Springer

KSME

Journal of Mechanical Science and Technology 33 (2) (2019) 695~709
www.springerlink.com/content/1738-494x(Print)/1976-3824(Online)
DOI 10.1007/s12206-019-0124-5

Diversity enhanced particle swarm optimization algorithm and its application in vehicle lightweight design †

Zhao Liu, Han Li and Ping Zhu*

State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, Shanghai, 200240, China

(Manuscript Received January 4, 2018; Revised October 1, 2018; Accepted October 20, 2018)

Abstract

Particle swarm optimization, a widely used metaheuristic algorithm, mimics the cooperation behavior among species. The PSO algorithm has become a new trend owing to its simplicity and strong optimization capacity. However, premature convergence problem is also a serious issue for PSO comparable with other evolutionary algorithms. Diversity loss is generally known as one of the major causes. For enhancing the diversity of swarms during optimization procedure, an improved PSO algorithm named OLAR-PSO-d is proposed, which incorporates design of experiment technique as well as adaptive reset operator into standard PSO. The OLAR-PSO-d algorithm is compared with other 10 heuristic algorithms. The numerical experiments' results demonstrate the priority of OLAR-PSO-d both in optimization ability and algorithm stability. The proposed algorithm is also used in a vehicle lightweight design problem. The auto-body achieves 20.25 kg weight reduction with meeting all the performance requirements of crashworthiness.

Keywords: Adaptive reset operator; Algorithm stability; Design of experiments; Global optimization; Particle swarm optimization

1. Introduction

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart [1], is a widely used population-based metaheuristic optimization method. PSO mimics the cooperation behavior among species such as blocks of birds, schools of fishes, etc. During optimization, the possible solutions are treated as particles in the design space. Based on particle's own best position that it has reached (*pbest*) and the best position of the entire swarm at each generation (*gbest*), position and velocity of every particle are updated iteratively [1-3]. The PSO algorithm has become a new trend and has been highly applied. It's mainly because of its briefness of implementation and powerful competence to get a fairly favorable solution rapidly [4-6].

While a fast convergence rate is the consequence the movement of particles in PSO algorithm, premature convergence problem also occurs. That's because the diversity among particles quickly loses [7-9]. Diversity is regarded as the degree of dispersion of all particles. During the PSO procedure, the particles' diversity is high after initialization. After the process started, the particles tend to be convergent and the diversity is declined. The process, which enhances the local search (exploitation) ability while reduces the global search (exploration) ability, is beneficial to the efficiency but may bring worse poor diversity. In another word, the premature convergence may occur.

The performance of PSO is deeply influenced by the diversity of particles. Consequently, maintaining a high diversity is a crucial work for PSO algorithm. Numbers of scholars have studied different diversity maintaining mechanisms for improve the optimization ability of PSO. They can be generally divided into three categories. First is the mutation operator: the mutation operator is a key operator used in some evolutionary algorithms to generate the offspring for further optimizing process. Ratnaweera et al. presented a mutation operator used for mutating velocity instead of the position of a particle [10]. And mutations were only generated when the global best solution have not changed for several iterations. Diversity-guided strategy is the second category. Diversity measures are conventionally used for analyzing evolutionary algorithms instead of guiding the optimization procedure. Riget proposed the ARPSO algorithm, in which a repulsion phase was defined to modify velocity updating equation [11]. This phase was used to prevent particles attracting too fast by the best positions. Pant et al. defined a middle phase based on Riget's study [12]. The middle phase was called positive conflict phase which was between attraction and repulsion phases. Every particle was attracted by local best position and repelled by the global best position in the middle phase. Sun et al. used a mutation operator to increase the swarm diversity when the

*Corresponding author. Tel.: +86 2134206787, Fax.: +86 2134206787
E-mail address: pzhu@sjtu.edu.cn
† Recommended by Associate Editor Gang-Won Jang

diversity was below the predefined criterion [13]. Wang et al. incorporated a trial particle to enhance the diversity of particle swarm which can save the computational cost for the diversity calculation [14]. Meng et al. employed the crisscross search method into the PSO algorithm and developed a crisscross search particle swarm optimization method [15]. This method can improve the global searching ability and accelerating the global convergence of PSO. The third technique is the neighborhood search method. The suboptimal may be near to the global optimization solution or the neighborhood individuals may contain the global optimal solution occasionally. Based on this, the neighborhood search strategies have been employed into PSO procedure. Kennedy designed circle, wheel, star and random topologies for testing [16]. The results showed that fewer connections may perform better on highly multimodal problems while highly interconnected populations were suitable for unimodal problems. Mendes et al. proposed a fully informed PSO (FIPS), which used the neighbors to update the velocity instead of the local and global best positions [17]. Peram used a fitness-distance-ratio to control the particles moving toward the best previous positions of its neighbors [18]. Li proposed an information sharing mechanism, which allows every particle to share its *pbest* with others, to improve the performance of PSO [19].

This article proposed an improved PSO algorithm, the OLHD sampling and adaptive reset operator enhanced PSO with disturbance particles (OLAR-PSO-d), to enhance the diversity and strengthen the searching ability of the standard PSO. The latin hypercube design (LHD) technique is an efficient DOE method to identify sampling locations in design domain [20, 21]. Since the initialization of PSO is a sampling process, the proposed mothed adopts the optimal LHD (OLHD) technique to generate the initial swarm with full coverage in design domain. After the particles being initialized, a stagnation judgment criterion is established to enhance diversity among the swarm. When both the stagnation criterion and a predefined probability are satisfied, the adoptive reset operator will be activated to modify particles' velocity. Otherwise, the disturbance particles are introduced to enhance diversity of the swarm. Therefore, the stagnation is broken up and the optimization process will continue to search the global optimum.

The remain parts of this article are organized as follows: Sec. 2 is the technical base in which the PSO algorithm and DOE technique are simply described. In Sec. 3, the proposed OLAR-PSO-d method is presented in detail. Sec. 4 presents mathematical experiments, results and discussions. Sec. 5 is the application of the proposed algorithm in a vehicle lightweight design problem. Finally, the conclusion is summarized in Sec. 6.

2. The technological base

2.1 The basis of particle swarm optimization

When the standard deals with a D-dimensions problem, position of each particle represents the potential solution. Vector x_k^i is the position of the i^{th} particle and vector v_k^i is the velocity. p_k^i stands for the pbest of each particle while p_k^g represent the *gbest* of all particles. In each iteration, particles move toward both *pbest* and *gbest* until better solutions are found. The velocity and position updating equations are as Eqs. (1) and (2).

$$v_{k+1}^i = \omega v_k^i + c_1 r_1 \left(p_k^i - x_k^i \right) + c_2 r_2 \left(p_k^g - x_k^i \right) \tag{1}$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \tag{2}$$

where ω is the inertia factor; c_1 is the cognitive scaling parameter and c_2 is the social scaling parameter [22]; r_1 and r_2 are two uniformly distributed random numbers within range [0,1]. ω is obtained by a linearly varying inertia weight equation [23], as shown in Eq. (3)

$$\omega(iter_{\text{current}}) = \frac{(iter_{\max} - iter_{\text{current}})}{iter_{\max}} \cdot (\omega_{\max} - \omega_{\min}) + \omega_{\min} \tag{3}$$

where $iter_{\text{current}}$ represents current iteration and $iter_{\max}$ is the maximum iteration. The inertia factor ω has made the standard PSO improved significantly. The second and third part of Eq. (1) is cognition component and social component respectively. Cognition component encourages particles to move toward their own *pbest* while social component makes use of cooperation behavior among particles [24]. Eq. (1) makes particles tend to move across the design space as well as to balance the exploitation and exploration abilities while optimization proceeding [25].

The bounce method is employed in this article to solve the boundary of a problem [26]. If a particle exceeds design domain, its position is replaced by the nearest boundary and its velocity is reversed at each dimension i:

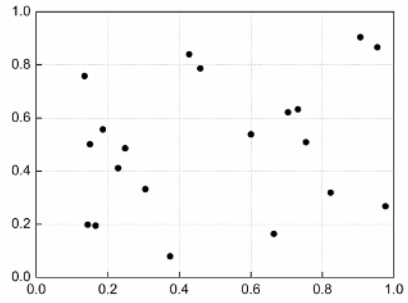
$$x_k^{i'} = \begin{cases} x_{\max}^i, x_k^i > x_{\max}^i \\ x_{\min}^i, x_k^i < x_{\min}^i \end{cases} \tag{4}$$

$$v_k^i = -v_k^i \,. \tag{5}$$

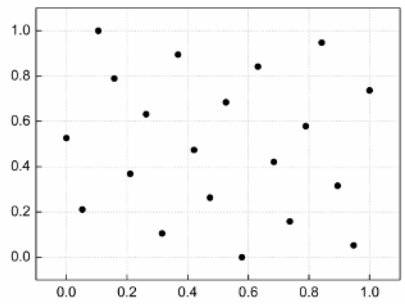
2.2 Enhanced stochastic evolutionary method for OLHD

In this article, the enhanced stochastic evolutionary (ESE) optimized latin hypercube design (LHD) is used [27]. LHD is a statistic mothed, which takes only one sample at each level. To obtain the optimal LHD (OLHD), ESE consists of double loops, i.e., the inner loop and the outer loop. The inner loop constructs new designs by element-exchanges randomly and if the new designs are better, accept them. Or the new designs will be accepted with probability. The outer loop controls the entire optimization process by adjusting the acceptance probability.

