Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients

Asanga Ratnaweera, Saman K. Halgamuge, *Member, IEEE*, and Harry C. Watson

***Abstract*—This paper introduces a novel parameter automation strategy for the particle swarm algorithm and two further extensions to improve its performance after a predefined number of generations. Initially, to efficiently control the local search and convergence to the global optimum solution, time-varying acceleration coefficients (TVAC) are introduced in addition to the time-varying inertia weight factor in particle swarm optimization (PSO). From the basis of TVAC, two new strategies are discussed to improve the performance of the PSO. First, the concept of "mutation" is introduced to the particle swarm optimization along with TVAC (MPSO-TVAC), by adding a small perturbation to a randomly selected modulus of the velocity vector of a random particle by predefined probability. Second, we introduce a novel particle swarm concept "self-organizing hierarchical particle swarm optimizer with TVAC (HPSO-TVAC)." Under this method, only the "social" part and the "cognitive" part of the particle swarm strategy are considered to estimate the new velocity of each particle and particles are reinitialized whenever they are stagnated in the search space. In addition, to overcome the difficulties of selecting an appropriate mutation step size for different problems, a time-varying mutation step size was introduced. Further, for most of the benchmarks, mutation probability is found to be insensitive to the performance of MPSO-TVAC method. On the other hand, the effect of reinitialization velocity on the performance of HPSO-TVAC method is also observed. Time-varying reinitialization step size is found to be an efficient parameter optimization strategy for HPSO-TVAC method. The HPSO-TVAC strategy outperformed all the methods considered in this investigation for most of the functions. Furthermore, it has also been observed that both the MPSO and HPSO strategies perform poorly when the acceleration coefficients are fixed at two.**

***Index Terms*—Acceleration coefficients, hierarchical particle swarm, mutation, particle swarm, reinitialization.**

I. INTRODUCTION

PARTICLE swarm optimization (PSO) is a population-based, self-adaptive search optimization technique first introduced by Kennedy and Eberhart [1] in 1995. The motivation for the development of this method was based on the simulation of simplified animal social behaviors such as fish schooling, bird flocking, etc.

Similar to other population-based optimization methods such as genetic algorithms, the particle swarm algorithm starts with the random initialization of a population of individuals (particles) in the search space [30]. However, unlike in other evolutionary optimization methods, in PSO there is no direct recombination of genetic material between individuals during the search. The PSO algorithm works on the social behavior of particles in the swarm. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual toward its own best location and toward the best particle of the entire swarm at each time step (generation) [1], [9], [10]. The PSO method is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution.

In the particle swarm algorithm, the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each particle, according to its own flying experience and the flying experience of the other particles in the search space. The position vector and the velocity vector of the ith particle in the d-dimensional search space can be represented as $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{\mathrm{id}})$ and $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{\mathrm{id}})$ respectively. According to a user defined fitness function, let us say the best position of each particle (which corresponds to the best fitness value obtained by that particle at time t) is $\mathrm{P}_i = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{\mathrm{id}})$, and the fittest particle found so far at time t is $\mathrm{P}_g = (p_{g1}, p_{g2}, \ldots, p_{\mathrm{gd}})$. Then, the new velocities and the positions of the particles for the next fitness evaluation are calculated using the following two equations:

$$\begin{aligned} v_{\mathrm{id}} &= v_{\mathrm{id}} + c_1 \times \mathrm{rand}(\cdot) \times (p_{\mathrm{id}} - x_{\mathrm{id}}) \\ &\quad + c_2 \times \mathrm{Rand}(\cdot) \times (p_{\mathrm{gd}} - x_{\mathrm{id}}) \qquad (1) \\ x_{\mathrm{id}} &= x_{\mathrm{id}} + v_{\mathrm{id}} \qquad (2) \end{aligned}$$

where c_1 and c_2 are constants known as acceleration coefficients, and $\mathrm{rand}(\cdot)$and $\mathrm{Rand}(\cdot)$are two separately generated uniformly distributed random numbers in the range [0,1].

The first part of (1) represents the previous velocity, which provides the necessary momentum for particles to roam across the search space. The second part, known as the "cognitive" component, represents the personal thinking of each particle. The cognitive component encourages the particles to move toward their own best positions found so far. The third part is known as the "social" component, which represents the collaborative effect of the particles, in finding the global optimal

Manuscript received July 11, 2002; revised August 27, 2003. The work of A. Ratnaweera was supported in part by the Asian Development Bank Project on Science and Technology Personnel Development, under the Ministry of Science and Technology, Sri Lanka, and in part by the Melbourne Research Scholarship Program.

A. Ratnaweera and S. K. Halgamuge are with the Mechatronics Research Group, Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria 3010, Australia (e-mail: asangar@mame.mu.oz.au; sam@mame.mu.oz.au).

H. C. Watson is with the Transport Energy Research Group, Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria 3010, Australia (e-mail: hw@mame.mu.oz.au).

Digital Object Identifier 10.1109/TEVC.2004.826071

solution. The social component always pulls the particles toward the global best particle found so far.

Initially, a population of particles is generated with random positions, and then random velocities are assigned to each particle. The fitness of each particle is then evaluated according to a user defined objective function. At each generation, the velocity of each particle is calculated according to (1) and the position for the next function evaluation is updated according to (2). Each time if a particle finds a better position than the previously found best position, its location is stored in memory. Generally, a maximum velocity ($V\max_d$) for each modulus of the velocity vector of the particles ($\mathbf{v}_{\mathrm{id}}$) is defined in order to control excessive roaming of particles outside the user defined search space. Whenever a v_{id} exceeds the defined limit, its velocity is set to $V\max_d$.

In this paper, we propose a novel parameter automation strategy and two new extensions to the particle swarm concept. The major objective of this development is to improve the performance after a predefined number of generations, through empirical simulations with well-known benchmarks.

Initially, we introduce the concept of time-varying acceleration coefficients (TVAC) c_1 and c_2 in addition to time-varying inertia weight factor [14], [15], to effectively control the global search and convergence to the global best solution. The major consideration of this modification is to avoid premature convergence in the early stages of the search and to enhance convergence to the global optimum solution during the latter stages of the search.

First, in addition to TVAC, we introduce a "mutation" operator to the particle swarm concept (MPSO-TVAC). Under this new development, when the global best value remains constant with increasing generations, a particle is randomly selected with a predefined probability (mutation probability). Then, a random perturbation is given to a randomly selected modulus of the velocity vector of the selected particle.

Second, we introduce a novel PSO concept "self-organizing hierarchical particle swarm optimizer with TVAC (HPSO-TVAC)." Under this method, the previous velocity term in (1) is kept constant at zero. The momentum for the particles to roam through the search space is maintained by reinitializing particles with random velocities, whenever they stagnate in the search space.

Finally, we apply the PSO method to estimate the operating parameters for optimum performance of internal combustion spark ignition engines. In this investigation, the objective function is defined through an engine simulation program, which evaluates the performance in terms of power output for a given set of input conditions and geometrical parameters.

The rest of this paper is organized as follows. In Section II, we summarize two significant previous developments to the original PSO methodology. One method was used as the basis for our novel developments, whereas the other one was selected as comparative measure of performance of the novel methods proposed in this paper. In Section III, we introduce the three new extensions to PSO proposed in this paper. Experimental settings for the benchmarks and simulation strategies are explained in Section IV and the results in comparison with the two previous developments are presented in Section V. In Section VI, we apply particle swarm methods to investigate their ability to find the design parameters for optimum performance of internal combustion engines.

II. Some Previous Work

Since the introduction of the PSO method in 1995, there has been a considerable amount of work done in developing the original version of PSO, through empirical simulations [1]–[8], [11]–[19], [22]–[26]. In this section, we summarize two significant previous developments, which serve as both a basis for and performance gauge of the novel strategies introduced in this paper.

In population-based optimization methods, proper control of global exploration and local exploitation is crucial in finding the optimum solution efficiently [8], [15]. Shi and Eberhart [15] introduced the concept of inertia weight to the original version of PSO, in order to balance the local and global search during the optimization process.

Generally, in population-based search optimization methods, considerably high diversity is necessary during the early part of the search to allow use of the full range of the search space. On the other hand, during the latter part of the search, when the algorithm is converging to the optimal solution, fine-tuning of the solutions is important to find the global optima efficiently.

Considering these concerns, Shi and Eberhart [17] have found a significant improvement in the performance of the PSO method with a linearly varying inertia weight (ω) over the generations. The mathematical representation of this concept is given by (3) and (4)

$$\mathbf{v}_{\mathrm{id}} = \omega \times \mathbf{v}_{\mathrm{id}} + c_1 \times \mathrm{rand}(\cdot) \times (p_{\mathrm{id}} - x_{\mathrm{id}}) + c_2 \times \mathrm{Rand}(\cdot) \times (p_{\mathrm{gd}} - x_{\mathrm{id}}) \quad (3)$$

where ω is given by

$$\omega = (\omega_1 - \omega_2) \times \frac{(\mathrm{MAXITER} - \mathrm{iter})}{\mathrm{MAXITER}} + \omega_2 \quad (4)$$

where ω_1 and ω_2 are the initial and final values of the inertia weight, respectively, iter is the current iteration number and $\mathrm{MAXITER}$ is the maximum number of allowable iterations.

