Structural design optimization using particle swarm optimization and its variants

By Doddy Prayogo

Structural design optimization using particle swarm optimization and its variants

D. Prayogo[1], Sebastian[*,1], W. Hauwing[1], F.T. Wong[1], Tjandra, D.[1]

Department of Civil Engineering, Petra Christian University, Jalan Siwalankerto 121-131, Surabaya 60236, Indonesia

Corresponding author email: m21416037@john.petra.ac.id

Abstract. Structural design optimization becomes an extremely challenging and more complex task for most real-world practical applications. A huge number of design variables and complex constraints has contributed to the complexity and nonlinearity of the problems. Mathematical programming and gradient-based search algorithms cannot be used to solve nonlinear problems. Thus, researchers have extensively conducted many experimental studies to address the growing complexity of these problems. Metaheuristic algorithms, which typically use nature as a source inspiration, have been developed over past decades. As one of the widely used algorithm, particle swarm optimization (PSO) has been studied and expanded to deal with many complex problems. Particle swarm optimization and its variants have great accuracy in finding the best solution while maintaining its fast convergence behavior. This study aims to investigate PSO and its variants to solve a set of complex structural optimization problems. Several complex benchmark studies of design problem were provided to study the performance of PSO, linearly decreasing inertia weight-PSO, and bare bones PSO. The results support the potential using of PSO and its variants as an alternative approach to solving structural design optimization problems.

Keywords: optimization, engineering, algorithm, convergency

1. Introduction

Optimization in the world of structural engineering has been used for minimizing building structure cost. Engineers are challenged to solve many calculations during the process of designing, which is time-consuming. Each calculation problem has its own constraint, variable, and parameters that are usually complicated to be solved manually [1]. Designing aims to obtain the best result in order to get the minimum weight design while meeting certain code requirements, which can be achieved with optimization [7].

Particle swarm optimization (PSO) is a metaheuristic calculation method that aims to find the most optimum answer or objective from a case which has different parameters and constraints. This algorithm imitates living organisms; it mimics a group of flock bird or school fish searching for food. When a bird finds its own best location, the pack eventually agree with a global best location [2].

Although PSO has been proven to solve many types of problems, the optimum answer depends on the parameter that is set in the beginning [2]. This paper, therefore, discusses two modified PSO algorithms: Linearly Decreasing Inertia Weight-PSO (LDW-PSO) [3] and Bare Bones-PSO (BB-PSO) [4]. Also, this paper compares three types of PSO in order to find the most reliable and the fastest type to obtain the convergent answer. This study includes three examples of engineering design problems with different constraints and parameters.

2. Particle Swarm Optimization

Particle swarm optimization imitates the movement of a living organism as a particle which can find its own best solution following its own best location. Each time an organism finds its own best location, the whole population immediately find a global best location. This global best location has the most optimum answer from all personal best solutions to get a global solution.

Our calculations start by defining parameters such as inertia weight parameter (w), the cognitive factor parameter (c_1), and the social factor parameter (c_2). Then, the location for each particle is generated randomly with defined upper bound and lower bound. Each particle later starts to search for the answer with a velocity (equation 1). For each iteration, the best location is updated using equation 2.

$$v_i = w \times v_i + rand(0,1) \times c_1 \times (pbest(x_i) - x_i) + rand(0,1) \times c_2 \times (gbest(x_i) - x_i) \quad (1)$$

$$x_i = x_i + v_i \quad (2)$$

Table 1. Particle Swarm Optimization Algorithm.

	Algorithm 1
1.	Initialize PSO parameters
2.	Initialize a population of random particles (solutions)
3.	Evaluate the objective value of each particle
4.	Determine initial pbeat X and gbest X
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 pbest X and pbest F if required
11.	**end** for
12.	Update gbest X and pbest F if required
13.	**end while**

3. Linearly Decreasing Inertia Weight Particles Swarm Optimization

Linearly decreasing inertia weight-PSO differs from the standard PSO. This modified PSO can linearly decrease its inertia weight (w) when each iteration finish [3]. When the value of w is high, the ability to find global search increase. On the other hand, when the value of w decrease the ability to find local search increase [5]. The functions of velocity and updated the location are the same as the original PSO, but the inertia weight is updated using equation 3.

$$w_i = w_1 - (w_1 - w_2)(\text{iter})/(\text{maxiter}) \quad (3)$$

where w_1 and w_2 are the initial and end value of inertia weight, respectively, iter is the number of iterations, and max iter is the maximum number of iterations.

