DE GRUYTER OPEN

JAISCR, 2014, Vol. 4, No. 3, pp. 189 – 204
10.1515/jaiscr-2015-0008

REPULSIVE SELF-ADAPTIVE ACCELERATION PARTICLE SWARM OPTIMIZATION APPROACH

Simone A. Ludwig

Department of Computer Science, North Dakota State University, Fargo, ND, USA

Abstract

Adaptive Particle Swarm Optimization (PSO) variants have become popular in recent years. The main idea of these adaptive PSO variants is that they adaptively change their search behavior during the optimization process based on information gathered during the run. Adaptive PSO variants have shown to be able to solve a wide range of difficult optimization problems efficiently and effectively. In this paper we propose a Repulsive Self-adaptive Acceleration PSO (RSAPSO) variant that adaptively optimizes the velocity weights of every particle at every iteration. The velocity weights include the acceleration constants as well as the inertia weight that are responsible for the balance between exploration and exploitation. Our proposed RSAPSO variant optimizes the velocity weights that are then used to search for the optimal solution of the problem (e.g., benchmark function). We compare RSAPSO to four known adaptive PSO variants (decreasing weight PSO, time-varying acceleration coefficients PSO, guaranteed convergence PSO, and attractive and repulsive PSO) on twenty benchmark problems. The results show that RSAPSO achives better results compared to the known PSO variants on difficult optimization problems that require large numbers of function evaluations.

1 Introduction

Particle Swarm Optimization (PSO) is one of the swarm intelligence methods [1]. The behavior of PSO is inspired by bird swarms searching for optimal food sources, where the direction in which a bird moves is influenced by its current movement, the best food source it ever experienced, and the best food source any bird in the swarm ever experienced. As for PSO, the movement of a particle is influenced by its inertia, its personal best position, and the global best position of the swarm.

PSO has several particles, and every particle maintains its current objective value, its position, its velocity, its *personal best value*, that is the best objective value the particle ever experienced, and its *personal best position*, that is the position at which the personal best value has been found. In addition, PSO maintains a *global best value*, that is the best objective value any particle has ever experienced, and a *global best position*, that is the position at which the global best value has been found. Basic PSO [1] uses the following equation to move the particles:

$$x^{(i)}(n+1) = x^{(i)}(n) + v^{(i)}(n+1), \quad n = 0,1,2,\ldots,-1, \quad (1a)$$

where $x^{(i)}$ is the position of particle i, n is the iteration number with $n = 0$ referring to the initialization, is the total number of iterations, and $v^{(i)}$ is the velocity of particle i, $i = 1,2,\ldots,n_p$, where n_p is the number of particles. Basic PSO uses the following equation to update the particle velocities:

$$v^{(i)}(n+1) = wv^{(i)}(n) + c_1 r_1^{(i)}(n)[x_p^{(i)}(n) - x^{(i)}(n)] + c_2 r_2^{(i)}(n)[x_g(n) - x^{(i)}(n)], \quad n = 0,1,2,\ldots,-1, \quad (1b)$$

where $x_p^{(i)}(n)$ is the personal best position of particle i, and $x_g^{(i)}(n)$ is the global best position of particle i, w is the inertia weight and is set to 1, and the acceleration constants are c_1 and c_2. Both $r_1^{(i)}$ and $r_2^{(i)}$ are vectors with components having random values uniformly distributed between 0 and 1. The notation $r^{(i)}(n)$ denotes that a new random vector is generated for every particle i at iteration n.

PSO can focus on either population/particle diversity or convergence of particles at any iteration. Diversity favors particles that are searching a large area coarsely, whereas convergence favors particles that are searching a small area intensively. A promising strategy is to promote diversity of the swarm in early iterations and convergence in later iterations [2, 3], or assigning attributes to individual particles to promote diversity or convergence [4].

Despite PSO's simplicity, the success of PSO largely depends on the selection of optimal values for the control parameters w, c_1, and c_2. Non-optimal values of the control parameters lead to suboptimal solutions, premature convergence, stagnation, or divergent or cyclic behaviour [5, 6]. However, the optimal setting of the control parameters is dependent on the problem and might be different for the particles within the swarm. Since finding the optimal control parameters manually is very time-consuming, therefore, related work has addressed this with PSO variants that adaptively change all or some of the control parameters. For example, decreasing weight PSO decreases the inertia weight $w(n)$ linearly over time [2], time-varying acceleration coefficients PSO changes not only the inertia weight but also $c_1(n)$ and $c_2(n)$ over time [2, 3], and guaranteed convergence PSO ensures that the global best particle searches within a dynamically adapted radius [7, 8].

Other variants include the linear reduction of the maximum velocity PSO [9], and non-linear adjusted inertia weight PSO [10]. PSO with dynamic adaption [11] uses an evolutionary speed factor that measures personal best value changes and an aggregation degree that measures the relative position of particles in the objective space to calculate the inertia weight w.

APSO in [12] adapts the inertia weight of every particle based on its objective value, the global best value, and the global worst value. APSO introduced in [13] changes its inertia weight based on swarm diversity to reduce premature convergence and hence to increase overall convergence. The swarm diversity is calculated as a function of positions. Different variations of the self-tuning APSO are discussed in [14, 15, 16].

Self-tuning APSO as described in [15] grants every particle its own personal best weight c_1^i and global best weight c_2^i. Self-tuning APSO initializes the personal best weights c_1^i and the global best weights c_2^i randomly for every particle, and moves the personal and global best weights towards values of particle i that yielded the most updates of the global best position, where the distance of the movement towards the personal best weight c_1^i and the global best weight c_2^i are based on the total number of iterations [14]. In an update of self-tuning APSO, the personal and global best weights are moved in ever smaller steps for increasing numbers of iterations [15].

It has been shown with past research that the adaptation of the velocity weights improve the converges speed of PSO compared to having fixed velocity weights. Therefore, our approach is basically inspired by other PSO variants that assign every particle its own velocity weights [15, 16]. These PSO variants usually adapt the velocity weights of a certain particle that is selected based on a measure of superior performance [16] and adopt these velocity weights for all other particles. This paper is an extension of the work published as a short paper in [17]. The organization of this paper is as follows: In Section 2, details of the five PSO variants against which we compare RSAPSO is given. Section 3 introduces and describes the proposed RSAPSO variant. In Section 4, the benchmark problems used to compare the variants with RSAPSO are outline. Section 5 lists the conclusions reached from this study.

2 Related Work and PSO Variants used for Comparison

We are interested in finding the global minimum of an objective function $f(x)$ in a D-dimensional search space of the form $[x_{\min}, x_{\max}]^D$. In order to assess the performance of RSAPSO, we utilize four related adaptive PSO variants: decreasing weight

PSO, time-varying acceleration coefficients PSO, guaranteed convergence PSO, and attractive and repulsive PSO.

2.1 Decreasing Weight PSO (DWPSO)

DWPSO is similar to basic PSO, but the inertia weight $w(n)$ is decreased linearly over time [2]. Thus, DWPSO promotes diversity in early iterations and convergence in late iterations. DWPSO uses Equation 1b to determine the velocities of the particles whereby the inertia weight $w(n)$ is calculated using:

$$w(n) = w_s - (w_s - w_e)\frac{n}{-1}, \tag{2}$$

where w_s is the inertia weight for the first iteration, and w_e is the inertia weight for the last iteration.

2.2 Time-Varying Acceleration Coefficients PSO (TVACPSO)

TVACPSO adapts the acceleration coefficients, i.e., the personal weight $c_1(n)$ and global best weight $c_2(n)$ over time besides the the inertia weight $w(n)$ [2, 3]. The idea is to have high diversity during early iterations and high convergence during late iterations. The inertia weight $w(n)$ is changed as in DWPSO using Equation (2). TVACPSO uses the following equation to determine the velocities:

$$\begin{aligned} v^{(i)}(n+1) = {} & w(n)v^{(i)}(n) \\ & + c_1(n)r_1^{(i)}(n)[x_p^{(i)}(n) - x^{(i)}(n)] \\ & + c_2(n)r_2^{(i)}(n)[x_g(n) - x^{(i)}(n)], \\ & n = 0,1,2,\ldots,-1, \end{aligned} \tag{3a}$$

where the personal best weight $c_1(n)$, and the global best weight $c_2(n)$ at iteration n are calculated using:

$$\begin{aligned} c_1(n) &= c_{1s} - (c_{1s} - c_{1e})\frac{n}{-1}, \\ c_2(n) &= c_{2s} - (c_{2s} - c_{2e})\frac{n}{-1}, \end{aligned} \tag{3b}$$

where c_{1s} is the personal best weight for the first iteration, c_{1e} is the personal best weight for the last iteration, c_{2s} is the global best weight for the first iteration, and c_{2e} is the global best weight for the last iteration.

