Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)

# On Trajectories of Particles in PSO

Stefan Janson and Martin Middendorf
Parallel Computing and Complex Systems Group
University of Leipzig, Johannisgasse 26, 04103 Leipzig, Germany
Email: {janson, middendorf}@informatik.uni-leipzig.de

***Abstract*— The moving behaviour of the particles in Particle Swarm Optimization (PSO) algorithms is studied in this paper. It is shown that particles in standard PSO have a clear bias in their movement direction that depends on the direction of the coordinate axes. This has the effect that the optimization behaviour of standard PSO is not invariant to rotations of the optimization function. A second problem of standard PSO is that non-oscillatory trajectories can quickly cause a particle to stagnate. A sidestep mechanism is proposed to improve the movement of the particles. A particle performs a sidestep with respect to a certain dimension when stagnation of movement along this dimension is observed. It is shown for simple test functions that the movement behaviour of sidestep PSO can prevent the unwanted bias and makes PSO less dependent on rotations of the optimization function. It is also shown for standard benchmark functions that sidestep PSO outperforms standard PSO.**

## I. Introduction

In this paper we investigate the moving behaviour of the particles in Particle Swarm Optimization (PSO) algorithms. PSO is a population based method for function optimization (see [5]) where a swarm of individuals, also called particles, iteratively explores a multidimensional search space. Each particle "flies" through the search space according to its velocity vector. This velocity vector is adjusted in every iteration so that prior personal successful positions (cognitive aspect) and the best position found by particles within a specific neighbourhood (social aspect) act as attractors.

Several theoretical studies have been done to analyze the trajectory of a single particle in PSO [1], [7], [8]. All of these works examine the convergence properties of a single particle. For several PSO parameters the values where the algorithm converges have been identified and guidelines for the selection of parameter values have been given. In [6], [1] the specific behaviour that is associated with certain domains of parameter values is characterized and a theoretical framework to examine PSO is provided. In the theoretical analysis, the random component in the movement behaviour of the particles is substituted by its expected value and the behaviour within a single dimension is observed.

In this paper the movement behaviour of particles in PSO is analyzed experimentally. We identify two distint biases of the optimization trajectories of the particles when the standard PSO movement rules are used. To illustrate our findings some simple test functions are used for the experiments. In particular, it is shown that in a two-dimensional search space a single particle has a distinct bias in its movement direction. Ideally, a particle should be influenced only by the good positions in the search space that have been found before. But it is shown here that a bias in the particles movement directions exists, which is influenced by the direction of the coordinate axes. As an effect the PSO optimization behaviour is not invariant to rotations of the optimization function. A second problem of standard PSO algorithms is investigated here, namely, that non-oscillatory trajectories can quickly cause a particle to stagnate.

Several explanatory experiments have been created to clearly illustrate the deficiencies in the particle's trajectories. Therefore the movement of a PSO with only one particle is studied first. Clearly, it does not make much sense for optimization purposes to use only a single particle but it is certainly useful for our investigations. Then it is shown that the same effects which have been observed for a single particle PSO are also relevant for standard multi-particle swarm PSO algorithms. Several adjustments of the standard movement rules are proposed in order to counteract the deficiencies that have been found. The influence of the new movement rules on the optimization behaviour are evaluated experimentally on standard benchmark functions.

In Section II an overview of PSO is given. Some results on the trajectories of PSO particles are described III. Moreover, several modifications of the standard PSO update rules are proposed. Further experimental results are presented in Section IV. Conclusions are given in V.

## II. PSO

In Particle Swarm Optimization [5] a swarm of $m$ particles moves through a multi-dimensional search space. A particle is defined by its current position $x$ and its velocity $v$. Each particle remembers the location $y$ at which it has found the so far best solution to the optimization function $f$. In PSO each particle updates its velocity in each dimension $d$ according to Equation (1), where $r_1$ and $r_2$ are uniformly drawn random numbers from $[0;1]$. The previous velocity is included in this formula, scaled by the inertia weight $w$, and the attraction potential of the personal best $y$ and neighbourhood best position $\hat{y}$, is varied with the respective weights $c_1$ and $c_2$. The position $\hat{y}$ is either the best position of all particles in the swarm (gbest) or the best positions within a local neighbourhood of the particle. One method to define a local neighbourhood (lbest) is to number the particles from 0 to $m-1$ and to define the neighbourhood of particle $i$ as consisting of particle $i$ itself and the two particles $i-1 \bmod m$ and $i+1 \bmod m$.

