

Dynamic multi-swarm particle swarm optimizer with harmony search

S.-Z. Zhao^a, P.N. Suganthan^{a,*}, Quan-Ke Pan^b, M. Fatih Tasgetiren^c

^a School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

^b College of Computer Science, Liaocheng University, Liaocheng 252059, PR China

^c Department of Industrial Engineering, Yasar University, Bornova, Izmir, Turkey

ARTICLE INFO

Keywords:

Particle swarm optimizer
Dynamic multi-swarm particle swarm optimizer
Harmony search
Dynamic sub-swarms
Numerical optimization
Multimodal optimization

ABSTRACT

In this paper, the dynamic multi-swarm particle swarm optimizer (DMS-PSO) is improved by hybridizing it with the harmony search (HS) algorithm and the resulting algorithm is abbreviated as DMS-PSO-HS. We present a novel approach to merge the HS algorithm into each sub-swarm of the DMS-PSO. Combining the exploration capabilities of the DMS-PSO and the stochastic exploitation of the HS, the DMS-PSO-HS is developed. The whole DMS-PSO population is divided into a large number of small and dynamic sub-swarms which are also individual HS populations. These sub-swarms are regrouped frequently and information is exchanged among the particles in the whole swarm. The DMS-PSO-HS demonstrates improved on multimodal and composition test problems when compared with the DMS-PSO and the HS.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Particle swarm optimizer (PSO) emulates flocking behavior of birds and herding behavior of animals to solve optimization problems. The PSO was introduced by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995). Typical single objective bound constrained optimization problems can be expressed as:

$$\begin{aligned} \text{Min}(f(x)), \quad x &= [x_1, x_2, \dots, x_D] \\ x &\in [x_{\min}, x_{\max}] \end{aligned} \quad (1)$$

where D is the number of parameters to be optimized. The x_{\min} and x_{\max} are the upper and lower bounds of the search space. In PSO, each potential solution is regarded as a particle. All particles fly through the D dimensional parameter space of the problem while learning from the historical information gathered during the search process. The particles have a tendency to fly towards better search regions over the course of search process. The velocity V_i^d and position X_i^d updates of the d th dimension of the i th particle are presented below:

$$\begin{aligned} V_i^d &= w * V_i^d + c_1 * \text{rand1}_i^d * (pbest_i^d - X_i^d) \\ &\quad + c_2 * \text{rand2}_i^d * (gbest^d - x_i^d) \end{aligned} \quad (2)$$

$$X_i^d = X_i^d + V_i^d \quad (3)$$

where c_1 and c_2 are the acceleration constants, rand1_i^d and rand2_i^d are two uniformly distributed random numbers in $[0, 1]$. $X_i = (X_1, X_2, \dots, X_D)$ is the position of the i th particle; $pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^D)$ is the best previous position yielding the best fitness value for the i th particle; $gbest = (gbest^1, gbest^2, \dots, gbest^D)$ is the best position discovered by the whole population; $V_i = (v_i^1, v_i^2, \dots, v_i^D)$ represents the rate of position change (velocity) for particle i . w is the inertia weight used to balance between the global and local search abilities.

In the PSO domain, there are two main variants: global PSO and local PSO. In the local version of the PSO, each particle's velocity is adjusted according to its personal best position $pbest$ and the best position $lbest$ achieved so far within its neighborhood. The global PSO learns from the personal best position $pbest$ and the best position $gbest$ achieved so far by the whole population. The velocity update of the local PSO is:

$$\begin{aligned} V_i^d &= w * V_i^d + c_1 * \text{rand1}_i^d * (pbest_i^d - X_i^d) \\ &\quad + c_2 * \text{rand2}_i^d * (lbest_i^d - x_i^d) \end{aligned} \quad (4)$$

where $lbest = (lbest_i^1, lbest_i^2, \dots, lbest_i^D)$ is the best historical position achieved within i th particle's local neighborhood.

Focusing on improving the local variants of the PSO, different neighborhood structures were proposed and discussed (Broyden, 1970; Clerc & Kennedy, 2002; Fletcher, 1970; Goldfarb, 1970; Kennedy, 1999; Shanno, 1970; van den Bergh & Engelbrecht, 2004; Zielinski & Laur, 2007). Except these local PSO variants, some variants that use multi-swarm (Davidon, 1959; Young, 1989), sub-population (Fletcher & Powell, 1963) can also be

* Corresponding author. Tel.: +65 67905404; fax: +65 67933318.

E-mail addresses: ZH0047NG@ntu.edu.sg (S.-Z. Zhao), epnsugan@ntu.edu.sg (P.N. Suganthan), qkpan@gmail.com (Q.-K. Pan), fatih.tasgetiren@yasar.edu.tr (M. Fatih Tasgetiren).

regarded as the local PSO variants if we view the sub-groups as special neighborhood structures. In the existing local versions of PSO with different neighborhood structures and the multi-swarm PSOs, the swarms are predefined or dynamically adjusted according to the distances. Hence, the freedom of sub-swarms is limited. In Liang and Suganthan (2005), a dynamic multi-swarm particle swarm optimizer (DMS-PSO) was proposed whose neighborhood topology is dynamic and randomized. DMS-PSO exhibits superior exploratory capabilities on multimodal problems than other PSO variants at the expense of its local search performance.