2.3 Guaranteed Convergence PSO (GCPSO)

GCPSO guarantees that the global best particle searches within a dynamically adapted radius [7, 8]. This addresses the problem of stagnation and increases local convergence by using the global best particle to randomly search within an adaptively changing radius at every iteration [8]. GCPSO, as described in [2], uses the following equation to update the position:

$$\begin{aligned} x^{(i_g)}(n+1) = {} & x_g(n) + w(n)v^{(i_g)}(n) \\ & + \rho(n)(1 - 2r_3(n)), \\ & n = 0,1,2,\ldots,-1, \end{aligned} \tag{4a}$$

GCPSO uses Equation (1b) to determine the velocities $v^{(i)}(n)$. The personal best weight c_1 and the global best weight c_2 are held constant. GCPSO uses the following equation to update the velocity of the global best particle:

$$\begin{aligned} v^{(i_g)}(n+1) = {} & -x^{(i_g)}(n) + x_g(n) + w(n)v^{(i_g)}(n) \\ & + \rho(n)(1 - 2r_3(n)), \\ & n = 0,1,2,\ldots,-1, \end{aligned} \tag{4b}$$

where i_g is the index of the particle that updated the global best value most recently. The expression $-x^{(i_g)}(n) + x_g(n)$ is used to reset the position of particle i_g to the global best position. $r_3(n)$ are random numbers uniformly distributed between 0 and 1. The search radius is controlled by the search radius parameter $\rho(n)$. The search radius parameter $\rho(n)$ is calculated using:

$$\rho(n+1) = \begin{cases} 2\rho(n), & \text{if } \sigma(n+1) > \sigma_c, \\ \frac{1}{2}\rho(n), & \text{if } \varphi(n+1) > \varphi_c, \\ \rho(n), & \text{otherwise,} \end{cases} \tag{4c}$$

where σ_c is the consecutive success threshold, and φ_c is the consecutive failure threshold defined below. Success means that using Equations (1) and (4b) to update the particle positions results in an improved global best value and position, and failure means it does not. The numbers of consecutive successes $\sigma(n)$ and failures $\varphi(n)$ are calculated us-

ing:

$$\sigma(n+1) = \begin{cases} 0, \text{ if } \varphi(n+1) > \varphi(n), \\ \sigma(n)+1, \text{ otherwise,} \end{cases} \quad (4d)$$

$$\varphi(n+1) = \begin{cases} 0, \text{ if } \sigma(n+1) > \sigma(n), \\ \varphi(n)+1, \text{ otherwise.} \end{cases} \quad (4e)$$

2.4 Attractive and Repulsive PSO (RPSO)

RPSO aims to overcome the problem of premature convergence [18]. It uses a diversity measure to control the swarm by alternating between phases of "attraction" and "repulsion". The attraction phase operates as basic PSO by the particles attracting each other (see Equation (1b)). The repulsion phase is done by inverting the velocity-update equation of the particles as follows:

$$\begin{aligned} v^{(i)}(n+1) = w(n)v^{(i)}(n) \\ - c_1 r_1^{(i)}(n)[x_p^{(i)}(n) - x^{(i)}(n)] \\ - c_2 r_2^{(i)}(n)[x_g(n) - x^{(i)}(n)], \\ n = 0,1,2,\ldots,-1, \end{aligned} \quad (5a)$$

In the repulsion phase, the individual particle is no longer attracted to, but instead repelled by the best known particle position and its own previous best position.

In the attraction phase, the swarm is contracting, and therefore the diversity decreases. Once the diversity drops below a lower bound, d_{low}, the repulsion phase is switched to, so that the swarm expands according to Equation (5a). When a diversity of d_{high} is reached, the attraction phase is switched on again. Therefore, there is an alternation between phases of exploitation and exploration (attraction and repulsion).

Equation (5b) sets the sign-variable dir to either 1 or -1 depending on the diversity values as given in Equation (5c):

$$dir = \begin{cases} -1, & \text{if } diversity(S) < d_{low}, \\ 1, & \text{if } diversity(S) > d_{high}, \end{cases} \quad (5b)$$

$$diversity(S) = \frac{1}{|S| \times |L|} \times \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^{N} (p_{ij} - \bar{p}_j)^2}, \quad (5c)$$

where S is the swarm, $|S|$ is the swarm size, $|L|$ is the length of the longest diagonal in the search space, N is the dimensionality of the problem, p_{ij} is the j^{th} value of the i^{th} particle, and $\bar{p}_j$ is the j^{th} value of the average point p. Finally, Equation (5d) is modified by multiplying the sign-variable dir by the social and personal components that decide whether the particles are attracted to, or repelled by each other:

$$\begin{aligned} v^{(i)}(n+1) = w(n)v^{(i)}(n) \\ + dir(c_1 r_1^{(i)}(n)[x_p^{(i)}(n) - x^{(i)}(n)] \\ + c_2 r_2^{(i)}(n)[x_g(n) - x^{(i)}(n)]), \\ n = 0,1,2,\ldots,-1, \end{aligned} \quad (5d)$$

2.5 Dealing with Search Space Violations

If a particle attempts to leave the search space, our strategy is to return it along its proposed path through a series of *correcting iterations*. In particular, we use:

$$\breve{x}^{(i)}(\breve{n}+1) = \breve{x}^{(i)}(\breve{n}) - \breve{v}^{(i)}(\breve{n}+1), \quad \breve{n} = 0,1,\ldots,\breve{}-1, \quad (6a)$$

where $\breve{x}^{(i)}(\breve{n}+1)$ is the corrected position, $\breve{v}^{(i)}$ is the corrected velocity, $\breve{n}$ is the count for the correcting iterations, and$\breve{}$is the total number of correcting iterations. The initial corrected position $\breve{x}^{(i)}(0)$ is set to the position $x^{(i)}(n+1)$, which is outside the search space. The corrected velocities $\breve{v}^{(i)}$ are calculated using:

$$\breve{v}^{(i)}(\breve{n}+1) = \alpha\breve{v}^{(i)}(\breve{n}), \quad \breve{n} = 0,1,\ldots,\breve{}-1, \quad (6b)$$

where α is the correction factor, and the initial corrected velocity $\breve{v}^{(i)}(0)$ is set to the velocity $v^{(i)}(n+1)$ that caused the particle to attempt to leave the search space. Equation (6a) is used until the corrected position $\breve{x}^{(i)}(\breve{n}+1)$ is in the search space or the limit on the total number of correcting iterations$\breve{}$is reached. If$\breve{}$is reached, the components of $\breve{x}^{(i)}(\breve{})$ still outside the search space are clamped to the boundary of the search space. Based on good performance in empirical experiments, the values chosen are $\alpha = 0.54$ and$\breve{} = 4$.

3 Proposed Variant: Repulsive Self-adaptive Acceleration PSO (RSAPSO)

Our RSAPSO variant is inspired by other PSO variants that assign every particle its own velocity weights [15, 16]. These variants typically move the velocity weights of all particles toward the velocity weights of a certain particle that is selected based on a measure of superior performance [16].

For example, self-tuning APSO moves the velocity weights towards the settings of the particle that yielded the most updates of the global best position [15, 16]. Controlled APSO [19] adaptively changes the personal best weights $c_1^{(i)}(n)$ and the global best weights $c_2^{(i)}(n)$ based on the distance between the positions and the global best position. Inertia weight APSO [12] allows every particle its own inertia weight $w^{(i)}(n)$ that is changed using a function of the objective values and the global best value. Optimized PSO [20] uses multiple PSO subswarms, each having their own parameter settings, in an inner iteration to solve the original optimization problem. The parameter settings are then optimized in an outer iteration of PSO for a fixed number of iterations.

Inspired by the optimized PSO variant [20], we treat the problem of finding good velocity weights as an optimization problem. In RSAPSO every particle has its own velocity weights, i.e., its inertia weight $w^{(i)}$, personal best weight $c_1^{(i)}$, and global best weight $c_2^{(i)}$. A particular setting of the velocity weights is referred to as the *position of the velocity weights*. An objective function for the velocity weights is used to quantify how well the positions of the velocity weights perform for solving the overall optimization problem. Using the calculated objective values of the velocity weights, RSAPSO takes a step toward optimizing the velocity weights. The velocity weights are optimized in a fixed auxiliary search space.

