Hindawi
Advances in Civil Engineering
Volume 2023, Article ID 9806200, 9 pages
https://doi.org/10.1155/2023/9806200

Research Article

Multistrategy Cooperation Particle Swarm Optimization for FEM Model Update of the Continuous Warren Truss Steel Railway Bridge

Yiqiang Li and Liming Zhou

School of Safety Engineering and Emergency Management, Shijiazhuang Tiedao University, Shijiazhuang 050043, Hebei, China

Correspondence should be addressed to Yiqiang Li; liyiq@stdu.edu.cn

Received 26 July 2022; Revised 14 December 2022; Accepted 6 January 2023; Published 22 February 2023

Academic Editor: Fabio Di Trapani

It is very difficult to obtain an accurate finite element method (FEM) model to further analyze structural mechanical properties. Therefore, as the main means of establishing accurate models, the model update has become a research hotspot in the dominion of bridge engineering. Particle swarm optimization (PSO) has the characteristics of being easy to implement, but it is easy to fall into the local optimum. Therefore, multistrategy cooperation particle swarm optimization (MCPSO) that balances exploration and exploitation of particle swarm is proposed. This algorithm achieves the goal of balancing exploration and exploitation by adopting different combinations of particle swarm velocity update strategies in different iteration stages. The application effects of MCPSO in the FEM model update of the continuous Warren truss steel railway bridge are compared and analyzed, and the results show that the algorithm proposed in this paper outperforms the standard PSO (SPSO) algorithm. This paper provides a more effective algorithm for bridge model updates.

1. Introduction

Particle swarm optimization (PSO) is a stochastic population-based optimization method proposed by Shi and Eberhart [1]. Since the algorithm was proposed, researchers have carried out long-term research and improvement work on it. It has the disadvantage that it is easy to fall into the local optimum [2–4]. There are unimodal and multimodal problems in engineering practice. Unimodal problems have only one extreme point, while multimodal problems have multiple extreme points. For unimodal problems, this is advantageous, but when encountering multimodal problems, it is easy to obtain results that deviate from the global optimum [5–7]. The current solution to this problem is often to use some velocity update strategies to balance the exploration and exploitation of particle swarms in the solution domain [7–9].

The article [10, 11] indicates that multiswarm-combining dynamical topology is an effective strategy to improve PSO. Li et al. [12] proposed four strategies to update the particles' positions called a self-learning particle swarm optimizer (SLPSO), in which each particle has four cooperation strategies implemented by an adaptive learning framework and can choose the optimal strategy according to its own local fitness landscape. Tang et al. [13] proposed multistrategy adaptive particle swarm optimization (MAPSO), which evaluates the population distribution, alternates strategy in real time, and has enhanced the research ability of PSO variants. Gülcü and Kodaz [14] proposed PSO variants, which set swarms as master and slave subswarms and make them work cooperatively and concurrently. Bonyadi and Michalewicz [15] conducted review research on PSO, and it is believed that the combination of multiple speed update strategies is one of the methods to improve the performance of PSO. Wang and Song [16] separated particles near the global best position and other particles and updated them in the population in different ways. It has good performance and high search precision than PSO and some other optimization algorithms. Xia et al. [17] presented dynamic multiswarm particle swarm optimization based on an elite learning strategy (DMS-PSO-EL), in which the whole computational process is divided into a former stage and a later stage. Tang et al. [18] presented dynamic multiswarm global particle swarm optimization (DMS-GPSO), which consists of two novel strategies balancing exploration and exploitation abilities.

However, benchmark functions are often used to test the pros and cons of most PSO variants [19–22], which are one sided. The ultimate goal of researching algorithms is to apply them to engineering practice and solve practical engineering problems [23]. In order to solve practical engineering problems, it is meaningful to design algorithms for this engineering problem. This paper proposes a PSO variant with combined strategies. The algorithm evaluates the PSO results through roulette and selects different particle speed update methods according to the evaluation results, to achieve the purpose of improving the performance of the algorithm.

At present, researchers have developed some bridge structure model update methods [24–27], and the use of swarm intelligence algorithms to carry out bridge structure model update research is also one of the current research hotspots. Ho et al. presented a multiphase model update approach to system identification of a real railway bridge using vibration test results [28]. Bayraktar et al. (2010) [29] obtained a railway bridge's dynamic characteristics experimentally, and according to them, the FEM model of the bridge was updated by changing some uncertain parameters, material properties and boundary conditions. Arisoy and Erol [30] compared an FEM model and experimental dynamic properties of a steel railway bridge and updated the FEM model by tuning material properties to match the real bridge. Tran-Ngoc et al. [31] updated a large-scale steel truss bridge using PSO and the genetic algorithm (GA) and found that PSO provides a better accuracy FE model and reduces the calculation cost compared to GA.