Harmony search (HS) algorithm (Lee & Geem, 2005) conceptualized a behavioral phenomenon of music players' improvisation process, where each player continues to improve its tune in order to produce better harmony in a natural musical performance processes. Originated in an analogy between music improvisation and engineering optimization, the engineers seek for a global solution as determined by an objective function, just like the musicians seek to find musically pleasing harmony as determined by aesthetics (Eberhart & Kennedy, 1995). In music improvisation, each player selects any pitch within the possible range, together making one harmony vector. If all the pitches make a good solution, that experience is stored in each variable's memory, and the possibility of generating a better solution is also increased next time.

Recently, some researchers have improved the HS algorithm by introducing the particle swarm concepts (Omran & Mahdavi, 2008) and reported improved performance. The particle swarm harmony search (PSHS), is presented in Geem (2009). The particle swarm concept is introduced to the original HS algorithm for the first time, and the modified HS algorithm is applied to solve the water-network-design problem.

In this paper, we improved the DMS-PSO by combining original harmony search (HS) algorithm, and evaluate the DMS-PSO-HS on multimodal and composition numerical optimization problems. In order to compare fairly, the original HS and the original DMS-PSO are included in the comparisons.

2. Dynamic multi-swarm particle swarm optimizer with harmony search

In this section, we will first introduce the dynamic multi-swarm particle swarm optimizer (DMS-PSO) and the harmony search (HS). Finally, we will explain how we combine the two approaches to form the proposed dynamic multi-swarm particle swarm optimizer with harmony search (DMS-PSO-HS).

2.1. The DMS-PSO algorithm

The dynamic multi-swarm particle swarm optimizer was constructed based on the local version of the PSO with a new neighborhood topology (Liang & Suganthan, 2005). Many existing evolutionary algorithms require larger populations, while PSO requires a comparatively smaller population size. A population with three to five particles can achieve satisfactory results for simple problems. According to many reported results on the local variants of the PSO (Broyden, 1970; Kennedy, 1999), PSO with small neighborhoods performs better on complex problems. Hence, in order to slow down the convergence speed and to increase diversity to achieve better results on multimodal problems, in the DMS-PSO, small neighborhoods are used. The population is divided into small sized swarms. Each sub-swarm uses its own members to search for better regions in the search space.

Since the small sized swarms are searching using their own best historical information, they can easily converge to a local optimum because of PSO's speedy convergence behavior. Further, unlike a co-evolutionary PSO, we allow maximum information exchange among the particles. Hence, a randomized regrouping schedule is introduced so that the particles will enhance their diversity by having dynamically changing neighborhood structures. Every R generations, the population is regrouped randomly and starts searching using a new configuration of small sub-swarms. Here R is called the regrouping period. In this way, the information obtained by each sub-swarm is exchanged among the whole swarm. Simultaneously the diversity of the population is also increased. In this algorithm, in order to constrain the particles within the search range, the fitness value of a particle is calculated and the corresponding $pbest$ is updated only if the particle is within the search range. Since all $pbest$ s and $lbest$ s are within the search bounds, all particles will eventually return within the search bounds.

For example, suppose that we have three sub-swarms with three particles in each sub-swarm. First, the nine particles are divided into three sub-swarms randomly. Then the three sub-swarms use their own particles to search for better solutions. In this period, they may converge to near a local optimum. Then the whole population is regrouped into three different sub-swarms. The new sub-swarms begin their search. This process is continued until a termination criterion is satisfied. With the periodically randomized regrouping process, particles from different sub-swarms are grouped in a new configuration so that each small sub-swarm's search space is enlarged and better solutions are possible to be found by the new small sub-swarms. This regrouping procedure is shown in Fig. 2, and the flowchart of the original DMS-PSO is given in Fig. 1.

2.2. The HS algorithm

The HS algorithm (Geem, 2009; Lee & Geem, 2005; Omran & Mahdavi, 2008) is based on natural musical performance processes that occur when a musician searches for a better state of harmony. The optimization operators of HS algorithm are specified as: the harmony memory (HM), which stores the solution vectors which are all within the search space, as shown in Eq. (5), the harmony memory size HMS , specifies the number of solution vectors stored in the HM, the harmony memory consideration rate ($HMCR$), the pitch adjustment rate (PAR) and the pitch adjustment bandwidth (bw).

$$HM = \left[\begin{array}{cccc|c} x_1^1 & x_2^1 & \dots & x_D^1 & fitness(x^1) \\ x_1^2 & x_2^2 & \dots & x_D^2 & fitness(x^2) \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_D^{HMS} & fitness(x^{HMS}) \end{array} \right] \quad (5)$$