The inner loop compares designs by the ϕ_p optimal criterion, which is firstly proposed by Morris and Mitchell [28]:



(a) Rand samples



(b) OLHD samples

Fig. 1. Initialization method.

$$\min \quad \phi_p = \left(\sum_{i=1}^{s} J_i d_i^{-p} \right)^{-p} \tag{6}$$

where d_i are distinct distance values with $d_1 < d_2 < \ldots < d_s$, J_i is the number of pairs of sites in the design separated by d_i, s is the number of distinct distance values, p is a positive integer.

3. The proposed algorithm

In this section, the OLAR-PSO-d is described in detail. There are three parts, the initialization method in OLAR-PSO-d are compared with the traditional method at first. Then the adaptive reset operator and disturbance particles proposed in this article is interpreted. At last, the technological process of the OLAR-PSO-d algorithm is presented.

3.1 Initialization by OLHD technique

To guarantee a complete coverage of the design domain, the OLHD, rather than the uniformly random distributed method is applied to acquire the first generation of particle swarm. Compare the distribution of particles developed by OLHD in Fig. 1(b) and this using traditional method in Fig. 1(a), it's can be concluded easily that the former is more acceptable.

3.2 Adaptive reset operator acted on velocity

It is understood that as for the premature convergence problem, the prevailing factor is the lack of diversification. In the most evolutionary optimization, mutation operators are highly applied. In this way, lack of diversity can be avoided. Thus, one algorithm is able to lookup a larger region of design space. Variants are created by mutation operator founded on current individuals to add diverseness to population and avert stagnation in local optima.

Various numerical experiments show that standard PSO rapidly gets a comparatively favorable solution. However, it may stagnate occasionally in the local optimum without future enhancement after considerable times of iterations. To enhance the exploration capability of PSO, an adaptive reset operator acted on velocity is now developed. The operator is similar to the mutation operator abovementioned to some extent. As PSO in process, if the particles stagnate in the local optimum, the proposed strategy will be active at predefined probability. Then the operator will reset particles' velocity and force the particles to further search the global optimum. The adaptive reset operator works as Eq. (7):

$$V_{reset} = \mu \cdot rw \cdot V_{rand} \tag{7}$$

where μ stands for a generation correlation coefficient and it is linearly declined during iteration as Eq. (8):

$$\mu = \sqrt{1 - \frac{iter_{current}}{iter_{\max}}} \,. \tag{8}$$

rw is a velocity correlation coefficient with range [$rw_{\min}$, $rw_{\max}$] as Eq. (9):

$$rw = \left(rw_{\max} - rw_{\min} \right) \mu + rw_{\min} \,. \tag{9}$$

V_{rand} represents particles' velocity matrix, generated randomly with range [$-V_{\max}$, $V_{\max}$].

With the iteration times increased, μ is decreased while the amplitude of μ and V_{reset} is shrunken. Consequently, the distribution of reset particles in consideration of exploitation and exploration ability is improved by rw and the convergence guaranteed. After the adaptive reset operator activated, the particles are dispersed from the previous stagnation position by Eq. (10):

$$P_p = P_{stagnation} + V_{reset} \,. \tag{10}$$

3.3 Disturbance particles

In order to enhance the diversity of the swarm, the disturbance particles are introduced to the proposed method. In each iteration, current particles would learn from both local disturbance particles and global disturbance particles besides *pbest* and *gbest*. The two types of disturbance particles are generated form the *pbest* and *gbest* respectively according following equations:

$$d_k^i = s_k r_{0.5} p_k^i \tag{11}$$

$$d_k^g = s_k r_{0.5} p_k^g \tag{12}$$

where s_k is the scale factor with linear decrease from 1 to 0 during the optimization; $r_{0.5}$ is randomly created between -0.5 to 0.5.

Then the velocity Eq. (1) is rewritten as Eq. (13):

$$v_{k+1}^i = \omega v_k^i + c_1 r_1 \left(p_k^i - x_k^i + d_k^i \right) + c_2 r_2 \left(p_k^g - x_k^i + d_k^g \right). \tag{13}$$

3.4 Convergence analysis

In this part, the convergence of the proposed method has been proved. Since the Mathematical expectation of the $r_{0.5}$ and the finial scale factor s is both 0, disturbance particle terms d_k^i and d_k^g in Eqs. (11)-(13) are not taken into following deduction.

Replaced the velocity updating Eq. (1) to position updating Eq. (2) and rearrange the velocity Eq. (1), the general velocity and position equations of a particle can be represented as follows:

$$x_{k+1}^i = x_k^i (1 - c_1 r_1 - c_2 r_2) + \omega v_k^i + c_1 r_1 p_k^i + c_2 r_2 p_k^g \tag{14}$$

$$v_{k+1}^i = -x_k^i (c_1 r_1 + c_2 r_2) + \omega v_k^i + c_1 r_1 p_k^i + c_2 r_2 p_k^g . \tag{15}$$

When the reset operator is activated, the Eq. (14) could be represented as:

$$x_{k+1}^i = x_k^i (1 - c_1 r_1 - c_2 r_2) + \omega v_k^i + c_1 r_1 p_k^i + c_2 r_2 p_k^g + \mu \cdot rw \cdot v_{rand}^i . \tag{16}$$

Combining Eqs. (15) and (16), written in matrix form as Eq. (17):

$$\begin{bmatrix} x_{k+1}^i \\ v_{k+1}^i \end{bmatrix} = \begin{bmatrix} 1 - c_1 r_1 - c_2 r_2 & \omega \\ -(c_1 r_1 + c_2 r_2) & \omega \end{bmatrix} \begin{bmatrix} x_k^i \\ v_k^i \end{bmatrix} + \begin{bmatrix} c_1 r_1 & c_2 r_2 \\ c_1 r_1 & c_2 r_2 \end{bmatrix} \begin{bmatrix} p_k^i \\ p_k^g \end{bmatrix} + \begin{bmatrix} \mu \cdot rw \cdot v_{rand}^i \\ 0 \end{bmatrix}. \tag{17}$$

According to Eq. (17), the PSO algorithm can be considered as a discrete-dynamic system [29]. $\begin{bmatrix} x_{k+1}^i & v_{k+1}^i \end{bmatrix}^T$ is the state subject of the input criterion $\begin{bmatrix} p_k^i & p_k^g \end{bmatrix}^T$. If there is no external excitation for the dynamic system, $\begin{bmatrix} p_k^i & p_k^g \end{bmatrix}^T$ is constant, which means there are no better positions identified by other particles, then the convergence can be maintained. In this case, when the iterations $k \to \infty$, then

$$\lim_{k \to \infty} \begin{bmatrix} x_{k+1}^i \\ v_{k+1}^i \end{bmatrix} = \begin{bmatrix} x_k^i \\ v_k^i \end{bmatrix} \tag{18}$$

and

$$\lim_{k \to \infty} \sqrt{1 - \frac{iter_{current}}{iter_{\max}}} \left((rw_{\max} - rw_{\min}) \sqrt{1 - \frac{iter_{current}}{iter_{\max}}} + rw_{\min} \right) v_{rand}^i = 0 \tag{19}$$