Compared to optimized PSO [20], the RSAPSO approach of optimizing the velocity weights after every (outer) PSO iteration is more efficient since only one additional PSO instance (for optimizing the velocity weights) is executed and only for one (inner) iteration. An advantage of RSAPSO is that the velocity weights can adapt themselves to dynamic changes, e.g., different particle distributions at different iterations.

RSAPSO uses the following equation, with the notation used in Equation (1b), to update the velocities of particles:

$$\begin{aligned} v^{(i)}(n+1) = {} & w^{(i)}(n)v^{(i)}(n) \\ & + c_1^{(i)}(n)r_1^{(i)}(n)[x_p^{(i)}(n) - x^{(i)}(n)] \\ & + c_2^{(i)}(n)r_2^{(i)}(n)[x_g(n) - x^{(i)}(n)], \\ & n = 0,1,2,\ldots,-1, \end{aligned} \quad (7a)$$

An auxiliary objective function is used to quantify the success of particles as a function of their velocity weights. There are reliable and directly employable entities to measure the success of particles. In particular, we use the improvement in the objective value of the particle [21], the number of updates of the global best position that the particle yielded [15, 16], and the number of updates of the personal best position that the particle yielded. We propose the following objective function for the velocity weights, selected based on good performance in empirical experiments:

$$\tilde{f}^{(i)}(n) = e^{(i)}(n)\big(1 + w_l u_l^{(i)}(n) + w_g u_g^{(i)}(n)\big), \quad n = 1,2,\ldots,-1, \quad (7b)$$

where $\tilde{f}^{(i)}(n)$ is the objective value of the velocity weights for particle i at iteration n, $e^{(i)}(n)$ is the normalized improvement described below, $u_l^{(i)}$ is the number of times particle i updated its personal best position, $u_g^{(i)}$ is the number of times particle i updated the global best position, w_l is the local weight factor used to weigh the number of personal best updates $u_l^{(i)}$, and w_g is the global weight factor used to weigh the number of global best updates $u_g^{(i)}$. The value of w_g is usually set to a larger number than the value of w_l because updates to the global best position are relatively more important. Equation (7b) is thus used to guide the evolution of the positions of the velocity weights towards optimal values. Alternative objective functions are possible, e.g., ones that use the normalized improvements $e^{(i)}(n)$, or the local and global best update counters individually. The normalized improvements $e^{(i)}(n)$ are calculated as follows, based on good performance in empirical experiments:

$$e^{(i)}(n) = \frac{\delta^{(i)}(n)}{\sigma(n)}, \quad (7c)$$

where $\sigma(n)$ is the normalization sum (which has to be greater than zero), and $\delta^{(i)}(n)$ is the difference in the objective values calculated using:

$$\delta^{(i)}(n) = f^{(i)}(n) - f^{(i)}(n-1), \tag{7d}$$

where $f^{(i)}$ is the objective value of particle i.

In practice, early iterations might yield large absolute values of $\delta^{(i)}$, whereas late iterations might only yield small absolute values of $\delta^{(i)}$. Therefore, we propose the following normalization to fairly account for the contribution of the velocity weights from late iterations:

$$\sigma(n) = \begin{cases} \sum_{i=1}^{n_p} -\delta^{(i)}(n), \text{ for } \delta^{(i)}(n) < 0, \\ 1, \text{ otherwise.} \end{cases} \tag{7e}$$

In other words, the normalization sum $\sigma(n)$ makes objective values of the velocity weights comparable for different n. This normalization is chosen based on good performance in empirical experiments.

The velocity weights are optimized using one step of PSO in an inner iteration, resulting in the following overall iteration to update the positions of the velocity weights:

$$\tilde{x}^{(i)}(n+1) = \tilde{x}^{(i)}(n) + \tilde{v}^{(i)}(n+1), \quad n = 1,2,\ldots,-1, \tag{7f}$$

$$\begin{aligned} \tilde{v}^{(i)}(n+1) = {} & \tilde{w}(n)\tilde{v}^{(i)}(n) \\ & + \tilde{c}_1(n)\tilde{r}_1^{(i)}(n)[\tilde{x}_p^{(i)}(n) - \tilde{x}^{(i)}(n)] \\ & + \tilde{c}_2(n)\tilde{r}_2^{(i)}(n)[\tilde{x}_g^{(i)}(n) - \tilde{x}^{(i)}(n)], \\ & n = 1,2,\ldots,-1, \end{aligned} \tag{7g}$$

where $\tilde{x}^{(i)}(n)$ is the position of the velocity weights, $\tilde{v}^{(i)}(n)$ is the velocity of the velocity weights, $\tilde{x}_p^{(i)}(n)$ is the personal best position of the velocity weights, $\tilde{x}_g^{(i)}(n)$ is the global best position of the velocity weights, $\tilde{w}(n)$ is the inertia weight for optimizing the velocity weights, $\tilde{c}_1(n)$ is the personal best weight for optimizing the velocity weights, $\tilde{c}_2(n)$ is the global best weight for optimizing the velocity weights, and $\tilde{r}_1^{(i)}(n)$ and $\tilde{r}_2^{(i)}(n)$ are random vectors with components that are uniformly distributed between 0 and 1 for every particle i and iteration n.

Equations (7f) and (7g) are used after Equation (1a) and (7a) have been used to update the positions of the particles, and the new objective values have been calculated. The first component of $\tilde{x}^{(i)}(n)$ is used as the inertia weight $w^{(i)}(n)$, the second component of $\tilde{x}^{(i)}(n)$ is used as the personal best weight $c_1^{(i)}(n)$, and the third component of $\tilde{x}^{(i)}(n)$ is used as the global best weight $c_2^{(i)}(n)$ as given in Equation (7a).

The proposed RSAPSO switches between phases based on the mean separation of particles. If RSAPSO is in the attractive phase and converges, it switches to the repulsive phase once it has reached a small enough mean separation value. This can counter the trapping in a local optimum. If RSAPSO is in the repulsive phase, it switches to the attractive phase once it has reached a large enough mean separation. Similarly, four-state APSO uses the mean separation to decide in which of the four states it is in as given in [22]. The attractive-repulsive PSO [18] switches between phases based on a calculated diversity factor that is calculated similarly to the mean separation.

We propose the following objective function for the velocity weights that adapt itself to the current phase:

$$\bar{f}^{(i)}(n) = \begin{cases} \tilde{f}^{(i)}(n), & \text{if } a(n) = 1, \\ -s^{(i)}(n), & \text{if } a(n) = 2, \end{cases} \tag{7h}$$

where $\bar{f}^{(i)}(n)$ is the objective value of the velocity weights, and $a(n)$ is the phase indicator. If RSAPSO is in the attractive phase $a(n) = 1$, the objective value of the velocity weights $\bar{f}^{(i)}(n)$ is set to $\tilde{f}^{(i)}(n)$ as calculated in Equation (7b). If RSAPSO is in the repulsive phase $a(n) = 2$, the objective value of the velocity weights $\bar{f}^{(i)}(n)$ is set to the negation of the mean separation $s^{(i)}(n)$. This objective function for the velocity weights was selected for RSAPSO since good performance of the velocity weights is indicated by $\tilde{f}^{(i)}(n)$ in the attractive phase, and $-s^{(i)}(n)$ in the repulsive phase. In particular, in the attractive phase we focus on convergence by rewarding good objective values of the velocity weights $\tilde{f}^{(i)}(n)$, and in the repulsive phase we focus on diversity by rewarding high mean separation values $s^{(i)}(n)$.

The attractive-repulsive PSO [18] switches to the repulsive phase if its diversity factor goes below an absolute lower threshold value and switches to the attractive phase if its diversity factor goes above

an absolute upper threshold value. We use the same mechanism but replace the diversity factor with the mean separation. Specifically, we use the following equation to switch between the phases:

$$a(n+1) = \begin{cases} 1, & \text{if } a(n) = 2 \wedge s(n) > s_u(n), \\ 2, & \text{if } a(n) = 1 \wedge s(n) < s_l(n), \\ a(n), & \text{otherwise,} \end{cases} \quad (7i)$$

where $s_l(n)$ is the mean separation absolute lower threshold, and $s_u(n)$ is the mean separation absolute upper threshold.
RSAPSO starts in the attractive phase $a(n) = 1$. If the mean separation $s(n)$ falls below the mean separation absolute lower threshold $s_l(n)$, RSAPSO changes from the attractive phase $a(n) = 1$ to the repulsive phase $a(n+1) = 2$. If the mean separation $s(n)$ rises above the mean separation absolute upper threshold $s_u(n)$, RSAPSO changes from the repulsive phase $a(n) = 2$ to the attractive phase $a(n+1) = 1$.