1-4244-0708-7/07/$20.00 ©2007 IEEE

$$v_d(t+1) = w\, v_d(t) + c_1 r_1 (y_d - x_d) + c_2 r_2 (\hat{y}_d - x_d) \quad (1)$$

After the velocity has been updated, all particles move one step with their newly determined velocity (2).

$$x_d(t+1) = x_d(t) + v_d(t+1) \quad (2)$$

This process is iterated until a sufficient solution quality or a maximum number of iterations has been reached.

Numerous modifications to the standard PSO have been proposed so far in the literature. Several modifications aim to avoid premature convergence to a sub-optimal location. In the original PSO definition a random term, called “craziness” [5], was added to the velocity update. Although it was omitted in subsequent versions of PSO algorithms, several recent PSO variants have re-introduced this random term, e.g., as “turbulence” [3]. In [4] normally distributed random values are used in the velocity update formula instead of values from a uniform distribution within $[0;1]$. This way also values outside of $[0; c_1\,(y_d - x_d) + c_2\,(\hat{y}_d - x_d)]$ become possible increment values of the velocity vector.

In [10] the PSO algorithm has been enhanced with a differential evolution operator. The differential evolution step is performed alternately with a PSO step, upon the current best positions of the swarm.

In [2] the influence of the personal best position on the movement of a particle is substituted with a random perturbation according to the positional difference of two randomly selected other particles. This way the variation is bounded by the current spatial extension of the population, as in differential evolution.

In [9] the velocity update formula is evaluated for whole vectors instead of a component-wise update as it is done usually. This vector-PSO has not the strong bias along a line that is parallel to one of the coordinate axes. Instead is has an analogous bias along $y - \hat{y}$. To avoid this, a random rotation matrix is applied.

All of these modifications introduce additional randomness to the process, to avoid stagnation of the optimization. Our *sidestep* modification, instead, is applied when a particle’s state resembles certain undesired conditions, that are identified in the following section. This simple adjustment is applied deterministically and punctually.

## III. Particle Trajectories

The trajectory of the particle in a single particle PSO is examined here in detail on a simple test function, namely, the two-dimensional Sphere function with $f(x, y) = x_1^2 + x_2^2$. The particle is initialized on a random point of the unit circle around $(0, 0)$ with a random velocity from $[-1; 1]^2$. The neighbourhood best position $\hat{y}$ is set to the personal best position $y$. Then a regular PSO algorithm is executed with only this particle for 100 iterations.

The trajectories of the particle from two test runs are shown in Figure 1. In the left part of Figure 1 the actual trajectories

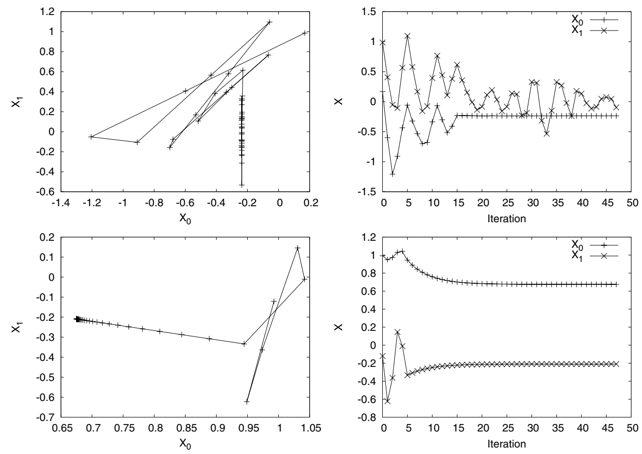


Fig. 1. Two sample trajectories of single particles; right the trajectory in two-dimensional space, left the separate dimensions over time.

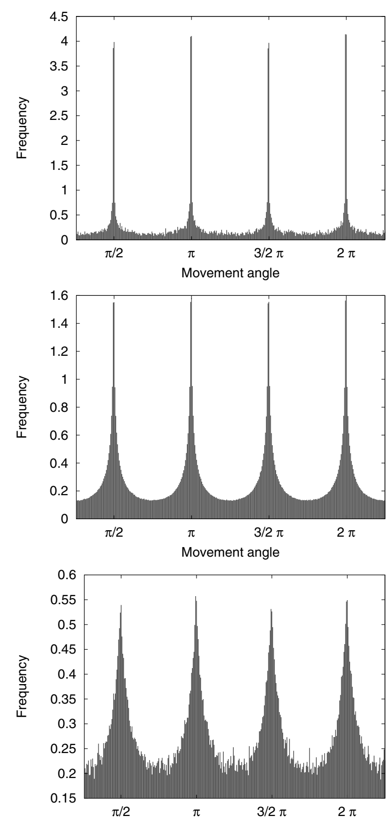


Fig. 2. Frequency of movement angles of particle for the Sphere function, average over 10000 runs; top: single particle standard PSO; middle: standard PSO with five particles; bottom: sidestep PSO with one particle.