Table 2. Linearly Decreasing Inertia Weight Particles Swarm Optimization Algorithm.

Algorithm 2
1. Initialize PSO parameters
2. Initialize a population of random particles (solutions)
3. Evaluate the objective value of each particle
4. Determine initial pbeat X and gbest X
5. **while** termination criteria are not satisfied **do**
6. Update inertia weight
7. **for** each particle **do**
8. Update the velocity for the particle
9. Update the new location for the particle
10. Determine the objective value for the particle in its new location
11. Update pbest X and pbest F if required
12. **end** for
13. Update gbest X and pbest F if required
14. **end while**

4. Bare Bones Particles Swarm Optimization

Different from the abovementioned types of PSO, Bare Bones PSO ignore all parameters and does not need to use velocity to find a new location. Bare Bones PSO mainly uses Gaussian distribution. The new location is updated based on the location, which is the mean between the personal best solution and the global best solution. The formula is shown in equation 4.

$$\mu = \frac{pbest+gbest}{2}$$

$$\sigma = |pbest - gbest| \tag{4}$$

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

Table 3. Bare Bones Particles Swarm Optimization Algorithm.

Algorithm 3
1. Initialize PSO parameters
2. Initialize a population of random particles (solutions)
3. Evaluate the objective value of each particle
4. Determine initial pbeat X and gbest X
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 pbest X and pbest F if required
9. **end** for
10. Update gbest X and pbest F if required
11. **end while**

5. Test Problem and Results Particles Swarm Optimization

This section presents three cases that were solved using three types of PSO. Each case addresses an engineering design problem, which has different constraint and parameters.

5.1. Case 1-A three-bar truss design

This case involves 3-bar planar truss structure, as shown in Figure 1 [6]. The weight of the structure minimizes subject to stress constraint on each bar element. The objective function of this case is to find the optimal value of cross-sectional areas (A_1, A_2). The function of this case is given as follows:

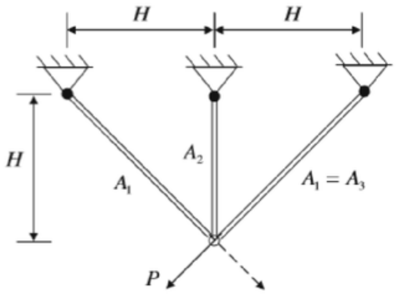

Figure 1. Three-bar truss.

Minimize: $f(A_1, A_2) = \left(2\sqrt{2}A_1 + A_2\right) \times l$

Subject to

$$g_1 = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P - \sigma \leq 0$$

$$g_2 = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2}P - \sigma \leq 0$$

$$g_3 = \frac{1}{A_1 + \sqrt{2}A_2}P - \sigma \leq 0$$

Where

$$0 \leq A_1 \leq 1 \text{ and } 0 \leq A_2 \leq 1;$$
$$l = 100 \text{ cm},$$
$$P = 2\,KN/cm^2$$
$$\sigma = 2\,KN/cm^2$$

Table 4 shows the statistical result for the best objective value by the three methods. Table 5 compares the result obtained by the three methods. Figure 2 shows the convergence behavior of each method.

Table 4. Statistical result for Case 1.

	Best	Avg	Worst	SD	Time (sec)
PSO	263.8959	263.9829	264.7531	0.187233	0.166044
LDW-PSO	263.8959	266.4239	282.8427	6.550067	0.159809
BB-PSO	263.8959	263.8985	263.9198	0.004515	0.085384

Table 5. Comparison of optimization result for Case 1.

	PSO	LDW-PSO	*BB-PSO*
A_1	0.79494	0.78789	0.78902
A_2	0.39080	0.41046	0.40727
$g1$	-0.22099	-0.20478	-0.09537
$g2$	-1.26311	-1.25670	-1.16958
$g3$	-0.95788	-0.94808	-0.92579
f	263.92410	263.89640	263.89593