From the literature research, it is known that the research on the improvement of particle swarm optimization is one of the research hotspots of the current swarm evolutionary algorithm. However, no one PSO variant is superior to all algorithms. Therefore, for specific problems, it is necessary to popularize the application of improved PSO algorithms and design targeted improved PSO algorithms. At the same time, the current standard for using benchmark functions to test algorithm performance is one sided. Attention should be paid to the comparative application effect of algorithms in practical engineering, and effectively improved algorithms should be recommended for solving practical engineering problems.

This study makes several contributions to the current literature. 1. It is not a simple mathematical average that the article introduces roulette into particle swarm velocity updating strategy evaluation but considers the distribution of the particle updating process, which is more objective. 2. The improvement of the proposed algorithm is embodied in the comprehensive strategy to realize an optimal combination of the existing PSO algorithm. The proposed algorithm is based on the evaluation result. This method can be used as a frame to replace the subswarm velocity update formula or introduce other optimization algorithms such as the ant colony algorithm, bee colony algorithm, wolf colony algorithm, and genetic algorithm to achieve the purpose of improving the competitiveness of the algorithm.

The remainder of the article is organized as follows: Section 2 introduces the basic concepts of PSO and the related PSO variants. Section 3 elaborates the improvement strategies in proposed MCPSO. Section 4 applies the MCPSO to the structural update problem for the continuous Warren truss steel railway bridge, compares the results of MCPSO and SPSO, and evaluates the effectiveness of MCPSO. Finally, Section 5 provides the summary and conclusion.

2. PSO and Related PSO Variants

Canonical PSO is a swarm intelligence algorithm inspired by birds' natural behavior in search of food. It imitates the behavior they move, the positions they change, and trajectories to search their destinations. In mathematics, a single bird is regarded as a particle. Therefore, in the simulation process, each particle has two position-related characteristics, namely, the current position and moving velocity. In the PSO algorithm, each particle describes a feasible solution to a problem, a single particle is seen as a point in a D-dimensional space, and a population of N particles is used for an optimization problem generally. The position of the ith particle is represented as $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{iD})$, and its velocity is represented as $\mathbf{v_i} = (v_{i1}, v_{i2}, \cdots, v_{ij}, \cdots, v_{iD})$. The best previous searched particle position with the best fitness value is saved and denoted as $\boldsymbol{p_i} = (p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots, p_{iD})$. Meanwhile, the particle position with the smallest objective function value among all the swarm particles is denoted as $\mathbf{g} = (g_1, g_2, \cdots, g_j, \cdots, g_D)$. The position and velocity of the ith particle at iteration t of the PSO algorithm are denoted as $x_i(t)$ and $v_i(t)$, respectively. Thus, the particle's position element is expressed as follows:

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \tag{1}$$

From a variety of velocity update formulas, the classic particle velocity update formulas that can cover all kinds of velocity update strategies are selected as the subswarms' velocity update formulas of the proposed algorithm in this article.

2.1. SPSO. Shi and Eberhart [1] introduced standard PSO, which added an inertia weight ω in PSO, to regulate the effect of the previous velocity on the updated velocity. The particles' velocities at iteration $t+1$ consist of three components. The first part is the velocity at iteration t, the second part is the individuality behaviors of the particles controlled by the random number $r_{1,ij}$ uniformly distributed in [0, 1] and c_1 referred to as a cognitive parameter, and the last part is the sociality behavior of the particle considering the random number $r_{2,ij}$ uniformly distributed in [0, 1] and c_2 called a social parameter. Thus, the jth-dimensional velocities of the particle $\mathbf{x}_i$ at t + 1 iteration are manipulated iteratively as follows:

$$\begin{aligned} v_{ij}(t+1) &= \omega(t)v_{ij}(t) + c_1(t)r_{1,ij}(t)\left[p_{ij}(t) - x_{ij}(t)\right] + c_2(t)r_{2,ij}(t)\left[g(t) - x_{ij}(t)\right], \\ \zeta(t) &= \zeta_e + (\zeta_b - \zeta_e)\frac{(t_N - t)}{t_N}, \end{aligned} \tag{2}$$

where ζ is w, c_1 or c_2; if ζ is w, $\zeta_b = 0.9$ and $\zeta_e = 0.2$; if ζ is c_1, $\zeta_b = 2.5$ and $\zeta_e = 0.5$; and if ζ is c_2, $\zeta_b = 0.5$ and $\zeta_e = 2.5$. t_N is total number of iterations.

This process of the velocity element $v_{ij}(t+1)$, position element $x_{ij}(t+1)$, individual best element $p_{ij}(t)$, and global best $g(t)$ is iterated until a predetermined stopping condition is met.

