

Reliability-based Design Optimization for Structures using Particle Swarm Optimization

D Prayogo¹, G Gaby¹, B H Wijaya¹, F T Wong¹, D Tjandra¹

Department of Civil Engineering, Petra Christian University, Jalan Siwalankerto 121-131, Surabaya 60236, Indonesia
m21416086@john.petra.ac.id

Abstract. Studies on structure design optimization have been conducted extensively over the past decades because it can increase structural efficiency and maximize the engineers' profit. One of the significant current discussions is the reliability aspect of structure design in addition to optimal design. This problem becomes more important especially in sizing and shaping optimization of truss structures. To address the problem, this paper applies Reliability-based Design Optimization (RBDO) by combining Latin Hypercube Sampling (LHS) and metaheuristic algorithms which are Particle Swarm Optimization (PSO) and Barebone PSO (BBPSO). The presence of uncertainty is modeled using LHS and the reliability constraint is added to measure the probability of structural failure. This study aims to investigate the performance of these metaheuristic algorithms in optimizing the structure design while still satisfying the reliability constraints. Two case studies, a welded beam design problem and 15-bar planar truss problem, are used in this study. The performance of these two algorithms will be compared. The obtained results indicate that BBPSO has a better performance than PSO.
Keywords: Optimization, PSO variants, Reliability, Truss structure

1. Introduction

In civil engineering, structure optimization has become an important and challenging topic because it can increase the structural efficiency. Structure optimization is the act of designing and developing structures to achieve maximum profit from available resources [1]. To minimize cost and the weight of structures, many researchers seek to optimize the diameter of steel pipe and the thickness or cross-sectional area of steel elements [3]. Moreover, the number of elements and the constraints in a structure's design increase the complexity of structure optimization. For this reason, the metaheuristic approach has become a popular method than the gradient method for solving structure optimization cases [9] because randomness is useful to find a global solution. Particle Swarm Optimization (PSO) is a frequently used metaheuristic algorithm in optimization problems [2].

In structure optimization, uncertainty is an inevitable problem because truss structures are sensitive to uncertain design variables, such as cross section, or uncertain parameters, such as force and the material's modulus of elasticity [5]. The robustness and safety of a structure are also affected by the changing of those variables and parameters, thus, uncertainty must be calculated in the design [5]. As a result, Reliability-based Design Optimization (RBDO) has become an important matter in structure optimization. There are three approaches used to analyze the probability of failure and structural reliability: the moment method, simulation method, and heuristic method [6].

This research aims to optimize a single variable—the structure's weight—by using Latin Hypercube Sampling (LHS) to model and simulate the uncertainty of variables with certain mean

value and standard deviation. Structural reliability was analyzed using LHS because it can converge with a smaller sample size than Simple Random Sampling or Monte Carlo Sampling [7]. In order to achieve a reliable smallest weight, we employed PSO and its variant, Barebone PSO (BBPSO), and then provided some constraints in the process with a probability of success not less than 99%.

2. Metaheuristic algorithms

2.1. Particle swarm optimization

The Particle Swarm Optimization (PSO) algorithm was developed by Kennedy and Eberhart in 1995 [2], inspired by the behavior of social organisms in groups such as bird and fish schooling. When bird flocks search for food, they will search for the best location. To find the best location, each bird will move with their own velocity based on its personal best location, its group's best location, and the previous location [10]. One of the advantages of PSO is that this algorithm is simple and it can be simply applied in a computer program [2]. The initialization of this method is random the population in a specified range. These particles have their own velocity so that they can move randomly from one place to another. Equation (1) is the particle's velocity and Equation (2) is used to update the location. When the particle discovers the optimal location, the location is saved in pbestX and the result is saved in pbestF. The optimal locations from all particles are saved in gbestX and the result is saved in gbestF, which represents the solution for the problem:

$$v_i(t+1) = wv_i(t) + r_1c_1(X_{pbest}(t) - X_i(t)) + r_2c_2(X_{gbest}(t) - X_i(t)), \quad (1)$$

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

where v_i is the velocity of particle i , w is the inertia weight parameter, c_1 is the cognitive factor parameter, X_{pbest} is the location coordinate of personal best, x_i is the coordinate of particle i , c_2 is the social factor parameter, X_{gbest} is the location coordinate of global best, and r_1 and r_2 are a random numbers from zero to one. Figure 1 displays the pseudo-code for PSO.