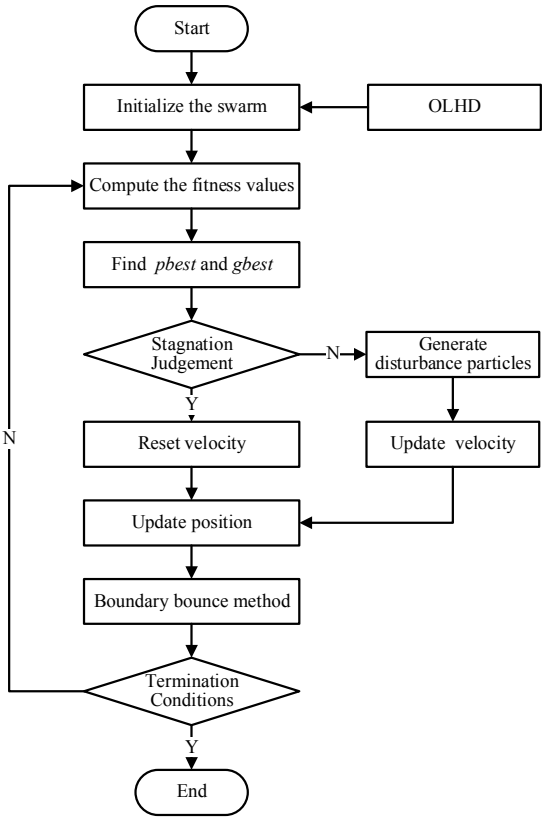


Fig. 2. The flowchart of OLAR-PSO-d.

$$\lim_{k \to \infty} \begin{bmatrix} \mu \cdot rw \cdot v_{rand}^i \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{20}$$

Eq. (17) can be reduced as follows:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -(c_1 r_1 + c_2 r_2) & \omega \\ -(c_1 r_1 + c_2 r_2) & \omega - 1 \end{bmatrix} \begin{bmatrix} x_k^i \\ v_k^i \end{bmatrix} + \begin{bmatrix} c_1 r_1 & c_2 r_2 \\ c_1 r_1 & c_2 r_2 \end{bmatrix} \begin{bmatrix} p_k^i \\ p_k^g \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} c_1 r_1 \left(p_k^i - x_k^i \right) + c_2 r_2 \left(p_k^g - x_k^i \right) + \omega v_k^i \\ c_1 r_1 \left(p_k^i - x_k^i \right) + c_2 r_2 \left(p_k^g - x_k^i \right) + (\omega - 1) v_k^i \end{bmatrix} \tag{22}$$

which is true only if $x_k^i = p_k^i = p_k^g$ and $v_k^i = 0$. The point found is not a local minimum or a global solution but an equilibrium point. Therefore, this point can move forward the optimum if there is a better *pbest* and *gbest* discovered by the optimization process.

3.5 The procedure of OLAR-PSO-d

The flowchart of proposed OLAR-PSO-d method is shown in Fig. 2. Seven steps are used for OLAR-PSO-d:

(1) To guarantee a through coverage of the design domain, the OLHD is used to perform initialization, which is various from standard PSO algorithm.

(2) Calculate fitness values of each particle, find the *pbest*

Table 1. Benchmark functions.

No.	Name	Formula	Range	Optimum
f1	Ackley	$f=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi x_i\right)+20+e$	[-32,32]	0
f2	Griewank	$f=\frac{1}{4000}\sum_{i=1}^{D-1}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600]	0
f3	Penalized	$f=\frac{\pi}{D}\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right\}+\sum_{i=1}^{D}u(x_i,10,100,4)$ $y_i=1+\frac{1}{4}(x_i+1)$	[-50,50]	0
f4	Penalized 2	$f=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{D-1}(x_i-1)^2\left[1+\sin^2(3\pi x_{i+1})\right]+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\right\}+\sum_{i=1}^{D}u(x_i,5,100,4)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\ 0-a\le x_i\le a\\ -k(-x_i-a)^m, x_i<-a\end{cases}$	[-50,50]	0
f5	Quartic Noise	$f=\sum_{i=1}^{D}ix_i^4+rand(0,1)$	[-1.28,1.28]	0
f6	Rastrigin	$f=\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	[-5.12,5.12]	0
f7	Rosenbrock	$f=\sum_{i=1}^{D-1}\left[100(x_{i+1}-x_i)^2+(x_i-1)^2\right]$	[-30,30]	0
f8	Schafferf 7	$f=\sum_{i=1}^{n-1}\left[\left(x_i^2+x_{i+1}^2\right)^{0.25}\left(\sin\left(50\left(x_i^2+x_{i+1}^2\right)^{0.1}\right)^2+1\right)\right]$	[-100,100]	0
f9	Schwefel	$f=\sum_{i=1}^{n}x_i\sin(\sqrt{\lvert x_i\rvert})$	[-500,500]	−418.9829×D
f10	Schwefel 1.2	$f=\sum_{i=1}^{D}\left(\sum_{j=1}^{i}x_j\right)^2$	[-100,100]	0
f11	Schwefel 2.21	$f=\max\left\{\lvert x_i\rvert, 1\le i\le D\right\}$	[-100,100]	0
f12	Schwefel 2.22	$f=\sum_{i=1}^{D}\lVert x_i\rVert+\prod_{i=1}^{D}\lVert x_i\rVert$	[-100,100]	0
f13	Sphere	$f=\sum_{i=1}^{D}x_i^2$	[-100,100]	0
f14	Step	$f=\sum_{i=1}^{D}(x_i+0.5)^2$	[-100,100]	0
f15	Zakharov	$f=\sum_{i=1}^{2}x_i^2+\left(\sum_{i=1}^{2}0.5ix_i\right)^2+\left(\sum_{i=1}^{2}0.5ix_i\right)^4$	[-100,100]	0

and *gbest*.

(3) Check the predefined stagnation criterion $G_{\text{stagnation}}$. If the requirement met, turn into step (4). In this article, $G_{\text{stagnation}}$ is maximum counts that *gbest* unchanged, which is set to $D\times 3$ via empirical observations.

(4) As the adaptive reset operator is activated, velocities are reset for all the particles if the probability judgment satisfied. In each iteration, the judgment would generate a number r between 0 and 1 randomly. If r is smaller than the given probability P, the reset operator based on Eq. (7) would proceed to break up stagnation. According to the mathematical experiments, the value of P used is 0.5, $rw_{\min}$ and $rw_{\max}$ are set to 0.1 and 0.9, respectively.

(5) If the adaptive reset operator is inactivate, the disturbance particles are generated by Eqs. (11) and (12). The velocity is updated based on Eq. (13).

(6) Renew particles' position according to Eq. (2) and proceed the boundary check process in case that the positions of

Table 2. Optimization results with 30 dimensions.

No.	Index	BBPSO	CLPSO	APSO	DMS-PSO	DE/best/1	ODE	ABC	GABC	IGHS	GDHS	OLAR-PSO-d
f1	M	20	1.42E-14	0.00757	6.06E-12	6.04E-15	6.04E-15	1.63E-09	1.63E-09	3.44E-06	0.00456	0
	SD	10.4	7.46E-15	0.000963	3.9E-13	1.67E-15	1.67E-15	1.44E-09	1.44E-09	1.79E-07	0.000394	0
f2	M	0.0108	0.0032	0.0287	0.00185	0.00283	0.00283	7.48E-10	7.48E-10	0.0108	0.00414	0
	SD	0.0132	0.00493	0.0258	0.00407	0.0066	0.0066	2.4E-09	2.4E-09	0.00974	0.00451	0
f3	M	0.228	1.36E-33	2.65E-06	0	0.00518	0.00518	3.66E-21	3.66E-21	2.04E-13	7.43E-07	4.63E-07
	SD	0.379	2.82E-33	7.17E-06	0	0.0232	0.0232	1.02E-20	1.02E-20	2.44E-14	1.37E-07	1.61E-07
f4	M	0.00769	1.65E-33	4.48E-06	6.16E-35	1040	1040	1.47E-19	1.47E-19	3.03E-12	0.000012	6.03E-09
	SD	0.0134	4.03E-33	1.14E-05	2.76E-34	3460	3460	4.04E-19	4.04E-19	4.49E-13	1.66E-06	2.81E-09
f5	M	0.41	0.00174	0.00286	0.000609	0.00465	0.00465	0.101	0.101	0.00422	0.000732	0.0000727
	SD	0.981	0.000783	0.000923	0.000418	0.00272	0.00272	0.033	0.033	0.00195	0.00025	0.0000404
f6	M	99.7	12.7	0.0498	14.9	13.1	13.1	3.2E-11	3.2E-11	4.23E-09	0.000376	0
	SD	34.4	4.22	0.222	3.62	3.91	3.91	5.28E-11	5.28E-11	4.68E-10	5.53E-05	0
f7	M	18100	41.8	70	34.9	20.3	20.3	0.483	0.483	24.2	26.5	23.8
	SD	36900	33.5	125	27.6	17.6	17.6	0.443	0.443	26.8	17.9	0.386
f8	M	98.2	0.43	3.02	0.0853	0.212	0.212	1.45	1.45	12.1	2.52	0.0068
	SD	28.6	0.37	2.07	0.0787	0.362	0.362	1.09	1.09	3.49	0.158	0.0417
f9	M	-10042.5	-12569.5	-12569.5	-12569.5	-12123.4	-12569.5	-12569.5	-12569.5	-12569.5	-12234.5	-8290
	SD	0.0561	0.0035	0.000109	0.0836	0.0286	0.0078	0.0787	0.0298	0.000912	0.000767	514
f10	M	12000	0.00932	7.7E-08	0.000295	3.59E-06	3.59E-06	7800	7800	1.07E-10	0.00154	9.99E-106
	SD	7040	0.0193	2.44E-07	0.00058	5.24E-06	5.24E-06	2580	2580	2.73E-11	0.000362	3.36E-105
f11	M	0.000185	0.0653	3.02E-05	0.00794	17.6	17.6	13.2	13.2	2.04E-06	0.0075	4.15E-104
	SD	4.22E-04	0.0379	5.85E-05	0.00762	4.86	4.86	4.76	4.76	1.98E-07	0.000767	5.1E-104
f12	M	440	5.5E-57	0.00538	218	2.73E-32	4.73E-32	5.18E-10	5.18E-10	0.0121	0.0676	1.06E-104
	SD	223	1.92E-56	0.0101	183	3.54E-32	5.54E-32	7.84E-10	7.84E-10	0.0333	0.00455	1.74E-104
f13	M	1.3E-216	8.06E-96	2.3E-12	1.5E-113	5.01E-57	5.01E-57	8.01E-19	8.01E-19	2.17E-11	0.000257	8.8E-108
	SD	0	3.53E-95	1.03E-11	5.1E-113	1.57E-56	1.57E-56	2.36E-18	2.36E-18	2.98E-12	3.67E-05	1.73E-107
f14	M	1.55	0	0	0	0.05	0.05	0	0	0	0	5.98E-06
	SD	3.12	0	0	0	0.224	0.224	0	0	0	0	2.29E-06
f15	M	20000	0.0185	87.8	0.969	0.433	0.433	29300	29300	5.19E-11	0.000732	2.65E-104
	SD	14200	0.0447	280	2.35	1.38	1.38	19400	19400	7.37E-12	0.000112	7.51E-104