Through empirical studies, Shi and Eberhart [17] have observed that the optimal solution can be improved by varying the value of ω from 0.9 at the beginning of the search to 0.4 at the end of the search for most problems. This modification to the original PSO concept has been considered as the basis for two novel strategies introduced in this paper. Hereafter, in this paper, this version of PSO is referred to as time-varying inertia weight factor method (PSO-TVIW).

Most early developments in PSO have been proven to be effective in optimizing static problems [31], [32], [36], [37]. However, most real-world applications are identified as nonlinear dynamic systems. Eberhart and Shi [31] found that the PSO-TVIW concept is not very effective for tracking dynamic systems. Instead, considering the dynamic nature of real-world applications, they have proposed a random inertia weight factor for tracking dynamic systems.

In this development, the inertia weight factor (ω) is set to change randomly according to the following equation:

$$\omega = 0.5 + \frac{\mathrm{rand}(\cdot)}{2} \tag{5}$$

where $\mathrm{rand}(\cdot)$ is a uniformly distributed random number within the range [0,1]. Therefore, the mean value of the inertia weight is 0.75. This modification was inspired by Clerc's constriction factor concept [10], [11], [34] in which the inertia weight is kept constant at 0.729 and both acceleration coefficients are kept constant at 1.494. Therefore, when random inertia weight factor method is used the acceleration coefficients are kept constant at 1.494. In the remainder of this paper, this method is referred to as PSO-RANDIW.

Through empirical studies with some of the well-known benchmarks, it has been identified that the random inertia weight method shows rapid convergence in the early stages of the optimization process and can find a reasonably good solution for most of the functions. Therefore, comparing the results reported in the literature with the same benchmarks, this method was selected to compare the effectiveness of the novel PSO strategies introduced in this paper. However, since two of the new strategies introduced in this development are based on the TVIW concept, the performance of novel strategies was also compared with the PSO-TVIW method.

III. Proposed New Developments

Even though the PSO-TVIW method is capable of locating a good solution at a significantly faster rate, when compared with other evolutionary optimization methods, its ability to fine tune the optimum solution is comparatively weak, mainly due to the lack of diversity at the end of the search [8]. On the other hand, in PSO, problem-based tuning of parameters is also a key factor to find the optimum solution accurately and efficiently [14]. Considering these concerns, in this paper, we propose three strategic developments to improve the performance of PSO.

A. Time-Varying Acceleration Coefficients (TVAC)

It is clear from (1) that, in PSO, the search toward the optimum solution is guided by the two stochastic acceleration components (the cognitive component and the social component). Therefore, proper control of these two components is very important to find the optimum solution accurately and efficiently.

Kennedy and Eberhart [1] described that a relatively high value of the cognitive component, compared with the social component, will result in excessive wandering of individuals through the search space. In contrast, a relatively high value of the social component may lead particles to rush prematurely toward a local optimum. Moreover, they suggested setting either of the acceleration coefficients at 2, in order to make the mean of both stochastic factors in (1) unity, so that particles would over fly only half the time of search. Since then, this suggestion has been extensively used for most studies.

Suganthan [19] tested a method of linearly decreasing both acceleration coefficients with time, but observed that the fixed acceleration coefficients at 2 generate better solutions. However, through empirical studies he suggested that the acceleration coefficients should not be equal to 2 all the time.

Generally, in population-based optimization methods, it is desirable to encourage the individuals to wander through the entire search space, without clustering around local optima, during the early stages of the optimization. On the other hand, during the latter stages, it is very important to enhance convergence toward the global optima, to find the optimum solution efficiently.

Considering those concerns, in this paper, we propose time-varying acceleration coefficients as a new parameter automation strategy for the PSO concept. The objective of this development is to enhance the global search in the early part of the optimization and to encourage the particles to converge toward the global optima at the end of the search.

Under this new development, we reduce the cognitive component and increase the social component, by changing the acceleration coefficients c_1 and c_2 with time. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the population best. On the other hand, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization. We suggest this method be run with a time-varying inertia weight factor as given in (4). Hereafter, this will be referred to as PSO-TVAC method.

This modification can be mathematically represented as follows:

$$c_1 = (c_{1f} - c_{1i})\frac{\mathrm{iter}}{\mathrm{MAXITR}} + c_{1i} \tag{6}$$

$$c_2 = (c_{2f} - c_{2i})\frac{\mathrm{iter}}{\mathrm{MAXITR}} + c_{2i} \tag{7}$$

where c_{1i}, c_{1f}, c_{2i}, and c_{2f} are constants, iter is the current iteration number and MAXITR is the maximum number of allowable iterations.

Simulations were carried out with numerical benchmarks (all benchmarks are discussed in Section IV), to find out the best ranges of values for c_1 and c_2. Results are presented in Section V. An improved optimum solution for most of the benchmarks was observed when changing c_1 from 2.5 to 0.5 and changing c_2 from 0.5 to 2.5, over the full range of the search. Therefore, these values are used for the rest of the work.

With this modification, a significant improvement of the optimum value and the rate of convergence were observed, particularly for unimodal functions, compared with the PSO-TVIW. However, it has been observed that the performance of the PSO-TVAC method is similar or poor for multimodal functions. In contrast, compared with the PSO-RANDIW method an improved performance has been observed with the PSO-TVAC for multimodal functions. However, for unimodal functions, the PSO-RANDIW method showed significantly quick convergence to a good solution compared with the PSO-TVAC method. The results are presented and discussed in Section V.

B. Particle Swarm Optimizer With "Mutation" and Time-Varying Acceleration Coefficients (MPSO-TVAC)

In PSO, lack of diversity of the population, particularly during the latter stages of the optimization, was understood as the dominant factor for the convergence of particles to local optimum solutions prematurely. Recently, several attempts on improving the diversity of the population have been reported in the literature, considering the behavior of the particles in the swarm during the search [22]–[25]. Further, possible use of the concept of "mutation" in PSO (as explained in genetic algorithms), as a performance enhancing strategy, has also been investigated [38].

In this paper, we introduce "mutation" to the particle swarm strategy (MPSO), to enhance the global search capability of the particles by providing additional diversity. Mutation is widely used in most evolutionary optimization methods, such as evolutionary programming and genetic algorithms, to guide and enhance the search toward the global best solution [8], [13], [31], [35].

In evolutionary programming, a mutation function is defined to control the search toward the global optimum solution. However, different forms of mutation functions are used in evolutionary programming and the severity of mutation is decided on the basis of the functional change imposed on the parents [8], [35]. On the other hand, in genetic algorithms, the search toward the global optimum solution is mostly guided by the crossover operation [31], [35]. However, in genetic algorithms, a mutation operator is defined to introduce new genetic material into the individuals to enhance the search in new areas within the search space.

In PSO, the search toward the global optimum solution is guided by the two stochastic acceleration factors (the cognitive part and the social part), of (1). Therefore, Angeline [8] related these two acceleration factors to the mutation function in evolutionary programming, whereas Shi and Eberhart [13] related these two factors to the crossover operation in genetic algorithms.

It has been observed through simulations with numerical benchmarks that PSO quickly finds a good local solution but it sometimes remains in a local optimum solution for a considerable number of iterations (generations) without an improvement. Therefore, to control this phenomenon, we enhance the global search via the introduction of a mutation operator, which is conceptually equivalent to the mutation in genetic algorithms. Under this new strategy, when the global optimum solution is not improving with the increasing number of generations, a particle is selected randomly and then a random perturbation (mutation step size), is added to a randomly selected modulus of the velocity vector of that particle by a predefined probability (mutation probability). However, in this paper, the mutation step size is set proportionally to the maximum allowable velocity. The pseudocode for the **MPSO** method is as follows.

```
begin
  initialize the population
  while (termination condition= false)
    do
      for (i = 1 to number of particles)
          evaluate the fitness:=f (x)
          update P_id and P_g
              for d = 1 to number of
              dimensions
                calculate new velocity := v_id
                update the position
              increase d
          increase i
          select a random particle := k
          select a random dimension := l
          if (Δglobal <= 0)
            if (rand1(·) < p_m)
            if (rand2(·) < 0.5)
              v_kl = v_kl + rand3(·)*vmax/m;
            else
            v_kl = v_kl − rand4(·)*vmax/m;
              end if
            end if
          end if
    end do
end
```

Where $\mathrm{randi}(\cdot)$, $i = 1, 2, \ldots, 4$ are separately generated, uniformly distributed random numbers in the range [0,1], p_m is the mutation probability, $\Delta\mathrm{global}$ is the rate of improvement of the global solution over the generations and m, k, and l are constants.

The effect of the mutation step size and mutation probability on the optimal solution of the MPSO method along with TVAC (MPSO-TVAC) was observed through empirical simulations. A significant improvement of performance with most of the chosen benchmarks was observed with the MPSO-TVAC method, when compared with the PSO-TVIW method. Further, compared with the PSO-RANDIW method, a significant improvement was observed for the Rastrigrin function with PSO-RANDIW strategy. However, competitive performance was observed with the MPSO-TVAC and the PSO-RANDIW for most of the other functions. Further, when compared with the MPSO-TVAC method, the PSO-RANDIW method showed significantly quick convergence to an optimum solution for unimodal functions. It has also been observed that proper selection of the mutation step size can enhance the performance for some functions. Alternatively, time-varying mutation step size was found to be an effective parameter automation strategy for most of the test functions. However, it has been observed that the performance of the MPSO method with fixed acceleration coefficients at 2 (MPSO-FAC) is significantly poor for most of the benchmarks. The results are presented and discussed in Section V.

C. Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients (HPSO-TVAC)

It has been understood that most of the previous empirical developments of PSO are based on either the inertia weight factor method, with a linear varying inertial weight factor, or the constriction factor method [18]–[20], [22]–[25]. However, Shi and Eberhart [14] suggested that for complex multimodal functions, the control of diversity of the population with a linearly varying inertia weight may lead the particles to converge to a local optimum prematurely. On the other hand, the work done by Eberhart and Shi [12] clearly shows that the constriction factor method is ineffective for complex multimodal functions, despite its ability to converge to stopping criteria at a significantly faster rate for unimodal functions.

By contrast, Kennedy and Eberhart [1] proposed a version of PSO without the velocity of the previous iteration in (1). Later, they concluded that since this version is very simple, it is quite ineffective in finding the global optima for most of the complex problems.

In this paper, we observed the behavior of the particles in the swarm without the previous velocity term in (1). Through simulations with some well-known benchmarks, we observed that in the absence of the previous velocity term, particles rapidly rush to a local optimum solution and stagnate due to the lack of momentum. Indeed, without the previous velocity term in (1), the optimum solution is highly dependent on the population initialization.

Considering these concerns, we introduce the novel concept "self-organizing hierarchical particle swarm optimizer (**HPSO**)" to provide the required momentum for particles to find the global optimum solution, in the absence of the previous velocity term in (1).

In this method, we keep the previous velocity term at zero, and reinitialize the modulus of velocity vector of a particle with a random velocity (reinitialization velocity) should it stagnate ($\mathbf{v}_{\mathrm{id}} = 0$) in the search space. Therefore, in this method, a series of particle swarm optimizers are automatically generated inside the main particle swarm optimizer according to the behavior of the particles in the search space, until the convergence criteria is met. In this paper, we set the reinitialization velocity proportional to the maximum allowable velocity ($V\max$).

The pseudocode for **HPSO** is as follows.

```
begin
  initialize the population
  while (termination condition= false)
    do
      for (i = 1 to number of particles)
          evaluate the fitness:=f (x)
          update P_id and P_gd
        for d = 1 to number of dimensions
          calculate the new velocity
          v_id = c1 * rand1(·) * (p_id − x_id) + c2*
          rand2(·) * (p_gd − x_id)
            if (v_id = 0)
              if (rand3(·) < 0.5)
              v_id = rand4(·) * v
            else
              v_id = −rand5(·) * v
              end if
            end if
            v_id = sign(v_id) * min(abs(v_id, vmax)
            update the position
      increase d
    increase i
  end do
end
```

Where $\mathrm{randi}(\cdot)$, $i = 1, 2, \ldots, 5$ are separately generated uniformly distributed random numbers in the range [0, 1] and v is the reinitialization velocity.

The effect of the reinitialization velocity on the optimal solution of HPSO along with TVAC (HPSO-TVAC) was observed through empirical simulations. To overcome the difficulties of selecting appropriate reinitialization velocities for different problems, a time-varying reinitialization velocity strategy was used. A significant improvement of the performance was observed with the HPSO-TVAC method. In contrast, it has been observed that the performance in terms of the optimum solution is significantly poor for most of the benchmarks, with the HPSO method when the acceleration coefficients are fixed at 2 (HPSO-FAC). The results are presented and discussed in Section V.

TABLE I
BENCHMARKS FOR SIMULATIONS

Name of the function	Mathematical representation
Sphere function	$f_1(x) = \sum_{i=1}^{n} x_i^2$
Rosenbrock function	$f_2(x) = \sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$
Rastrigrin function	$f_3(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$
Griewank function	$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Schaffer's f6 function	$f_6(x) = 0.5 - \frac{\left(\sin\sqrt{x^2+y^2}\right)^2 - 0.5}{\left(1.0 + 0.001\left(x^2+y^2\right)\right)^2}$

TABLE II
INITIALIZATION RANGE AND DYNAMIC RANGE OF THE SEARCH FOR BENCHMARKS

Function	Range of search	Range of initialization
f_1	$(-100, 100)^n$	$(50, 100)^n$
f_2	$(-100, 100)^n$	$(15, 30)^n$
f_3	$(-10, 10)^n$	$(2.56, 5.12)^n$
f_4	$(-600, 600)^n$	$(300, 600)^n$
f_6	$(-100, 100)^2$	$(15, 30)^2$

Where n is the number of dimensions

TABLE III
MAXIMUM VELOCITY FOR BENCHMARKS

Function	Vmax
f_1	100
f_2	100
f_3	10
f_4	600
f6	100

IV. EXPERIMENTAL SETTINGS AND SIMULATION STRATEGIES FOR BENCHMARK TESTING

A. Benchmarks

Five of the well-known benchmarks used in evolutionary optimization methods, were used to evaluate the performance, both in terms of the optimum solution after a predefined number of iterations, and the rate of convergence to the optimum solution, of all the new developments introduced in this paper. These benchmarks are widely used in evaluating performance of PSO methods [5], [7], [12], [15]–[21]. The performance of all new

TABLE IV
CALIBRATION OF ACCELERATION COEFFICIENTS FOR TIME VARYING ACCELERATION COEFFICIENT METHOD

function	Dimension	Gmax	Average optimum value / (Standard deviation) for 50 trials for different ranges of c_1 and c_2				
			$c_1 = 2$ $c_2 = 2$	$c_1 = 2 - 0$ $c_2 = 0 - 2$	$c_1 = 2 – 0.25$ $c_2 = 0.25 - 2$	$c_1 = 2 – 0.5$ $c_2 = 0.5 - 2$	$c_1 = 2 – 0.75$ $c_2 = 0.75 - 2$
f_1	30	2000	0.000 (0.000)	1511.514 (1315.094)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
f_2	30	2000	297.627 (402.585)	14362.517 30082.886	149.604 (148.300)	83.184 (88.041)	58.147 (64.714)
f_3	30	2000	39.426 (9.471)	85.295 (19.511)	58.985 (19.455)	56.295 (13.223)	56.0361 (12.716)
f_4	30	2000	0.0178 (0.0222)	10.804 (7.591)	0.041 (0.048)	0.021 (0.030)	0.016 (0.021)

(a)

function	Dimension	Gmax	Average optimum value / (Standard deviation) for 50 trials for different ranges of c_1 and c_2				
			$c_1 = 2 - 1$ $c_2 = 1 - 2$	$c_1 = 2.25 – 0.5$ $c_2 = 0.5 – 2.25$	$c_1 = 2.5 – 0.5$ $c_2 = 0.5 – 2.5$	$c_1 = 2.75 – 0.5$ $c_2 = 0.5 – 2.75$	$c_1 = 2.5 – 0.75$ $c_2 = 0.75 – 2.5$
f_1	30	2000	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)	0.000 (0.000)
f_2	30	2000	69.126 (96.294)	72.061 (74.796)	68.462 (85.505)	68.708 (68.230)	58.998 (90.981)
f_3	30	2000	57.110 (15.280)	53.489 (13.400)	45.907 (11.454)	49.131 (12.724)	52.036 (11.613)
f_4	30	2000	0.017699 (0.0190)	0.016 (0.015)	0.016 (0.019)	0.015 (0.023)	0.014 (0.017)

(b)

methods is compared with the PSO-TVIW method, as well as the PSO-RANDIW method.

The first two functions are simple unimodal functions whereas the next two functions are multimodal functions designed with a considerable amount of local minima. All functions have the global minimum at the origin or very close to the origin [17]. Simulations were carried out to find the global minimum of each function. All benchmarks used are given in Table I.

B. Population Initialization

During the early stages of the development of the PSO method, symmetric initialization was widely used, where the initial population is uniformly distributed in the entire search space. Later, Angeline [8] introduced the asymmetric initialization method, in which the population is initialized only in a portion of the search space.

Since most of the benchmarks used in this paper have the global minimum at or close to the origin of the search space, we use the asymmetric initialization method to observe the performance of the new developments introduced in this paper. The most common dynamic ranges used in the literature for the benchmarks considered in this paper were used and the same dynamic range is used in all dimensions [11], [21]. Table II shows the range of population initialization and the dynamic range of the search for each function.

It is quite common in PSO methods to limit the maximum velocity of each modulus of the velocity vector of a particle $(\mathbf{v}_{\mathrm{id}})$ to a maximum allowable velocity, in order to limit excessive searching outside the predefined search space. Through empirical studies on numerical benchmarks, Eberhart and Shi [12] suggested that it is good to limit the maximum velocity $V\max$ to the upper value of the dynamic range of search $X\max$. Therefore, this limitation was used for the simulation in this paper. Table III shows the maximum velocity with the limitation of $X\max = V\max$ for the benchmarks considered in this paper.

C. Simulation Strategies

Simulations were carried out to observe the rate of convergence and the quality of the optimum solution of the new methods introduced in this investigation in comparison with both PSO-TVIW and PSO-RANDIW. All benchmarks with the exception of Schaffer's f6 function, which is two-dimensional (2-D), were tested with dimensions 10, 20, and 30. For each function, 50 trials were carried out and the average optimal value and the standard deviation (inside the brackets) are presented. A different number of maximum generations (Gmax) is used according to the complexity of the problem under consideration.