2.2. UPSO. Focusing on the balance abilities between global exploration and local exploitation, Parsopoulos and Varhatis [8] proposed unified PSO (UPSO). It distinguishes the local and global velocity update of PSO. The equations of global and local velocity updates are shown as follows:

$$\begin{aligned} G_{ij}(t+1) &= \xi\left\{v_{ij}(t) + c_3 r_{3,ij}(t)\left[p_{ij}(t) - x_{ij}(t)\right] + c_4 r_{4,ij}(t)\left[g(t) - x_{ij}(t)\right]\right\}, \\ L_{ij}(t+1) &= \xi\left\{v_{ij}(t) + c_5 r_{5,ij}(t)\left[p_{ij}(t) - x_{ij}(t)\right] + c_6 r_{6,ij}(t)\left[p_{n,ij}(t) - x_{ij}(t)\right]\right\}, \end{aligned} \tag{3}$$

where ξ denotes the constriction factor, $p_{n,ij}(t)$ denotes the j element of the personal best position of the best neighborhood n of the particle $\mathbf{x}_i(t)$, $G_{ij}(t+1)$ is the global version velocity of the ith particle on the jth dimension, and $L_{ij}(t+1)$ is the local version velocity.

The velocity update of UPSO is calculated as follows:

$$v_{ij}(t+1) = u r_{7,ij}(t) G_{ij}(t+1) + (1-u)L_{ij}(t+1), \tag{4}$$

where u denotes the unification factor controlling the global and local influence on the velocity update, and

$$r_{7,ij}(t) \sim N(\mu, \sigma^2). \tag{5}$$

2.3. CLPSO. Comprehensive learning particle swarm optimization (CLPSO) [32] was proposed for better exploration. The velocity updates are as follows:

$$v_{ij}(t+1) = \omega(t)v_{ij}(t) + c_7 r_{8,ij}(t)\left[p_{fi(j)}(t) - x_{ij}(t)\right], \tag{6}$$

where f_i determines which p_{ij} is used to guide the particle x_{ij}. The decision depends on the learning probability p_{ci}. The learning probability p_{ci} is expressed as follows:

$$p_{ci} = a + b\frac{\exp((10(i-1)/N - 1) - 1)}{\exp(10) - 1}, \tag{7}$$

where a and b are the two parameters tuning the learning probability, $a = 0$ and $b = 0.5$ in this article.

3. Strategy for Proposed PSO

The proposed strategies improve PSO as follows: 1. Three algorithms are combined with different advantages, and the characteristics of different algorithms are integrated to work together to improve the performance of PSO variants. 2. The iterative process of the algorithm is divided into several calculation periods to prevent the accumulation of the limitations of a single algorithm strategy. 3. After a calculation period, each particle update position is evaluated by objective function value change and the evaluation result guides the number of particles allocated to different particles' velocity update strategies in the next calculation period.

The swarm grouping strategy is shown in Figure 1. The strategy with the best performance obtains the subswarm with the largest number of particles, and it is the green part in Figure 1. The strategy with the second best performance obtains the subswarm with a larger number of particles and is shown as the blue part in Figure 1. The strategy with the worst performance obtains the subswarm with the least number of particles, and it is the yellow part in Figure 1. The pros and cons of the strategy are evaluated in a computing cycle, and the strategy with the largest change in the objective function in this cycle is the optimal strategy, and vice versa. Figure 2 shows the graphical flowchart of the MCPSO algorithm.

This research proposes a comprehensive strategy for the PSO algorithm that dynamically adjusts the number of subswarm particles. The algorithm divides all particles into S subswarms ($S = 1$, 2, or 3 in this article) with strategies S and adjusts the particles in subswarms after each calculation period (C_p). The subswarm with the strategy that significantly reduces the objective function value is allocated a larger number of particles. The objective function value's average change is expressed as

$$\Delta f_s(t) = \frac{\sum_{j=1}^{T_s}\left[f\left(\mathbf{x}_j(t)\right) - f\left(\mathbf{p}_j(t)\right)\right]}{T_s}, \tag{8}$$

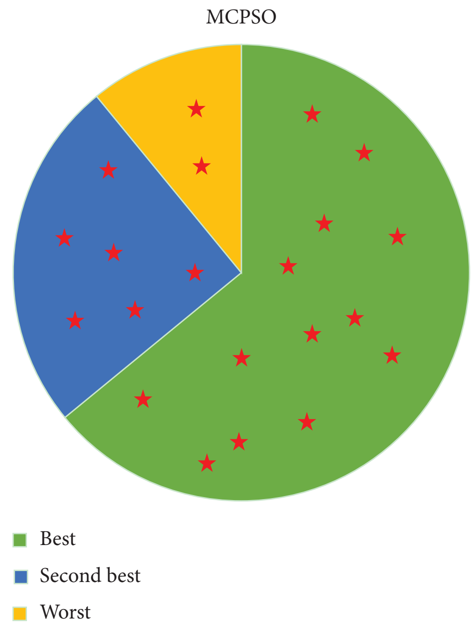


Figure 1: Particle swarm grouping to strategies.