updating particles are beyond the boundary of the optimization problem.

(7) Steps (2)-(6) are proceeded repeatedly until meeting the termination conditions. In this work, the predefined maximum function evaluation is chosen as the termination criterion.

4. Experimental verifications

In this section, the proposed method is compared with 10 PSO variants or other efficient heuristic algorithm by 15 wildly used benchmark functions. The compared algorithms are BBPSO [30], CLPSO [31], APSO [32], DMS-PSO [33], DE/best/1 [34], ODE [35], ABC [36], GABC [37], IGHS [38] and GDHS [39].

For fair competition, all the other algorithms are run with their recommended hyperparameters. Part of the experimental data are come from article [40].

15 widely used benchmark functions are tested as shown in Table 1. Dimension (D) of each function is set to 30, 50 and 100, respectively. The inertia weight is linearly decreased from 0.9 to 0.1 while two cognitive factors c_1 and c_2 are set to 1.75 and 1.5. The population size is 40 and the maximum function evaluation is $D\times10^4$. All the benchmark functions are tested 50 times for each algorithm.

The mean final fitness values (M) and standard deviation (SD) of all the problems are listed in Tables 2-4, which demonstrates the optimization ability and the ability stability of algorithms.

From the experimental results of Tables 2-4, the proposed OLAR-PSO-d method works better than the others in functions *Ackley*, *Griewank*, *Quartic Noise*, *Rastrigin*, *Schafferf 7 Schwefel 1.2*/*2.2.1*/*2.22*, and *Zakharov*, and a litter worse in functions *Penalized*, *Rosenbrock*, *Sphere* and *Step*. In order to make a clearly and comprehensive comparison among these

Table 3. Optimization results with 50 dimensions.

No.	Index	BBPSO	CLPSO	APSO	DMS-PSO	DE/best/1	ODE	ABC	GABC	IGHS	GDHS	OLAR-PSO-d
f1	M	18	1.01E-14	1.84	7.75E-08	0.385	7.32E-13	1.01E-11	3.24	4.24E-06	0.00687	0
	SD	4.19	3.32E-15	0.885	2.79E-07	0.564	2.13E-12	1.6E-11	3.05	1.98E-07	0.000335	0
f2	M	18.1	0.000493	4.56	0.00074	0.00627	0.0174	6.29E-13	3.07	0.00222	0.000806	0
	SD	47.2	0.0022	20.3	0.00228	0.011	0.0318	1.15E-12	5.24	0.00466	7.48E-05	0
f3	M	1.28E+07	0.00311	0.306	0.00311	372	5.01E-27	3.67E-25	0.559	2.86E-13	1.6E-06	0.0000591
	SD	5.72E+07	0.0139	0.37	0.0139	1150	1.61E-26	1.33E-24	1.28	3.57E-14	2.28E-07	0.000413
f4	M	0.00659	0.000549	0.00439	8.63E-34	29700	0.0022	7.12E-25	0.311	7.64E-12	4.17E-05	3.1E-09
	SD	0.00552	0.00246	0.00552	2.3E-33	108000	0.00983	1.14E-24	1.39	7.94E-13	7.3E-06	1.3E-09
f5	M	3.64	0.00331	0.408	0.00115	0.0164	0.00138	0.303	0.861	0.00604	0.00121	0.0000519
	SD	7.5	0.000905	1.31	0.000392	0.00847	0.00117	0.15	0.581	0.00186	0.000461	0.0000241
f6	M	308	24.2	179	29.9	32.6	1.11E-12	1.27E-06	19.2	1.16E-08	0.00137	0
	SD	56.4	6.4	53.2	6.3	6.49	4.39E-12	5.52E-06	9.08	9.64E-10	0.000146	0
f7	M	22700	69.7	4520	70.3	64.3	7.16E-25	0.292	23900	59.7	50.3	43.7
	SD	39900	44.5	20100	44.9	38.3	1.56E-24	0.255	77700	32.9	21.3	0.611
f8	M	285	1.28	157	1.26	6.87	8.18	0.625	9.88	19.2	5.54	0.0000651
	SD	57.2	0.971	47.8	2.4	3.78	4.67	0.583	4.49	3.23	0.236	0.000133
f9	M	-17765.7	-20949.1	-20949.1	-20949.1	-20685.1	-20949.1	-20949.1	-20949.1	-20949.1	-20049	-12400
	SD	103	0.0397	0.00129	0.0105	0.655	0.498	0.00998	0.0645	0.00798	0.699	1090
f10	M	40400	54.3	8830	2.54	0.0676	6.64E-22	23200	19800	1.24E-09	0.0324	3.53E-106
	SD	19700	31.4	9320	2.79	0.0423	1.13E-21	9000	9770	2.76E-10	0.00771	1.34E-105
f11	M	62.2	1.49	0.0131	0.995	28.6	0.0612	37	34.7	3.19E-06	0.0145	4.48E-105
	SD	21.5	0.547	0.015	0.523	5.23	0.0336	20.9	15.8	2.32E-07	0.00169	6.39E-105
f12	M	980	4.47E-52	2.94E-38	309	2.62E-33	1.03E-42	8.91E-13	48.1	0.149	0.167	7.08E-106
	SD	177	1.22E-51	1.31E-37	28.7	6.71E-33	3.53E-42	4.85E-13	26.8	0.252	0.0109	8.93E-106
f13	M	1000	2.28E-85	9.5E-140	8.6E-103	1.39E-53	1E-165	7.97E-25	180	5.96E-11	0.000929	8.17E-109
	SD	3080	3.87E-85	3.6E-139	2.8E-102	2.2E-53	0	1.25E-24	405	5.63E-12	0.000137	1.32E-108
f14	M	3010	0	2.5	0	5.65	0	0	592	0	0	0.0000176
	SD	5710	0	8.24	0	6.43	0	0	723	0	0	5.93E-06
f15	M	51000	355	517	794	8760	6.97E-15	55900	58900	1.72E-10	0.00453	1.34E-104
	SD	19600	259	2310	362	3640	2.91E-14	38900	37600	3.14E-11	0.000809	1.7E-104

listed algorithms, a ranking procedure is conducted. For each function, the fitness values from different algorithms are sorted from smallest to biggest, and the order is as their scores. That means the better performance obtains the higher score. For each algorithm, scores of every functions are summed and the accumulation column graph are shown in Fig. 3.

The optimization ability of the proposed method performs best among all the algorithms from Fig. 3, which proves the effectiveness of our improvements. For function *Schwefel*, the proposed method didn't acquire satisfied results, which indicate that the OLAR-PSO-d is not applicable for some extreme multimodal problems.

