorithm and formulate recommendations based on the findings.

3.1 Validation of OpenSeesPy Structural Analysis Framework

In this research paper, OpenSeesPy is used to performanalysis of the truss structure, including the determination of node displacements and axial forces in the bars. OpenSees is an open-source software framework which is quite popular in the field of earthquake engineering. Usually, it is used to simulate the response of a structure under seismic loads. However, it is only available for Linux operating system users.Moreover, building models and performing analysis using OpenSees require

knowledge of TCL, a scripting language. Thedevelopment of OpenSeesPy, an OpenSees Python interpreter, has broadly extended the user base to include the Windows operating systemuser community (Zhu, 2018).

To validate the OpenSeesPy software framework, a truss consisting of 25 members, as shown inFigure 2,was analyzed using both STAAD Pro and OpenSeesPy. Different software is built upon different methods of analysis and design, which can lead to variations in the results among them. Nevertheless, when the differences in results are minor, it indicates the reliability and consistency of the software's outputs. During the analysis, all members were assumed to have an equal cross-sectional area of 10 $in^2$and a modulus of elasticity of 10,000 ksi. Four pin supports were assigned at node 7 to node 10. Two concentrated loads with a value of 100 kips were applied at node 1 acting in the positive X direction and positive Y direction, respectively.

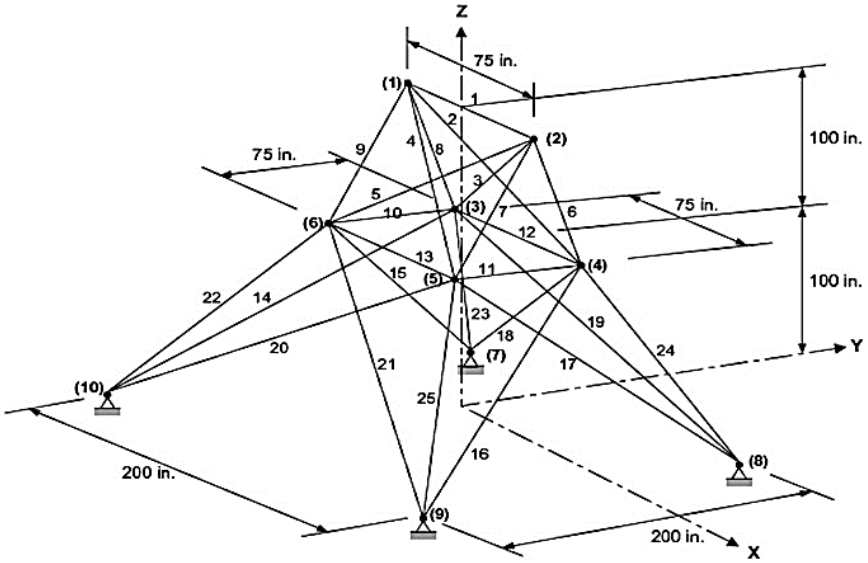

Figure 2: 25 bar truss structure for validation of OpenSeesPy framework. Source: Li et al. (2007)

The obtained results are presented in and Table 2. shows the axial forces denoted as P, obtained from the structural analysis using StaadPro and OpenSeesPy, whereas Table 2 shows the nodal displacements denoted as ΔX, ΔY, and ΔZ in the X axis, Y axis, and Z axis, respectively. In , positive values indicate tensile forces in the members, while negative values represent compressive forces. In both software, maximum compressive force and maximum tensile force were found inmember 24 andmember 9, respectively. Remarkably, both software packages showed identical results with very small deviations in both scenarios. From the analysis of both software, the maximum compressive force was 118.233 kip, while the maximum tensile force was 128.233 kip. The largest deviation in axial forces, comparing OpenSeesPy to StaadPro, was only 0.003% inmember 14, with an average variation of just 0.001% across all members.InTable 2, the maximum nodal displacement was found at node 1, having a value of 0.579 inches in the positive y-axis. Interestingly, the deviations in the obtained nodal displacements between OpenSeesPy and StaadPro were slightly more pronounced than thoseseen in axial forces. The most significant variation in nodal displacement, when comparing the results obtained from OpenSeesPy to the results obtained from StaadPro, was 2.09%. On average, the deviation in displacement across all nodes was 0.42%.

