Particle Swarm Optimization (PSO) #2

Particle Swarm Optimizer Incorporating a Weighted Particle (EPSOWP) -- An Enhanced Variant

Outline

- Introduction
- Preliminaries of Particle Swarm Optimization (PSO)
- Weighted Particle
- Enhanced Particle Swarm Optimizer Incorporating a Weighted Particle (EPSOWP)
- Simulation Results
- Conclusions

Introduction

- Particle swarm optimization (PSO) was introduced by Kennedy and Eberhart in 1995. Based on simulation of simplified animal social behaviors such as fish schooling, bird flocking, etc.
- In order to improve the performances of conventional PSO, many variants have been, considering the selection of control parameters for optimality and convergence.

Preliminaries of Particle swarm optimization

- ***Particle swarm optimization***
  - PSO searches a space by adjusting the trajectories of individual vectors, called "particles", which are conceptualized as moving points in multidimensional space.
  - Each particle ***i*** moves through a ***n***-dimensional search space with two associated vectors,

    position vector $x_i(t) = \{x_{i1}(t), x_{i2}(t),\ldots, x_{in}(t)\}$ and

    velocity vector $v_i(t) = \{v_{i1}(t), v_{i2}(t),\ldots,v_{in}(t)\}$, $i = 1, 2, \ldots, P$,

    in a population ***P*** for the current evolutionary iteration ***t***.

Preliminaries of Particle swarm optimization

- The particle behavior in a PSO can be modeled as:

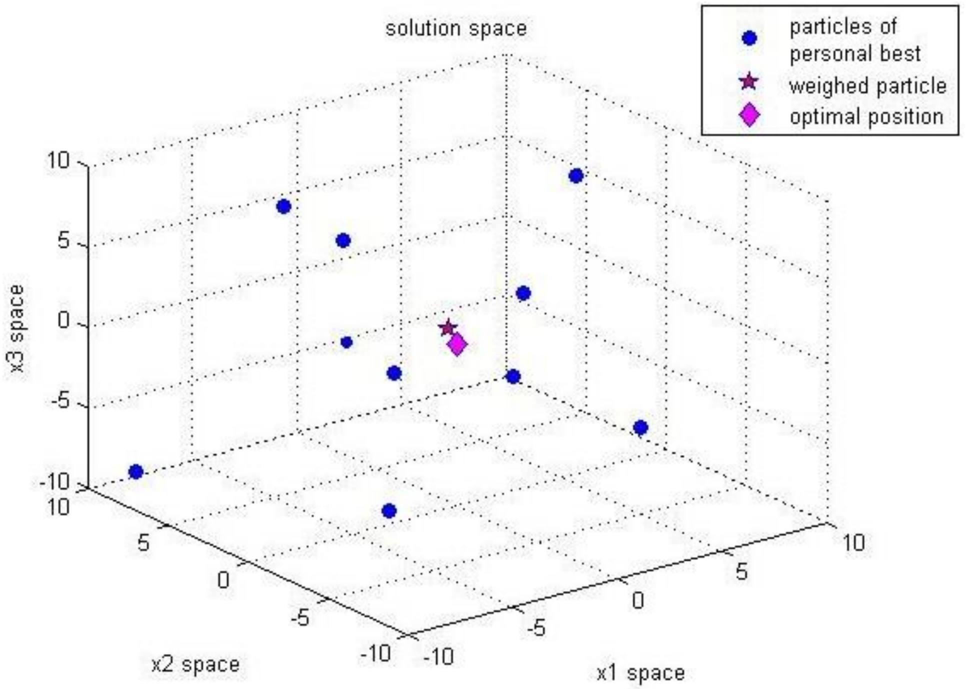
$$v_i(t+1) = w \times v_i(t) + c_1 \times rand \times (pbest_i - x_i(t)) + c_2 \times rand \times (Gbest - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

- c_1 and c_2 are acceleration constants;
- ***rand*** is random number between 0 and 1;
- w is inertia weight factor
- G_{best} is the best previous position among all the particles and $pbest_i$ is the best previous position of particle *i*.

Preliminaries of Particle swarm optimization

- The swarm, which is initialized by a random population to search the best solution, is associated with movement toward G_{best} and $pbest_i$ locations of particles. However, G_{best} and $pbest_i$ represent two directions of movement for each particle.