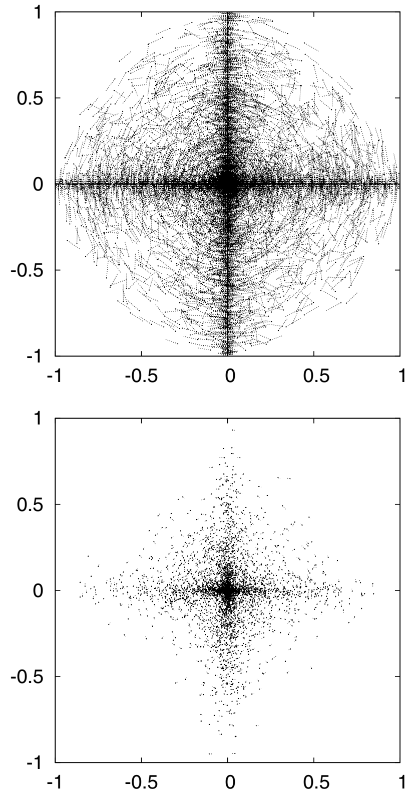

Fig. 3. Best positions and major movement directions of a single particle PSO (10000 runs); upper: standard PSO; lower: sidestep-PSO

can be seen. The right part of the figure shows the current positions $x(t)$ for each dimension separately. In the upper subfigures it can be seen that the particle converged to position (-0.2,0.0) — which is not optimal. For this trajectory, the characteristic oscillation behaviour of particles in PSO could only be observed in the second dimension. The value of $x_1(t)$ vanished after about 15 iterations and subsequently the algorithm only tries to optimize $x_2$. In the second example run, the particle moved towards (0.65,-0.2) and soon finished its progress. When looking at the dimensions of $x(t)$ separately, one can see that both the values $x_1(t)$ and $x_2(t)$ approach their final value from one side instead of performing oscillations around this value.

In the first example run, the particle finishes its movement directly above its current best position and the $v_1$ component of the velocity vector is afterwards set to a value close to 0. In the second example run, the particle improves its personal best position $y$ with every single move. This (actually desired) optimization behaviour prevents the particle from performing an oscillatory movement. Therefore, the length of the movement vector is reduced very quickly.

The previous experiment was repeated 10000 times with different initial positions and velocities. For each run all angles between $x(t+1) - x(t)$ and the $x_1$-axis have been recorded during the run of the particle. The angles were

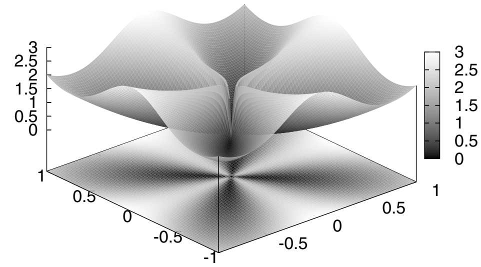

Fig. 4. Rotatable function $R(\omega)$, $\omega = 0$

measured in steps of size $2\pi/360$. Moreover, the final position of the particle was recorded. Figure 2 (upper) shows for the 10000 runs the average angle of during the movement of each particle. It can be seen that most movement steps occurred parallel to one of the coordinate axes. The same movement bias is still present, when 5 particles are used in the PSO (gbest), see Figure 2 (middle), even though the preference for only few angles is smaller.

Figure 3 (upper) shows for the 10000 runs the final position of each particle together with a vector that shows the most frequent angle during the movement of the particle to its final position. The length of the vector corresponds to the relative frequency of the most frequent angle. The figure shows that most of the particles end up at a position next to one of the coordinate axes and the major directions of movement are parallel to the respective other axis.