To the best of our knowledge, the adaptive change of the mean separation absolute lower $s_l(n)$ and upper threshold $s_u(n)$ is novel. This concept allows for increased accuracy and convergence as the algorithm proceeds. Furthermore, it can be used if good values for the mean separation absolute lower and the mean separation absolute upper threshold are not known. The mean separation absolute lower and upper threshold, $s_l(n)$ and $s_u(n)$ respectively, are adapted as follows:

$$s_l(n+1) = \begin{cases} \frac{s_l(n)}{\check{s}_l}, & \text{if } a(n) = 2 \wedge s(n) > s_u(n), \\ s_l(n), & \text{otherwise,} \end{cases} \quad (7j)$$

$$s_u(n+1) = \begin{cases} \frac{s_u(n)}{\check{s}_u}, & \text{if } a(n) = 2 \wedge s(n) > s_u(n), \\ s_u(n), & \text{otherwise,} \end{cases} \quad (7k)$$

where $\check{s}_l$ is the mean separation absolute lower divisor and $\check{s}_u$ is the mean separation absolute upper divisor. The mean separation absolute lower threshold $s_l(n)$ is divided by the mean separation absolute lower divisor $\check{s}_l$, and the mean separation absolute upper threshold $s_u(n)$ is divided by the mean separation absolute upper divisor $\check{s}_u$ if the algorithm switches from the repulsive phase to the attractive phase at iteration n. Both the mean separation absolute lower $s_l(n+1)$ and upper threshold $s_u(n+1)$ remain the same if the algorithm does not switch from the repulsive phase to the attractive phase; i.e., the mean separation absolute lower threshold $s_l(n+1)$ and upper threshold $s_u(n+1)$ are only changed after a full cycle through the attractive and repulsive states.

Algorithm 1 outlines our RSAPSO variant. RSAPSO calculates the mean separation after the local and global best positions are updated. RSAPSO requires the mean separation to decide whether a phase switch is required. If so, the objective function for the velocity weights, and the search space for the velocity weights are switched to their counterparts in the new phase. The search space for the velocity weights in the attractive phase must mainly yield positive velocity weights. The search space for the velocity weights in the repulsive phase must mainly yield negative velocity weights. All velocity weights have to be reinitialized in the new search space for the velocity weights if a phase switch occurs. The personal best positions and values of the velocity weights and the global best position and value of the velocity weights are reset since if their values were discovered in the attractive phase, they cannot be used in the repulsive phase and vice versa. In case a switch from the repulsive to the attractive phase occurs, i.e., one phase cycle is finished, the mean separation absolute lower and upper threshold are updated using Equations (7j) and (7k). If no phase switch occurs, RSAPSO follows the flow of optimizing the velocity weights; however, it uses Equation (7h) instead of Equation (7b) as the objective function for the velocity weights.

Algorithm 1 Description of RSAPSO

```
initialize positions and velocities
initialize positions of velocity weights
calculate objective values
update local and global best positions and values
repeat
    update positions
    calculate objective values
    update local and global best positions and values
    calculate mean distance
    if phase changed then
        update mean absolute lower/upper threshold if necessary
        switch objective function/search space for velocity weights
        reinitialize positions of velocity weights
        reset local/global best positions/values of velocity weights
    else
        calculate objective values of velocity weights
        update local/global best positions/values of velocity weights
        update positions of velocity weights
    end if
until maximum number of generations reached
report final results
```

4 Experiments and Results

4.1 Benchmark Problems

Twenty optimization benchmark problems are used to compare our RSAPSO algorithm with the chosen PSO variants. All the benchmark problems from the semi-continuous challenge [23] are used, including the Ackley, Alpine, Griewank, Parabola, Rosenbrock, and Tripod test problems. Some of the optimization problems from [24] have been selected based on their shapes to guarantee a diverse set of problems, including the Six-hump Camel Back, De Jong 5, Drop Wave, Easom, Goldstein–Price, Axis Parallel Hyper-ellipsoid, Michalewicz, and Shubert test problems [23]. We also use optimization problems from [25] to expand our benchmark set. These include the Generalized Penalized, Non-continuous Rastrigin, Sphere, Rastrigin, and Step test problems [25]. In addition, Schaffer's F6 test problem from [20] is used. For ease of comparison, we normalized the benchmark problems in order for all to have a global optimum of 0.0.

Table 1 lists the benchmark functions, their properties, bounds on x, and the search space dimensions.

4.2 Parameter Settings

The parameters are set to the values described as follows:

- search space for velocity weights: $[-0.5, 2.0]$;
- search space for personal best weights: $[-1.0, 4.2]$;
- search space for global best weights: $[-1.0, 4.2]$;
- $w_l = 1$;
- $w_g = 6$;
- $w_s = 0.9$;
- $w_e = 0.4$;
- $c_{1s} = 2.5$;
- $c_{1e} = 0.5$;
- $c_{2s} = 0.5$;
- $c_{2e} = 2.5$;
- percentage of particles selected for mutation of their velocity weights: 33;
- iterations before resetting best positions and velocity weights: 50;
- initialization space for inertia weights: $[0.4, 0.9]$;
- initialization space for personal best weights: $[0.5, 2.5]$;
- initialization space for global best weights: $[0.5, 2.5]$;
- reinitialization space for
- inertia weights: $[0.5, 0.8]$;
- reinitialization space for personal best weights: $[0.6, 2.4]$;
- reinitialization space for global best weights: $[0.6, 2.4]$;
- $\tilde{\alpha} = 0.5$;
- $\breve{m} = 10$;
- $\breve{m}_u = 2.5$.

4.3 Experimental Setup

We compare the PSO variants on four experiments using four different numbers of function evaluations (FE) including initialization.

- The first experiment uses $n_p = 10$ particles and $= 100$ iterations resulting in a total of $1,000$ FE.
- The second experiment uses $n_p = 20$ particles and $= 500$ iterations resulting in a total of $10,000$ FE.
- The third experiment uses $n_p = 40$ particles and $= 2,500$ iterations resulting in a total of $100,000$ FE.
- The fourth experiment uses $n_p = 100$ particles and $= 10,000$ iterations resulting in a total of $1,000,000$ FE.

If the FE are the dominant expense, all the variants considered require approximately the same CPU time for a given number of FE. All calculations are performed in double precision. The results reported are best, mean and standard deviation from 30 runs performed.

Table 1. Description of Test Problems.

Function	Name	$[\mathbf{x}_{min}, \mathbf{x}_{max}]$	D
F1	Ackley	[-30,30]	30
F2	Alpine	[-10,10]	10
F3	Six-hump Camel Back	[-2,2]	2
F4	De Jong 5	[-65.536,65.536]	2
F5	Drop Wave	[-5.12,5.12]	2
F6	Easom	[-100,100]	2
F7	Generalized Penalized	[-50,50]	30
F8	Griewank	[-300,300]	30
F9	Goldstein–Price	[-2,2]	2
F10	Axis Parallel Hyper-ellipsoid	[-5.12,5.12]	100
F11	Michalewicz	$[0, \pi]$	10
F12	Non-continous Rastrigin	[-5.12,5.12]	30
F13	Parabola	[-20,20]	200
F14	Rastrigin	[-10,10]	30
F15	Rosenbrock	[-10,10]	30
F16	Schaffer's F6	[-100,100]	2
F17	Shubert	[-10,10]	2
F18	Sphere	[-100,100]	100
F19	Step	[-100,100]	30
F20	Tripod	[-100,100]	2

4.4 Results

Analyzing the results, as shown in Tables 2 to 5 (best mean values are given in bold), reveal that RSAPSO improves with increasing numbers of FE, scoring best compared to the other variants for 100,000 and 1,000,000 FE. Figure 1 shows the results counting the number of wins, i.e., number of times an algorithm scored best in terms of best mean value on the benchmark functions.

For 1,000 FE (Table 2), DWPSO, GCPSO, and RSAPSO score best on 2 benchmark functions, RPSO scores best on only 1 benchmark function, and TVACPSO outperforms the other algorithms scoring best on 12 benchmark functions.

For 10,000 FE (Table 3), TVACPSO and RPSO score best on 5 benchmark functions, DWPSO and GCPSO score best on 7 benchmark functions, and RSAPSO scores best on 8 benchmark functions.