Use of different stopping criteria for different benchmarks is reported in the literature [5], [20]. However, all benchmarks have the global optimum value of 0.00. Therefore, for all of the benchmarks (excluding Schaffer's f6), the stopping criteria are set to 0.01. However, for Schaffer's f6, widely used error limit of 0.000 01 was used for this investigation [5], [20].

D. Population Size

Work done by Eberhart and Shi [12] indicated that the effect of population size on the performance of the PSO method is of minimum significance. However, it is quite common in PSO research to limit the number of particles to the range 20 to 60 [12]–[14]. van den Bergh and Engelbrecht [20] suggested that even though there is a slight improvement of the optimal value with increasing swarm size, it increases the number of function evaluations to converge to an error limit. Therefore, in this paper, all empirical experiments were carried out with a population size of 40.

V. Results From Benchmark Simulations

Initially, we observed the performance in terms of quality of the average optimum value for 50 trials, of each new development in comparison with the PSO-TVIW method, as well as the PSO-RANDIW method. For the HPSO-TVAC method, the effect of the reinitialization velocity on the average optimum solution was also observed. In addition, for the MPSO-TVAC method, the effects of mutation step size and mutation probability on the average optimum solution were investigated. Finally, the rate of convergence of all methods was observed for all the functions in 30 dimensions. The average and the standard deviation (inside brackets) of the optimum solution for 50 trials are presented in Tables IV–IX. In Table X, the number of trials, which converged to the stopping criteria, and the average number of generations for convergence (inside brackets) is presented. In all the tables, figures in bold represent the comparatively best values.

TABLE V
Average Value and the Standard Deviation of the Optimal Value for 50 Trials

Function	Dimen -sion	Gmax	Average (Standard Deviation)		
			PSO-TVIW	PSO-RANDW	PSO-TVAC
f_1	10	1000	**0.01**	**0.01**	**0.01**
	20	2000	**0.01**	**0.01**	**0.01**
	30	3000	**0.01**	**0.01**	**0.01**
f_2	10	3000	27.11 (58.312)	**2.102 (3.218)**	9.946 (32.127)
	20	4000	51.56 (119.79)	28.1788 (73.072)	**17.944 (46.296)**
	30	5000	63.35 (71.210)	35.277 (55.751)	**28.97 (51.638)**
f_3	10	3000	**2.069 (1.152)**	4.63 (2.366)	2.268 (1.333)
	20	4000	**11.74 (3.673)**	26.293 (8.176)	15.323 (5.585)
	30	5000	**29.35 (6.578)**	69.7266 (20.700)	36.236 (8.133)
f_4	10	3000	0.0675 (0.029)	0.0661 (0.030)	**0.05454 (0.025)**
	20	4000	0.0288 (0.023)	**0.0272 (0.025)**	0.0293 (0.027)
	30	5000	**0.0167 (0.013)**	0.0175 (0.018)	0.0191 (0.015)
f_6	2	1000	0.0039 (0.0019)	**0.0029 (0.004)**	0.0039 (0.0019)

A. Time-Varying Acceleration Coefficients (TVAC)

The optimum range for the acceleration coefficients was empirically investigated through benchmark simulations. All the benchmarks, except Schaffer's f6 function, were used in 30 dimensions for this investigation. The maximum number of iterations was set to 2000. For all simulations, the inertia weight factor was set to change from 0.9 to 0.4 over the generations. The results are presented in Table IV.

Note: the initial value and the final value of the acceleration coefficients c_1 and c_2 are presented in the table and Gmax is the maximum number of generations

It has been identified from the results that the best ranges for c_1 and c_2 are 2.5–0.5 and 0.5–2.5, respectively.

The performance of the PSO-TVAC method was then observed in comparison with both the PSO-TVIW and the PSO-RANDIW methods. Results are presented in Table V.

It is clear from the results that all of the methods perform well for the Sphere function and Schaffer's f6 function. However, for the Rosenbrock function, the introduction of TVAC has improved the average optimum solution significantly when compared with the PSO-TVIW strategy, but its performance is competitive with the PSO-RANDIW in all dimensions for most of the other functions. However, for the Rastrigrin function a slight reduction of the quality of the average optimum solution was observed with the PSO-TVAC strategy compared with the PSO-TVIW method, even though it performed significantly well compared with the PSO-RANDIW method. Further, for the Griewank function and Schaffer's f6 function, competitive results were observed with all methods.

In general, using the time-varying acceleration coefficients along with the time-varying inertia weight factor, consistent performance has been observed for all benchmarks considered in this investigation.

B. Particle Swarm Optimizer With "Mutation" and Time-Varying Acceleration Coefficients (MPSO-TVAC)

The effect of mutation probability and mutation step size on the best solution was investigated through empirical simulations. Further, the effect of linearly decreasing mutation step size with increasing generations was also investigated.

The maximum velocity for each function was set to the upper limit of the dynamic range of the search $(V\max = X\max)$ and the mutation step size is presented as a percentage of maximum velocity. In the case of the time-varying mutation step size

TABLE VI
VARIATION OF THE AVERAGE AND THE STANDARD DEVIATION OF THE OPTIMAL VALUE FOR 50 TRIALS WITH MUTATION STEP SIZES FOR MPSO-TVAC METHOD

Function	Dimension	Gmax	Average Optimum value / (Standard deviation) With respect to the mutation step size as a % of Vmax						
			100%	50%	20%	10%	5%	2%	Time varying (100% - 10%)
f_1	10	1000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	20	2000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	30	3000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
f_2	10	3000	8.784 (19.311)	9.80 (20.832)	7.437 (20.077)	8.622 (18.912)	6.3035 (16.59)	**4.75 (17.462)**	12.536 (30.081)
	20	4000	24.678 (49.025)	14.261 (26.473)	21.656 (40.432)	**10.908 (17.229)**	15.026 (32.479)	22.587 (58.138)	18.974 (25.512)
	30	5000	37.623 (42.727)	43.757 (64.956)	30.888 (63.357)	33.965 (57.921)	**27.91 (40.167)**	28.871 (54.056)	31.55 (46.412)
f_3	10	3000	**0.01**	**0.01**	**0.01**	0.9172 (0.847)	2.05026 (1.353)	2.567 (1.363)	0.00669 (0.003)
	20	4000	**0.044 (0.1959)**	0.0436 (0.196)	0.421 (1.378)	8.1785 (3.329)	14.665 (4.644)	14.685 (5.335)	0.361 (0.558)
	30	5000	**0.579 (0.696)**	0.669 (0.808)	2.4092 (2.559)	20.814 (8.097)	33.768 (9.1154)	36.316 (10.524)	1.712 (1.956)
f_4	10	3000	0.050 0.022	**0.0441 (0.022)**	0.0470 (0.020)	0.0446 (0.020)	0.0513 (0.022)	0.055 (0.027)	0.0445 (0.022)
	20	4000	0.0254 (0.187)	0.0238 (0.021)	0.0220 (0.015)	**0.0217 (0.018)**	0.0222 (0.02)	0.0301 (0.023)	0.0239 (0.015)
	30	5000	0.0185 (0.016)	**0.0146 (0.008)**	0.017 (0.014)	0.0165 (0.011)	0.017 (0.011)	0.0149 (0.013)	0.0188 (0.019)
f_6	2	1000	**0.0002 (0.001)**	0.00039 (0.002)	0.00078 (0.002)	0.00058 (0.002)	0.0078 (0.002)	0.0078 (0.002)	**0.0002 (0.001)**

strategy, the mutation step size was set to change from $V\max$ to $0.1\ V\max$ over the search. Table VI displays the effect of mutation step size on the optimum solution. In this investigation, the mutation probability was set to 0.8.

The results indicate that the effect of mutation step size on the optimum solution is insignificant for the sphere function, the Griewank function and Schaffer's f6 function.

Only a slight variation of the optimum value was observed for the Rosenbrock function for different mutation step sizes. However, it is clear from the results that for the Rosenbrock function, it is good to keep the mutation step size at 5%–10% of the maximum velocity. Further, high standard deviation for 50 trials was observed with the MPSO-TVAC strategy for the Rosenbrock function. Therefore, there is a higher chance for premature convergence to a local optimum solution with the MPSO-TVAC method, despite the low average optimum value.

In contrast, the performance for the Rastrigrin function is found to be highly sensitive to the mutation step size. The best performance with the Rastrigrin function was observed when mutation step size was equal to $V\max$.

Therefore, it is good to keep the mutation step size high for the Rastrigrin function. Moreover, for the Rastrigrin function with ten dimensions, the MPSO-TAVC method converged to the stopping criteria (0.01) in all 50 trials when mutation step size is higher that 20% of $V\max$.

In conclusion, it is clear from the results that the average optimum solution can be dependent on the mutation step size, for some problems (f_3). Therefore, to obtain an improved solution, proper selection of the mutation step size may be a key factor for some problems. However, the use of a time-varying mutation step size can be identified as a good strategy to overcome the difficulties of selecting a proper mutation step size.

Table VII shows the effect of the mutation probability on the optimum solution. In this investigation, the mutation step size was set to change linearly from $V\max$ to 10% of the value of $V\max$. A slight variation of the average optimum solution was observed with different mutation probabilities for most of the benchmarks. However, it is clear from the results that it is good to set the mutation probability in the range of 0.8–0.4 for most of the functions.