Preliminaries of Particle swarm optimization

- Problems of conventional PSOs
  - The convergence of optimization becomes staggered, especially in later stage during the evolution process.
  - Other issues include proper control of global exploration and local exploitation as well as sensitivity of the control parameters

Weighed Particle

- The weighed particle x_w in a swarm of a PSO algorithm plays a critical role during the optimization process, where the position of the weighed particle can be calculated as:

$$x_w = \sum_{i=1}^{P} \bar{c}_{w\_i} x_{pbest\_i} \, , \qquad (3)$$

$$\bar{c}_{w\_i} = \frac{\hat{c}_{w\_i}}{\sum_{j=1}^{P} \hat{c}_{w\_j}} \, , \; i = 1, 2, \ldots, P \, , \qquad (4)$$

$$\hat{c}_{w\_i} = \frac{\max\limits_{1 \le i \le P}(f(x_{pbest\_i})) - f(x_{pbest\_i})}{\max\limits_{1 \le i \le P}(f(x_{pbest\_i})) - \min\limits_{1 \le i \le P}(f(x_{pbest\_i}))} \, , \; i = 1, 2, \ldots, P \, , \qquad (5)$$

- $f(\cdot)$ presents a fitness value of the benchmark function.

Weighed Particle

- The weighed particle x_w is generally closer to the optimal position than G_{best}
- The weighed particle is employed in PSO to allow a swarm to adapt in the search space during the optimization process.
- More importantly, the weighed particle often attracts other particles and guides the search direction of the whole swarm.
- The weighed particle is generated when a random value is lower than an attraction value α

Enhanced Particle Swarm Optimizer Incorporating a Weighted Particle (EPSOWP)

- Mechanism I in EPSOWP:

If random number $\leq \alpha$

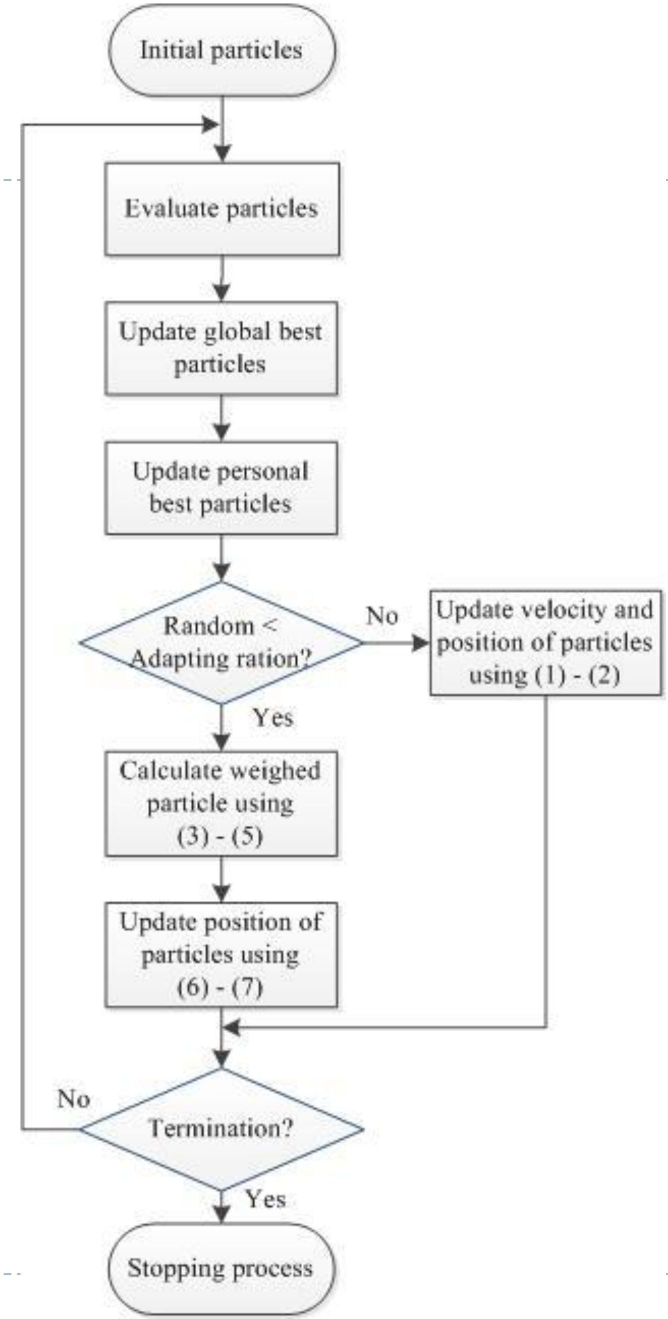
$$v_i(t+1) = w \times v_i(t) + c_3 \times rand \times (pbest_i - x_i(t)) + c_4 \times rand \times (\boxed{x_w} - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