From Tables 2-4 for standard deviation, the proposed method performs better in most functions as mentioned before. And it is easy to find the proposed OLAR-PSO-d performs the best in aspect of the algorithm stability.

From the mathematical results of this part, it can be concluded that the reset operator cooperating with the stagnation judgment criterion and the disturbance particles are effective and efficient in improving the performance of standard PSO algorithm. The proposed OLAR-PSO-d algorithm outperforms several modified PSO method and other heuristic algorithms in both optimization ability and algorithm stability.

5. The lightweight design of an auto-body considering crashworthiness using OLAR-PSO-d

In this section, the advantage of the proposed OLAR-PSO-d algorithm is further illustrated with the crashworthiness constrained vehicle lightweight design problem. Five crash cases are considered in this article. A Kriging surrogates model technique is employed to reduce the computational consumption. The proposed mothed is proceeded as following procedure:

Table 4. Optimization results with 100 dimensions.

No.	Index	BBPSO	CLPSO	APSO	DMS-PSO	DE/best/1	ODE	ABC	GABC	IGHS	GDHS	OLAR-PSO-d
f1	M	19.7	2.74E-14	6.6	0.991	7.31	0.000172	3.96E-11	7.03	5.91E-06	0.0144	0
	SD	0.309	5.17E-15	3.72	4.43	7.16	0.000544	3.22E-11	2.47	1.63E-07	0.000671	0
f2	M	176	3.33E-17	9.05	0.00074	0.0371	0.00074	1.05E-12	21.4	0.00228	0.0045	0
	SD	129	7.29E-17	27.8	0.00228	0.0731	0.00234	2.05E-12	30	0.00427	0.000518	0
f3	M	8.96E+07	0.00933	0.298	0.0156	365000	3.49E-25	2.53E-23	0.293	5.26E-13	6.83E-06	0.0159
	SD	1.50E+08	0.0287	0.41	0.0423	546000	8.66E-25	7.7E-23	0.927	4.25E-14	6.46E-07	0.00602
f4	M	1.03E+08	0.0011	0.00604	0.00549	210000	1.07E-15	5.35E-24	204000	1.86E-06	0.000333	8.78E-10
	SD	1.82E+08	0.00338	0.0104	0.0126	327000	2.23E-15	7.5E-24	444000	5.89E-06	4.66E-05	3.77E-10
f5	M	102	0.007	13.5	0.0064	0.0712	0.00183	0.806	0.857	0.018	0.00287	0.0000219
	SD	100	0.00153	30	0.00256	0.0197	0.00202	0.681	0.627	0.00217	0.000574	0.0000113
f6	M	89.3	7.02	50.3	19.5	108	4.52E-08	0.0108	46.4	4.41E-08	0.0116	0
	SD	28.9	100	26.1	25.9	15.3	1.41E-07	0.0181	16.3	3.47E-09	0.00114	0
f7	M	3.18E+07	146	4860	120	198	9.69E-12	0.342	607000	132	107	93.9
	SD	4.72E+07	47.8	20100	36.2	83.9	3.06E-11	0.223	823000	44.9	23.9	1.09
f8	M	747	3.01	549	11.6	136	12.3	6.94	31.7	40.6	19.4	2.79E-06
	SD	49.4	1.71	53.6	5.19	31.6	7.7	7.84	5.15	7.7	1.2	4.25E-06
f9	M	-38965.4	-38995	-39087.1	-41898.3	-41675.7	-41709.3	-41898.3	-41898.3	-39898.3	-41875.3	-23000
	SD	4660	451	2360	0.00207	0.183	0.0435	0.0198	0.0213	0.976	0.0457	1540
f10	M	127000	7830	33100	1270	95	1.81E-05	88100	89400	2.72E-06	6.17	6.81E-116
	SD	42900	1680	18900	494	27.7	4.69E-05	34700	35100	9.49E-07	0.77	1.67E-115
f11	M	95.9	9.02	35.2	13.2	43.5	0.0927	55.1	53.4	3.53	0.0671	7.76E-108
	SD	1.16	1.7	6.41	2.01	5.54	0.0233	21	19.3	0.267	0.0105	1.41E-107
f12	M	2300	3.45E-45	15	583	8.38E-24	3.60E-06	2.34E-12	84.7	0.396	0.72	7.6E-109
	SD	504	1.52E-44	36.6	35.4	2.27E-23	1.13E-05	6.21E-13	30.6	0.307	0.0416	1.66E-108
f13	M	19500	4.16E-75	2500	4.89E-77	7.68E-37	5.67E-81	1.29E-23	858	2.14E-10	0.00807	7.2E-120
	SD	15400	1.8E-74	4440	1.4E-76	2.22E-36	9.55E-81	2.91E-23	1520	1.22E-11	0.000887	1.31E-119
f14	M	18300	0.1	1010	0.35	260	0	0	2350	0	0	0.724
	SD	16000	0.308	3080	0.745	248	0	0	2900	0	0	0.587
f15	M	273000	28900	17700	32100	131000	429	211000	218000	31.5	0.133	9.91E-108
	SD	45400	6680	10400	5690	12800	819	47200	24200	68.3	0.0161	2.82E-107

5.1 Definition of the optimization problem

A full-vehicle finite element model is established with strict meshing criteria to guarantee the simulation accuracy. The model contains 1038131 elements with 10 mm element size averagely, 7793 beam elements, 16476 solid elements, 1013862 shell elements and 77312 triangular elements in particular. The complete vehicle mass is 2090.9 kg. The comparison of structure deformation of simulation and experiments is shown in Fig. 4. It indicates that the model established is valid for further study.

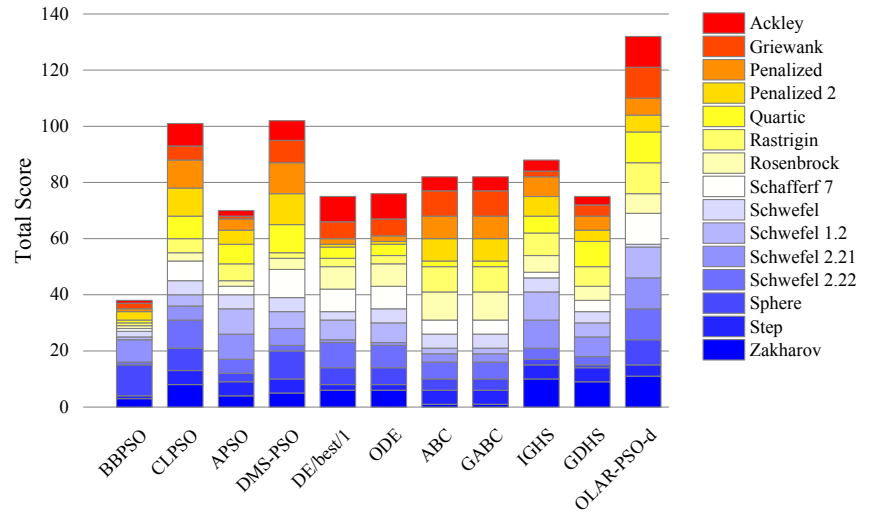
The objective of the optimization is the mass reduction of auto-body, and the thickness of each component is defined as design variable within the range of 0.6 mm to 1.1 times the original thickness value. The constraints are derived from the crash cases working conditions. A brief introduction of FEM simulations is as follows:

According to Chinese Standard *GB11551-2003*, a FM model for frontal impact simulation is established as shown in Fig. 5(a). Fig. 5(b) illustrates the selected 15 design variables which represent the thickness of sheet related to this case.

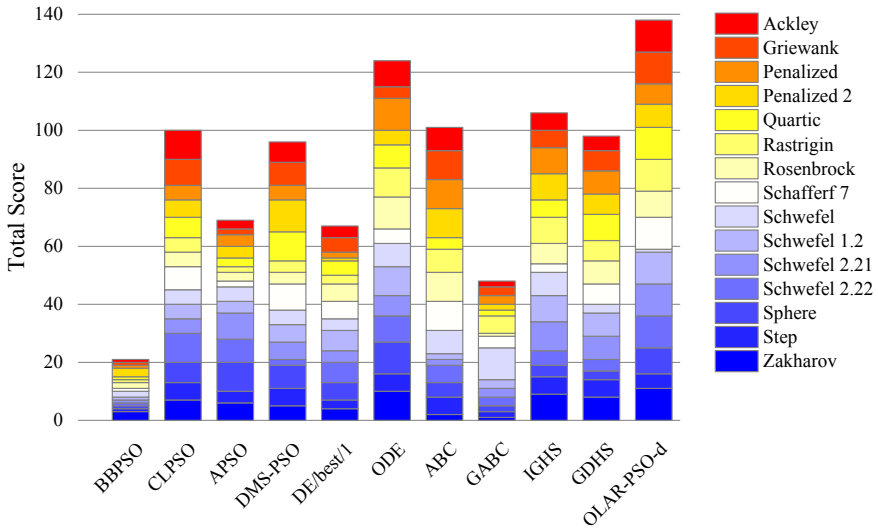
A FM model for frontal offset impact simulation based on the Chinese Standard *GB11551-2003* is established as shown in Fig. 6(a). Fig. 6(b) illustrates the selected 21 design variables which represent the thickness of sheet related to this case.