Algorithm 1 Particle Swarm Optimization

```

1 Initialize PSO parameters
2 Initialize a population of random particles (solutions)
3 Evaluate the objective value of each particle
4 Determine initial pbestX and gbestX
5 while termination criteria are not satisfied do
6   for each particle do
7     Update the velocity for the particle
8     Update the new location for the particle
9     Determine the objective value for the particle in its new location
10    Update pbestX and pbestF if required
11  end for
12  Update gbestX and pbestF if required
13 end while

```

Figure 1. Pseudo-code for particle swarm optimization.

2.2. Barebone particle swarm optimization

Barebone Particle Swarm Optimization (BBPSO) is a variant of PSO that has been simplified. However, it can be trapped into local optima for high-dimensional and complicated optimization problems [4]. This algorithm ignores all parameters and does not need to use velocity to find a new location. BBPSO mainly uses a jump strategy, which is implemented based on a Gaussian distribution [4]. The particle's next position is only calculated by its personal best location and swarm global best location:

$$\mu = \frac{pi+gbest}{2} \quad (3)$$

$$\sigma = |p_i - gbest| \quad (4)$$

$$x(i + 1) = \begin{cases} N(\mu, \sigma) & \text{if } (\omega > 0.5) \\ p_i & \text{else} \end{cases} \quad (5)$$

where p_i is the personal best location of each particle, $gbest$ is the best location of the whole swarm, and ω is a random number from zero to one. Figure 2 displays the pseudo-code for BBPSO.

Algorithm 2 Barebone Particle Swarm Optimization

```

1 Initialize PSO parameters
2 Initialize a population of random particles (solutions)
3 Evaluate the objective value of each particle
4 Determine initial pbestX and gbestX
5 while termination criteria are not satisfied do
6   for each particle do
7     Determine the objective value for the particle in its new location
8     Update pbestX and pbestF if required
9   end for
10  Update gbestX and pbestF if required
11 end while

```

Figure 2. Pseudo-code for barebone particle swarm optimization.

3. Latin Hypercube Sampling

Latin Hypercube Sampling (LHS) was used to assure a good estimation of the statistical moments of response function [11]. In this method, sample points are well spread out when projected onto a subspace spanned by several coordinate axes. First, we need to select n different values of k variables where the range of each variable is divided into n nonoverlapping intervals on the basis of equal probability. Then, select a value randomly from each interval. The sampled cumulative probability can be written as Equation (6):

$$Probi = \left(\frac{1}{N}\right)r_u + \frac{(i-1)}{N} \quad (6)$$

where r_u is a uniformly distributed random number ranging from zero to one. Then, the probability of failure can be obtained from Equation (7):

$$Pf \cong \frac{N_H}{N}, \quad (7)$$

where N_H is the number of failures and N is the number of simulations.