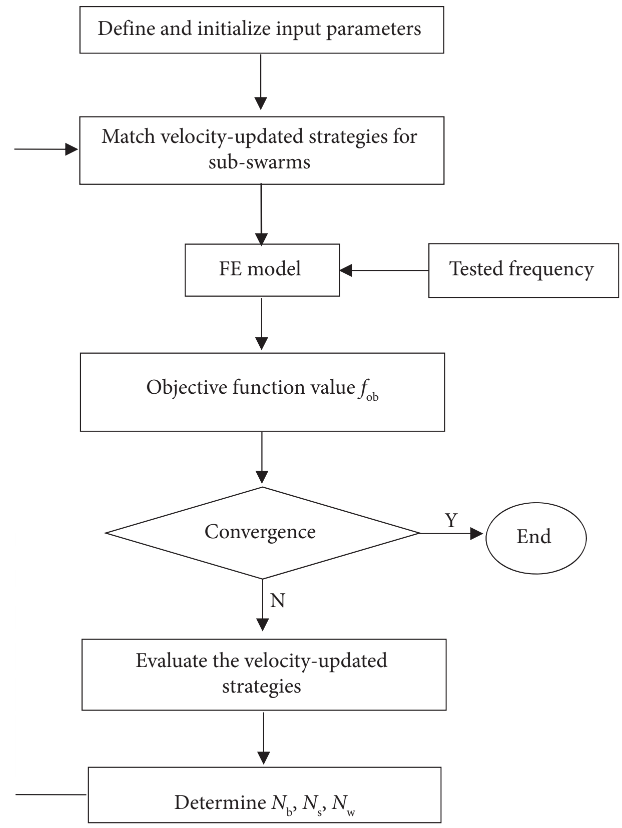


Figure 2: Graphical flowchart of the MCPSO algorithm.

where T_s is calculation times for the subswarm S with the strategy S. The best strategy is the strategy S resulting in maximum $\Delta f_s(t)$, the earlier stage of the calculation period, or else it is that resulting in minimum $\Delta f_s(t)$, the later stage of the calculation period.

The pseudocode of the MCPSO algorithm is shown in Table 1. N_s ($S \in \{1, 2, 3\}$) is the proportion of subswarms S in particles, N is the population size, and C_p is the calculation period. It outputs the optimal solution g, and it is also the particle position with the smallest objective function value among all the swarm particles.

4. Engineering Application

In this section, MCPSO proposed in this paper is applied to the FEM model update of the continuous Warren truss steel railway bridge.

4.1. Objective Function. Natural frequencies and mode shapes are well known as the sensitive factors for the changes in structure properties. The problem of FEM model update is transformed into the optimization problem of minimizing the objective function describing the gaps between the measured responses and the calculated ones.

The objective function for FEM model update is expressed as [33]

$$f_{ob} = \sum_{j=1}^{NF} w_{\omega j} \Delta\omega_j^2, \qquad \Delta\omega_j = \left| \frac{\omega_j^C - \omega_j^M}{\omega_j^M} \right|, \tag{9}$$

where $w_{\omega j}$ is the weight factor corresponding to the jth natural frequency, NF is the number of frequencies and mode shapes used in calculation, and ω_j^C and ω_j^M are the jth calculated and measured natural frequencies, respectively.

4.2. Introduction of the Bridge. The example bridge is a continuous Warren truss steel railway bridge, with a span of $(48 + 3 \times 64 + 48)$ m, and the span layout diagram is shown in Figure 3. The main truss pattern is the Warren type, the height is 11 m, the internode length is 8 m, and the center distance of the main truss is 5.75 m. The upper and lower chords of the main truss are welded H-shaped sections, some diagonal bars are box-shaped sections, the bridge deck structure is vertical and horizontal beams, and the upper and the lower horizontal longitudinal connections are welded I-shaped sections. The section parameters are shown in Table 2. The bridge bearing adopts a steel bearing, and the movable bearing is a roller bearing. The bridge was built in 1996 and has been in service for more than 25 years. Currently, the main trains in operation are C70 trains, with an annual transportation volume of about 50 million tons. Due to the long service period of the bridge and the existence of microdamages, force analysis can provide suggestions for later maintenance, which is conducive to the safe operation of the bridge.

Compared with other simply supported bridges, continuous truss steel girder bridges have greater vertical and lateral stiffness and their deflection curves are relatively

Table 1: Pseudocode of the MCPSO algorithm.

Algorithm 1 The pseudocode of MCPSO algorithm
Input:
(1) Initialize the parameters for all subswarms: N, t_N, $x_{\max}$, $x_{\min}$, $v_{\max}$, $v_{\min}$, set $C_p = 10$, $N_S = (0.33N, 0.33N, 0.33N)$
(2) Initialize the parameters of SPSO: $w_b = 0.9$, $w_e = 0.2$, $c_{1b} = 2.5$, $c_{1e} = 0.5$, $c_{2b} = 0.5$, $c_{2e} = 2.5$
(3) Initialize the parameters of UPSO: $\xi = 0.729$, $c_3 = c_4 = c_5 = c_6 = 2.05$, $u = 0.1$, $\mu = 0$, $\sigma = 0.01$
(4) Initialize the parameters of CLPSO: $c_7 = 1.49445$, $a = 0$, $b = 0.5$
(5) Evaluate the objective function values $f(x_i)$, $i = 1, 2, \ldots, N$
(6) Evaluate the objective function values change using (8)
(7) **while** $(t < t_N)$ or (stop criterion) **do**
(8) **if** $\mathrm{mod}(t, C_p) == 0$
(9) Find the best strategy and allocate the most particles to it
(10) Particle population size with the best strategy, the second best strategy, and the worst strategy are $N_b = 0.7N$, $N_s = 0.2N$, and $N_w = 0.1N$, respectively
(11) **end if**
(12) Update velocity of particles in subswarms by MCPSO
(13) Update positions for all the subswarms
(14) Evaluate the objective value f of each particle i in the population
(15) Find out the best previous positions of individuals within one subswarm and the global best position, respectively
(16) **end while**
Output: g