Table 1: Validation of axial forces in the members of 25-bar spatial truss structure

Member ID	**A (in^2)**	$P_{StaadPro}$ **(kip)**	$P_{OpenSeesPy}$ **(kip)**	$P_{variation}$	**Average Variation**
1	10	-44.778	-44.778	0.000%	0.001%
2		-113.400	-113.400	0.000%	
3		44.319	44.318	0.001%	
4		17.311	17.311	0.001%	

Member ID	A (in^2)	$P_{StaadPro}$ (kip)	$P_{OpenSeesPy}$ (kip)	$P_{variation}$	Average Variation
5		33.597	33.597	0.001%	
6		-36.269	-36.268	0.000%	
7		-27.495	-27.495	0.000%	
8		-49.597	-49.597	0.000%	
9		128.233	128.233	0.000%	
10		-14.017	-14.017	0.001%	
11		14.222	14.222	0.001%	

Table 1: (continued).

Member ID	A (in^2)	$P_{StaadPro}$ (kip)	$P_{OpenSeesPy}$ (kip)	$P_{variation}$	Average Variation
12		9.732	9.731	0.002%	
13		-5.357	-5.357	0.000%	
14		9.870	9.869	0.003%	
15		45.712	45.711	0.001%	
16		-40.246	-40.246	0.000%	
17		-15.707	-15.707	0.001%	
18		-18.200	-18.200	0.001%	
19		-38.947	-38.947	0.001%	
20		-11.223	-11.223	0.000%	
21		69.904	69.904	0.001%	
22		109.423	109.423	0.000%	
23		4.768	4.768	0.002%	
24		-118.233	-118.233	0.000%	
25		3.186	3.186	0.001%	

Table 2: Validation of nodal displacements of 25-bar spatial truss structure

	Staad Pro			OpenSeesPy			Variation		
Node	Δx (in)	Δy (in)	Δz (in)	Δx (in)	Δy (in)	Δz (in)	Δx (in)	Δy (in)	Δz (in)
1	0.4210	0.5790	0.1290	0.4209	0.5787	0.1293	0.03%	0.05%	0.25%
2	0.3870	0.1990	-0.1210	0.3873	0.1985	-0.1206	0.08%	0.25%	0.36%
3	0.0680	0.0120	-0.0270	0.0681	0.0119	-0.0266	0.21%	0.45%	1.37%
4	0.0750	0.0390	-0.1390	0.0754	0.0393	-0.1389	0.59%	0.70%	0.07%
5	-0.0210	0.0290	-0.0250	-0.0213	0.0286	-0.0255	1.19%	1.36%	1.94%
6	-0.0170	0.0220	0.1920	-0.0172	0.0225	0.1916	1.37%	2.09%	0.19%
7	0	0	0	0	0	0	0.00%	0.00%	0.00%
8	0	0	0	0	0	0	0.00%	0.00%	0.00%
9	0	0	0	0	0	0	0.00%	0.00%	0.00%
10	0	0	0	0	0	0	0.00%	0.00%	0.00%

3.2 Analysis & Optimization of Spatial Truss Structures

In this article, the performance of the PSO algorithm is evaluated using two spatial truss structures. Both the structural analysis module and the PSO algorithm itself were implemented using thePython programming language and executed on the Google Colab Notebook platform.The study includes analysis and optimization of a 72-bar spatial truss structure as illustrated in and a120-bar dome-shaped truss structure as illustrated in Figure 4.At first, in both examples, the values ofalgorithm parameters such asc_1, c_2, w and the number of particles were varied and corresponding changes in simulation runtime, weight of the structure, and the number of iterations required to achieve the minimum weight were observed.Based on these observations, optimal values of these parameterswere selected. Finally, the two truss structures were analyzed and optimized again using the selected values

of algorithm parameters and the results were compared with outcomes obtained by other researchers using alternative algorithms.