For 100,000 FE, as shown in Table 4, reveal that TVACPSO and RPSO score best on 7 benchmark functions, DWPSO, and GCPSO score best on 8 benchmark functions, and RSAPSO scores best on 10 benchmark functions.

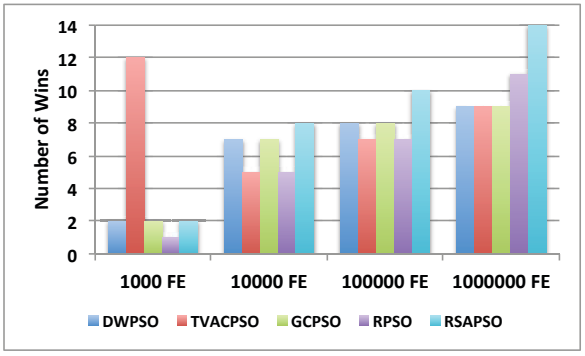
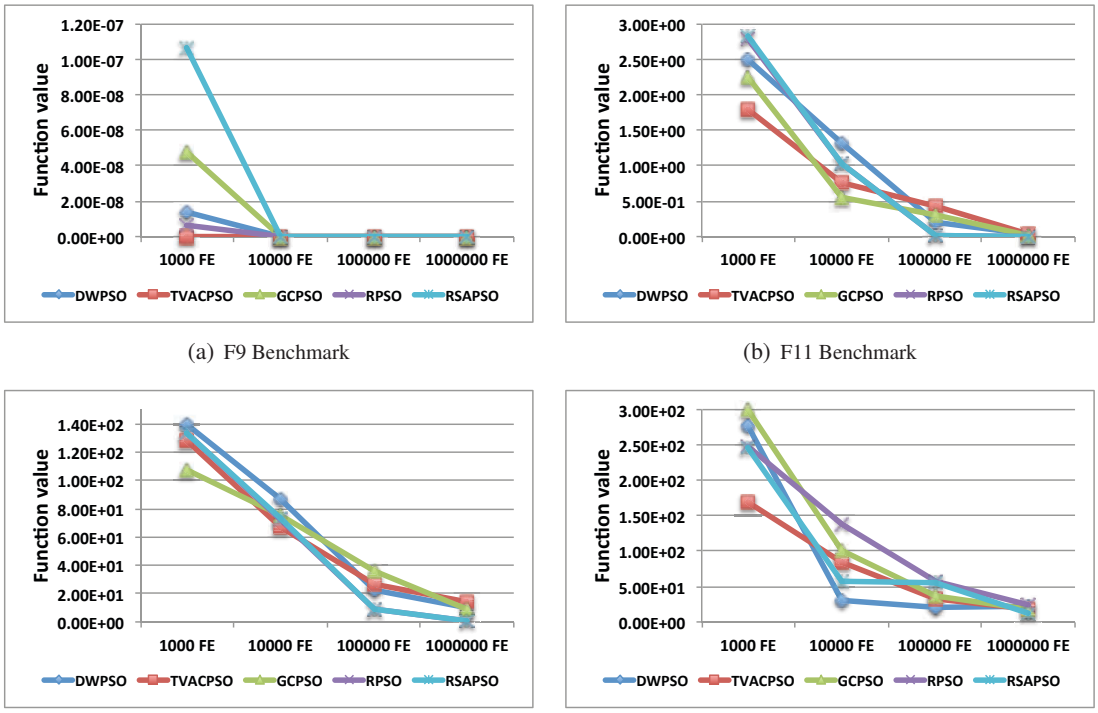


Figure 1. Number of wins versus number of function evaluations

For 1,000,000 FE (Table 5) DWPSO, TVACPSO and GCPSO have the best mean value for 9 benchmark functions, RPSO for 11 benchmark functions, and RSAPSO scores best on 14 benchmark functions.

For 1,000,000 FE, the optimum value of 0.0 was achieved by all PSO variants, measuring the best value on 7 benchmark functions, for 100,000 FE only on 6 benchmark functions, and for 10,000 FE only on 2 benchmark functions. This demonstrates



(a) F9 Benchmark

(b) F11 Benchmark

(c) F12 Benchmark

(d) F14 Benchmark

Figure 2. Function value versus FE for different benchmark functions.

that with increasing numbers of FE more benchmark functions are solved optimally.

In terms of the average values achieved, for 1,000,000 FE, 0.0 was achieved by the PSO variants 41 times, for 100,000 FE 28 times, and for 10,000 FE 16 times.

A Friedman ranking test [26] was applied on the average results for the four different FE. Table 6 shows the average ranks obtained by each PSO variant. All the results for all four different FE are not statistically significant at the 5% significance level. The post hoc procedures of Bonferroni-Dunn and Hochberg confirmed this. However, as the previous discussion outlined, our approach has the best rank for 1,000,000 FE, even though the results are not statistically significant.

Figure 2 shows the function value for 1,000, 10,000, 100,000 and 1,000,000 FE for benchmark functions F9, F11, F12, and F14. It confirms once more that our proposed RSAPSO first performs poorly for 1,000 FE, however, showing improved values with increasing numbers of FE by scoring best for 100,000 and 1,000,000 FE.

Overall, the experiments have shown that for increasing FE our proposed RSAPSO variant scored better than the other PSO variants. Looking at the particular benchmark functions that are multimodal, which are F4, F5, F7, F12, F16, and F17, RSAPSO as expected scored best on these functions with the exception of F17. As mentioned in literature [18], RPSO has shown to work particularly well on multimodal functions, which is most likely due to the switching between attractive and repulsive phases. This allows the algorithm to adopt good velocity values and together with the repulsive and attractive phase it helps to move the particle towards better solutions. The results on the benchmark functions confirms this by the implemented RPSO as well as our proposed RSAPSO variant showing the better results.

5 Conclusion

We proposed a repulsive and adaptive PSO variant, named RSAPSO, for which every particle has its own velocity weights, i.e., inertia weight, personal best weight and global best weight. An objective function for the velocity weights is used to measure the suitability of the velocity weights for solving the overall optimization problem. Due to the calculated objective values of the velocity weights, RSAPSO is able to improve the optimization process. In particular, the RSAPSO variant adapts the velocity weights before it optimizes the solution of the problem (e.g., benchmark function). The advan-

Table 2. Performance for benchmark F1 to F20 for 1,000 FE.