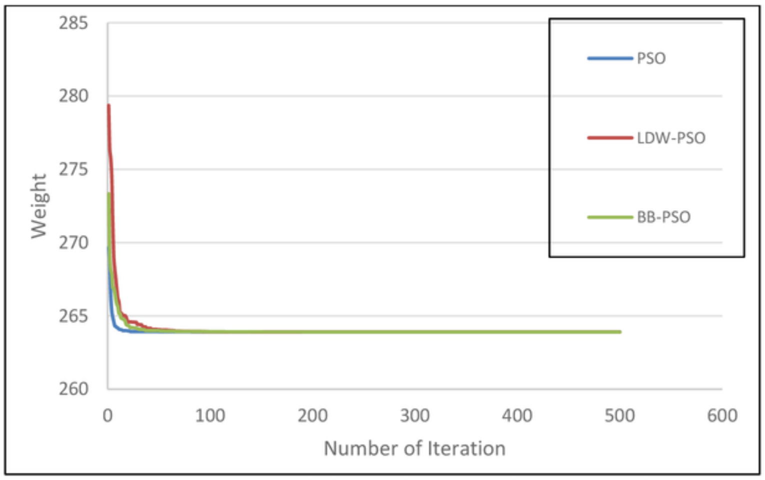

Figure 2. Convergence behavior for Case 1 for each algorithm.

5.2. Case 2-Tubular column design

Figure 3 shows a tubular column that receives an axial load (*P*) of 2500 kg [7]. The column material has a yield stress (ry) of 500 kg/cm^2, a modulus of elasticity (E) of 0.85 9 106 kg/cm^2, and a density (q) of 0.0025 kg/cm3. The length (L) of the column is 250 cm. This case is aimed to find the minimum cost of material and construction cost (f). The constraint and the optimization function are as follows:

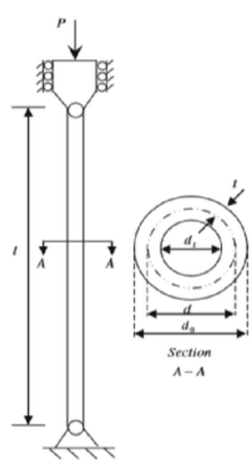

Figure 3. The tubular column.

Minimize: $f(\mathrm{d}, \mathrm{t}) = 9.8dt + 2d$
Subject to:

$$g_1 = \frac{P}{\pi d t \sigma_y} - 1 \le 0$$

$$g_2 = \frac{8PL^2}{\pi^3 E d t (d^2 + t^2)} - 1 \le 0$$

$$g_3 = \frac{2.0}{d} - 1 \le 0$$

$$g_4 = \frac{d}{14} - 1 \le 0$$

$$g_5 = \frac{0.2}{t} - 1 \le 0$$

$$g_6 = \frac{t}{0.8} - 1 \le 0$$

Table 6 represents the statistical result for the best objective value by the three methods. Table 7 compares the result obtained by the three methods. Figure 4 shows the convergence behaviour of each algorithm.

Table 6. Statistical result for Case 2.

	Best	Avg	Worst	SD	Time (sec)
PSO	26.4995	26.4995	26.4995	4.28E-11	0.168175
LDW-PSO	26.4995	26.8337	31.5127	1.271885	0.164555
BB-PSO	26.4995	26.4995	26.4995	3.79E-08	0.082097

Table 7. Comparison of optimization result for Case 2.

	PSO	LDW-PSO	*BB-PSO*
d	5.45116	5.45116	5.45116
t	0.29196	0.29196	0.29196
g1	-3.33066e-16	-2.22045e-16	-1.80071e-09
g2	0	-2.22045e-16	-2.15574e-10
g3	-0.63310	-0.63310	-0.63310
g4	-0.61063	-0.61063	-0.61063
g5	-0.31499	-0.31499	-0.31499
g6	-0.63504	-0.63504	-0.63504
f	26.49950	26.49950	26.49950

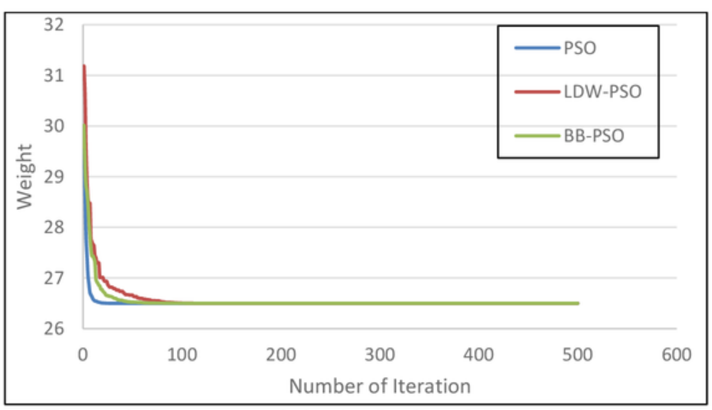

Figure 4. Convergence behavior for Case 2 for each algorithm.