3.2.1 Optimization of 72-bar Spatial Truss Structure

illustrates a spatial truss structure that consists of 72 members and 16 nodes. Four pin supports were assigned at node 1 to node 4. Additionally, three concentrated loads having amagnitude of 5 kipswere applied at node 17 along the positive x-axis, positive y-axis, and negative z-axis,respectively. The material's modulus of elasticity was set at 10,000 ksi and density of 0.1 lb/in³. All members were subjected to a maximum allowable stress within the range of ±25 ksi. Additionally, maximum nodal displacements were set to a range of ±0.25 in all directions — x-axis, y-axis, and z-axis. The members of the truss were categorized into 16 groups namely (1) $A_1 - A_4$, (2) $A_5 - A_{12}$, (3) $A_{13} - A_{16}$, (4) $A_{17} - A_{18}$, (5) $A_{19} - A_{22}$, (6) $A_{23} - A_{30}$, (7) $A31 - A34$, (8) $A35 - A36$, (9) $A37 - A40$, (10) $A41 - A48$, (11) $A_{49} - A_{52}$, (12) $A_{53} - A_{54}$, (13) $A_{55} - A_{58}$, (14) $A_{59} - A_{66}$ (15) $A_{67} - A_{70}$, and (16) $A_{71} - A_{72}$.

illustrates the influences of PSO algorithm parameters, including the number of particles, cognitive coefficient c_1 and social coefficient c_2, and inertia weight w, on the simulation runtime, the weight of the structure, and the number of iterations required to achieve the minimum weight. The table illustrates that as the number of particles increased up to 40 particles, better results were obtained and any further increase showed no significant improvement; however, corresponding simulation runtimes also increased somewhat proportionally. The influence of cognitive coefficient c_1 and social coefficient c_2 on the weight of the structure showed no particular pattern. Nevertheless, minimum weights were obtained when both c_1 and c_2 lied within the ranges of 0.6 to 0.8 and 1.5 to 4. Notably, the range of 1.5 to 4 required more than twice the number of iterations compared to the range of 0.6 to 0.8 to achieve the same result. The table additionally demonstrates the impact of inertia weight (w) on the model's solution, indicating that as w increased up to 0.8, the model converged toward a more optimal solution. However, any further increase in w led to a deviation from the best solution of the model. Based on these observations, the optimal values for the algorithm parameters were chosen as follows: c_1 = 0.8, c_2 = 0.8, w = 0.8, and the number of particles = 50 for the final optimization of the truss structure.

The 72-bar truss structure has also been optimized by Le et al. (2019), Li et al. (2007), Pierezan et al. (2021), and Schmit & Farshi (1974). Initially, the weight of the structure was found to be 750.57 lb. After optimization, the weight of the structure was reduced to 369.65 lb, indicating a 50.75% reduction compared to its initial weight. The results obtained by the PSO algorithm showed a similar pattern to those found by previous researchers mentioned in Table 4. Li et al. (2007)have applied the PSO algorithm to optimize the structure and have found the optimized weight of the structure is 6818.67 lb, which has significantly differed from those of other researchers. In contrast, this study achieved notably improved results compared to the findings of Li et al. (2007) using the same PSO algorithm and found the optimized weight of the structure is 369.654 lb. The consideration of the minimum andmaximum cross-sectional area of the members,along with thefine-tuning of algorithm parameterscould be the reasons for this significant improvement. In this research, the minimum and maximum cross sectional area were set to 0.1 and 5 in^2, respectively.However, the influence of initial consideration on the weight of the optimized structure is out of scope the of this research.