Algorithm		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DWPSO	Best	4.94e+00	2.09e-01	1.12e-12	1.76e-07	1.51e-05	6.00e-05	4.95e+04	4.52e+00	2.56e-10	4.82e+03
	Mean	5.98e+00	5.50e-01	4.71e-10	1.66e+00	2.13e-02	**1.45e-01**	1.00e+06	**5.74e+00**	1.41e-08	5.33e+03
	Std	1.26e+00	2.36e-01	6.45e-19	2.30e+00	1.35e-03	6.29e-02	2.04e+12	1.26e+00	5.16e-16	5.82e+05
TVACPSO	Best	5.56e+00	1.29e-02	2.13e-14	9.94e-01	6.37e-12	2.51e-04	6.10e+02	2.99e+00	2.44e-14	3.31e+03
	Mean	6.08e+00	**2.49e-01**	**2.00e-13**	3.29e+00	**2.13e-02**	3.27e-01	**9.59e+03**	6.98e+00	**1.44e-11**	**4.00e+03**
	Std	3.11e-01	4.49e-02	3.86e-26	6.11e+00	1.35e-03	3.18e-01	9.84e+07	1.24e+01	5.35e-22	7.06e+05
GCPSO	Best	4.51e+00	3.55e-01	3.55e-13	5.88e-10	7.92e-06	1.79e-01	1.02e+02	5.29e+00	7.51e-10	2.92e+03
	Mean	5.50e+00	5.28e-01	6.88e-10	3.47e-01	4.25e-02	7.26e-01	1.39e+04	8.00e+00	4.76e-08	4.13e+03
	Std	7.32e-01	7.70e-02	1.18e-18	3.62e-01	1.35e-03	2.25e-01	3.17e+08	2.02e+01	2.93e-15	1.29e+06
RPSO	Best	5.06e+00	8.06e-02	0.00e+00	3.99e-13	6.38e-02	5.70e-09	2.33e+03	5.93e+00	7.46e-14	4.98e+03
	Mean	**5.35e+00**	7.97e-01	4.62e-09	3.22e-01	6.38e-02	3.33e-01	1.82e+05	9.29e+00	6.72e-09	6.02e+03
	Std	6.66e-02	3.95e-01	6.41e-17	3.26e-01	1.89e-25	3.33e-01	7.08e+10	1.03e+01	1.35e-16	1.42e+06
RSAPSO	Best	5.72e+00	8.06e-02	6.56e-09	3.94e-13	6.38e-02	5.70e-09	2.33e+03	5.93e+00	6.47e-09	4.98e+03
	Mean	6.13e+00	7.97e-01	1.28e-08	**3.31e-01**	6.38e-02	3.33e-01	1.82e+05	9.29e+00	1.07e-07	6.02e+03
	Std	2.37e-01	3.95e-01	3.33e-17	3.29e-01	4.04e-21	3.33e-01	7.08e+10	1.03e+01	1.26e-12	1.42e+06
Algorithm		**F11**	**F12**	**F13**	**F14**	**F15**	**F16**	**F17**	**F18**	**F19**	**F20**
DWPSO	Best	1.92e+00	1.23e+02	5.15e+03	2.52e+02	6.10e+03	9.36e-03	1.80e-07	4.76e+04	1.82e+03	3.37e-03
	Mean	2.50e+00	1.40e+02	5.77e+03	2.77e+02	1.03e+04	9.60e-03	2.55e-04	4.91e+04	3.39e+03	6.93e-01
	Std	3.52e-01	8.93e+02	7.34e+05	8.67e+02	1.32e+07	4.16e-08	1.89e-07	3.10e+06	4.50e+06	1.28e+00
TVACPSO	Best	1.21e+00	1.11e+02	4.17e+03	1.23e+02	1.47e+03	9.72e-03	1.27e-07	3.17e+04	1.10e+03	3.64e-06
	Mean	**1.80e+00**	1.28e+02	**4.73e+03**	**1.69e+02**	**3.22e+03**	9.72e-03	1.14e-06	**3.78e+04**	**1.21e+03**	1.33e+00
	Std	3.49e-01	2.29e+02	4.53e+05	2.96e+03	2.86e+06	3.06e-19	3.07e-12	9.71e+07	8.86e+03	1.33e+00
GCPSO	Best	1.77e+00	8.00e+01	5.13e+03	2.29e+02	5.75e+03	9.72e-03	1.12e-08	4.22e+04	1.27e+03	1.27e-04
	Mean	2.25e+00	**1.08e+02**	5.67e+03	2.99e+02	9.23e+03	9.72e-03	**1.10e-07**	5.06e+04	3.09e+03	1.00e+00
	Std	2.67e-01	8.45e+02	2.51e+05	7.73e+03	3.29e+07	5.94e-27	1.92e-14	5.75e+07	2.57e+06	1.00e+00
RPSO	Best	2.17e+00	1.12e+02	5.61e+03	2.29e+02	2.80e+03	3.15e-03	2.84e-13	3.84e+04	2.24e+03	1.46e-08
	Mean	2.80e+00	1.33e+02	6.28e+03	2.49e+02	5.67e+03	5.34e-03	1.36e-07	4.62e+04	3.38e+03	3.33e-01
	Std	4.36e-01	5.33e+02	3.57e+05	2.28e+02	1.25e+07	1.48e-05	5.59e-14	4.87e+07	1.43e+06	3.33e-01
	Best	2.18e+00	1.12e+02	5.63e+03	2.28e+02	2.81e+03	3.05e-03	4.09e-07	3.81e+04	2.24e+03	9.67e-08

Table 3. Performance for benchmark F1 to F20 for 10,000 FE.

Algorithm		**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
DWPSO	Best	1.17e-01	2.32e-07	0.00e+00	2.22e-16	0.00e+00	0.00e+00	1.21e+00	4.85e-01	7.77e-14	4.97e+02
	Mean	**4.65e-01**	1.59e-06	**0.00e+00**	**2.22e-16**	**0.00e+00**	**0.00e+00**	**1.93e+00**	7.01e-01	7.70e-14	6.81e+02
	Std	1.02e-01	5.53e-12	0.00e+00	9.12e-64	0.00e+00	0.00e+00	6.24e-01	3.75e-02	1.64e-30	2.86e+04
TVACPSO	Best	3.31e-02	2.36e-10	0.00e+00	2.22e-16	0.00e+00	0.00e+00	1.78e+00	1.13e-02	7.82e-14	4.42e+02
	Mean	9.63e-01	2.43e-07	**0.00e+00**	**2.22e-16**	2.13e-02	**0.00e+00**	3.79e+00	**2.73e-02**	7.49e-14	5.41e+02
	Std	6.57e-01	1.11e-13	0.00e+00	9.12e-64	1.35e-03	0.00e+00	4.89e+00	2.07e-04	8.15e-30	9.61e+03
GCPSO	Best	1.82e-01	3.91e-08	0.00e+00	2.22e-16	0.00e+00	0.00e+00	1.06e+00	2.53e-01	7.77e-14	1.34e+02
	Mean	8.44e-01	6.70e-04	**0.00e+00**	**2.22e-16**	**0.00e+00**	**0.00e+00**	1.97e+00	5.45e-01	7.61e-14	**2.58e+02**
	Std	3.78e-01	1.34e-06	0.00e+00	9.12e-64	0.00e+00	0.00e+00	2.04e+00	6.40e-02	4.40e-30	1.27e+04
RPSO	Best	2.42e+00	1.11e-05	0.00e+00	2.22e-16	0.00e+00	0.00e+00	2.60e+00	1.65e-01	7.68e-14	9.86e+02
	Mean	3.15e+00	2.14e-05	**0.00e+00**	**2.22e-16**	**0.00e+00**	**0.00e+00**	5.08e+00	4.86e-01	7.37e-14	1.09e+03
	Std	4.51e-01	2.37e-10	0.00e+00	9.12e-64	0.00e+00	0.00e+00	1.09e+01	3.09e-01	1.99e-29	1.40e+04
RSAPSO	Best	2.58e+00	1.11e-05	0.00e+00	2.22e-16	0.00e+00	0.00e+00	3.31e+00	3.68e-01	7.64e-14	9.86e+02
	Mean	3.20e+00	2.14e-05	**0.00e+00**	**2.22e-16**	**0.00e+00**	**0.00e+00**	6.11e+00	7.37e-01	**7.30e-14**	1.09e+03
	Std	3.40e-01	2.37e-10	0.00e+00	9.12e-56	0.00e+00	0.00e+00	7.61e+00	1.44e-01	1.98e-29	1.40e+04

Algorithm		**F11**	**F12**	**F13**	**F14**	**F15**	**F16**	**F17**	**F18**	**F19**	**F20**
DWPSO	Best	9.04e-01	5.07e+01	1.75e+03	1.11e+01	1.11e+02	5.04e-08	2.84e-14	6.02e+03	2.00e+00	0.00e+00
	Mean	1.33e+00	8.72e+01	2.18e+03	2.97e+01	1.36e+02	5.63e-03	4.74e-14	8.47e+03	**3.00e+00**	3.33e-01
	Std	2.79e-01	9.99e+02	1.68e+05	8.43e+02	9.29e+02	2.54e-05	2.69e-28	4.71e+06	1.00e+00	3.33e-01
TVACPSO	Best	2.64e-01	5.60e+01	1.40e+03	7.01e+01	6.58e+01	0.00e+00	2.84e-14	3.51e+03	9.00e+00	0.00e+00
	Mean	7.61e-01	**6.73e+01**	1.63e+03	8.38e+01	1.15e+02	6.48e-03	5.68e-14	6.56e+03	2.33e+01	3.33e-01
	Std	6.97e-01	1.21e+02	4.26e+04	1.62e+02	1.86e+03	3.15e-05	8.08e-28	7.60e+06	2.26e+02	3.33e-01
GCPSO	Best	2.57e-01	5.65e+01	1.43e+03	7.39e+01	9.61e+01	0.00e+00	5.68e-14	6.46e+03	7.00e+00	0.00e+00
	Mean	**5.62e-01**	7.55e+01	**1.52e+03**	1.02e+02	1.12e+02	3.24e-03	6.63e-14	1.02e+04	1.10e+01	3.33e-01
	Std	8.42e-02	3.56e+02	8.85e+03	1.53e+03	7.14e+02	3.15e-05	2.69e-28	1.06e+07	1.30e+01	3.33e-01
RPSO	Best	4.50e-01	4.51e+01	2.08e+03	1.24e+02	1.65e+02	5.04e-08	5.68e-14	3.22+03	1.90e+02	0.00e+00
	Mean	1.03e+00	7.18e+01	2.73e+03	1.38e+02	2.34e+02	5.63e-03	7.58e-14	**3.80e+03**	3.11e+02	8.59e-11
	Std	4.32e-01	5.34e+02	3.44e+05	4.54e+02	5.40e+03	2.54e-05	2.69e-28	2.96e+05	1.10e+04	2.21e-20
RSAPSO	Best	4.50e-01	4.51e+01	2.08e+03	4.22e+01	7.72e+01	0.00e+00	5.68e-14	6.46e+03	2.90e+01	0.00e+00
	Mean	1.03e+00	7.28e+01	2.47e+03	**5.75e+01**	2.05e+02	**0.00e+00**	7.58e-14	1.02e+04	2.00e+02	**0.00e+00**
	Std	4.32e-01	5.77e+02	1.70e+05	1.76e+02	1.40e+04	0.00e+00	2.69e-28	1.06e+07	3.12e+04	0.00e+00

Table 4. Performance for benchmark F1 to F20 for 100,000 FE.