4. Methodology

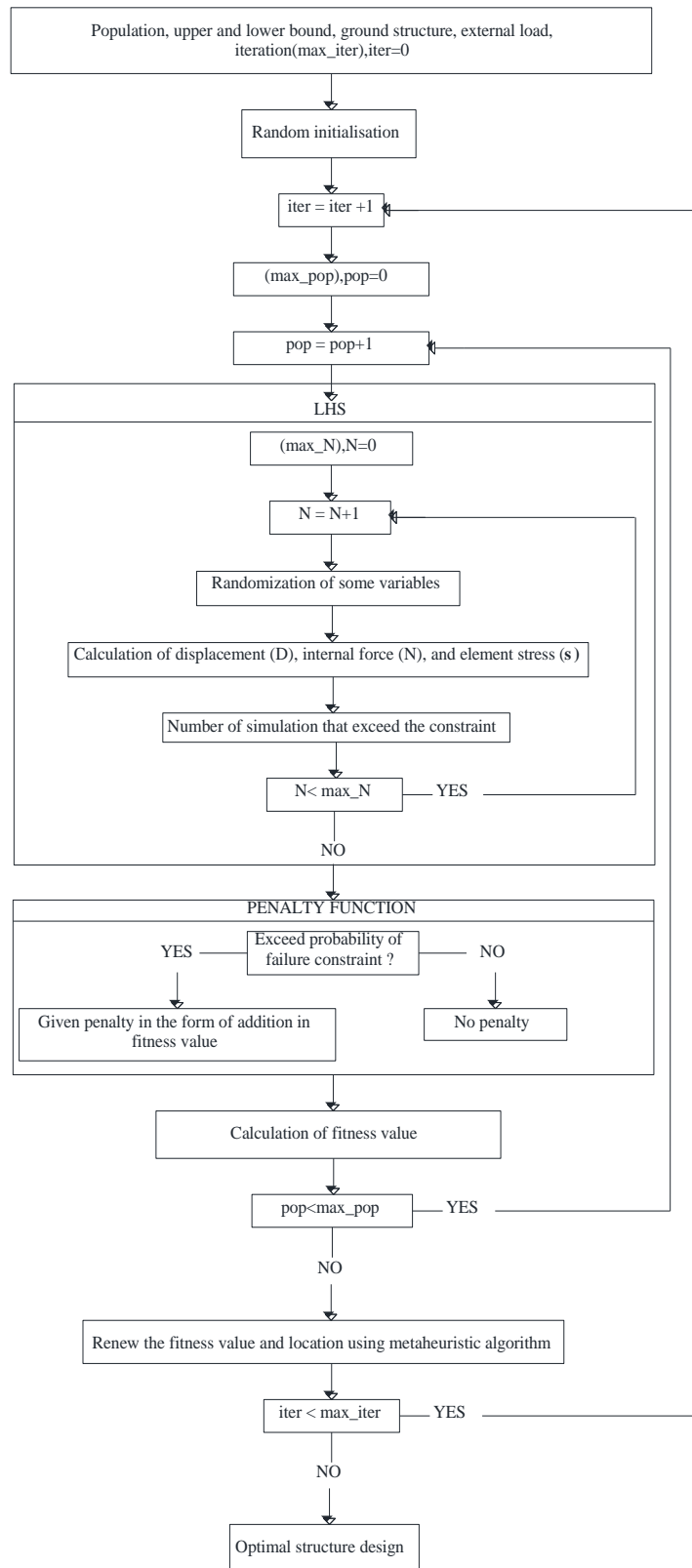


Figure 3. Flow chart for structure optimization.

Reliability-based Design Optimization was modeled by combining two metaheuristic algorithms, Direct Stiffness Method (DSM) and LHS. The metaheuristic algorithms used in this research are PSO and BBPSO, which were used to find the optimal structure design, DSM was used to obtain the displacement, axial force, and stress of each element in the truss structure, and LHS was used to model the uncertainty. The outputs were utilized to detect the number of structures that failed after some number of simulations (N). The probability of failure was obtained from LHS; i.e., if the structure is not reliable, then a penalty is given to the calculation of weight as the fitness value. This process will be repeated until reaching the maximum number of iterations that has been set before. DSM, PSO, and BBPSO are all written using MATLAB 2018b. A flow chart of the optimization process is presented in Figure 3.

5. Test Problems and Results

5.1. Welded beam problem

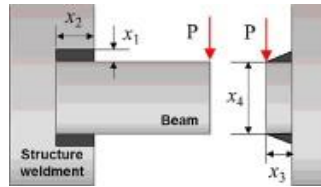


Figure 4. Welded beam structure [5].