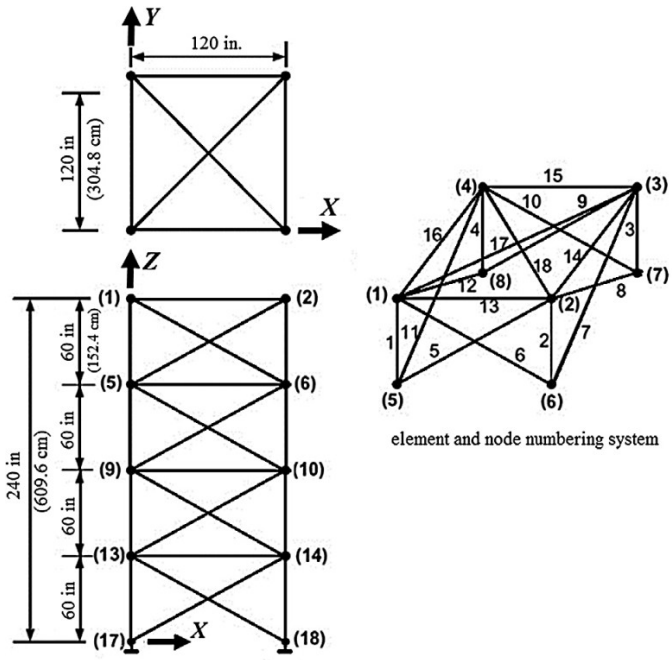

Figure 3: 72-bar spatial truss structure. Source: Li et al. (2007)

Table 3: Influence of the PSO algorithm parameters on optimization results for 72-bar spatial truss

Influence of Number of Particles			**Influence of c_1 and c_2**			**Influence of w**		
Number of Particles	**Weight (lb)**	**Runtime (sec)**	**c_1 and c_2**	**Weight (lb)**	**Iteration for Converge to Solution**	**w**	**Weight (lb)**	**Iteration for Converge to Solution**
5	440.78	24	0.0	756.09	101	0.0	653.136	13
10	389.75	43	0.1	398.44	280	0.1	644.582	35
15	370.08	61	0.2	412.52	238	0.2	637.342	39
20	369.96	79	0.4	372.67	406	0.4	630.467	64
25	369.93	102	0.6	369.92	336	0.6	464.06	437
30	369.67	119	0.8	369.69	451	0.8	369.939	437
35	377.25	160	1.0	376.67	581	1.0	370.207	178
40	369.66	181	1.5	369.92	749	1.5	372.350	639
50	369.65	200	2.0	369.92	652	2.0	372.879	1384
60	369.64	243	3.0	369.97	945	3.0	375.417	1469
70	369.64	286	4.0	369.90	1915	4.0	414.588	117
80	369.91	325	5.0	370.05	2412	5.0	421.645	749
100	369.64	406	10.0	370.31	938			

Table 4: Optimization results for 72-bar spatial truss

Variables	**Optimal Cross-Sectional Areas (in^2)**				
	This Article	**Li et al. (2007)**	**Schmit & Farshi (1974)**	**Pierezan et al. (2021)**	**Le et al. (2019)**
	Particle Swarm Optimization	**Particle Swarm Optimization**	**Structural Synthesis Concept**	**Chaotic Coyote Algorithm**	**Hybrid Method combining Electromagnetism**

	(PSO)	(PSO)			and Firefly Algorithms
$A_1 - A_4$	1.856	41.794	2.078	1.990	1.990
$A_5 - A_{12}$	0.510	0.195	0.503	0.563	0.563
$A_{13} - A_{16}$	0.100	10.797	0.100	0.111	0.111
$A_{17} - A_{18}$	0.100	6.861	0.100	0.111	0.111
$A_{19} - A_{22}$	1.250	0.438	1.107	1.228	1.228
$A_{23} - A_{30}$	0.505	0.286	0.579	0.442	0.442
$A_{31} - A_{34}$	0.100	18.309	0.100	0.111	0.111
$A_{35} - A_{36}$	0.100	1.220	0.100	0.111	0.111
$A_{37} - A_{40}$	0.491	5.933	0.264	0.563	0.563
$A_{41} - A_{48}$	0.505	19.545	0.548	0.563	0.563
$A_{49} - A_{52}$	0.100	0.159	0.100	0.111	0.111
$A_{53} - A_{54}$	0.100	0.151	0.151	0.111	0.111
$A_{55} - A_{58}$	0.100	10.127	0.158	0.196	0.196
$A_{59} - A_{66}$	0.520	7.320	0.594	0.563	0.563
$A_{67} - A_{70}$	0.399	3.812	0.341	0.391	0.391
$A_{71} - A_{72}$	0.535	18.196	0.608	0.563	0.563
Total Weight (lb)	369.654	6818.670	388.630	389.334	389.334