Fig. 7(a) is the FE model of lateral impact case according to Chinese Standard *GB20071-2006*. Fig. 7(b) illustrates the selected 15 design variables which represent the thickness of sheet related to this case.

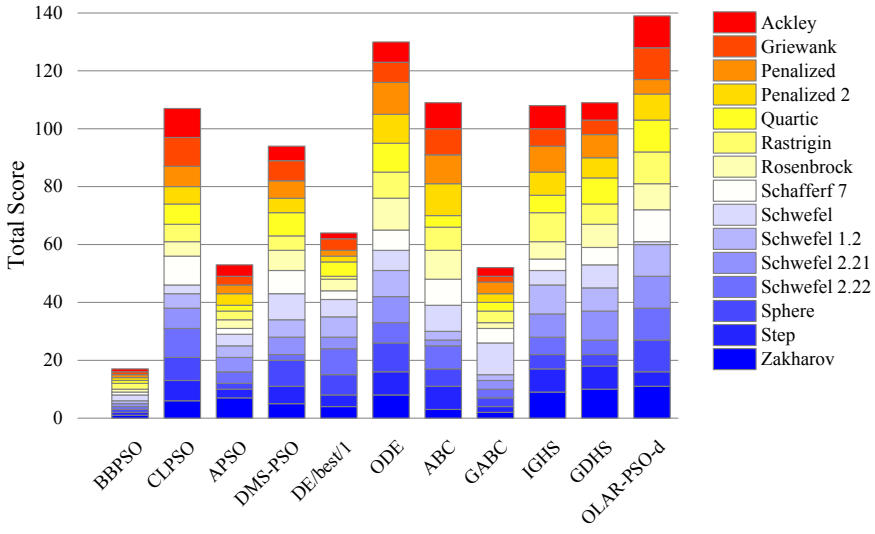
Rear impact simulation based on Chinese standard *GB20072-2006* is established as shown in Fig. 8(a). In this work, the hydrogen storage bottles are placed behind the car. So the performance indicators are related with hydrogen bottle. Fig. 8(b) illustrates the selected 8 design variables which rep-



(a) $D = 30$

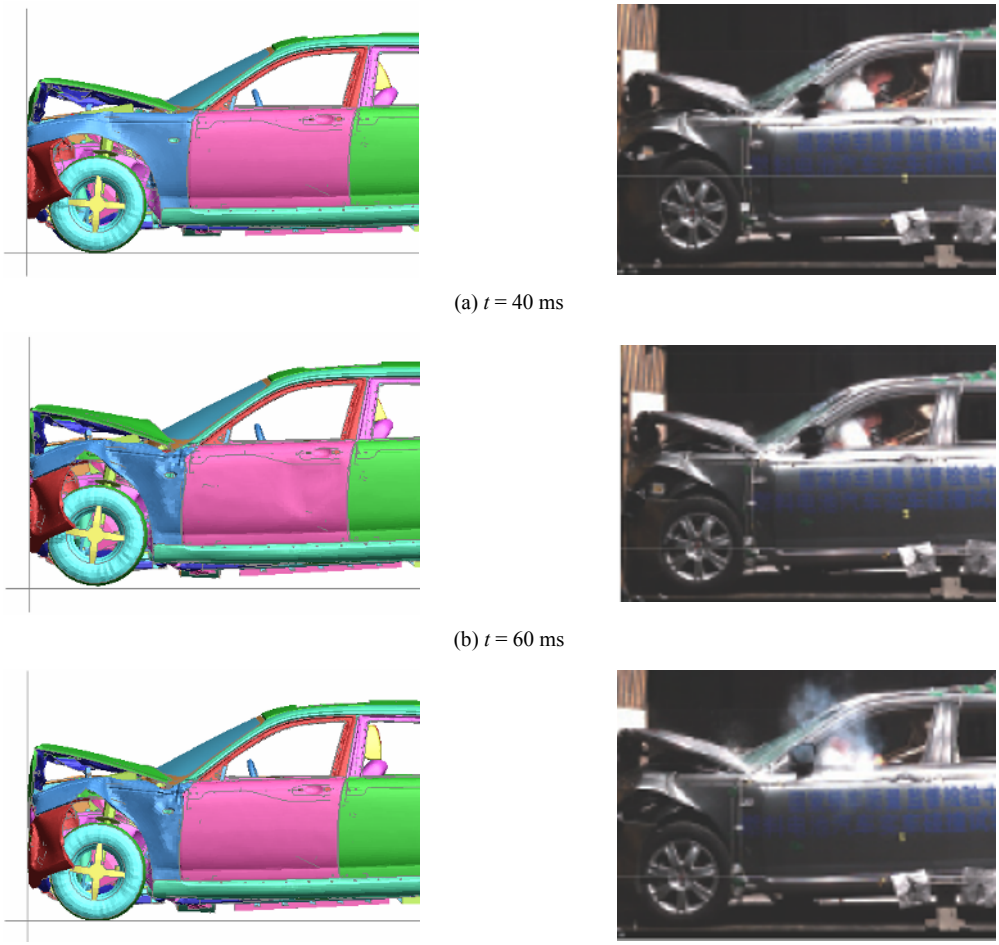


(b) $D = 50$



(c) $D = 100$

Fig. 3. The accumulation column graph of fitness value with different dimensions.

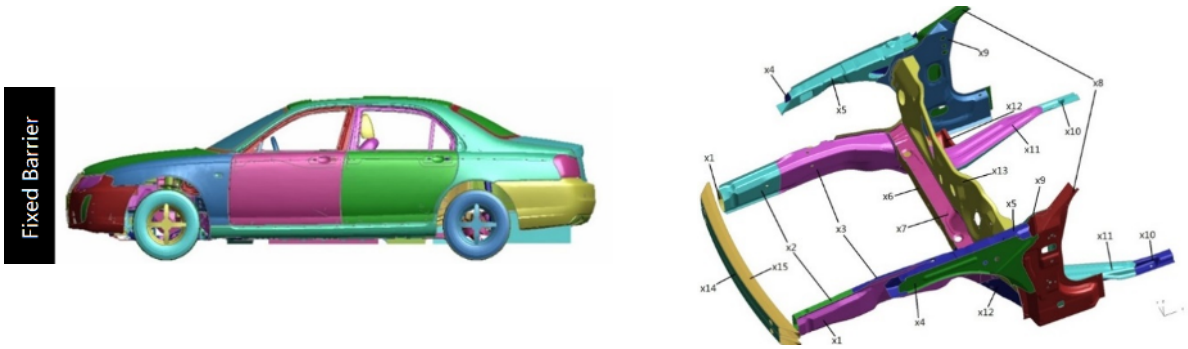


(a) t = 40 ms

(b) t = 60 ms

(c) t = 100 ms

Fig. 4. Comparison between experiment and simulation of frontal impact.



(a) FE model

(b) Design variables

Fig. 5. Case of frontal impact.

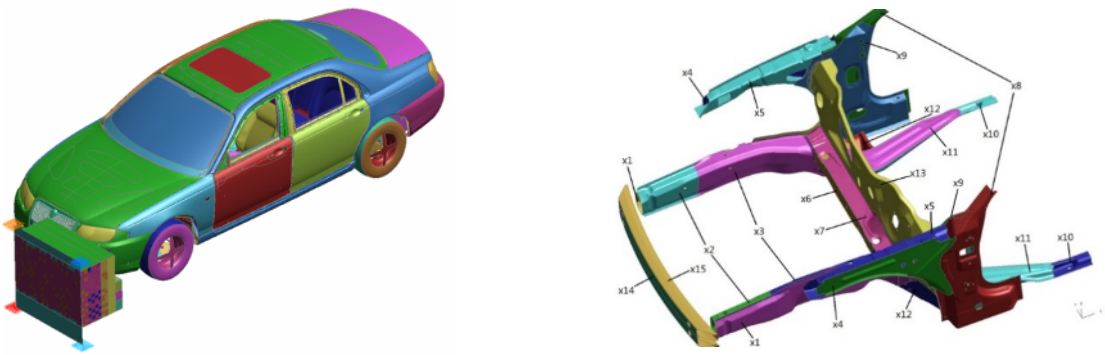
resent the thickness of sheet related to this case.