### *A. Rotation Variance*

A consequence of the bias to the coordinate axes in the movement behaviour of the particles is that the optimization behaviour of PSO changes with rotations of the optimization function around the point of origin. To illustrate this behaviour we define a rotatable function $R(\alpha)$ (3), where $\omega(x_1, x_2) \in [0; 2\,\pi]$ is the angle between $(1, 0)$ and $(x_1, x_2)$. In Figure 4 $R(0)$ is displayed.

$$R1(x_1, x_2) = (\cos\left(4 \cdot (\omega(x_1, x_2) + \alpha\right) + 1) + (x_1^2 + x_2^2) \quad (3)$$

The same experiment as for the Sphere function was performed 10000 times with a single particle PSO. Again the particle has been randomly placed on the unit circle with a random initial velocity. The final positions and major movement directions are given in Figure 5 for the $R(0)$ (left) and $R(\pi/4)$ function(right). One can clearly see that the two shapes differ. For $R(\pi/4)$, as for the Sphere function, most of the particles ended their move on one of the axes. The fanned out ends of the lines along which the final positions are placed are caused by the particles eventually only moving in either $x_1$ or $x_2$ direction.

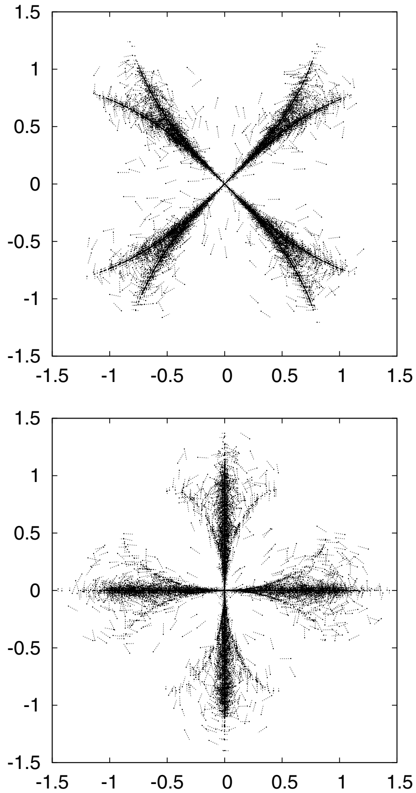


Fig. 5. Best positions and major movement directions of a single particle (1000 runs) for functions $R(0)$ (upper) and $R(\pi/4)$ (lower)

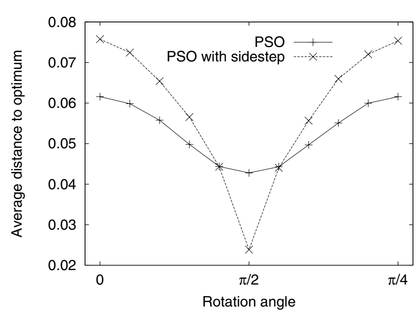


Fig. 6. Average distance to the optimum $(0,0)$ of the function $R(\omega)$, with different $\alpha \in [0;\pi/2]$, for 10000 runs of a standard PSO and a sidestep PSO, each with 5 particles.

### B. Sidestep-PSO

In order to compensate for the above described movement bias we investigate a simple method that does not introduce any further randomness into the procedure. A criterion is defined, that tries to capture the above described situation of stagnation along one dimension $d$ at a threshold level $\rho$. The condition, given in criterion (4), is reached, when, for dimension $d$, the particle is sufficiently close to its personal best position $y_d$ and the global best position $\hat{y}_d$, and also the velocity in this dimension is smaller than the threshold $\rho$. If the criterion (4) is satisfied within dimension $d$, the particle performs a sidestep with respect to this dimension. Thus, the velocity $v_d$ is changed by adding $\lambda \cdot \rho$ if $v_d < 0$ and adding $-\lambda \cdot \rho$ otherwise, where $\lambda > 0$ is a parameter. Afterwards the threshold is adjusted as $\rho := \rho/2$. Initially the threshold is set to some fraction of the search space. We used $\rho := 0.01\,(X_{max} - X_{min})$ as threshold value at the beginning, where $X_{max}$ and $X_{min}$ describe the extension of the search space. The scaling parameter $\lambda$ of the stepwidth was set to $\lambda = 10$.