TABLE VII
VARIATION OF THE AVERAGE AND THE STANDARD DEVIATION OF THE OPTIMAL VALUE FOR 50 TRIALS WITH MUTATION PROBABILITY FOR MPSO-TVAC METHOD

Function	Dimension	Gmax	Average Optimum Solution / (Standard Deviation) With respect to the mutation probability				
			1	0.8	0.6	0.4	0.2
f_1	10	1000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	20	2000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	30	3000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
f_2	10	3000	19.729 (38.67)	12.536 (30.08)	4.727 (13.414)	**4.247 (7.961)**	6.256 (25.74)
	20	4000	18.163 (30.434)	18.974 (25.512)	10.899 (24.415)	17.7148 (60.306)	**10.034 (23.978)**
	30	5000	37.623 (48.492)	31.550 (46.412)	35.555 (52.708)	**18.633 (25.122)**	18.957 (36.234)
f_3	10	3000	**0.01**	0.01 (0.0031)	0.01 (0.003)	0.01 (0.0033)	0.027 (0.1401)
	20	4000	**0.183 (0.4779)**	0.361 (0.558)	0.1827 (0.4337)	0.3415 (0.588)	0.797 (0.875)
	30	5000	**1.534 (1.831)**	1.712 (1.956)	1.990 (2.145)	2.050 1.910	3.621 (3.176)
f_4	10	3000	0.047 (0.0233)	**0.0445 (0.0219)**	0.0519 (0.0197)	0.0469 (0.0256)	0.0517 (0.0245)
	20	4000	0.0276 (0.0252)	**0.0239 (0.0154)**	0.0247 (0.0193)	**0.0239 (0.0172)**	0.0258 (0.019)
	30	5000	0.0164 (0.0146)	0.0188 (0.0199)	0.0199 (0.0202)	0.0169 (0.0149)	**0.0159 (0.0129)**
f_6	2	1000	0.0005 (0.0023)	**0.0002 (0.0013)**	0.0012 (0.00312)	0.0012 (0.00311)	0.0011 (0.00302)

C. Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients (HPSO-TVAC)

The effect of the reinitialization velocity on the average optimum solution for 50 trials was observed. The results are presented in Table VIII.

The reinitialization velocity is presented as a percentage of the maximum velocity ($V\max$). In the case of time-varying reinitialization velocity, the reinitialization velocity was set to decay from $V\max$to 0.1 $V\max$during the search. The maximum velocity is set to the upper limit of the dynamic range of the search $(V\max = X\max)$ for all benchmarks. Only a small variation of the average optimum solution was observed with different reinitialization velocities for most of the benchmarks.

Moreover, for the Rastrigrin function in 10 dimensions, as well as Griewank function in 30 dimensions, the HPSO-TAVC method converged to the stopping criteria (0.01) irrespective of the reinitialization velocity. Further, it is clear from the results that the time-varying reinitialization velocity is an effective strategy to overcome the difficulties of proper selection of reinitialization velocities for different problems.

Finally, we summarize the performance for all PSO concepts introduced in this paper, in Tables IX and X.

In order to make a fair conclusion, the performance of the MPSO method and the HPSO method together with the PSO-TVIW method (MPSO-FAC and HPSO-FAC), where the acceleration coefficients are fixed at 2, are also observed. Results are compared with the performance of both PSO-TVIW and PSO-RANDIW methods. In this investigation for the MPSO strategy, the mutation probability was set to 0.4 and the mutation step size was set to decay from $V\max$ to 0.1 $V\max$ during the search. For the HPSO strategy, reinitialization velocity was set to change from $V\max$ to 0.1 $V\max$ during the search. From the results presented in Table IX, it has been understood that the HPSO-TVAC method is superior to all the other methods for most of the benchmarks considered in this investigation. However, for the Rosenbrock function in small dimensions (ten dimension), the performance of the HPSO-TVAC was found to be poor compared with most of the other methods.

However, the performance in terms of average optimum solution for Schaffer's f6 function was found to be significantly poor with the HPSO-TVAC strategy.

On the other hand, with the HPSO-TVAC concept, the standard deviation of the final solution for 50 trials is found to be significantly low for most of the functions in all dimensions

TABLE VIII
VARIATION OF THE AVERAGE AND THE STANDARD DEVIATION OF THE OPTIMAL VALUE FOR 50 TRIALS WITH THE RE-INITIALIZING VELOCITY FOR HPSO-TVAC METHOD

Function	Dimension	Gmax	Average Optimum Solution / (Standard Deviation) With respect to the re-initialization velocity as a % of maximum velocity						
			100%	50%	20%	10%	200%	500%	Time varying (100% - 10%)
f_1	10	1000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	20	2000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	30	3000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
f_2	10	3000	14.834 (9.863)	11.7067 (11.126)	11.252 (8.736)	**10.046 (7.158)**	16.527 (12.713)	17.984 (20.923)	12.967 (11.538)
	20	4000	15.537 (10.295)	12.93 (6.734)	11.378 (8.254)	**9.389 (5.364)**	14.700 (10.968)	12.319 (8.494)	14.093 (9.641)
	30	5000	13.980 (7.93)	15.979 (12.5227)	12.640 (6.899)	**9.855 (6.725)**	14.748 (8.972)	16.172 (10.446)	13.666 (11.006)
f_3	10	3000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	20	4000	**0.01**	0.043 (0.1961)	**0.01**	0.023 (0.140)	**0.01**	0.083 (0.337)	**0.01**
	30	5000	0.124 (0.4319)	0.103 (0.361)	0.302 (1.225)	0.361 (1.403)	0.1827 (0.625)	0.163 (0.544)	**0.044 (0.196)**
f_4	10	3000	0.056 (0.0272)	0.055 (0.0242)	0.057 (0.022)	**0.048 (0.0255)**	0.05 (0.0252)	0.060 (0.0291)	0.057 (0.0264)
	20	4000	0.0157 (0.016)	**0.011 (0.005)**	**0.011 (0.005)**	0.012 (0.0065)	0.012 (0.008)	0.012 (0.009)	**0.011 (0.005)**
	30	5000	**0.01 (0.001)**	0.01 (0.004)	0.01 (0.002)	0.01 (0.003)	0.011 (0.0051)	0.01 (0.003)	0.01 (0.0035)
f_6	2	1000	0.008 (0.007)	0.010 (0.007)	**0.007 (0.003)**	0.01 (0.008)	0.008 (0.005)	0.007 (0.003)	0.007 (0.007)

compared with all of the other methods. Therefore, the HPSO-TVAC method was identified as a highly consistent strategy in finding the optimum solution compared with the other methods. Further, for the Rastrigrin function in all the dimensions, both the MPSO-TVAC method, as well as the HPSO-TVAC method significantly outperformed all the other methods and converged to the stopping criteria or very close to it at each trial. However, the performance in terms of the optimum solution for both the MPSO-FAC and the HPSO-FAC methods are found to be significantly poor.

In Table X, we summarize the results related to the convergence of each method to the stopping criteria. From the results, it is clear that even though all the methods have converged to the stopping criteria for the Sphere function, the PSO-RANDIW method is significantly faster than all of the other methods. However, with the Rastrigrin function except in 30 dimensions, HPSO-TVAC method has converged to the stopping criteria in all 50 trials. Further, for the Rastrigrin function in 30 dimensions, 48 out of 50 trials have converged to the stopping criteria. In addition, for the Griewank function, particularly at higher dimensions, the HPSO-TVAC method significantly outperformed the all the other methods.

Fig. 1 displays the variation of the optimum solution over the generations for PSO strategies considered in this investigation. All the benchmarks except Schaffer's f6 function, which is 2-D, are considered in 30 dimensions for this paper. For MPSO-TVAC method, mutation probability was set to 0.4 and mutation step size was set to change from $V\max$ to 0.1 $V\max$ during the search. For the HPSO-TVAC method, reinitialization velocity was set to change from $V\max$ to 0.1 $V\max$ during the search.

From Fig. 1, it is clear that the PSO-RANDIW method converges significantly faster than all of the other methods at the early stage of the optimization process for all the functions. However, its improvement was found to be significantly poor at the latter stages of the process for all the benchmarks, except for the sphere function. In contrast, the HPSO-TVAC method has shown to have improved the solution continuously throughout the simulation and has found better solutions for most of the functions except for Schaffer's f6 function. Further, for

TABLE IX
AVERAGE AND THE STANDARD DEVIATION OF THE OPTIMAL VALUE FOR 50 TRIALS FOR DIFFERENT METHODS DISCUSSED IN THIS STUDY