3.2.2 Optimization of 120-bar Dome-Shaped Truss Structure

Figure 4 and Figure 5 illustrate a dome-shaped truss structure consisting of 120 members and 49 nodes. 12 pin supports were assigned at node 38 to node 49. Vertical loads with magnitudes of 13.49 kips, 6.744 kips, and 2.248 kips were applied at node 1, nodes 2 to 14, and the remaining unsupported joints,respectively.The modulus of elasticity of the truss material wasset to30,450 ksi and the density of the material was set to 0.288 lb/in³. The displacements of each node in the structure were limited to a maximum of 0.1969 inches along all three axes: x-axis, y-axis, and z-axis. Allowable axial stress on each member was set according to the Allowable Stress Design (ASD) guidelines by AISC (1989) (AISC, 1989). Allowable tensile stress is outlined in equation (7) and allowable compressive stress is outlinedin equations (8) and (9).

$$\sigma_t = 0.6 \times F_y$$ 88* MERGEFORMAT ()

$$\sigma_c = \frac{12\pi^2 E}{23\lambda_i^2} \quad \text{for} \quad \lambda_i \geq C_c$$ 99* MERGEFORMAT ()

$$\sigma_c = \frac{\left(1 - \frac{\lambda_i^2}{2C_c^2}\right) \times F_y}{\frac{5}{3} + \frac{3\lambda_i}{8C_c} - \frac{\lambda_i^3}{8C_c^3}} \quad \text{for} \quad \lambda_i < C_c$$ 1010* MERGEFORMAT ()

Here, E = modulus of elasticity of material; F_y = yield stress of material; $C_c = \sqrt{2\pi^2 E ¿ F_y}$ = the slenderness ratio dividing the elastic and inelastic buckling regions; $\lambda_i = k L_i / r_i$ = slenderness ratio of member; k = effective length factor (for both end pin supported member, $k = 1$); L_i = length of the member; and r_i = radius of gyration of the member. In this simulation, the members were consideredto be circular solids. Themaximum slenderness ratio was limited to 300 for members

subjected to tensile stress and to 200 for members subjected to compressive stress. For all members, 0.775 in^2 and 20 in^2were assigned as minimum cross-sectional area and maximum cross-sectional area, respectively.