Algorithm		**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
DWPSO	Best	5.68e-10	2.22e-16	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.81e-18	8.79e-14	7.82e-14	8.89e-02
	Mean	1.19e-09	3.52e-16	**0.00e+00**	1.48e-16	**0.00e+00**	**0.00e+00**	1.60e-17	1.39e-02	7.80e-14	4.12e-01
	Std	1.01e-18	5.03e-32	0.00e+00	1.64e-32	0.00e+00	0.00e+00	3.18e-34	1.83e-04	6.57e-32	1.16e-01
TVACPSO	Best	4.24e-11	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	6.99e-15	7.82e-14	2.77e-01
	Mean	**5.29e-11**	**0.00e+00**	**0.00e+00**	7.40e-17	**0.00e+00**	**0.00e+00**	5.67e-20	**8.21e-03**	7.82e-14	6.25e-01
	Std	1.50e-22	0.00e+00	0.00e+00	1.64e-32	0.00e+00	0.00e+00	6.63e-39	7.47e-05	2.39e-58	9.09e-02
GCPSO	Best	5.87e-11	3.89e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00	6.00e-20	9.86e-03	7.95e-14	4.77e-02
	Mean	1.83e-10	4.16e-15	**0.00e+00**	1.48e-16	**0.00e+00**	**0.00e+00**	4.80e-19	2.46e-02	7.86e-14	6.54e-02
	Std	3.68e-20	9.55e-32	0.00e+00	1.64e-32	0.00e+00	0.00e+00	2.69e-37	1.88e-04	5.92e-31	3.62e-04
RPSO	Best	1.18e-09	3.22e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00	2.06e-17	2.22e-16	7.84e-14	4.33e+00
	Mean	8.59e-01	5.03e-14	**0.00e+00**	7.40e-17	**0.00e+00**	**0.00e+00**	3.46e-02	1.07e-02	**7.80e-14**	4.47e+01
	Std	6.81e-01	6.33e-27	0.00e+00	1.64e-32	0.00e+00	0.00e+00	3.58e-03	8.70e-05	7.57e-32	4.29e+03
RSAPSO	Best	1.18e-09	3.22e-15	0.00e+00	2.22e-16	0.00e+00	0.00e+00	2.06e-17	2.22e-16	7.82e-14	2.04e-01
	Mean	9.76e-02	5.03e-14	**0.00e+00**	2.22e-16	**0.00e+00**	**0.00e+00**	3.46e-02	1.07e-02	**7.80e-14**	4.67e+00
	Std	2.86e-02	6.33e-27	0.00e+00	9.12e-64	0.00e+00	0.00e+00	3.58e-03	8.70e-05	6.57e-32	2.16e+01

Algorithm		**F11**	**F12**	**F13**	**F14**	**F15**	**F16**	**F17**	**F18**	**F19**	**F20**
DWPSO	Best	4.91e-03	1.20e+01	4.29e+01	1.69e+01	2.37e+01	0.00e+00	2.84e-14	2.83e+00	0.00e+00	0.00e+00
	Mean	1.96e-01	2.30e+01	1.05e+02	**2.06e+01**	4.36e+01	**0.00e+00**	**2.84e-14**	3.57e+00	**0.00e+00**	**0.00e+00**
	Std	2.98e-02	2.73e+02	4.78e+03	1.22e+01	8.93e+02	0.00e+00	0.00e+00	1.28e+00	0.00e+00	0.00e+00
TVACPSO	Best	2.90e-01	2.20e+01	9.24e+01	2.69e+01	9.38e+00	0.00e+00	2.84e-14	3.36e+00	0.00e+00	0.00e+00
	Mean	4.22e-01	2.70e+01	1.25e+02	3.32e+01	3.50e+01	**0.00e+00**	5.68e-14	2.93e+01	3.33e-01	3.33e-01
	Std	2.78e-02	2.50e+01	9.89e+02	4.19e+01	1.21e+03	0.00e+00	8.08e-28	5.19e+02	3.33e-01	3.33e-01
GCPSO	Best	4.67e-02	3.05e+01	7.28e+01	3.38e+01	2.39e+01	0.00e+00	2.84e-14	2.02e+00	0.00e+00	0.00e+00
	Mean	3.11e-01	3.58e+01	**7.99e+01**	3.65e+01	2.52e+01	**0.00e+00**	**2.84e-14**	4.36e+00	**0.00e+00**	**0.00e+00**
	Std	8.86e-02	2.17e+01	1.28e+02	9.24e+00	2.48e+00	0.00e+00	0.00e+00	8.60e+00	0.00e+00	0.00e+00
RPSO	Best	1.78e-15	5.06e+00	2.38e+02	3.78e+01	2.32e+01	0.00e+00	2.84e-14	5.68e+00	0.00e+00	0.00e+00
	Mean	1.39e-02	**9.02e+00**	2.73e+02	5.77e+01	2.32e+01	**0.00e+00**	5.68e-14	2.13e+02	3.33e+00	**0.00e+00**
	Std	5.87e-04	1.28e+01	1.25e+03	3.45e+02	3.16e-04	0.00e+00	8.08e-28	1.29e+01	1.23e+01	0.00e+00
RSAPSO	Best	0.00e+00	5.06e+00	5.46e+01	3.28e+01	2.32e+01	0.00e+00	2.84e-14	1.89e-02	0.00e+00	0.00e+00
	Mean	**1.33e-02**	**9.02e+00**	1.14e+02	5.60e+01	2.32e+01	**0.00e+00**	5.68e-14	**4.78e-01**	**0.00e+00**	**0.00e+00**
	Std	5.29e-04	1.28e+01	6.74e+03	4.53e+02	3.16e-04	0.00e+00	8.08e-28	1.79e-01	0.00e+00	0.00e+00

Table 5. Performance for benchmark F1 to F20 for 1,000,000 FE.

Algorithm		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DWPSO	Best	7.55e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	7.40e-03	7.95e-14	9.94e-15
	Mean	7.55e-15	9.99e-16	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	1.07e-02	7.86e-14	6.26e-14
	Std	0.00e+00	3.00e-30	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	8.09e-06	5.92e-31	7.75e-27
TVACPSO	Best	7.55e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	7.82e-14	7.59e-10
	Mean	**9.92e-15**	**0.00e+00**	**0.00e+00**	7.40e-17	**0.00e+00**	**0.00e+00**	**0.00e+00**	6.56e-03	7.82e-14	1.18e-08
	Std	1.68e-29	0.00e+00	0.00e+00	1.64e-32	0.00e+00	0.00e+00	0.00e+00	1.29e-04	2.39e-58	2.00e-16
GCPSO	Best	7.55e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	8.04e-14	2.76e-16
	Mean	7.55e-15	1.30e-15	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**4.93e-03**	7.89e-14	4.63e-16
	Std	0.00e+00	5.03e-30	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	7.28e-05	1.64e-30	6.43e-32
RPSO	Best	7.55e-15	2.22e-16	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	8.04e-14	2.40e-19
	Mean	9.92e-15	2.61e-15	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	6.57e-03	7.89e-14	2.32e-17
	Std	1.68e-29	6.60e-30	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.85e-05	1.64e-30	1.32e-33
RSAPSO	Best	7.55e-15	2.22e-16	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	8.04e-14	2.40e-19
	Mean	9.92e-15	2.61e-15	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	6.57e-03	**7.61e-14**	2.32e-17
	Std	1.68e-29	6.60e-30	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.85e-05	1.64e-30	1.32e-33