If random number $> \alpha$

$$v_i(t+1) = w \times v_i(t) + c_1 \times rand \times (pbest_i - x_i(t)) + c_2 \times rand \times (Gbest - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$


Initial particles
Evaluate particles
Update global best particles
Update personal best particles
Random < Adapting ration?
No
Update velocity and position of particles using (1) - (2)
Yes
Calculate weighed particle using (3) - (5)
Update position of particles using (6) - (7)
No
Termination?
Yes
Stopping process

Enhanced Particle Swarm Optimizer Incorporating a Weighted Particle (EPSOWP)

- Mechanism II in EPSOWP:

If random number $\leq \alpha$

$v_i(t+1)$ (using the original velocity updating rule)

$$x_i(t+1) = x_i(t) + c_4 \times rand \times (\boxed{x_w} - x_i(t))$$

If random number $> \alpha$

$$\begin{aligned} v_i(t+1) = {} & w \times v_i(t) + c_1 \times rand \times (\boxed{pbest_j} - x_i(t)) \\ & + c_2 \times rand \times (Gbest - \boxed{pbest_j}) + c_3 \times rand \times (\boxed{x_w} - \boxed{pbest_j}) \end{aligned}$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Enhanced Particle Swarm Optimizer Incorporating a Weighted Particle (EPSOWP)

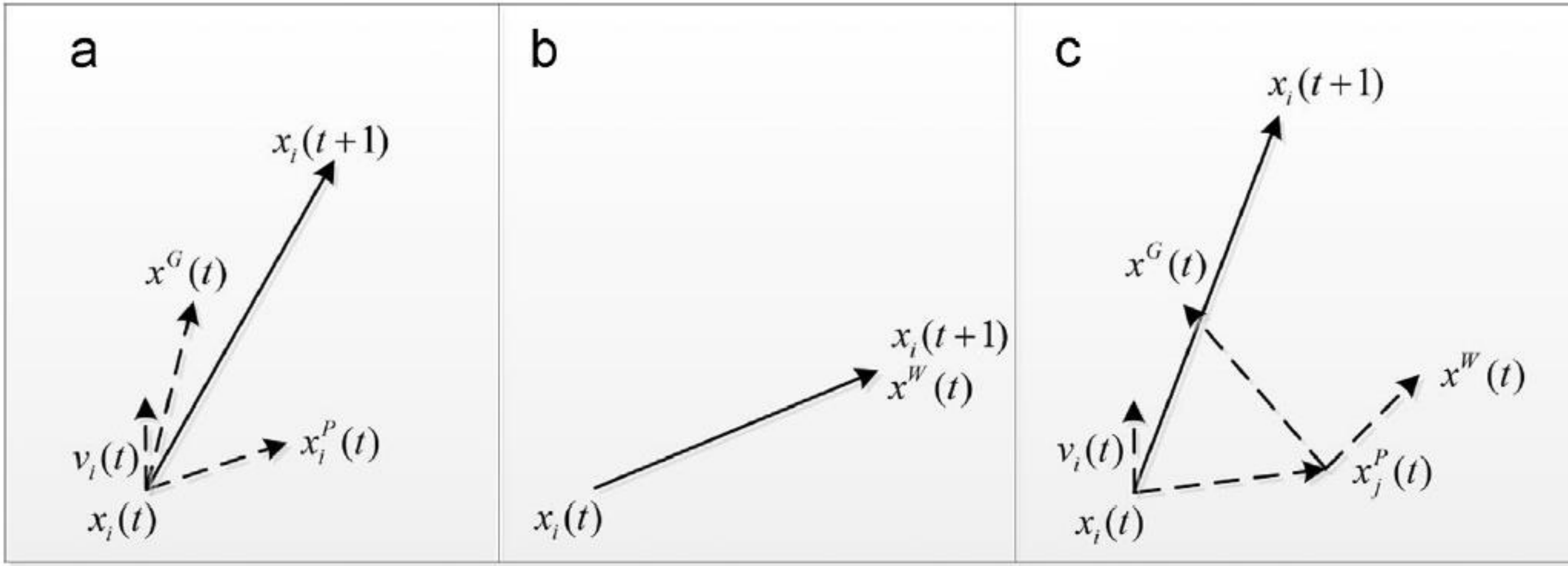


Fig. 1. Geometric views of (a) PSO (b) EPSOWP with $rand_i \le \alpha$ (c) EPSOWP $rand_i > \alpha$.