5.3. Case 3-Tension/Compression Spring

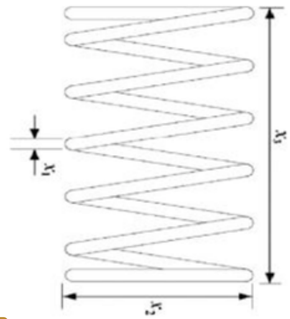

Figure 5. Tension/compression spring design problem.

Figure 5 shows a spring design with three variable, which are wire diameter (x_1), mean coil diameter (x_2), and the number of the active coils (x_3). The objective of this case is to find the minimum tension/compression spring weight. The function and constraint are defined as follows:

Minimize: $f(X) = (x_3 + 2)x_2 x_1^2$

Subject to:

$$g_1(X) = 1 - \frac{x_2^2 x_3}{71785 x_1^4} \leq 0$$

$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566 x_2 x_1^3 - x_1^4} + \frac{1}{5108 x_1^2} - 1 \leq 0$$

$$g_3(X) = 1 - \frac{140.45 x_3}{x_2^2 x_3} \leq 0$$

$$g_4(X) = \frac{x_2 + x_1}{1.5} - 1 \leq 0$$

with boundary conditions: $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$.

Table 8 represents the statistical result for the best objective value by the three methods. Table 9 compares the result obtained by the three methods. Figure 6 shows the convergence behaviour of each algorithm.

Table 8. Statistical result for Case 3.

	Best	Avg	Worst	SD	Time (sec)
PSO	0.004895	0.00467	0.00574	0.000418	0.1689
LDW-PSO	0.004869	0.00510	0.00574	0.000391	0.1773
BB-PSO	0.004869	0.00487	0.00488	2.57E-06	0.0881

Table 9. Comparison of optimization result for Case 3.

	PSO	LDW-PSO	*BB-PSO*
x_1	0.05000	0.05000	0.05000
x_2	0.37389	0.37443	0.37401
x_3	3.20934	3.20012	3.20744
g1	0.99972	0.99972	0.99972
g2	-0.82619	-0.82619	-0.82619
g3	-56179	-56179	-56179
g4	-0.93333	-0.93333	-0.93333
f	0.004895	0.004869	0.004869

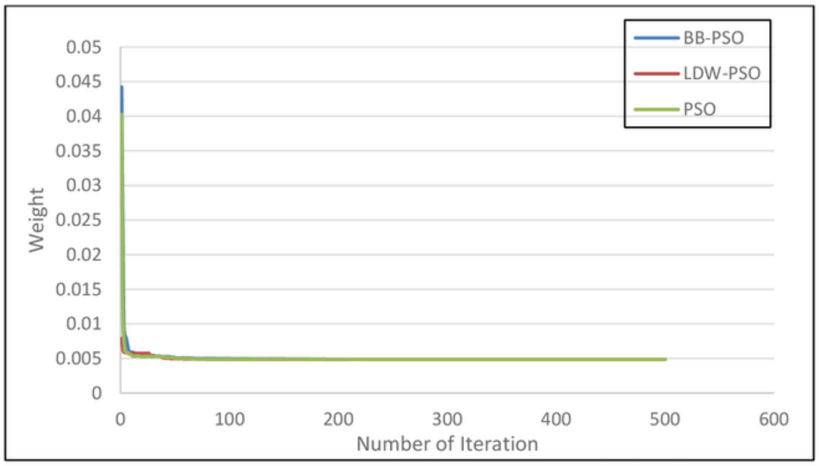

Figure 6. Convergence behavior for Case 3 for each algorithm.

6. Conclusion

This paper compared the result of three problem cases that were optimized with three different types of PSO. The results showed that, with the same number of iterations for each case and each PSO, BB-PSO has the fastest calculation time and always gives the best result. Meanwhile, the standard PSO does not give the best results. The BB-PSO case also has the smallest standard deviation, which means that each iteration has a stable result and is the fastest to obtain convergence in the results. Since the normal PSO has the smallest standard deviation, the performance of the algorithm still depends on the cases and the parameter of each case.