Algorithm		F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
DWPSO	Best	1.78e-15	7.00e+00	8.55e-04	4.97e+00	1.40e+01	0.00e+00	0.00e+00	1.05e-13	0.00e+00	0.00e+00
	Mean	2.81e-02	1.00e+01	1.65e-03	2.12e+01	1.80e+01	**0.00e+00**	**9.47e-15**	2.13e-12	**0.00e+00**	**0.00e+00**
	Std	6.13e-04	1.30e+01	4.82e-07	2.40e+01	1.19e+01	0.00e+00	2.69e-28	8.66e-24	0.00e+00	0.00e+00
TVACPSO	Best	4.91e-03	8.00e+00	2.27e-02	1.19e+01	6.14e+00	0.00e+00	0.00e+00	5.54e-08	0.00e+00	0.00e+00
	Mean	3.62e-02	1.37e+01	5.52e+00	1.82e+01	1.04e+01	**0.00e+00**	4.74e-14	1.63e-07	**0.00e+00**	**0.00e+00**
	Std	8.37e-04	2.43e+01	8.88e+01	5.97e+01	3.40e+01	0.00e+00	1.88e-27	1.77e-14	0.00e+00	0.00e+00
GCPSO	Best	0.00e+00	7.00e+00	1.01e-04	1.09e+01	8.70e+00	0.00e+00	5.68e-14	7.64e-16	0.00e+00	0.00e+00
	Mean	1.72e-02	9.00e+00	2.86e-04	1.76e+01	1.64e+01	**0.00e+00**	5.68e-14	2.79e-15	**0.00e+00**	**0.00e+00**
	Std	6.58e-04	3.00e+00	2.57e-08	3.70e+01	4.41e+01	0.00e+00	0.00e+00	1.02e-29	0.00e+00	0.00e+00
RPSO	Best	0.00e+00	0.00e+00	8.56e-07	4.97e+00	1.73e-03	0.00e+00	0.00e+00	4.25e-18	0.00e+00	0.00e+00
	Mean	5.71e-06	**3.33e-01**	**5.36e-05**	2.12e+01	**1.36e+00**	**0.00e+00**	3.79e-14	8.31e-16	**0.00e+00**	**0.00e+00**
	Std	9.78e-11	3.33e-01	8.13e-09	2.40e+01	5.25e+00	0.00e+00	1.88e-27	1.08e-30	0.00e+00	0.00e+00
RSAPSO	Best	0.00e+00	0.00e+00	4.42e-04	1.09e+01	1.73e-03	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Mean	**1.18e-15**	**3.33e-01**	7.03e+00	**1.23e+01**	**1.36e+00**	**0.00e+00**	3.79e-14	**0.00e+00**	**0.00e+00**	**0.00e+00**
	Std	1.05e-30	3.33e-01	1.48e-02	2.31e+00	5.25e+00	0.00e+00	1.88e-27	0.00e+00	0.00e+00	0.00e+00

Table 6. Average rankings of the algorithms (Friedman)

Algorithm	1,000 FE	10,000 FE	100,000 FE	1,000,000 FE
DWPSO	3.275	2.85	2.65	3.275
TVACPSO	2	2.7	3.15	3.475
GCPSO	2.975	2.7	2.75	2.8
RPSO	3.15	3.375	3.55	2.8
RSAPSO	3.6	3.375	2.9	2.65

tage of RSAPSO is that the velocity weights adapt themselves to dynamic changes, e.g., different particle distributions at different iterations.

We evaluated our RSAPSO algorithm on twenty benchmark functions and compared it with four PSO variants, namely decreasing weight PSO, time-varying acceleration coefficient PSO, guaranteed convergence PSO, and attractive and repulsive PSO. Our RSAPSO variant achieves better results than the other variants for higher numbers of FE in particular for 1,000,000 FE. A possible reason for RSAPSO's poorer performance for 1,000 and 10,000 FE is that the optimization of the velocity weights takes several iterations to have a beneficial effect since more knowledge of the optimization problem is acquired by then. In addition, RSAPSO has shown to work particularly well on multimodal functions due to the incorporated attractive and repulsive phases for the optimization of the velocity weights.

Since RSAPSO has longer running times depending on the difficulty and the dimensionality of the problem, future work will parallelize the algorithm using Hadoop's MapReduce methodology in order to speed up the optimization process. Furthermore, we would like to extend RSAPSO to integrate the idea of moving bound behavior that would allow expert knowledge about the search space for the velocity weights to be incorporated.

References

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

[2] A. Engelbrecht, *Computational Intelligence: An Introduction, 2nd Edition*, Wiley, 2007.

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

[4] X. Li, H. Fu, and C. Zhang, A self-adaptive particle swarm optimization algorithm, *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, 186–189, 2008.

[5] I. C. Trelea, The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection, *Information Processing Letters*, 85(6), 317-325, 2003.

[6] F. Van den Bergh and A. P. Engelbrecht, A Study of Particle Swarm Optimization Particle Trajectories, *Information Sciences*, 176(8), 937-971, 2006.

[7] C. K. Monson, K. D. Seppi, Exposing Origin-Seeking Bias in PSO, *Proceedings of GECCO'05*, pp. 241-248, 2005.

[8] F. Van den Bergh and A. P. Engelbrecht, A new locally convergent particle swarm optimiser, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 3:94–99, 2002.

[9] F. Schutte, A. A. Groenwold, A Study of Global Optimization using Particle Swarms, *Journal of Global Optimization*, 31, 93-108, 2005.

[10] A. Chatterjee, P. Siarry, Nonlinear inertial weight variation for dynamic adaptation in particle swarm optimization, *Journal of Computer Operation Research*, 33(3), 859-871, 2004.

[11] X. Yang, J. Yuan, J. Yuan, and H. Mao, A modified particle swarm optimizer with dynamic adaptation, *Applied Mathematics and Computation*, 189(2):1205–1213, 2007.

[12] J. Zhu, J. Zhao, and X. Li, A new adaptive particle swarm optimization algorithm, *International Workshop on Modelling, Simulation and Optimization*, 456–458, 2008.

[13] Y. Bo, Z. Ding-Xue, and L. Rui-Quan, A modified particle swarm optimization algorithm with dynamic adaptive, *2007 Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2:346–349, 2007.

[14] T. Yamaguchi and K. Yasuda, Adaptive particle swarm optimization; self-coordinating mechanism with updating information, *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*, 3:2303–2308, 2006.

[15] T. Yamaguchi, N. Iwasaki, and K. Yasuda, Adaptive particle swarm optimization using information about global best, *IEEE Transactions on Electronics, Information and Systems*, 126:270–276, 2006.

[16] K. Yasuda, K. Yazawa, and M. Motoki, Particle swarm optimization with parameter self-adjusting mechanism, *IEEE Transactions on Electrical and Electronic Engineering*, 5(2):256–257, 2010.

[17] S. A. Ludwig, Towards A Repulsive and Adaptive Particle Swarm Optimization Algorithm, Proceedings of Genetic and Evolutionary Computation Conference (ACM GECCO), Amsterdam, Netherlands, July 2013 (short paper).

[18] J. Riget and J. S. Vesterstrø, A diversity-guided particle swarm optimizer — The ARPSO, *EVALife Technical Report no. 2002-2*, 2002.

[19] A. Ide and K. Yasuda, A basic study of adaptive particle swarm optimization, *Denki Gakkai Ronbunshi, Electrical Engineering in Japan*, 151(3):41–49, 2005.

[20] M. Meissner, M. Schmuker, and G. Schneider, Optimized particle swarm optimization (OPSO) and its application to artificial neural network training, *BMC Bioinformatics*, 7(1):125, 2006.

[21] X. Cai, Z. Cui, J. Zeng, and Y. Tan, Self-learning particle swarm optimization based on environmental feedback, *Innovative Computing, Information and Control, 2007. ICICIC '07*, page 570, 2007.

[22] Z. H. Zhan and J. Zhang, *Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence*, pages 227–234, 2008.

[23] M. Clerc, Semi-continuous challenge, http://clerc.maurice.free.fr/pso/semi-continuous_challenge/, April 2004.

[24] M. Molga and C. Smutnicki, Test functions for optimization needs, http://zsd.iiar.pwr.wroc.pl/, 2005.

[25] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, Adaptive particle swarm optimization, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1362–1381, 2009.

[26] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association, vol. 32, no. 200, pp. 675–701, 1937.