The first test problem is a welded beam problem as diagrammed in Figure 4. This problem has four random variables and five probabilistic constraints. The problem objective is to find the minimum welding cost. The probabilistic constraints are limitation of shear stress, bending stress, buckling, and displacement. All random variables are statistically independent and of normal distribution. This test problem is run 30 times with 1000 iterations, 30 populations, and 100 simulations. The mathematical RBDO model of the welded beam problem is formulated as:

$$\begin{aligned}
 &\text{find} && x = \{x_1, x_2, x_3, x_4\} \\
 &\text{minimize} && f(x) = c_1 x_1^2 x_2 + c_2 x_3 x_4 (z_1 + d_2) \\
 &\text{subject to} && \text{Prob. } \{g_j(x) < 0\} \geq 99\%, j = 1, \dots, 5 \\
 &\text{where} && g_1 = \tau/z_6 - 1; \quad g_2 = \sigma/z_7 - 1; \quad g_3 = x_1/x_4; \quad g_4 = \delta/z_5 - 1; \quad g_5 = 1 - \text{Pc}/z_1; \\
 &&& \tau = (t^2 + 2t \cdot \text{tt} \cdot x_2 / 2R + \text{tt}^2)^{1/2} \\
 &&& t = \frac{z_1}{\sqrt{2} x_1 x_2}; \quad \text{tt} = M \cdot R / J \\
 &&& J = \sqrt{2} x_1 x_2 \{x_2^2 / 12 + (x_1 + x_3)^2 / 4\} \\
 &&& M = z_1 J (z_2 + \frac{x_2}{2}); \quad R = \frac{\sqrt{x_2^2 + (x_1 + x_3)^2}}{2} \\
 &&& \sigma = \frac{6 \cdot z_1 \cdot z_2}{x_3^2 x_4}; \quad \delta = \frac{4 \cdot z_1 \cdot z_2^2}{z_3 x_3^3 x_4} \\
 &&& \text{Pc} = \frac{4.013 x_4^3 x_3}{6 z_2^2} \sqrt{z_3 z_4} (1 - \frac{x_3}{4 z_2} \sqrt{z_3 / z_4}) \\
 &&& x_i \sim N(x_i, 0.1693^2) \text{ for } i = 1, 2 \\
 &&& x_i \sim N(x_i, 0.0107^2) \text{ for } i = 3, 4 \\
 &&& 3.175 \leq x_1 \leq 50.8; \quad 0 \leq x_2 \leq 254; \quad 0 \leq x_3 \leq 254; \quad 0 \leq x_4 \leq 50.8
 \end{aligned}$$

$$z1 = 2.6688 \times 10^4 \text{ (N)}; z2 = 3.556 \times 10^2 \text{ (mm)}; z3 = 2.0685 \times 10^5 \text{ (MPa)}$$

$$z4 = 8.274 \times 10^4 \text{ (Mpa)}; z5 = 6.35 \text{ (mm)}; z6 = 9.377 \times 10 \text{ (Mpa)}$$

$$z7 = 2.0685 \times 10^2 \text{ (Mpa)}$$

$$c1 = 6.74135 \times 10^{-5} / \text{mm}^3; c2 = 2.93585 \times 10^{-6} / \text{mm}^3$$

Table 1. Optimization results for welded beam problem

Design Variable	PSO	BBPSO	Ho-Huu, et al. [5]
x_1 (mm)	5.7833	5.8706	5.730
x_2 (mm)	179.4049	181.5337	201.00
x_3 (mm)	217.2732	210.3381	210.63
x_4 (mm)	6.1813	6.2499	6.240
Best f	2.5140	2.4948	2.5926
Average f	3.1000	2.6400	-
Standard deviation	0.4734	0.1832	-

The RBDO results of this test problem are summarized in Table 1. It is seen that BBPSO obtains the smallest best cost, which is 2.4948 compared with PSO. From Tabel 1, it is seen that BBPSO also obtains the average cost and standard deviation so that BBPSO has a better performance compared to PSO in this test problem. Thus, the convergence behaviors of these two algorithms areas graphed in Figure 5. As seen in Figure 5, PSO can converge faster than BBPSO but it cannot find the best solution.