According to American standard *FMVSS216*, a FM model for roof crush simulation is established as shown in Fig. 9(a). Fig. 9(b) is the selected 10 design variables which represent the thickness of sheet related to this case.

The performance indicators of aforementioned 5 cases are listed in Table 5. Part of the indicators are derived from the previous work [41]. In this problem, the target is to reduce the structural mass. And there are totally 58 design variables optimized together.

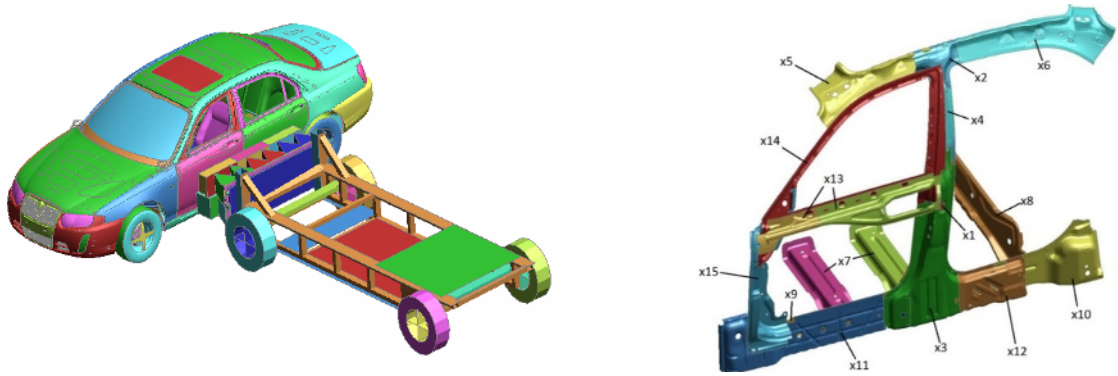
5.2 Construct the surrogate models of each crash case

Kriging surrogate model technique is chosen in this article. Aforementioned OLHD is applied to sample in the design domain. In order to achieve accurate surrogate models, the



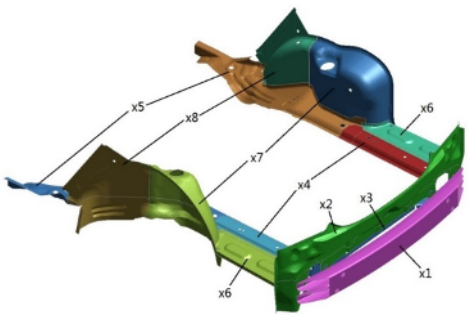
(a) FE model (b) Design variables

Fig. 6. Case of frontal offset impact.



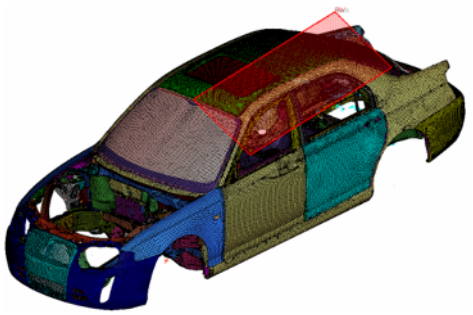
(a) FE model (b) Design variables

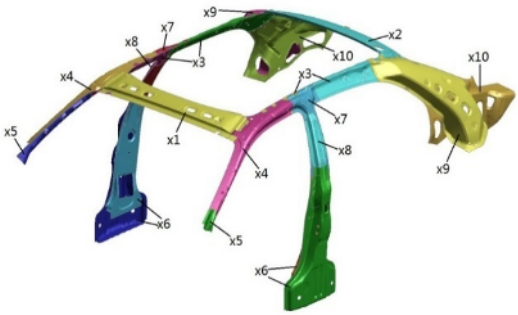
Fig. 7. Case of lateral impact.



(a) FE model (b) Design variables

Fig. 8. Case of rear impact.





(a) FE model (b) Design variables

Fig. 9. FEM model of roof crush.

Table 5. Load cases and constraints.

Load cases	Design variables	Performance indicators	Constraints
Frontal impact	15	Left B-pillar acceleration (g)	≤ 40
		Left toe-board intrusion (mm)	≤ 80
		Right toe-board intrusion (mm)	≤ 80
Frontal offset impact	21	Left B-pillar acceleration (g)	≤ 40
		A-pillar deformation (mm)	≤ 80
		Left toe-board intrusion (mm)	≤ 80
		Right toe-board intrusion (mm)	≤ 80
Lateral impact	15	Low rib deflection (mm)	≤ 32
		B-pillar intrusion velocity (m/s)	≤ 9
		Door deformation velocity (m/s)	≤ 9
		Abdomen acting force (kN)	≤ 1.5
		Pubic symphysis acting force (kN)	≤ 4
Rear impact	8	Left contact force of hydrogen bottle (kN)	≤ 50
		Middle contact force of hydrogen bottle (kN)	≤ 50
		Right contact force of hydrogen bottle (kN)	≤ 50
Roof crush	10	Resistance force (kN)	≥ 50

Table 6. Surrogate model.

Load cases	Performance indicators	Training points	Added points	R^2
Frontal impact	Left B-pillar acceleration (g)	100	19	0.9014
	Left toe-board intrusion (mm)		11	0.9120
	Right toe-board intrusion (mm)		--	0.9012
Frontal offset impact	Left B-pillar acceleration (g)	160	13	0.9035
	A-pillar deformation (mm)		11	0.9243
	Left toe-board intrusion (mm)		19	0.9007
	Right toe-board intrusion (mm)		--	0.9124
Lateral impact	Low rib deflection (mm)	120	10	0.9185
	B-pillar intrusion velocity (m/s)		15	0.9374
	Door deformation velocity (m/s)		11	0.9002
	Abdomen acting force (kN)		--	0.9156
	Pubic symphysis acting force (kN)		--	0.9324
Rear impact	Left contact force of hydrogen bottle (kN)	100	--	0.9052
	Middle contact force of hydrogen bottle (kN)		--	0.9346
	Right contact force of hydrogen bottle (kN)		6	0.9156
Roof crush	Resistance force (kN)	80	--	0.9211

target oriented sequential sampling technique is used [42]. The coefficient of determination (R^2) is employed to verify the accuracy of surrogate models.

The result of the built surrogate models is listed in Table 5. Since all the R^2 is greater than 0.9, the model is valid.

5.3 Optimization process based on OLAR-PSO-d and verification

The lightweight design problem is optimized by the proposed OLAR-PSO-d method. The widely used non-stationary penalty function method is applied to transform the constraints into a sequence of unconstrained optimization problems [43].

6. Conclusions

In this article, the OLAR-PSO-d algorithm is proposed, in which the optimal LHD technique is used for swarm initialization and the adaptive reset operator acted on velocity is adopted to enhance the diversity of particles and prompt the optimization program jumping out from stagnation. From the numerical experiments, conclusions are summarized as follows:

• Compared with the standard PSO with linearly decreased

Table 7. Performance indicators after optimization.

Load cases		Performance indicators	Constraints	Before	After
Crash cases	Frontal impact	Left B-pillar acceleration (g)	≤ 40	35.69	37.70
		Left toe-board intrusion (mm)	≤ 80	81.47	79.10
		Right toe-board intrusion (mm)	≤ 80	81.83	75.41
	Frontal offset impact	Left B-pillar acceleration (g)	≤ 40	28.90	32.36
		A-pillar deformation (mm)	≤ 80	31.19	67.83
		Left toe-board intrusion (mm)	≤ 80	117.26	72.16
		Right toe-board intrusion (mm)	≤ 80	71.31	73.85
	Lateral impact	Low rib deflection (mm)	≤ 32	30.26	30.41
		B-pillar intrusion velocity (m/s)	≤ 9	8.00	8.38
		Door deformation velocity (m/s)	≤ 9	8.93	8.12
		Abdomen acting force (kN)	≤ 1.5	1.53	1.35
		Pubic symphysis acting force (kN)	≤ 4	2.05	3.16
	Rear impact	Left contact force of hydrogen bottle (kN)	≤ 50	41.18	42.36
		Middle contact force of hydrogen bottle (kN)	≤ 50	52.78	47.16
		Right contact force of hydrogen bottle (kN)	≤ 50	39.94	45.37
	Roof crush	Resistance force (kN)	≥ 50	66.37	62.18
Mass (kg)				214.67	194.15

weight factor and with constriction factor versions, the proposed OLAR-PSO-d demonstrates its advantage in global optimal searching. Due to the scatter distribution of the initialized swarm and the proposed reset process, the convergence rate of proposed method is a slightly lower than the standard versions. However, the outcomes of OLAR-PSO-d present the effectiveness and efficiency to enhance the performance of the standard PSO.