7. References

[1] Cheng M-Y, Prayogo D, Wu Y-W, and Lukito M M 2016 *Autom. Constr.* **69** 21–33

[2] Kennedy J and Eberhart R 1995 *Proc. of IEEE Int. Conf. on Neural Networks (Perth)* (New York: IEEE) pp 1942–8

[3] Xin J, Chen G and Hai Y 2009 *Joint Conf. on Computational Sciences and Optimization, CSO (Sanya)* (New York: IEEE) pp 505–8

[4] Guo J and Sato Y 2017 *Int. J. Networked Distrib. Comput.* **5** 143

[5] Shi Y and Eberhart R 1998 *IEEE World Cong. on Computational Intelligence Evolutionary Computation Proc. (Anchorage)* (New York: IEEE) pp 69–73

[6] Nowcki H 1974 *Computer Applications in the Automation of Shipyard Operation and Ship Design* ed Y Fujita *et al* (New York: Elsevier) pp 327–38

[7] Rao S S 1996 *Engineering Optimization: Theory and Practice* 3rd edn (Chichester: John

Wiley & Sons).
[8] Hare W, Nutini J and Tesfamariam S 2013 *Adv. Eng. Softw.* **59** 19–28

Structural design optimization using particle swarm optimization and its variants

ORIGINALITY REPORT

15%

SIMILARITY INDEX

PRIMARY SOURCES

1	Amir Hossein Gandomi, Xin-She Yang, Amir Hossein Alavi. "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", Engineering with Computers, 2011 Crossref	41 words — 2%
2	www.cs.uoi.gr Internet	33 words — 1%
3	"Swarm, Evolutionary, and Memetic Computing", Springer Science and Business Media LLC, 2016 Crossref	30 words — 1%
4	Rao. "Introduction to Optimization", Engineering Optimization, 07/02/2009 Crossref	29 words — 1%
5	M. Senthil Arumugam. "On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems", Discrete Dynamics in Nature and Society, 2006 Crossref	28 words — 1%
6	open.library.ubc.ca Internet	24 words — 1%
7	mafiadoc.com Internet	18 words — 1%
8	www.matec-conferences.org Internet	17 words — 1%

9 Singiresu S Rao. "Engineering Optimization Theory and Practice", Wiley, 2019
Crossref
15 words — 1%

10 Doddy Prayogo, Min-Yuan Cheng, Yu-Wei Wu, Albertus Arief Herdany, Handy Prayogo. "Differential Big Bang - Big Crunch algorithm for construction-engineering design optimization", Automation in Construction, 2018
Crossref
14 words — 1%

11 ijcsi.org
Internet
11 words — < 1%

12 Kyle Robert Harrison, Andries P. Engelbrecht, Beatrice M. Ombuki-Berman. "Inertia weight control strategies for particle swarm optimization", Swarm Intelligence, 2016
Crossref
11 words — < 1%

13 Communications in Computer and Information Science, 2015.
Crossref
10 words — < 1%

14 Bahriye Akay, Dervis Karaboga. "Artificial bee colony algorithm for large-scale problems and engineering design optimization", Journal of Intelligent Manufacturing, 2010
Crossref
9 words — < 1%

15 Efrén Mezura-Montes. "Useful Infeasible Solutions in Engineering Optimization with Evolutionary Algorithms", Lecture Notes in Computer Science, 2005
Crossref
9 words — < 1%

16 Jaberipour, M.. "Two improved harmony search algorithms for solving engineering optimization problems", Communications in Nonlinear Science and Numerical Simulation, 201011
Crossref
9 words — < 1%

17 Debao Chen, Renquan Lu, Suwen Li, Feng Zou, Yajun Liu. "An enhanced colliding bodies optimization and its application", Artificial Intelligence Review, 2019
8 words — < 1%

Crossref

18	Yongquan Zhou, Ying Ling, Qifang Luo. "Lévy flight trajectory-based whale optimization algorithm for engineering optimization", Engineering Computations, 2018 Crossref	8 words — < 1%
19	A. Ellabib, A. Nachaoui. "On the numerical solution of a free boundary identification problem", Inverse Problems in Engineering, 2001 Crossref	8 words — < 1%
20	Engineering Computations, Volume 29, Issue 5 (2014-09-16) Publications	8 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		