Figure 5. Convergence behavior of PSO and BBPSO for welded beam problem.

5.2. 15-bar planar truss structure problem

The second test problem is a 15-bar planar truss structure as diagrammed in Figure 6. The goal is to minimize the cross-sectional area so that the minimum weight can be obtained without violating any constraints. The constraints used in this research are for reliability, stress, and shape. The mathematical formulation of this optimization problem can be performed as:

$$\text{find } X = \{A_1, A_2, \dots, A_m, \xi_1, \xi_2, \dots, \xi_n\}$$

$$\text{minimize } f(x) = \sum_{i=1}^m A_i \rho_i L_i$$

$$\text{subjected to } g_1: \text{Check probability of success} \geq 99\%$$

$$g_2: \text{Stress constraints, } |\sigma_i| - |\sigma_i^{max}| \leq 0$$

$$g_3: \text{Shape constraints, } \xi_j^{lower} \leq \xi_j \leq \xi_j^{upper}$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. A_i , ρ_i , L_i , and σ_i are cross-sectional area, weight density, length, and stress of element (i), respectively.

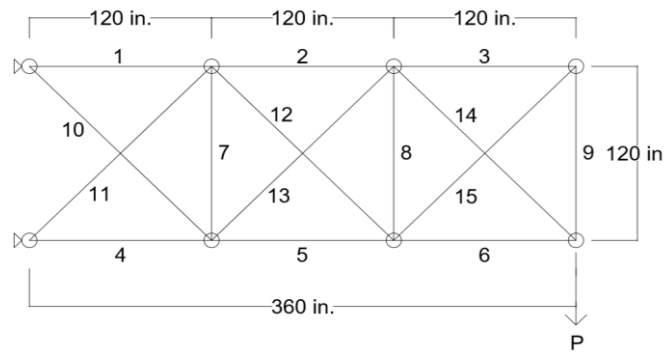


Figure 6. 15-bar problem [8].

Thirty experimental runs with 1000 iterations and 30 populations resulted in the same 120000 function evaluations. This case is also simulated 100 times with modulus of elasticity (E) = 10^4 ksi, weight density (ρ) = 0.1 lb/in.³, and available cross-sectional areas $D = [0.111, 0.141, 0.174, 0.220, 0.270, 0.287, 0.347, 0.440, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.800, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.300, 10.850, 13.330, 14.290, 17.170, 19.180]$ in². Stress limits in tension or compression were 25 ksi. There were 23 design variables in this problem: 15 cross-sectional area variables and eight configuration variables. The configuration variables were the x - and y -coordinates of nodes 2, 3, 6, and 7 and y -coordinates of nodes 4 and 8. However, nodes 6 and 7 were constrained to have the same x -coordinates as nodes 2 and 3, respectively. The side constraints for the configuration variables were $100 \text{ in} \leq x_2 \leq 140 \text{ in}$, $220 \text{ in} \leq x_3 \leq 260 \text{ in}$, $100 \text{ in} \leq y_2 \leq 140 \text{ in}$, $100 \text{ in} \leq y_3 \leq 140 \text{ in}$, $50 \text{ in} \leq y_4 \leq 90 \text{ in}$, $-20 \text{ in} \leq y_6 \leq 20 \text{ in}$, $-20 \text{ in} \leq y_7 \leq 20 \text{ in}$, and $20 \text{ in} \leq y_8 \leq 60 \text{ in}$.

This case is given a non-deterministic load (P) using normal distribution with mean 10 kips on node 8. The sectional area A_i of the i -th element is treated as a normal random design variable and its mean value is the design variable. All random variables are following normal distribution with dispersion $\pm 5\%$. These random variables are modeled by LHS. The results of this test problem are summarized in Table 2, which shows BBPSO also obtains smaller best weight, i.e., 96.583 lb compared with PSO. Other than that, BBPSO also obtains smaller average weight and standard deviation which means BBPSO has a better performance in the second test problem. Figure 7 shows the iteration process of a 15-bar truss structure optimization. In terms of consistency, the convergence behaviors of PSO and BBPSO are graphed in Figure 8. It is seen in Figure 8 that in this case BBPSO can converge faster than PSO and obtain the best solution.