Simulation Results

- Basis for comparison
  - An initial population is randomly generated from the search space of the optimization problem
  - c_1, c_2, c_3 and c_4 are set as 2.
  - The attraction value is set 0.4.
  - w is a random factor between 0.5 and 0.55 during the iteration.
  - The population size P is set as 20 and all algorithms are run for the maximum function evaluation, 180000, as the computation cost

Simulation Results

- The best fitness value of the six 10-dimensional benchmark functions below is minimum value for searching the best solution.

1. The Sphere function (unimodal)

$$f_1(x) = \sum_{k=1}^{10} x_k^2 \tag{8}$$

2. The Schwefel function (unimodal)

$$f_2(x) = \sum_{k=1}^{10} |x_k| + \prod_{k=1}^{10} |x_k| \tag{9}$$

3. The Rosenbrock function (unimodal)

$$f_3(x) = \sum_{k=1}^{10} 100 \times (x_{k+1} - x_k^2)^2 + (1 - x_k)^2 \tag{10}$$

Simulation Results

4. The Ackley function (multimodal)

$$f_4(x) = -20e^{-0.2\sqrt{\frac{1}{10}\sum_{k=1}^{10} x_k^2}} - e^{(1/10)\sum_{k=1}^{10}\cos(2\pi x_k)} + 20 + e \qquad (11)$$

5. The Grewank function (multimodal)

$$f_5(x) = \sum_{k=1}^{10} \frac{x_k^2}{4000} - \prod_{k=1}^{10} \cos\left(\frac{x_k}{\sqrt{k}}\right) \qquad (12)$$

6. The Rastrigin function (multimodal)

$$f_6(x) = \sum_{k=1}^{10} (x_k^2 - 10\cos(2\pi x_k) + 10) \qquad (13)$$

Simulation Results

- Parameters
  - c_1, c_2, c_3 and c_4 are set as 2.
  - The attraction value is set 0.4.
  - w is a random factor between 0.5 and 0.55 during the iteration.
  - The population size P is set as 20 and all algorithms are run for the maximum function evaluation, 180000, as the computation cost
  - 100 independent runs
- Define

$$accu.dist(G_{best}, x_{opt}) = \sum_{k=1}^{T} \left\| G_{best}(k) - x_{opt} \right\|$$

$$accu.dist(x_w, x_{opt}) = \sum_{k=1}^{T} \left\| x_w(k) - x_{opt} \right\|$$

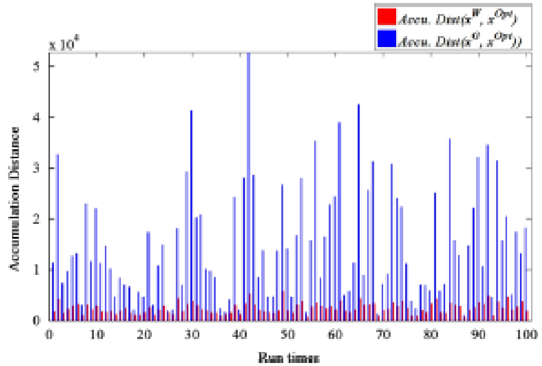


Fig. 3. The comparison of accumulation distance between *Accu. dist*(x^G, x^{Opt}) and *Accu. dist*(x^W, x^{Opt}) for $F_1(x)$.

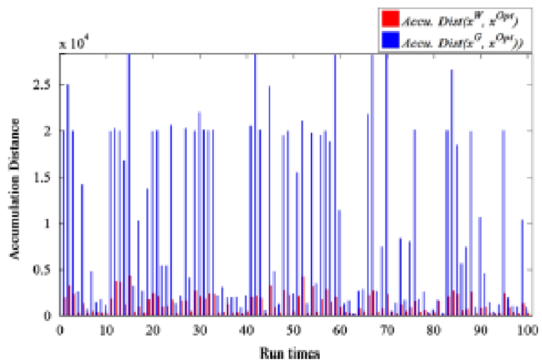


Fig. 4. The comparison of accumulation distance between *Accu. dist*(x^G, x^{Opt}) and *Accu. dist*(x^W, x^{Opt}) for $F_2(x)$.