Function	Dimension	Gmax	Average (Standard Deviation)						
			PSO-TVIW	PSO-RANDIW	PSO-TVAC	MPSO-TVAC	MPSO-FAC	HPSO-TVAC	HPSO-FAC
f_1	10	1000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	20	2000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
	30	3000	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.230 (0.173)
f_2	10	3000	27.11 (58.312)	**2.102 (3.218)**	9.946 (32.127)	4.247 (7.961)	11.12 (14.243)	12.967 (11.538)	12.693 (14.397)
	20	4000	51.56 (119.79)	28.1788 (73.072)	17.944 (46.296)	17.7148 (60.306)	54.402 (92.88)	**14.093 (9.641)**	101.126 (129.56)
	30	5000	63.35 (71.210)	35.277 (55.751)	28.97 (51.638)	18.633 (25.122)	135.08 (306.07)	**13.666 (11.006)**	706.28 (951.95)
f_3	10	3000	2.069 (1.152)	4.63 (2.366)	2.268 (1.333)	0.01 (0.0033)	19.54 (36.577)	**0.01**	0.0671 (0.237)
	20	4000	11.74 (3.673)	26.293 (8.176)	15.323 (5.585)	0.3415 (0.588)	2.786 (1.808)	**0.01**	11.391 (6.489)
	30	5000	29.35 (6.578)	69.7266 (20.700)	36.236 (8.133)	2.050 1.910	12.477 (5.990)	**0.044 (0.196)**	36.847 (10.626)
f_4	10	3000	0.0675 (0.029)	0.0661 (0.030)	0.05454 (0.025)	**0.0469 (0.0256)**	0.065 (0.2373)	0.057 (0.026)	0.057 (0.023)
	20	4000	0.0288 (0.023)	0.0272 (0.025)	0.0293 (0.027)	0.0239 (0.017)	0.027 0.025	**0.011 (0.005)**	0.055 (0.085)
	30	5000	0.0167 (0.013)	0.0175 (0.018)	0.0191 (0.015)	0.0169 (0.0149)	0.018 (0.051)	**0.01 (0.0035)**	0.116 (0.193)
f_6	2	1000	0.0039 (0.0019)	0.0029 (0.004)	0.0039 (0.0019)	**0.0012 (0.0031)**	0.004 (0.026)	0.01 (0.007)	0.003 (0.0047)

Schaffer's f6 function, the MPSO-TVAC strategy outperformed all of the other methods.

VI. APPLICATION TO PARAMETER SELECTION OF INTERNAL COMBUSTION ENGINES

As a case study, the PSO strategies discussed in this paper were used to find the best set of operating parameters of an internal combustion engine for optimum power output.

In general, performance of spark ignition engines depends on operating parameters such as spark timing, air-fuel ratio, exhaust gas recirculation, valve timing, and geometrical parameters such as compression ratio, combustion chamber shape, piston ring geometry, spark plug location, and so on. In addition, performance of internal combustion engines is highly restricted to constraints such as abnormal combustion like knocking [40]; a penalty is introduced if the engine in knocking.

Seven different engine operation parameters are identified as input parameters for the present study. They are as follows: compression ratio, equivalence ratio, spark timing, inlet valve opening timing, inlet valve duration, exhaust valve opening timing, exhaust valve duration, and the maximum allowable inlet valve lift. A detailed description about these input variables is given in [39] and [40]. The engine power output (fitness), for a given set of input variables is evaluated using a thermodynamic engine simulation model developed by the authors.

The fitness function of the engine power optimization was formulated as follows:

$$\text{Engine power} = f(\text{operation parameters})$$

$$\text{Fitness} = \text{Engine Power}.$$

If the engine knocks for a particular set of operating parameters, a penalty is introduced as follows:

$$\text{Fitness} = 0.2 * \text{Engine Power}.$$

The PSO methods discussed in this paper were used to find the optimal values of input variables, which maximize the engine power. The structure of the implementation of particle swarm algorithms to find the maximum power through engine simulation model is given in Fig. 2.

TABLE X
NUMBER OF TRIALS THAT CONVERGED TO THE STOPPING CRITERIA AND AVERAGE NUMBER OF GENERATIONS FOR CONVERGENCE FOR 50 TRIALS

Function	Dimension	Gmax	No of trails converged to the stopping criteria (Average number of generations)						
			PSO-TVIW	PSO-RANDIW	PSO-TVAC	MPSO-TVAC	MPSO -FAC	HPSO-TVAC	HPSO-FAC
f_1	10	1000	50 (554.2)	**50 (136.1)**	50 (290.1)	50 (572.5)	50 573	50 (245.1)	50 (175.8)
	20	2000	50 (1274.5)	**50 (274.1)**	50 (592.3)	50 (1305.7)	50 (1315.7)	50 (547.7)	50 (1808.2)
	30	3000	50 (2060.1)	**50 (452.1)**	50 (909.6)	50 (977.1)	50 (2122.9)	50 (862.1)	0
f_2	10	3000	0	**3 (2906.5)**	7 (2926.7)	4 (2945.0)	1 (2973.4)	1 (2997.2)	0
	20	4000	0	**2 (3941.88)**	1 (2972.0)	1 (3995.8)	0	0	0
	30	5000	0	0	**2 (4976.1)**	0	0	0	0
f_3	10	3000	2 (2965.2)	1 (2947.7)	2 (2928.7)	50 (1269.8)	47 (2197.5)	**50 (1249.7)**	47 (1637.7)
	20	4000	0	0	0	36 (2991.3)	5 (3956.8)	**50 (2467.3)**	0
	30	5000	0	0	0	12 (4690.8)	0	**48 (3752.4)**	0
f_4	10	3000	1 (2974.2)	2 (2897.6)	2 (2926.0)	**3 (2907.8)**	0	3 (2930.7)	1 (2989.9)
	20	4000	14 (3521)	17 (2752.3)	13 (3233.3)	17 3013.5	19 (3405.4)	**40 (1550.7)**	11 (3535.9)
	30	5000	29 (3933.9)	30 (2298.7)	24 (3252.0)	27 (3155.7)	29 (4029.1)	**39 (2202.7)**	0
f_6	2	1000	45 (520.2)	46 (462.1)	**48 (374.9)**	48 (386.8)	46 (521.5)	19 (750.8)	12 (618.9)

The maximum allowable velocity ($V\max$) is set to the upper limit of the dynamic range of the search. The dynamic range of the search and the maximum velocity for each input variable are given in Table XI.

All the simulations were carried out with a population size of 40. The maximum number of iterations was set to 150.

As stated above, it is common practice in PSO to limit the maximum velocity of each dimension ($\mathbf{v}_{\rm id}$) to a maximum allowable velocity ($V\max_d$) to avoid excessive searching outside the predefined search space. Even though this is generally limited to the upper limit of the search ($V\max_d = X\max_d$), it does not confine the search to the predefined search space and overshoots can occur at any time.

In the engine optimization problem, particles should be highly confined to the defined search space as input parameters outside the search space are not practically viable. Therefore, whenever a dimension of a particle moves away from the predefined search space it is replaced with a corresponding random value inside the search space.

The performance of the newly developed PSO methods applied to engine optimization, were observed in comparison with two previous developments considered in this research. Results are shown in Fig. 3 and parameters for optimum performance for each method are listed in Table XII for an engine speed of 1500 rev/min. Spark timing and valve open angles are presented in reference to the top most piston position (top dead center) or bottom most piston position (bottom dead center).

From the results, it is clear that most of the methods considered in this paper are competitive in finding the optimal solution. However, the performance of PSO-TVIW and PSO-TVAC