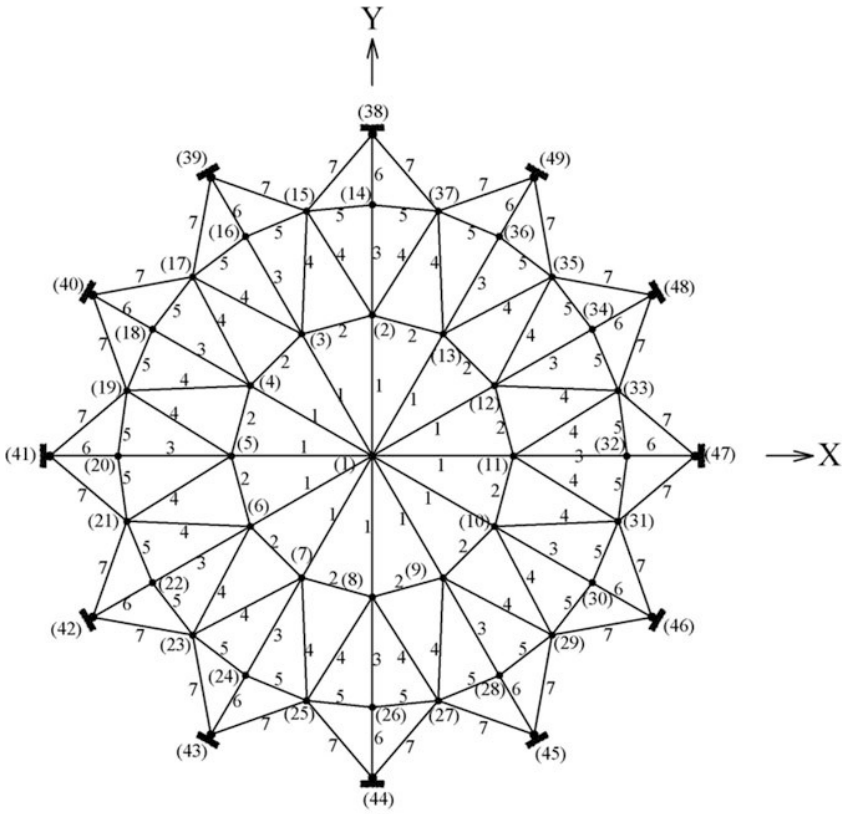

Figure 4: Plan view of 120 bar dome-shaped truss structure. Source: Kaveh (2017)

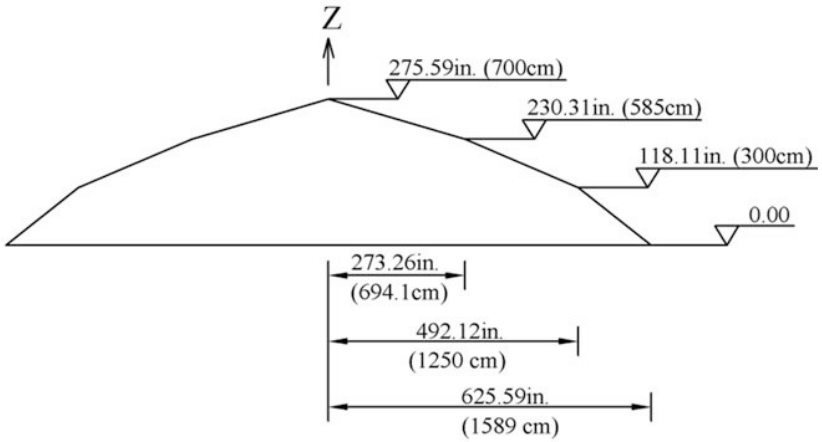

Figure 5: Side view of 120 bar dome-shaped truss structure. Source: Kaveh (2017)

Table 5 illustrates the influences of PSO algorithm parameters, including the number of particles, cognitive coefficient c_1 and social coefficient c_2, and inertia weight w on the simulation runtime, the weight of the structure, and the number of iterations required to achieve the minimum weight. The table indicates that an increase in the number of particles up to 50 particles led to an improved result and any further increase showed no significant improvement; however, there was a somewhat proportional increase in simulation runtimes. The influence of cognitive coefficient (c_1) and social coefficient (c_2) on the structure's weight didn't follow a specific pattern. Nevertheless, the minimum weights were achieved when both c_1 and c_2 were within the ranges of 0.8 to 1.0 and 2.0 to 3.0, respectively. It is noteworthy that the range of 2.0 to 3.0 required more than twice the number of iterations compared to the range of 0.8 to 1.0 to attain similar results. The table also indicates how

inertia weight (w) influenced the optimal solution. It reveals that as w increased up to 0.8, better solutions were achieved. Nevertheless, any further increase in w resulted in a departure from the model's best solution. Based on these observations, the optimal values for the algorithm parameters were chosen as follows: c_1 = 0.8, c_2 = 0.8, w = 0.8, and the number of particles = 50 for the final optimization of the truss structure.