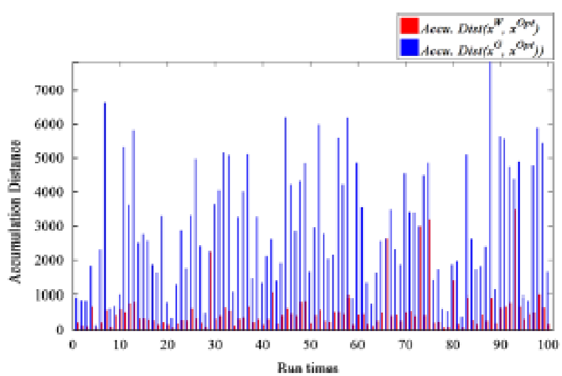


Fig. 5. The comparison of accumulation distance between *Accu. dist*(x^G, x^{Opt}) and *Accu. dist*(x^W, x^{Opt}) for $F_3(x)$.

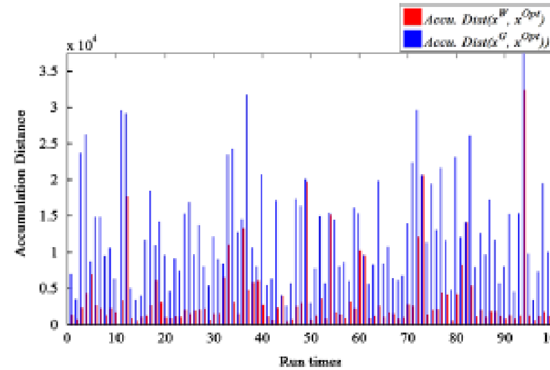


Fig. 6. The comparison of accumulation distance between *Accu. dist*(x^G, x^{Opt}) and *Accu. dist*(x^W, x^{Opt}) for $F_4(x)$.

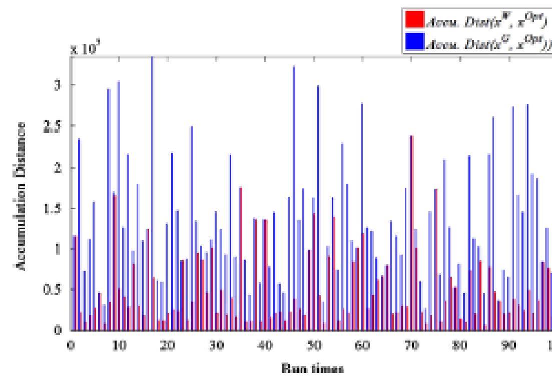


Fig. 7. The comparison of accumulation distance between *Accu. dist*(x^G, x^{Opt}) and *Accu. dist*(x^W, x^{Opt}) for $F_5(x)$.

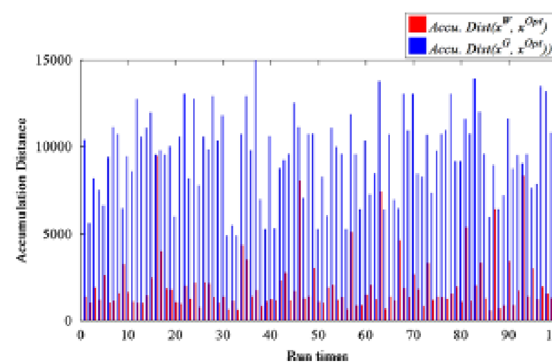


Fig. 8. The comparison of accumulation distance between *Accu. dist*(x^G, x^{Opt}) and *Accu. dist*(x^W, x^{Opt}) for $F_6(x)$.

The population size M is set as 20 and the termination condition

Simulation Results

PSO, CRPSO, GA, DE and EPSOWP for six benchmark functions.

itness	PSO	CRPSO	GA	DE
Mean	1.0675e−024	6.1992e−039	2.1074e−008	2.7970e−016
td.	4.2664e−024	1.4934e−038	1.0306e−008	1.7708e−016
Mean	2.5503e−017	2.2927e−021	1.3332e−001	5.0090e−010
td.	1.0104e−016	3.4088e−021	1.4673e−001	1.8440e−010
Mean	2.9552e+001	1.4372e+001	9.7191e+001	**2.5418e−011**
td.	2.2389e+001	2.4899e+000	1.8175e+001	**2.5069e−012**
Mean	4.5581e−013	1.8971e−014	4.9296e−012	5.7289e−009
td.	1.4563e−012	5.1062e−015	2.0079e−012	2.0900e−009
Mean	1.2388e−002	6.3070e−003	1.2744e−009	**6.3192e−012**
td.	1.3601e−002	8.8200e−003	6.1488e−010	**2.6574e−011**
Mean	3.1122e+001	1.3823e+002	8.8949e+000	1.3089e+002
td.	9.1057e+000	5.1833e+001	3.7033e+000	9.8771e+000

Simulation Results

Table 3
Simulation results of LPSO, MPSO, LDWPSO and EPSOWP for five benchmark functions.