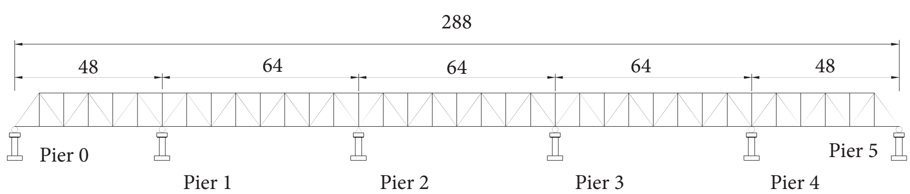
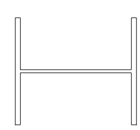
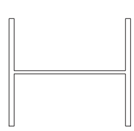
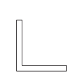
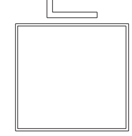


Figure 3: Span layout of through continuous steel truss girder (m).

Table 2: Transverse section attribute (mm).

Truss	Section form	Section composition	Truss	Section form	Section composition
Bottom chord		2-420 × 12 1-436 × 12 2-420 × 12 1-412 × 16	Side diagonal member		2-450 × 12 1-436 × 10 2-460 × 16 1-428 × 12
Top chord		2-420 × 12 1-436 × 12 2-420 × 20 1-420 × 123 2-420 × 16 1-428 × 12	Portal strut		1-220 × 16 1-220 × 12 1-772 × 10 1-180 × 12 1-180 × 12 1-296 × 10
Hip vertical		2-260 × 16 1-436 × 12	Floor beam		2-240 × 24 1-1252 × 12
Top lateral bracing		2-180 × 10 1-220 × 8	Brake frame		2-140 × 12 1-196 × 10
Bottom lateral bracing		2-180 × 10 1-220 × 10	Stringer connection		1-100 × 100 × 10 1-90 × 90 × 10
Stringer		2-240 × 20 1-1020 × 12	Inclined end post		2-460 × 20 2-420 × 20

smooth. Continuous truss steel girder bridges usually have only one fixed support for the whole bridge. In this article, the fixed support of the continuous steel truss bridge is set at pier 2. The braking force of the continuous steel truss bridge is all here, so at the fixed support, the stress on the pier and foundation of the seat will be greater than that of other parts.

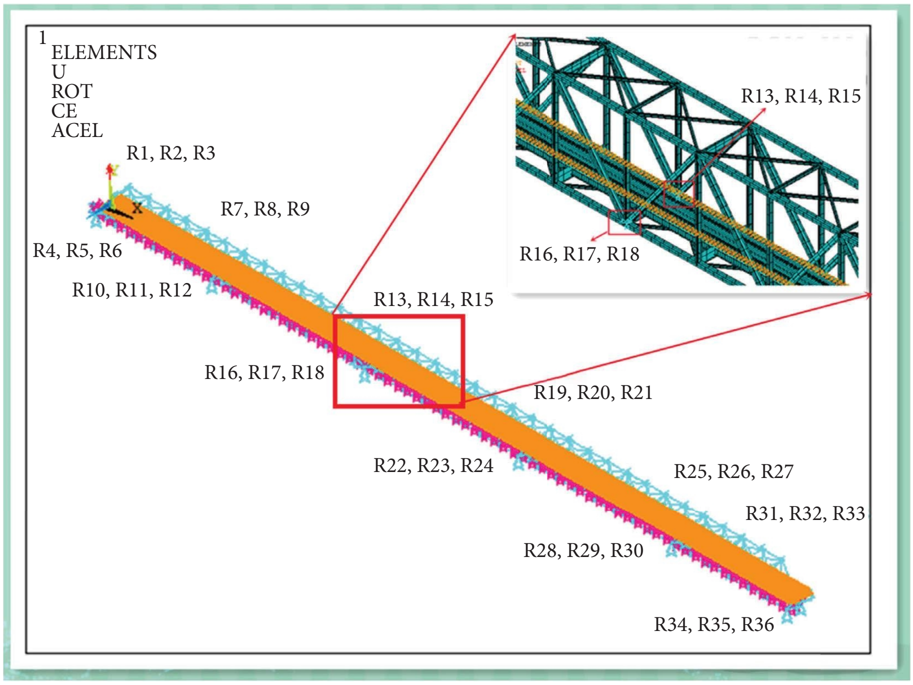


Figure 4: Finite element model of the bridge structure.