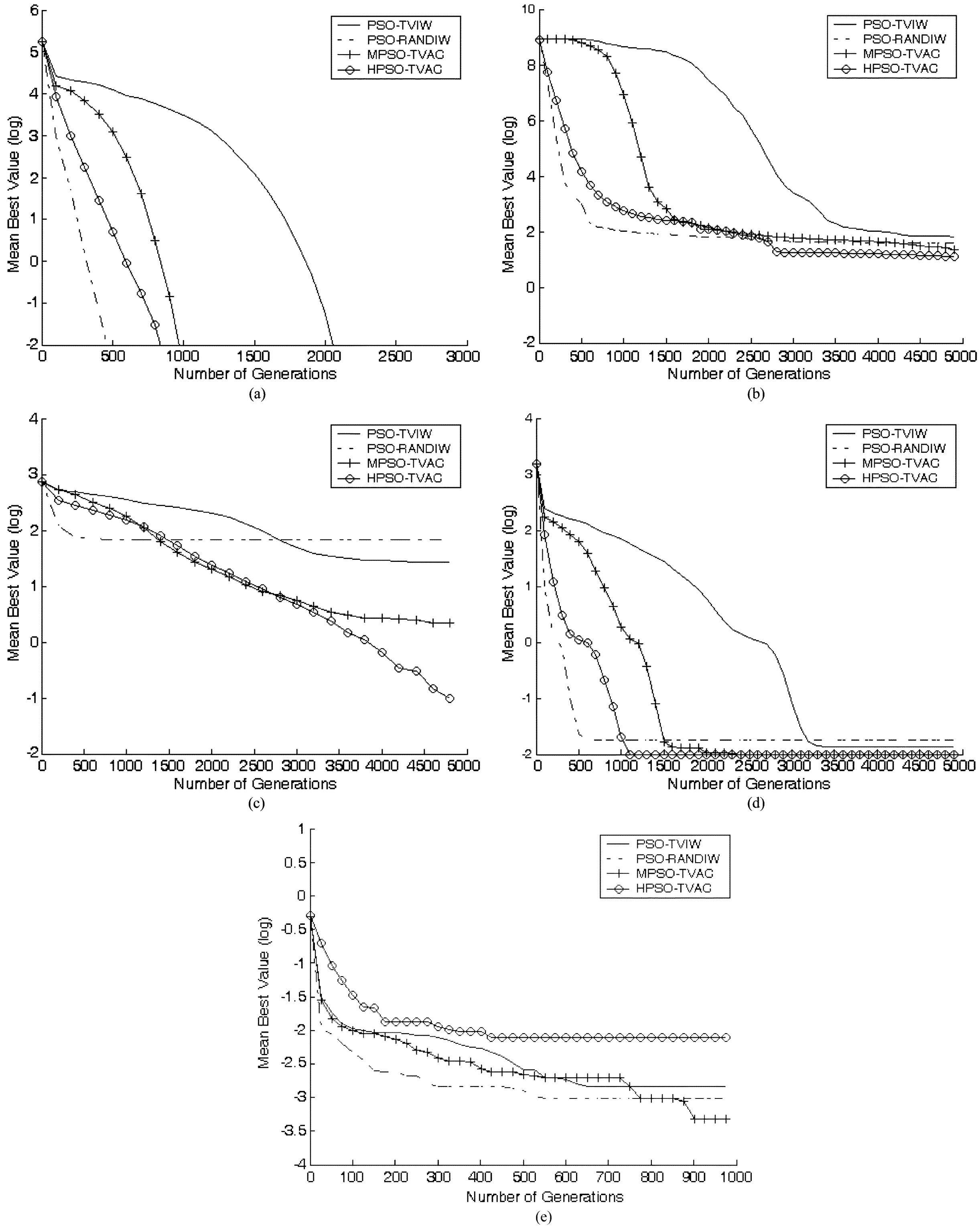


Fig. 1. Variation of the average optimum value with time. (a) Sphere function. (b) Rosenbrock function. (c) Rastrigrin function. (d) Griewank function. (e) Schaffer's f6 function.

method was found to be relatively poor in finding the optimum solution within the predefined number of generations.

VII. Conclusion

We have described a novel parameter automation strategy and two further extensions to the PSO method aiming to improve the performance in terms of the optimal solution within a reasonable number of generations. Then, we applied these novel PSO strategies to select the design parameters for the optimum power output of an internal combustion engine.

Initially, we introduced time-varying acceleration coefficients (PSO-TVAC) to reduce premature convergence in

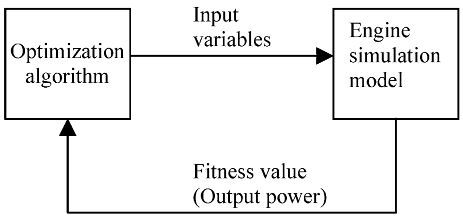


Fig. 2. Data flow diagram between two programs.

TABLE XI
DYNAMIC RANGE OF SEARCH AND THE MAXIMUM VELOCITY

Variable	Dynamic rage	Vmax
Compression ratio	(5, 15)	15
Spark timing	(-30, 10)	10
Equivalence ratio	(0.5, 1.5)	1.5
Inlet valve opening angle	(-50, 50)	50
Inlet valve duration	(-180, 100)	100
Exhaust valve opening angle	(-50, 50)	50
Exhaust valve duration	(100, -100)	100

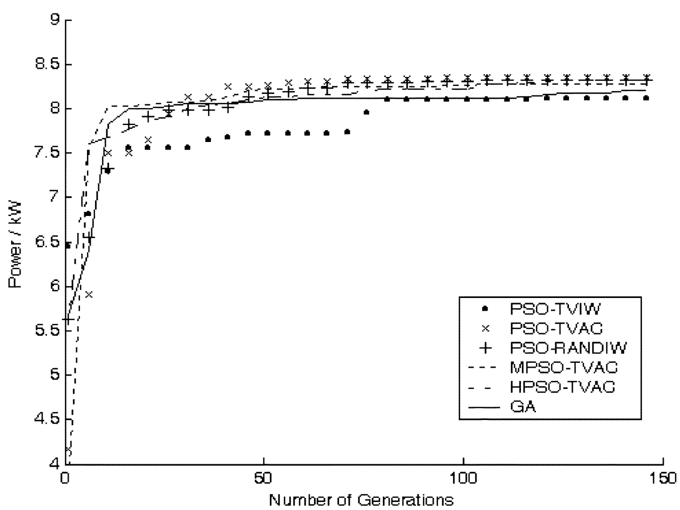


Fig. 3. Variation of optimum power with generations.

the early part of the search and to improve convergence at the end of the optimization. From the results of empirical simulations with five of the well-known benchmarks, a significant improvement of the optimum solution, especially for unimodal functions was observed with PSO-TVAC compared with the PSO with time-varying inertia weight method (PSO-TVIW) method. In contrast, it has also been observed that for multimodal functions, especially for the Rastrigrin function, the introduction of time-varying acceleration coefficients encourages the particles to converge to a local optimum prematurely. However, the performance in terms of optimum solution of the PSO-TVAC method was found to be competitive with PSO method with a random inertia weight factor (PSO-RANDIW) for unimodal functions. In contrast, compared with the PSO-RANDIW method a significant improvement was observed with PSO-TVAC method for the Rastrigrin function. From the basis of varying acceleration coefficients, we introduced two new strategies to improve the performance of PSO.

First, we introduced the concept of "mutation" to the PSO (MPSO) aiming to improve the performance by providing additional diversity into the swarm. A significant improvement in terms of the optimum solution was observed with the introduction of mutation along with time-varying acceleration coefficients (MPSO-TVAC) for all the benchmarks in comparison with the PSO with time-varying inertia weight (PSO-TVIW). The performance of the MPSO-TVAC method on the Rastrigrin function in all dimensions showed significant improvement compared with both the PSO-TVIW and the PSO-RANDIW methods. However, the performance of MPSO-TVAC with all of the other functions was found to be competitive with the PSO-RANDIW method.

Further, the mutation step size and the mutation probability were found to be less sensitive to the performance for most of the test functions considered. However, for the Rastrigrin function, a high mutation step size and mutation probability are preferred. Therefore, to address the difficulties of selecting the appropriate mutation step size for different problems, we successfully introduced a strategy of time-varying mutation step size. On the other hand, it has been observed that the rate of convergence at the early stages of the simulation of the MPSO-TVAC method is significantly similar to the PSO-TVAC for most of the functions despite the significant improvement of the optimum value. Furthermore, the performance of the MPSO-FAC where the acceleration coefficients are fixed at 2 is significantly poor for most of benchmarks.

Second, we introduced another novel concept "self-organizing hierarchical particle swarm optimizer" (HPSO) as a performance improvement strategy. A significant improvement of performance, compared with both PSO-TVIW and PSO-RANDIW methods, was observed with this method along with time-varying acceleration coefficients (HPSO-TVAC) for most benchmarks considered in this paper. However, the performance of the HPSO-TVAC method on Schaffer's f6 function, and the Rosenbrock function in small dimensions was found to be relatively poor. The effect of the reinitialization velocity on the performance was also studied, but a significant variation was not observed for most functions. However, for a more generalized conclusion, we introduced the concept of time-varying reinitialization velocity as a parameter independent strategy of the HPSO-TVAC method. It has also been observed that the performance of the HPSO method is significantly poor with fixed acceleration coefficients $(c_1 = c_2 = 2)$.

Therefore, in conclusion, we propose the HPSO-TVAC method with time-varying reinitialization velocity as a robust and consistent optimization strategy. However, compared with all of the methods the PSO-RANDIW method showed significantly faster convergence at the early stages of the optimization process.

Finally, we applied these new PSO methods to select the design parameters for the optimum performance of an internal combustion spark ignition engine. Seven operating parameters were considered and PSO methods were used to obtain the best set of parameter values for maximum power output. Competitive results were observed with all the methods considered in

TABLE XII
OPTIMUM DESIGN PARAMETERS

Optimization strategy	Spark timing BTDC / CAD	Equivalence ratio	Compression ratio	Inlet valve open angle BTDC/ CAD	Inlet valve duration /CAD	Exhaust valve open angle BBDC/CAD	Exhaust valve duration /CAD	Power / kW
Normal operating conditions	23	1.0	9.65	12	264	58	262	7.45
PSO-TVIW	3.71	1.14	11.95	11.82	196.62	108.0	316.34	8.106
PSO-RANDIW	**9.85**	**1.13**	**11.95**	**14.55**	**191.04**	**75.0**	**276.40**	**8.415**
PSO-TVAC	4.02	1.12	14.87	19.87	199.98	65.37	262.22	8.410
MPSO-TVAC	9.35	1.13	11.97	18.70	196.63	53.60	252.35	8.352
HPSO-TVAC	7.57	1.13	12.82	13.42	190.70	53.65	264.01	8.405

Note: BTCD—before top dead center, BBDC—before bottom dead center, CAD—crank angle degrees.

this paper. Further, PSO methods were found to be a promising technique to optimize the performance of internal combustion engines.