TABLE I
TEST FUNCTIONS

| | |
|---|---|
| **Sphere:** $[-100;100]^n$ | $F_{Sph}(x) = \sum_{i=1}^{n} x_i^2$ |
| **Rosenbrock:** $[-30;30]^n$ | $F_{Ros}(\overrightarrow{x}) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$ |
| **Rastrigin:** $[-5.12;5.12]^n$ | $F_{Ras}(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ |
| **Griewank:** $[-600;600]^n$ | $F_{Gri}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ |
| **Ackley:** $[-32;32]^n$ | $F_{Ack}(x) = -20 \cdot \exp\left(-0.2\sqrt{1/n \cdot \sum_{i=1}^{n} x_i^2}\right) - \exp\left(1/n \cdot \sum_{i=1}^{n}\cos\left(2\pi \cdot x_i\right)\right) + 20 + e$ |

$$(|x_d - y_d| < \rho)\ \texttt{AND}\ (|x_d - \hat{y}_d| < \rho)\ \texttt{AND}\ (v_d < \rho) \qquad (4)$$

With this sidestep PSO the same experiment with a single particle as for the standard PSO was performed. The final best positions and major move directions of the particle for 10000 repetitions are shown in Figure 3 (lower). Compared with the standard PSO, it can be seen that for the sidestep PSO the final positions of the particles are much closer to the origin and less aligned along the axes. But also the major movement direction is not as dominant (i.e., the relative frequency of the primary movement angle is much smaller compared to other directions) as for the standard PSO. In Figure 2 (right) the relative frequency of the different movement angles is given. The figure shows that there also for sidestep PSO is a bias towards movement along the axes, but it is much less distinct than for the standard PSO.

To investigate a situation with more than one particle, experiments have also been done with $m = 5$ particles on the rotatable function $R(\omega)$. In Figure 6 the average distance to the optimum is shown for the standard PSO and the sidestep-PSO at different rotation angles $\omega$. The dependence of the optimization quality from the rotation angle of the optimization function, that is very strong for the standard PSO, is much weaker for the sidestep PSO.

## IV. Results on Classical Benchmark Functions

In this section we compare the standard PSO and sidestep PSO, in order to investigate whether their different optimization behaviour and the results of the last section also

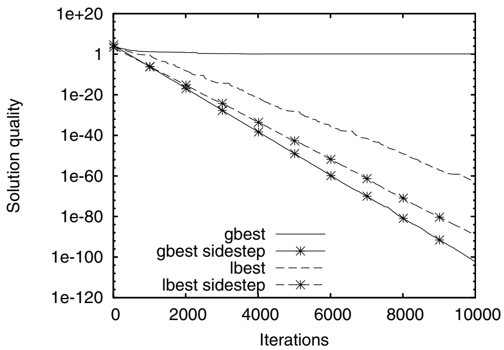


Fig. 7. Solution quality of standard PSO (lbest/gbest) and sidestep PSO (lbest/gbest) for the Sphere function.

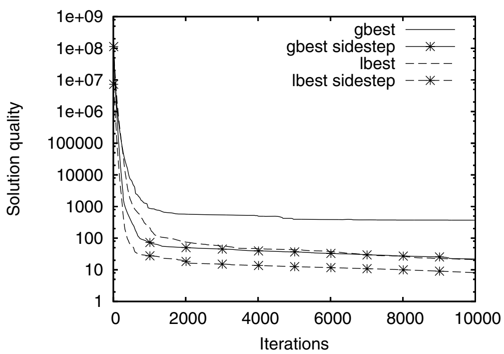


Fig. 8. Solution quality of standard PSO (lbest/gbest) and sidestep PSO (lbest/gbest) for the Rosenbrock function.

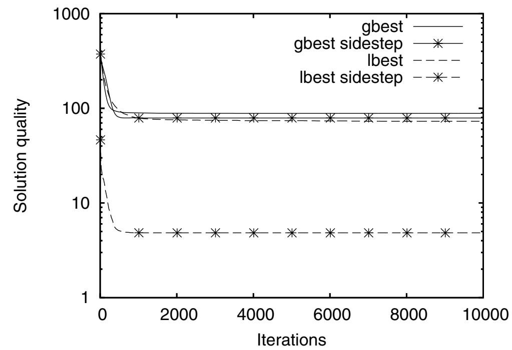


Fig. 9. Solution quality of standard PSO (lbest/gbest) and sidestep PSO (lbest/gbest) for the Rastrigin function.