Initially, the weight of the structure was around 86,000 lb. After the optimization, the weight of the optimized structure decreased to 31,974.9 lb, signifying a substantial 62.82% reduction from its initial weight. The results obtained through the PSO method in this article exhibited a slightly better resultthan those found by previous researchers mentioned in Error: Reference source not found.

Table 5: Influence of the PSO algorithm parameters on optimization results for the 120-bar dome truss

Influence of Number of Particles			**Influence of c_1 and c_2**			**Influence of w**		
Number of Particles	**Weight (lb)**	**Runtime (sec)**	**c_1 and c_2**	**Weight (lb)**	**Iteration for Converge to Solution**	**w**	**Weight (lb)**	**Iteration for Converge to Solution**
5	34570.81	28	0.0	67455.48	74	0.0	87013.57	33
10	32170.85	57	0.1	42396.91	94	0.1	89225.17	32
15	33571.85	85	0.2	36732.30	134	0.2	86394.78	71
20	32050.47	115	0.4	32865.75	290	0.4	58574.88	101
25	32006.64	145	0.6	33478.04	285	0.6	33420.46	288
30	32030.15	172	0.8	32028.04	280	0.8	31977.44	433
35	32085.00	201	1.0	32088.10	364	1.0	32295.89	837
40	32043.83	226	1.5	33408.24	400	1.5	32573.26	1378
50	31999.31	288	2.0	31974.92	778	2.0	33499.93	1081
60	31983.17	347	3.0	32086.12	1860	3.0	35333.30	1092
80	31974.19	460	4.0	33649.42	2801	4.0	37349.52	26

Table 6: Optimization results for 120-bar dome-shaped truss structure

Element Group	**Optimal cross-sectional areas (in²)**					
	This Article	**Kaveh (2017)**	**Kaveh et al. (2013)**	**Talatahari et al. (2013)**	**Kaveh & Mahdavi (2014)**	**Kaveh et al. (2017)**
	Particle Swarm Optimization (PSO)	**Charged System Search (CSS) Algorithm**	**Improved Ray Optimization (IRO) Algorithm**	**Multi-Stage Particle Swarm Optimization (MSPSO)**	**Colliding Bodies Optimization**	**Vibrating ParticlesSystem (VPS)**
1 - (A_1¿	1.952	3.027	3.0252	3.024	3.027	3.0244
2 - (A_2¿	15.317	14.606	14.835	14.780	15.172	14.754

3 - (A_3 ¿	5.485	5.044	5.114	5.057	5.234	5.079
4 - (A_4 ¿	2.219	3.139	3.131	3.136	3.119	3.137
5 - (A_5 ¿	9.207	8.543	8.404	8.483	8.104	8.483
6 - (A_6 ¿	5.170	3.367	3.332	3.310	3.417	3.301
7 - (A_7 ¿	1.853	2.497	2.499	2.498	2.492	2.496
Weight (lb)	31974.9	33,251.9	33,256.5	33,251.2	33,286.3	33,249.9

4. CONCLUSIONS

This article presents OpenSeesPy, a Python interpreter designed for one of the widely utilized structural analysis modules known as OpenSees. A comparative analysis was conducted between the structural analysis outcomes obtained through OpenSeesPy and those derived from another widely used structural analysis software named StaadPro. The differences in results between these software applications were minimal, which confirms the effectiveness of the OpenSeesPy framework in solving three-dimensional truss structures.