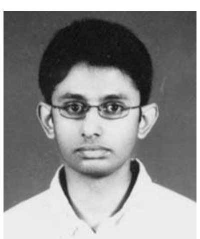
ACKNOWLEDGMENT

The authors are highly appreciative for the assistance extended by a fellow researcher G. Coath in proofreading this paper.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942–1948.
[2] J. Kennedy, "The particle swarm: Social adaptation of knowledge," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1997, pp. 303–308.
[3] ——, "Behavior of particles," in *Lecture Notes in Computer Science*, vol. 1447, Proc. 7th Int. Conf. Evolutionary Programming—Evolutionary Programming VII, Mar. 1998, pp. 581–589.
[4] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics, Computational Cybernetics, Simulation*, vol. 5, 1997, pp. 4104–4108.
[5] J. Kennedy, "Stereotyping: Improving particle swarm performance with cluster analysis," in *Proc. IEEE Int. Conf. Evolutionary Computation*, vol. 2, 2000, pp. 303–308.
[6] J. Kennedy and W. M. Spears, "Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multi-modal problem generator," in *Proc. IEEE World Congr. Computational Intelligence*, May 1998, pp. 78–83.
[7] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proc. IEEE Int. Conf. Computational Intelligence*, 1998, pp. 84–89.
[8] ——, "Evolutionary optimization verses particle swarm optimization: Philosophy and the performance difference," in *Lecture Notes in Computer Science*, vol. 1447, Proc. 7th Int. Conf. Evolutionary Programming—Evolutionary Programming VII, Mar. 1998, pp. 600–610.
[9] M. Clerc, "The swarm and the queen: Toward a deterministic and adaptive particle swarm optimization," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 3, 1999, p. 1957.
[10] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 58–73, Feb. 2002.
[11] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. IEEE Int. Conf. Evolutionary Computation*, vol. 1, 2001, pp. 81–86.
[12] ——, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 1, 2000, pp. 84–88.
[13] Y. Shi and R. C. Eberhart, "Comparison between genetic algorithms and particle swarm optimization," in *Lecture Notes in Computer Science—Evolutionary Programming VII*, vol. 1447, Proc. 7th Int. Conf. Evolutionary Programming, Mar. 1998, pp. 611–616.
[14] ——, "Parameter selection in particle swarm optimization," in *Lecture Notes in Computer Science—Evolutionary Programming VII*, vol. 1447, Proc. 7th Int. Conf. Evolutionary Programming, Mar. 1998, pp. 591–600.
[15] ——, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1998, pp. 69–73.
[16] ——, "Fuzzy adaptive particle swarm optimization," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 1, 2001, pp. 101–106.
[17] ——, "Empirical study of particle swarm optimization," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 3, 1999, pp. 101–106.
[18] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm optimizer with breeding and subpopulation," in *Proc. 3rd Genetic Evolutionary Computation Conf. (GECCO-2001)*, San Francisco, CA, July 2001, pp. 469–476.
[19] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 3, 1999, pp. 1958–1962.
[20] F. van den Bergh and A. P. Engelbrecht, "Effect of swarm size on cooperative particle swarm optimizers," in *Proc. Genetic Evolutionary Computation Conf. (GECCO-2001)*, San Francisco, CA, July 2001, pp. 892–899.
[21] T. M. Blackwell and P. bentley, "Improvised music with swarms," in *Proc. IEEE Congr. Evolutionary Computation 2002*, vol. 2, Honolulu, HI, May 2002, pp. 1462–1467.
[22] M. Lovbjerg and T. Krink, "Extending particle swarm optimizers with self-organized critically," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 2, Honolulu, HI, May 2002, pp. 1588–1593.
[23] J. S. Vesterstrom, J. Riget, and T. Krink, "Division of labor in particle swarm optimization," in *Proc. IEEE Int. Congr. Evolutionary Computation 2002*, vol. 2, Honolulu, HI, May 2002, pp. 1570–1575.
[24] T. Krink, J. S. Vesterstrom, and J. Riget, "Particle swarm optimization with spatial particle extension," in *Proc. IEEE Congr. Evolutionary Computation 2002*, vol. 2, Honolulu, HI, May 2002, pp. 1474–1479.
[25] X. F. Xie, W. J. Zhang, and Z.-L.Zhi-Lian Yang, "A dissipative particle swarm optimization," in *Proc. IEEE Congr. Evolutionary Computation 2002*, vol. 2, Honolulu, HI, May 2002, pp. 1456–1461.
[26] A. Carlisle and G. Dozier, "Tracking changing extrema with particle swarm optimizer," Auburn Univ., Auburn, AL, Tech. Rep. CSSE01-08, 2001.
[27] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method in multiobjective problems," in *Proc. ACM Symp. Applied Computing 2002 (SAC 2002)*, 2002, pp. 603–607.
[28] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for minimax problems," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 2, Honolulu, HI, May 2002, pp. 1576–1581.
[29] ——, "Particle swarm optimization for integer programming," in *Proc. IEEE Int. Congr. Evolutionary Computation*, vol. 2, Honolulu, HI, May 2002, pp. 1582–1587.

[30] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[31] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. IEEE Congr. Evolutionary Computation 2001*, Seoul, Korea, 2001, pp. 94–97.
[32] Y. Fukuyama and H. A. Yoshida, "Particle swarm optimization for reactive power and voltage control in electric power systems," in *Proc. IEEE Congr. Evolutionary Computation 2001 (CEC 2001)*, vol. 1, May 2001, pp. 87–93.
[33] J. Kennedy and R. C Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. World Multiconf. Systemics, Cybernetics, Informatics*, Orlando, FL, 1997, pp. 4104–4109.
[34] E. Ozcanand and C. K. Mohan, "Particle swarm optimization: Surfing the waves," in *Proc. IEEE Congr. Evolutionary Computation 1999*, vol. 3, Washington, DC, 1999, pp. 1944–1999.
[35] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage stability," in *Proc. Int. Conf. Intelligent System Application to Power System*, Rio de Janeiro, Brazil, 1999, pp. 117–121.
[36] V. Tandon, H. E. Mounayri, and H. Kishawy, "NC end milling optimization using evolutionary computation," *Int. J. Mach. Tools Manuf.*, vol. 42, no. 5, pp. 595–605, 2002.
[37] G. Ciuprina, D. Ioan, and I. Munteanu, "Use of intelligent-particle swarm optimization in electromagnetics," *IEEE Trans. Magn.*, vol. 38, pp. 1037–1040, Mar. 2002.
[38] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation," in *Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003)*, Indianapolis, IN, 2003, pp. 72–79.
[39] A. Ratnaweera, H. C. Watson, and S. K. Halgamuge, "Optimization of valve timing events of internal combustion engines with particle swarm optimization," in *Proc. 2003 Congr. Evolutionary Computation (CEC 2003)*, vol. 4, Canberra, Australia, Dec. 2003, pp. 2411–2418.
[40] J. B. Heywood, *Internal Combustion Engine Fundamentals*. New York: McGraw-Hill, 2000.



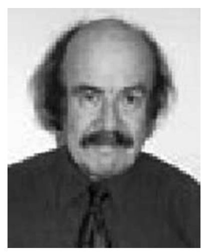
Asanga Ratnaweera received the B.Sc. degree in production engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 1997.

He is a Lecturer with the Department of Mechanical Engineering, University of Peradeniya. He is on leave from the University of Peradeniya and is currently with the Mechatronics Research Group, Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria, Australia. His major research interests are artificial intelligence, internal combustion engines, hybrid vehicles, etc.

Saman K. Halgamuge (M'85) received the B.Sc. degree in electronic and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 1985, and the Dipl.-Ing. and Dr.-Ing. degrees in computer engineering from Darmstadt University (TUD), Darmstadt, Germany, in 1990 and 1995, respectively.

In 1984, he worked as an Engineer with the Ceylon Electricity Board, Colombo, Sri Lanka. From 1990 to 1995, he was a Research Associate at TUD. He was a Lecturer in computer systems engineering associated with the Institute for Telecommunications Research, University of South Australia, Adelaide, Australia, from 1996 to July 1997. He joined the Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria, Australia, as a Senior Lecturer in mechatronics in 1997. Since 2002, he is an Associate Professor and Reader in the Department of Mechanical and Manufacturing Engineering. He is a Member of the Editorial Board of the *Journal on Neural Information Processing: Letters and Reviews*. He has coauthored over 150 research papers and contributed to books in the areas of data mining and analysis, mechatronics, soft computing, and bioinfomatics. His research interest also includes vehicle engineering, communication networks, bioinfomatics, and bio-inspired computing. Over 150 journal papers have cited his research papers during the last ten years (ISI citation database).

Dr. Halgamuge is an Executive Committee Member of the Asia Pacific Neural Network Assembly, and he has Co-chaired and Chaired the Program Committees of the International Conference on Neural Information Processing (ICONIP) 1999 and the International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) 2002, respectively.



Harry C. Watson received the B.Sc. (Eng), DIC, and Ph.D. degrees from Imperial College, London, U.K., in 1962, 1964, and 1969 respectively.

In 2000, after four years as Head of the Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria, Australia, he stepped down to devote more time to his research specialty in Transport Energy and Emissions. He leads a research team with an international reputation for its research investigations spanning combustion fundamentals through to the modeling of the behavior of vehicles in traffic. He is a Personal Chair of in the Department of Mechanical and Manufacturing Engineering, University of Melbourne, Victoria, Australia. He is a Senior Adviser and Consultant to the Australian Greenhouse Office and Environment, Australia. The Transport Energy and Emissions group's patents cover oxygen enrichment of the combustion air in diesel engines and the application of hydrogen assisted jet ignition to spark ignition engines and fueling systems for hydrogen engines. Model predictions of the impact of vehicle emissions on urban air pollution have been used as the basis for new motor vehicle emission control regulations and at the local level, in impact studies for road systems. The group's engine and vehicle test facilities and suite of computational models are unique in Australia. The group is one of a few in the world to be able to deliver prototype power plants designed using CAD/CAE and manufactured paperlessly in on-site facilities. He is a Member of FTSE, FSAE-A, I.Mech.E., SAE, and HEA.