4.3. Truss Steel Railway Bridge FEM Model. The continuous Warren truss steel railway bridge model is established by using general finite element software ANSYS, and the main truss, longitudinal and transverse beams, and connection systems are all simulated by three-dimensional Timoshenko beam element BEAM189, as shown in Figure 4. The connections between the main truss, beams, and connecting system are all rigid connections. In the design, to reduce the height of the bridge building, the longitudinal and transverse beams are set at unequal heights and rigid beam unit MPC184 is used to simulate the rigid connection between the longitudinal beams and transverse beams. The sleepers and rails are simulated by three-dimensional Euler beam unit BEAM4, with the sleeper interval being 0.4 m and the rails 60 kg/m. The element sizes of the stringer, the rail, and the rest component are 0.2 m, 0.1 m, and 1.0 m, respectively. The steel is 16 Mnq, the elastic modulus is 2.06×10^5 MPa, the density is 7850 kg/m^3, and Poisson's ratio is 0.3. According to the quality of the gusset plate and high-strength bolts of the continuous steel truss bridge given by the design drawings, MASS21 mass elements are used, which are evenly distributed at each node, and other auxiliary masses are evenly distributed on the vertical and horizontal beams by using MASS21 mass elements. The element COMBIN14 spring simulates the base plate between the beam and rail fastening or sleeper. Structural connections are achieved through shared nodes and rigid connections. The longitudinal stiffness of the bearing is R_1, R_4, R_7, . . ., R_{28}, R_{31}, R_{34}, the vertical stiffness of the bearing is R_2, R_5, R_8, . . ., R_{29}, R_{32}, R_{35}, and the lateral stiffness of the bearing is R_3, R_6, R_9, . . ., R_{30}, R_{33}, R_{36}.

4.4. Results. Vertical and transverse acceleration sensors are arranged in the first, second, and third midspans to collect the response data when the train passes through. The free attenuation data segment after the train passes through is used for spectrum analysis to obtain the natural frequency of the structure. The natural vibration frequency measured of the bridge is the frequency corresponding to the first-order transverse mode. The range of parameters is determined according to empirical values. The reasonable parameter range is $[1 \times 10^8, 11 \times 10^{10}]$ N/m.

The FEM update purpose of the bridge based on the test is to find an FEM model whose mechanical property is close to the reality. The updated results are shown in Table 3. The objective function values of SPSO and MCPSO are average values with 30 analyses performed under the same conditions. The tested frequency is 2.48, while the calculated frequencies after the FEM model update by SPSO and MCPSO are 2.463 and 2.464. Figure 5 shows the convergence of SPSO and MCPSO, and it shows that proposed MCPSO updated the FEM model and finally obtained a more optimized FEM model. The updated FEM model can be used to study the bridge response in a variant environment.

Table 3: Parameters of the range and update N/m.

Stiffness	SPSO	MCPSO
R_1	$0.00E+00$	$0.00E+00$
R_2	$1.02E+11$	$1.03E+11$
R_3	$1.92E+10$	$1.90E+10$
R_4	$0.00E+00$	$0.00E+00$
R_5	$6.07E+10$	$6.06E+10$
R_6	$5.38E+10$	$5.30E+10$
R_7	$0.00E+00$	$0.00E+00$
R_8	$1.05E+11$	$1.07E+11$
R_9	$9.23E+10$	$9.28E+10$
R_{10}	$0.00E+00$	$0.00E+00$
R_{11}	$8.55E+10$	$8.82E+10$
R_{12}	$7.30E+10$	$7.50E+10$
R_{13}	$3.54E+10$	$3.42E+10$
R_{14}	$9.36E+10$	$9.66E+10$
R_{15}	$1.64E+10$	$1.72E+10$
R_{16}	$8.84E+10$	$8.29E+10$
R_{17}	$9.68E+10$	$9.64E+10$
R_{18}	$5.10E+09$	$5.12E+09$
R_{19}	$0.00E+00$	$0.00E+00$
R_{20}	$1.03E+11$	$1.01E+11$
R_{21}	$5.79E+10$	$5.81E+10$
R_{22}	$0.00E+00$	$0.00E+00$
R_{23}	$9.29E+10$	$9.14E+10$
R_{24}	$2.71E+10$	$2.70E+10$
R_{25}	$0.00E+00$	$0.00E+00$
R_{26}	$9.47E+10$	$9.34E+10$
R_{27}	$3.43E+10$	$3.45E+10$
R_{28}	$0.00E+00$	$0.00E+00$
R_{29}	$5.74E+10$	$5.72E+10$
R_{30}	$6.41E+10$	$6.53E+10$
R_{31}	$0.00E+00$	$0.00E+00$
R_{32}	$1.43E+10$	$1.42E+10$
R_{33}	$8.59E+08$	$8.51E+08$
R_{34}	$0.00E+00$	$0.00E+00$
R_{35}	$2.37E+10$	$2.39E+10$
R_{36}	$1.15E+09$	$1.11E+09$