Fun.	Dim.	LPSO				MPSO				LDWPSO				EPSOWP			
		Best	Worst	Average Deviation	Time (s)	Best	Worst	Average Deviation	Time (s)	Best	Worst	Average Deviation	Time (s)	Best	Worst	Average Deviation	Time (s)
F_1	10	5.93e−77	8.20e−73	2.56e−75	1.22	4.22e−30	3.18e−28	3.71e−29	1.02	1.01e−29	4.53e−28	2.50e−28	**0.98**	**0**	**0**	**0**	0.99
	20	4.33e−32	1.13e−29	4.66e−30	1.31	1.68e−18	2.15e−12	6.49e−17	1.26	6.99e−12	3.97e−11	1.30e−11	**1.22**	**1.52e−274**	**4.38e−247**	**0**	2.09
	30	1.95e−14	5.82e−10	2.26e−10	1.64	1.48e−10	6.81e−07	7.33e−08	1.57	2.89e−06	4.63e−05	1.70e−5	**1.48**	**7.44e−139**	**5.43e−114**	**5.43e−115**	2.56
F_3	10	0.004	**0.20**	**0.068**	1.74	0.005	2.13	1.05	**1.66**	0.009	3.27	1.36	1.77	**2.77e−08**	2.98	1.30	2.57
	20	0.007	11.32	4.48	2.12	2.96	69.54	23.57	2.43	3.99	72.78	24.68	**2.05**	**4.98e−07**	**3.99**	**1.57**	3.45
	30	0.62	66.73	23.10	2.77	13.48	91.71	25.63	2.94	12.35	78.23	24.29	**2.68**	**2.42e−05**	**42.91**	**6.93**	4.39
F_4	10	1.54e−09	1.54e−09	0	1.34	1.54e−09	1.54e−09	0	2.26	1.54e−9	1.54e−9	0	**1.25**	**8.88e−16**	**8.88e−16**	**0**	1.76
	20	1.54e−09	1.54e−09	0	2.21	5.63e−07	8.33e−07	0.78e−07	1.99	3.28e−07	4.97e−07	7.10e−07	**1.72**	**8.88e−16**	**8.88e−16**	**0**	2.23
	30	1.58e−09	4.44e−09	1.13e−09	2.47	1.30e−04	3.11e−04	0.99e−04	2.37	1.20e−04	3.92e−04	1.10e−04	**2.08**	**8.88e−16**	**8.88e−16**	**0**	2.69
F_5	10	0.014	**0.07**	**0.03**	0.98	0.014	0.13	0.05	0.90	0.014	0.11	0.04	0.86	**0**	0.13	0.04	**0.42**
	20	0.017	0.056	**0.017**	1.17	0.03	0.34	0.17	1.09	0.03	**0.036**	0.023	0.91	**0**	0.28	0.05	**0.33**
	30	0.012	**0.022**	**0.0048**	1.39	0.05	0.07	0.017	1.07	0.05	0.07	0.018	1.05	**0**	0.48	0.07	**0.55**
F_6	10	0.09	**1.08**	**0.69**	1.22	1.68	2.98	0.94	**1.13**	1.02	2.98	1.46	1.28	**0**	14.92	4.39	1.54
	20	8.95	**12.94**	6.75	1.59	8.13	26.02	8.43	1.64	8.96	21.89	**6.38**	1.42	**0**	23.86	7.52	**1.14**
	30	20.81	37.78	6.52	1.89	22.33	52.18	13.25	1.88	23.88	75.04	10.46	1.43	**0**	**28.79**	**5.96**	**0.56**

Conclusions

- This paper presents a novel strategy where weighed particles help guiding particles of swarm to optimal solution.

- Simulation results show the effectiveness of the EPSOWP to solve high-dimension benchmark functions.

- In light of the satisfactory results obtained in optimizing the benchmark functions, the proposed optimization method has the potential to tackle more complex practical real-world applications.