- The modified PSO algorithm using the most common used single point crossover and single point mutation operators is chosen as comparative case. From the tests results, the OLAR-PSO-d algorithm outperforms the modified PSO method in both optimization ability and algorithm stability.
- Combined with the Kriging surrogate model technique and the non-stationary penalty functions method, the proposed method is applied to solve a vehicle lightweight design problem. The weight of the auto-body is successfully decreased by 9.56 % with satisfying all the crashworthiness requirements.

Acknowledgments

The research leading to the above results was supported by National Natural Science Foundation of China (Grant No. 11772191), National Science Foundation for Young Scientists of China (Grant No. 51705312) and National Postdoctoral Foundation of China (Grant No. 2017M61156).

References

[1] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, 4 (1995) 1942-1948.

[2] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, *Proceedings of the 1999 Congress on Evolutionary Computation*, 3 (2002) 1957.

[3] M. Clerc and J. Kennedy, The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6 (1) (2002) 58-73.

[4] V. Miranda, J. D. H. Martins and V. Palma, Optimizing large scale problems with metaheuristics in a reduced space mapped by autoencoders—application to the wind-hydro coordination, *IEEE Transactions on Power Systems*, 29 (6) (2014) 3078-3085.

[5] Y. Marinakis, An improved particle swarm optimization algorithm for the capacitated location routing problem and for the location routing problem with stochastic demands, *Applied Soft Computing*, 37 (C) (2015) 680-701.

[6] B. Haddar, M. Khemakhem, S. Hanafi and C. Wilbaut, A hybrid quantum particle swarm optimization for the multidimensional knapsack problem, *Engineering Applications of Artificial Intelligence*, 55 (C) (2016) 1-13.

[7] R. C. Eberhart and Y. H. Shi, Guest editorial special issue on particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, 8 (3) (2004) 201-203.

[8] C. K. Monson and K. D. Seppi, Adaptive diversity in PSO, *Proceedings of Genetic and Evolutionary Computation Conference*, *GECCO 2006*, Seattle, Washington, Usa, July (2006) 59-66.

[9] S. M. A. Salehizadeh, P. Yadmellat and M. B. Menhaj, Local optima avoidable particle swarm optimization, *Swarm In-*

telligence Symposium, IEEE (2009) 16-21.

[10] A. Ratnaweera, S. K. Halgamuge and H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation*, 8 (3) (2004) 240-255.

[11] J. Riget and J. S. Vesterstrøm, A diversity-guide particle swarm optimizer - the ARPSO, *EVALife Technical Report* (2002) 2.

[12] M. Pant, T. Radha and V. P. Singh, A simple diversity guided particle swarm optimization, *IEEE Congress on Evolutionary Computation CEC 2007*, 8 (2007) 3294-3299.

[13] J. Sun, W. Xu and W. Fang, A diversity-guided quantum-behaved particle swarm optimization algorithm, *Simulated Evolution and Learning, International Conference*, China (2006) 497-504.

[14] H. Wang, H. Sun, C. Li, S. Rahnamayan and J. S. Pan, Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, 223 (2) (2013) 119-135.

[15] A. Meng, Z. Li, H. Yin, S. Chen and Z. Guo, Accelerating particle swarm optimization using crisscross search, *Information Sciences*, 329 (C) (2016) 52-72.

[16] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance, *Proceedings of the 1999 Congress on Evolutionary Computation*, 3 (1999) 1938.

[17] R. Mendes, J. Kennedy and J. Neves, The fully informed particle swarm: Simpler, maybe better, *IEEE Transactions on Evolutionary Computation*, 8 (3) (2004) 204-210.

[18] T. Peram, K. Veeramachaneni and C. K. Mohan, Fitness-distance-ratio based particle swarm optimization, *Proceedings of the Swarm Intelligence Symposium* (2003).

[19] Y. Li, Z. H. Zhan, S. Lin, J. Zhang and X. Luo, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, *Information Sciences*, 293 (3) (2015) 370-382.

[20] M. D. McKay, R. J. Beckman and W, J. Conover, A comparison of three methods for selecting values of input variables from a computer code, *Technometrics*, 21 (1979) 239-245.

[21] R. L. Iman and W. J. Conover, Small sample sensitivity analysis techniques for computer models with an application to risk assessment, *Communications in Statistics - Theory and Methods*, 9 (17) (1980) 1749-1842.

[22] R. C. Eberhart and Y. Shi, Particle swarm optimization: developments, applications and resources, *Proceedings of the 2001 Congress on Evolutionary Computation*, 1 (2002) 81-86.

[23] Y. H. Shi and R. C. Eberhart, Empirical study of particle swarm optimization, *IEEE International Conference on Evolutionary Computation*, Washington, DC, USA, 3 (1997) 101-106.

[24] J. Kennedy, The particle swarm: Social adaptation of knowledge, *IEEE International Conference on Evolutionary Computation*, Indianapolis, Indiana (1997) 303-308.

[25] Y. H. Shi and R. C. Eberhart, A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation*, Anchorage, AK (1998) 69-73.

[26] M. Clerc, *Confinements and biases in particle swarm optimization*, http://clerc. maurice.free.fr/pso (2006).

[27] R. Jin, W. Chen and A. Sudjianto, An efficient algorithm for constructing optimal design of computer experiments, *Journal of Statistical Planning and Inference*, 134 (1) (2005) 268-287.

[28] M. D. Morris and T. J. Mitchell, Exploratory designs for computational experiments, *Journal of Statistical Planning and Inference*, 43 (3) (1995) 381-402.

[29] R. E. Perez and K. Behdinan, Particle swarm approach for structural design optimization, *Computers Structures*, 85 (19-20) (2007) 1579-1588.

[30] J. Kennedy, Bare bones particle swarms, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium* (2003) 80-87.

[31] 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,* 10 (2006) 281-295.

[32] Z. H. Zhan, J. Zhang, Y. Li and H. S. H. Chung, Adaptive particle swarm optimization, *IEEE Transactions on Systems Man & Cybernetics Part B*, 39 (2009) 1362-1381.

[33] J. J. Liang and P. N. Suganthan, Dynamic multi-swarm particle swarm optimizer with local search, *Proceedings of The 2005 IEEE Congress on Evolutionary Computation*, 1 (2005) 522-528.

[34] R. Storn and K. Price, Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization,* 11 (1997) 341-359.

[35] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, Opposition-based differential evolution, *IEEE Transactions on Evolutionary Computation,* 12 (2008) 64-79.

[36] D. Karaboga and B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing,* 8 (2008) 687-697.

[37] G. P. Zhu and S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics & Computation,* 217 (2010) 3166-3173.

[38] E. A. Mohammed, An improved global-best harmony search algorithm, *Applied Mathematics & Computation,* 222 (2013) 94-106.

[39] M. Khalili, R. Kharrat, K. Salahshoor and M. H. Sefat, Global dynamic harmonysearch algorithm: GDHS, *Applied Mathematics & Computation,* 228 (2014) 195-219.

[40] H. B. Ouyang, L. Q. Gao and S. Li, Improved global-best-guided particle swarm optimization with learning operation for global optimization problems, *Applied Soft Computing*, 52 (C) (2016) 987-1008.

[41] L. Zhao, Z. Ping and C. Wei, Multidisciplinary optimization of auto-body lightweight design using modified particle swarm optimizer, *11th World Congress on Structural*

and Multidisciplinary Optimisation, Sydney Australia (2015).

[42] D. R. Jones, M. Schonlau and W. J. Welch, Efficient global optimization of expensive black-box functions, *Journal of Global Optimization*, 13 (4) (1998) 455-492.

[43] J. M. Yang, Y. P. Chen, J. T. Horng and C. Y. Kao, Applying family competition to evolution strategies for constrained optimization, *Evolutionary Programming Vi, International Conference*, Ep97, Indianapolis, Indiana, Usa, April 13-16, 1213 (1997) 201-211.

Zhao Liu is a post-doctor of Shanghai Jiao Tong University. He got his Ph.D. degree from Shanghai Jiao Tong University in 2016. Zhao's research areas include: a) Intelligent optimization algorithm; b) Multidisciplinary optimization design; c) Lightweight design of Auto-body; d) Multi-scale optimization design. He is responsible for two national foundation projects of China which are related to the improvement of PSO algorithm.