Table 2. Final design of size and shape for the 15-bar truss problem

Variable	PSO	BBPSO	Variable	PSO	BBPSO
A1 (in ²)	1.333	1.081	A14 (in ²)	0.539	0.539
A2 (in ²)	0.954	0.954	A15 (in ²)	0.174	0.111
A3 (in ²)	0.111	0.111	X2 (in)	118.997	119.0362
A4 (in ²)	1.174	1.333	X3 (in)	232.9781	220
A5 (in ²)	0.954	0.954	Y2 (in)	130.1803	108.9745
A6 (in ²)	0.44	0.44	Y3 (in)	119.835	100
A7 (in ²)	0.287	0.111	Y4 (in)	77.5552	54.5177
A8 (in ²)	0.174	0.111	Y6 (in)	6.5362	19.9913
A9 (in ²)	0.111	0.111	Y7 (in)	-13.1163	-17.9019
A10 (in ²)	0.27	0.539	Y8 (in)	59.985	51.0149
A11 (in ²)	0.539	0.141	Best weight (lb)	104.0628	96.5828
A12 (in ²)	0.347	0.141	Average (lb)	1.30	1.070
A13 (in ²)	0.347	0.539	Standard deviation	14.9555	6.0024

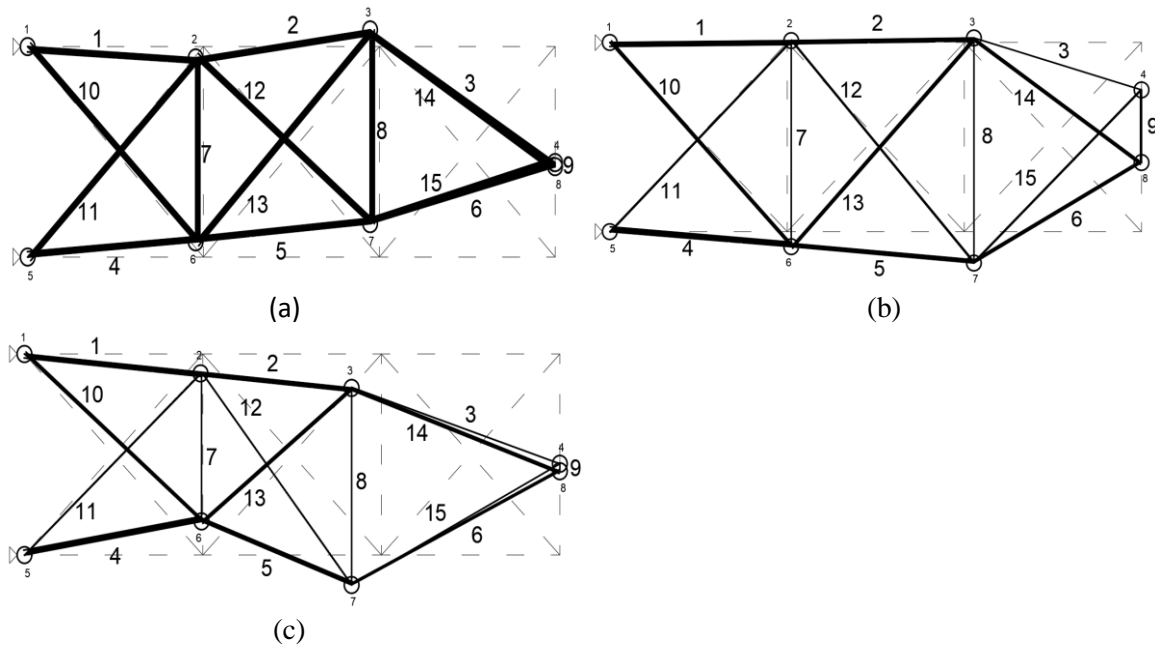


Figure 7. Iteration process of 15-bar truss structure: (a) iteration number 1; (b) iteration number 100; (c) iteration number 1,000.