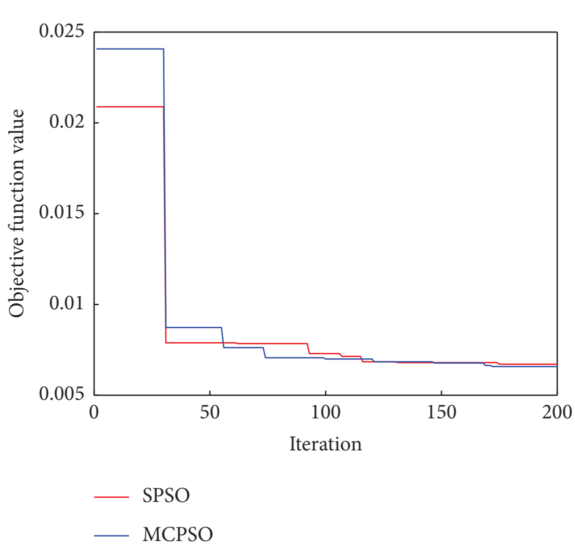


Figure 5: Convergence curve.

5. Conclusions

An MCPSO algorithm method is presented in this article, based on the PSO algorithm and the multiswarm strategy. The experiment adopted a new technique for dividing the population into three subswarms and used strategies with different specialties to balance exploration and exploitation searches. The MCPSO algorithm method is not easy to trap in the local optima, so it has a strong exploration and exploitation characteristic compared with that of SPSO. The following conclusions can be drawn:

(1) This paper proposes MCPSO, which divides the total particle swarm into three subswarms. The number of subswarm particles is determined according to the evaluation of the objective function value change. The results show that the comprehensive evaluation strategy optimizes the computational performance of the PSO variant.

(2) MCPSO algorithms achieved better performance than the SPSO algorithms in terms of the estimation accuracy and convergence velocity. A more accurate and effective structural FEM model considering damage possibility was obtained.

(3) The velocity update performance of a single strategy of the PSO variant algorithm with multistrategy combination is more critical to the algorithm. If the selected velocity update strategy is not suitable, the performance improvement of the PSO variant algorithm with the combined strategy will be not obvious compared to that of a single strategy particle.

(4) The structural parameters that affect the response of the bridge structure include the section size change caused by component damage and the stiffness reduction caused by fatigue. Therefore, the model update problem is essentially an inverse problem, which is ill posed, and the optimized bearing stiffness value may not necessarily be the true value of bridge structural stiffness. More possible monitoring bridge structure data are needed to make the calculation model closer to the real state of the bridge.

Data Availability

All data, models, and codes generated or used during the study appear in the submitted article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key Research and Development Program of Hebei Province (19210804D), State Key Laboratory of Mechanical Behavior and System Safety of Traffic Engineering Structures (ZZ2020-19), Science and

Technology Research Youth Fund Project of Higher School in Hebei Province (QN2018108), Natural Science Foundation for Youths of Hebei Province of China (E2021210055), Science and Technology Research Project of Colleges and Universities in Hebei Province (ZD2022073), Technology Development Project of National Energy Group Shuohuang Railway Development Co., Ltd (GJNY-20-230), and Science and Technology Innovation Project of China Energy Investment Corporation (SHGF-15-41).

References

[1] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, Anchorage, AK, USA, May 1998.

[2] J. Jakubik, A. Binding, and S. Feuerriegel, “Directed particle swarm optimization with Gaussian-Process-based function forecasting,” *European Journal of Operational Research*, vol. 295, no. 1, pp. 157–169, 2021.

[3] M. F. Ab Aziz, S. A. Mostafa, C. F. Mohd Foozy, M. A. Mohammed, M. Elhoseny, and A. Z. Abualkishik, “Integrating elman recurrent neural network with particle swarm optimization algorithms for an improved hybrid training of multidisciplinary datasets,” *Expert Systems with Applications*, vol. 183, no. 2, Article ID 115441, 2021.

[4] A. Kaveh, “Particle swarm optimization,” in *Advances in Metaheuristic Algorithms for Optimal Design of Structures*-Springer, Cham, Champa, 2021.

[5] M. M. Noel, “A new gradient based particle swarm optimization algorithm for accurate computation of global minimum,” *Applied Soft Computing*, vol. 12, no. 1, pp. 353–359, 2012.

[6] C.-F. Wang and K. Liu, “An improved particle swarm optimization algorithm based on comparative judgment,” *Natural Computing*, vol. 17, no. 3, pp. 641–661, 2018.

[7] K. M. Ang, W. H. Lim, N. A. M. Isa, S. S. Tiang, and C. H. Wong, “A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems,” *Expert Systems with Applications*, vol. 140, Article ID 112882, 2020.

[8] K. E. Parsopoulos and M. N. Vrahatis, “UPSO: a unified particle swarm optimization scheme,” in *Proceedings of the International Conference of Computational Methods in Sciences and Engineering 2004 (ICCMSE 2004)*, pp. 868–873, CRC Press, London, UK, January 2019.