This paper also discussed the application of the Particle Swarm Optimization (PSO) algorithm in the optimization of truss structures and provided a necessary discussion of its processes. The algorithm was tested for the optimization of two 3-dimensional truss structures. In the case of the 72-bar spatial truss structure, the algorithm showed similar results compared to outcomes obtained by other researchers employing different optimization methods. However, for the 120-bar dome-shaped truss structure, there was a slight improvement in the outcomes achieved by the algorithm. Furthermore, the effects of different parameters within the Particle Swarm Optimization (PSO) algorithm on the optimized results of the structures were also studied. The optimization of both structures indicated that the most favorable outcomes could be obtained when the cognitive coefficient (c_1) and social coefficient (c_2) were within the range of 0.8 to 1.0, the inertia weight (w) was equal to 0.8, and the number of particles was equal to 50. One notable advantage of employing the PSO algorithm was its simplicity and few number of parameters within the algorithm to calibrate the model. Furthermore, due to fast convergence towards the solution, the simulation runtime was also low. The effectiveness and simplicity of the PSO algorithm, combined with its fast convergence rate, make it a strong option when compared to other optimization methods.

In future research, it would be interesting to see if the PSO algorithm is applicable to optimization of even larger and more complex structures with numerous constraints in the construction industry. Additionally, exploring the combination of various optimization methods at different stages and portions of the structure could be an intriguing study.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Dr. Abu Zakir Morshed, Professor in the Department of Civil Engineering at Khulna University of Engineering & Technology (KUET), for his valuable guidance while writing this paper.

REFERENCES

AISC. (1989). *AISC Manual of Steel Construction: Allowable Stress Design 9th Edition, ASD,* (9th edition). American Inst. Of Steel Construction.

Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, *29*(5), 2531–2561. https://doi.org/10.1007/s11831-021-09694-4

Kaveh, A. (2017). Charged System Search Algorithm. In A. Kaveh, *Advances in Metaheuristic Algorithms for Optimal Design of Structures* (pp. 45–89). Springer International Publishing. https://doi.org/10.1007/978-3-319-46173-1_3

Kaveh, A., Ilchi Ghazaan, M., & Bakhshpoori, T. (2013). An improved ray optimization algorithm for design of truss structures. *Periodica Polytechnica Civil Engineering*, *57*(2), 97. https://doi.org/10.3311/PPci.7166

Kaveh, A., Kaveh, A., & Ilchi Ghazaan, M. (2017). A new meta-heuristic algorithm: Vibrating particles system. *Scientia Iranica*, *24*(2), 551–566. https://doi.org/10.24200/sci.2017.2417

Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & Structures*, *139*, 18–27. https://doi.org/10.1016/j.compstruc.2014.04.005

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for Optimization*. MIT Press.

Le, D. T., Bui, D.-K., Ngo, T. D., Nguyen, Q.-H., & Nguyen-Xuan, H. (2019). A novel hybrid method combining electromagnetism-like mechanism and firefly algorithms for constrained design optimization of discrete truss structures. *Computers & Structures*, *212*, 20–42. https://doi.org/10.1016/j.compstruc.2018.10.017

Li, L. J., Huang, Z. B., Liu, F., & Wu, Q. H. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures*, *85*(7–8), 340–349. https://doi.org/10.1016/j.compstruc.2006.11.020

Pierezan, J., Dos Santos Coelho, L., Cocco Mariani, V., Hochsteiner De Vasconcelos Segundo, E., & Prayogo, D. (2021). Chaotic coyote algorithm applied to truss optimization problems. *Computers & Structures*, *242*, 106353. https://doi.org/10.1016/j.compstruc.2020.106353

Schmit, L. A., & Farshi, B. (1974). Some Approximation Concepts for Structural Synthesis. *AIAA Journal*, *12*(5), 692–699. https://doi.org/10.2514/3.49321

Talatahari, S., Kheirollahi, M., Farahmandpour, C., & Gandomi, A. H. (2013). A multi-stage particle swarm for optimum design of truss structures. *Neural Computing and Applications*, *23*(5), 1297–1309. https://doi.org/10.1007/s00521-012-1072-5

Tejani, G., Savsani, V., & Bureerat, S. (2018). *Truss Topology Optimization: A Review*.

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Computing*, *22*(2), 387–408. https://doi.org/10.1007/s00500-016-2474-6

Zhu, M. (2018). *OpenSeesPy: Python library for the OpenSees finite element framework*.