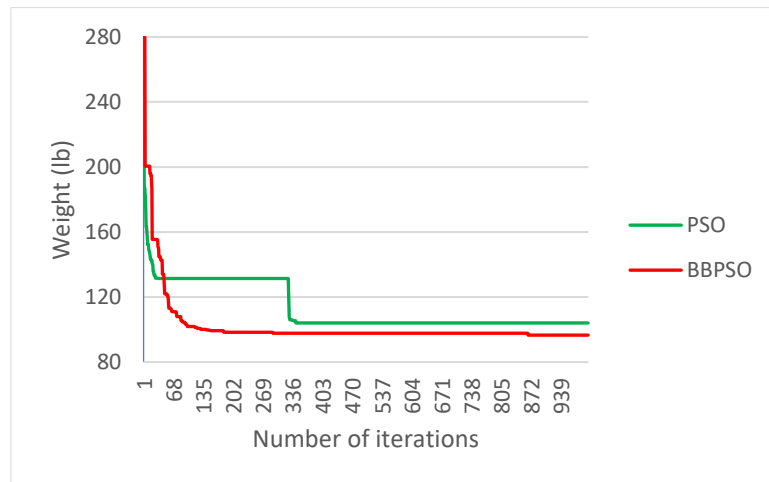


Figure 8. Convergence behavior of PSO and BBPSO for 15-bar truss problem.

6. Conclusion

This research compared the performance of PSO and BBPSO in solving optimization problems. By reviewing these two test problems, it can be seen in Table 1 and Table 2 that with the same number of iterations, BBPSO can obtain a smaller best solution, average solution, and standard deviation compared with PSO which leads to a conclusion that BBPSO has a better performance compared to PSO. It is also shown that combining PSO and LHS can deliver a reliable and robust truss structure design. Moreover, this research shows that RBDO is important in structure design and cannot be neglected.

Acknowledgements

Authors wishing to acknowledge the present research is financially supported by The Ministry of Research, Technology and Higher Education of the Republic of Indonesia under the Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) 2019 Research Grant Scheme.

References

- [1] Assimi H, Jamali A and Nariman-zadeh N 2017 *Swarm and Evolutionary Computation* **37** 90–103
- [2] Eberhart R and Kennedy J 1995 *Proceedings of the Sixth International Symposium on Micro Machine and Human Science* doi:10.1109/mhs.1995.494215
- [3] Gunawan S and Papalambros P Y 2006 *Journal of Mechanical Design* **129** 158–65
- [4] Guo J and Sato Y 2017 *International Journal of Networked and Distributed Computing* **5** 625–38
- [5] Ho-Huu V, Nguyen-Thoi T, Nguyen-Thoi M H and Le-Anh L 2015 *Expert Systems with Applications* **42** 7057–69
- [6] Kaveh A and Mahdavi V R 2015 *Structural Engineering and Mechanics* **53** 847–65
- [7] Matala A 2008 *Sample Size Requirement for Monte Carlo Simulations using Latin Hypercube Sampling* (Helsinki University of Technology, Department of Engineering Physics and Mathematics, Systems Analysis Laboratory)
- [8] Miguel L F F, Lopez R H and Miguel L F F 2013 *Advances in Engineering Software* **56** 23-37
- [9] Prayogo D, Cheng M-Y, Wong F T, Tjandra D and Tran D-H 2018 *Asian Journal of Civil Engineering* **19** 625–38
- [10] Ren Y and Bai G 2010 *Determination of Optimal SVM Parameters by using GA/PSO* **5** 1160-68
- [11] Viana F A, Venter G and Balabanov V 2010 *International Journal for Numerical Methods in Engineering* **82** 135-56