[9] Y. Ning, Z. Peng, Y. Dai, D. Bi, and J. Wang, “Enhanced particle swarm optimization with multi-swarm and multi-velocity for optimizing high-dimensional problems,” *Applied Intelligence*, vol. 49, no. 2, pp. 335–351, 2019.

[10] S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, and J. C Bansal, “A survey on parallel particle swarm optimization algorithms,” *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 2899–2923, 2019.

[11] Z. Liu and T. Nishi, “Strategy dynamics particle swarm optimizer,” *Information Sciences*, vol. 582, pp. 665–703, 2022.

[12] C. Li, S. Yang, and T. Nguyen, “A self-learning particle swarm optimizer for global optimization problems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 627–646, 2012.

[13] K. Tang, Z. Li, L. Luo, and B. Liu, “Multi-strategy adaptive particle swarm optimization for numerical optimization,” *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 9–19, 2015.

[14] Ş. Gülcü and H. Kodaz, “A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization,” *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 33–45, 2015.

[15] M. R. Bonyadi and Z. Michalewicz, “Particle swarm optimization for single objective continuous space problems: a review,” *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, 2017.

[16] C. Wang and W. Song, “A modified particle swarm optimization algorithm based on velocity updating mechanism,” *Ain Shams Engineering Journal*, vol. 10, no. 4, pp. 847–866, 2019.

[17] X. Xia, Y. Tang, B. Wei, and L. Gui, “Dynamic multi-swarm particle swarm optimization based on elite learning,” *IEEE Access*, vol. 7, pp. 184849–184865, 2019.

[18] Y. Tang, L. Xiong, Y. Zhang, X. Xia, and L. Gui, “Dynamic multi-swarm global particle swarm optimization,” in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1030–1037, Wellington, New Zealand, June 2019.

[19] X. Xia, L. Gui, and Z.-H. Zhan, “A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting,” *Applied Soft Computing*, vol. 67, pp. 126–140, 2018.

[20] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. S. Hone, and X. Liu, “A dynamic neighborhood-based switching particle swarm optimization algorithm,” *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9290–9301, 2022.

[21] Q. Zhao and C. Li, “Two-stagemulti-swarm particle swarm optimizer for unconstrained and constrained global optimization,” *IEEE Access*, vol. 8, pp. 124905–124927, 2020.

[22] W. Li, X. Meng, Y. Huang, and Z. H. Fu, “Multipopulation cooperative particle swarm optimization with a mixed mutation strategy,” *Information Sciences*, vol. 529, pp. 179–196, 2020.

[23] M. Elbes, S. Alzu’bi, T. Kanan, A. Al-Fuqaha, and B. Hawashin, “A survey on particle swarm optimization with emphasis on engineering and network applications,” *Evolutionary Intelligence*, vol. 12, no. 2, pp. 113–129, 2019.

[24] I. G. Araujo, E. Maldonado, and G. C. Cho, “Ambient vibration testing and updating of the finite element model of a simply supported beam bridge,” *Frontiers of Architecture and Civil Engineering in China*, vol. 5, no. 3, pp. 344–354, 2011.

[25] D. S. Jung and C.-Y. Kim, “Finite element model updating on small-scale bridge model using the hybrid genetic algorithm,” *Structure and Infrastructure Engineering*, vol. 9, no. 5, pp. 481–495, 2013.

[26] J. P. Han and Y. P. Luo, “Static and dynamic finite element model updating of a rigid frame-continuous girders bridge based on response surface method,” *Advanced Materials Research*, vol. 640, pp. 992–997, 2013.

[27] H. Tran-Ngoc, H. Ho-Khac, T. Le-Xuan et al., “Model updating for a railway bridge using a hybrid optimization algorithm combined with experimental data,” *Lecture Notes in Civil Engineering*, vol. 362, pp. 19–30, 2020.

[28] D. D. Ho, J. T. Kim, J. H. Park, and D. S. Hong, “Field vibration tests-based model update for system identification of railway bridge,” in *Proceedings of the SPIE, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2010*, p. 764731p. 764731, 2010.

[29] A. Bayraktar, A. C. Altunışık, B. Sevim, and T. Türker, "Finite element model updating of Kömürhan highway bridge based on experimental measurements," *Smart Structures and Systems*, vol. 6, no. 4, pp. 373–388, 2010.

[30] B. Arisoy and O. Erol, "Finite element model calibration of a steel railway bridge via ambient vibration test," *Steel and Composite Structures*, vol. 27, no. 3, pp. 327–335, 2018.

[31] H. Tran-Ngoc, S. Khatir, G. D. Roeck, T. Bui-Tien, L. Nguyen-Ngoc, and M. A. Wahab, *Model Updating for Nam O Bridge Using Particle Swarm Optimization Algorithm and Genetic Algorithm*, Sensors, (Basel, Switzerland), 2018.

[32] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[33] Z. H. Ding, M. Huang, and Z. R. Lu, "Structural damage detection using artificial bee colony algorithm with hybrid search strategy," *Swarm and Evolutionary Computation*, vol. 28, pp. 1–13, 2016.