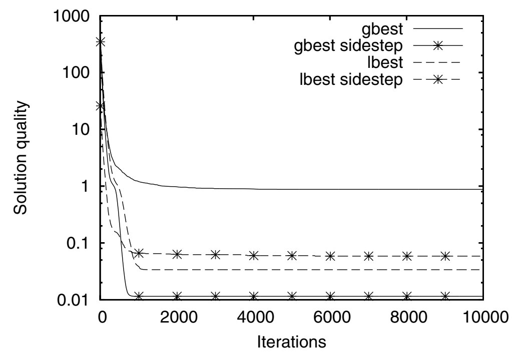


Fig. 10. Solution quality of standard PSO (lbest/gbest) and sidestep PSO (lbest/gbest) for the Griewank function.

have relevance for their optimization performance on standard benchmark functions. The test functions are shown in Table I together with their domain $[X_{min}; X_{max}]^n$. All test functions have dimension $n = 30$. For the tests, parameter values $w = 0.729$ and $c_1 = c_2 = 1.494$ [7] have been used. The number of particles in each run was $m = 10$. The gbest and the lbest neighbourhoods were used, both for the standard PSO and the sidestep PSO.

The results are shown in Figures 7–11. For each test function the sidestep PSO obtains better results than the standard PSO. For all test functions the gbest PSO with sidestep mechanisms is better than the gbest standard PSO. Similarly, for all but one case the lbest version of the sidestep PSO with is better than the lbest standard PSO (only for the Griewank function the lbest standard PSO is slightly better than the sidestep PSO).

## V. Conclusion

In this paper we have formulated two potential weaknesses of the movement behaviour of particles in standard PSO algorithms. It was shown that particles have a distinct bias in their movement direction which is influenced by the direction of the coordinate axes. An unwanted effect is that the optimization behaviour of PSO is not invariant to rotations of the optimization function. A second problem of standard PSO algorithms is that non-oscillatory trajectories can quickly cause a particle to stagnate. A so called sidestep mechanism was proposed which tries to detect and correct the stagnation condition, without adding any further randomness into the method. When a certain stagnation along one dimension is observed for a particles movement then the particle performs a sidestep with respect to this dimension. A PSO where particles perform these sidesteps has been called sidestep PSO. It was shown for simple test functions that the movement behaviour of the sidestep PSO can prevent the bias to a significant degree and also the optimization behaviour is less dependent on rotations of the objective functions. It was also shown for standard benchmark functions that the sidestep PSO outperforms the standard PSO (no matter whether the global best or local best neighbourhood was used for velocity update). An interesting research question is to find other improved methods that prevent the observed deficiencies. As a possible enhancement the particles trajectories might be monitored along the different and it could be ensured that oscillations remain present, e.g., by enforcing sinusoidal trajectories.

## References

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

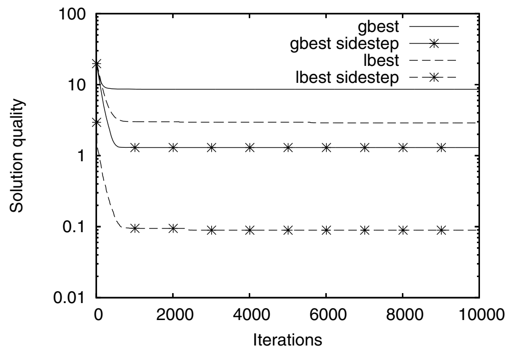


Fig. 11. Solution quality of standard PSO (lbest/gbest) and sidestep PSO (lbest/gbest) for the Ackley function.


[2] Swagatam Das, Amit Konar, and Uday K. Chakraborty. Improving particle swarm optimization with differentially perturbed velocity. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 177–184, New York, NY, USA, 2005. ACM Press.

[3] Jonathan E. Fieldsend and Sameer Singh. A multi-objective algorithm based upon particle swarm optimisation. In *Proceedings of the 2002 U.K. Workshop on Computational Intelligence*, pages 37–44, Birmingham, UK, 2002.

[4] James Kennedy. Dynamic-probabilistic particle swarms. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 201–207, New York, NY, USA, 2005. ACM Press.

[5] James Kennedy and Russell Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks (ICNN'95)*, volume 4, pages 1942–1947, Perth, Western Australia, November-December 1995. IEEE.

[6] Ender Ozcan and Chilukuri K. Mohan. Particle swarm optimization: surfing the waves. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3, 6-9 July 1999.

[7] Ioan C. Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *IPL: Information Processing Letters*, 85:317–325, 2003.

[8] Frans van den Bergh and Andries P. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971, April 2006.

[9] D. N. Wilke. Analysis of the particle swarm optimization algorithm. Master's thesis, University of Pretoria, Pretoria, South Africa, February 2005.

[10] Wen-Jun Zhang and Xiao-Feng Xie. DEPSO: hybrid particle swarm with differential evolution operator. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2003*, pages 3816–3821, 2003.