When a musician improvises one pitch, usually one of three rules are used: (1) playing any one pitch from musician's memory known as the harmony memory consideration rate ($HMCR$), (2) playing an adjacent pitch of one pitch in Musician's memory, and (3) playing totally random pitch from feasible ranges. Similarly, when each decision variable chooses one value in the HS algorithm, it can apply one of the above three rules in the whole HS procedure. If a new harmony vector is better than the worst harmony vector in the HM, the new harmony vector replaces the worst harmony vector in the HM. This procedure is repeated until a stopping criterion is satisfied. The computational procedure of the basic HS algorithm can be summarized as follows:

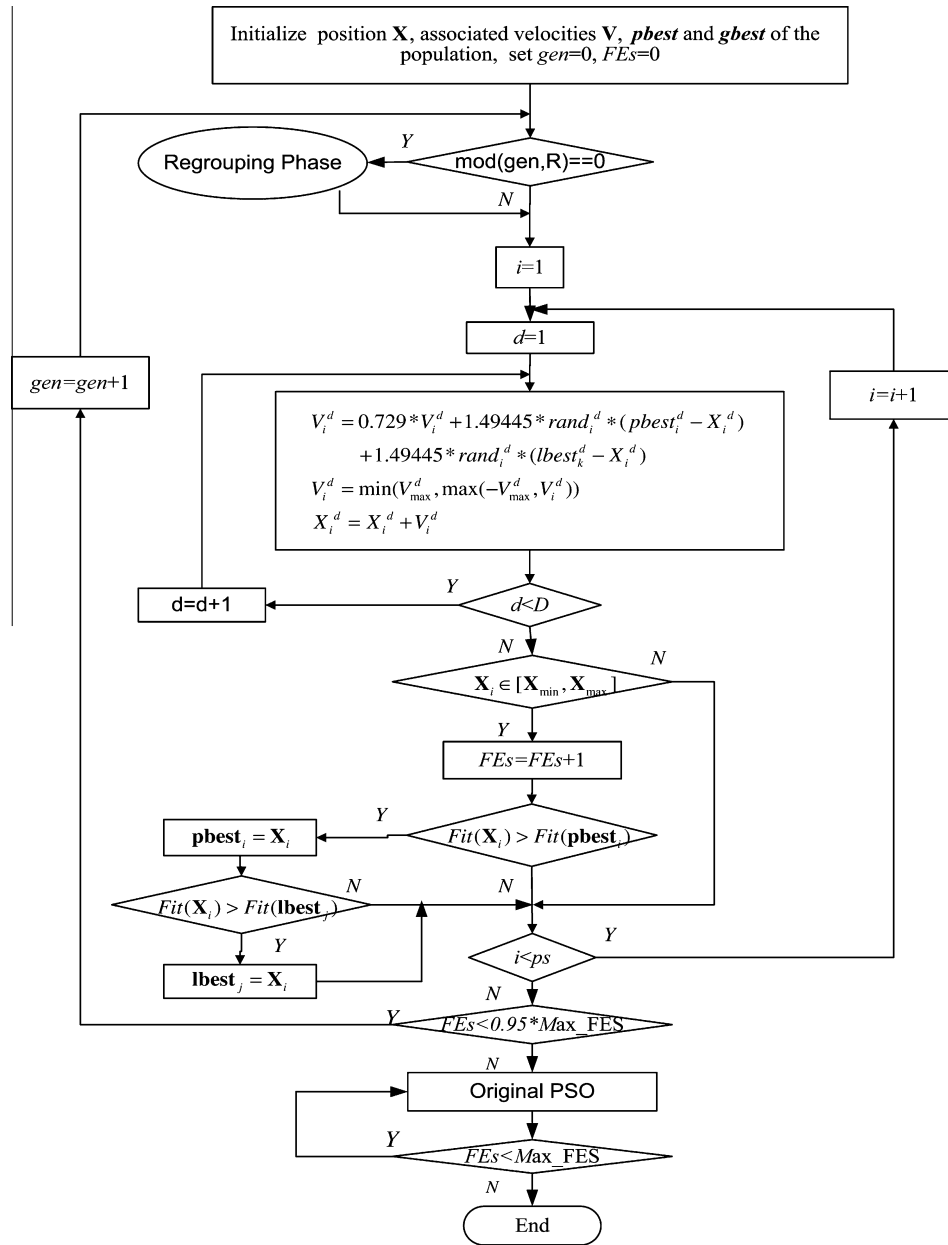


Fig. 1. The flowchart of the DMS-PSO.

Step 1: Set the parameters and initialize the HM.

Step 2: Improvise a new harmony X_{new} as follows:

for ($d = 1$ to D) do

if ($\text{rand}() < \text{HMCR}$) then // memory consideration

$x_{new}(d) = x_a(d)$ where $a \in (1, 2, \dots, \text{HMS})$

if ($\text{rand}() < \text{PAR}$) then // pitch adjustment

$x_{new}(d) = x_{new}(d) \pm \text{rand}() \times \text{BW}$

endif

else // random selection

$x_{new}(d) = x_{\min,d} + \text{rand}() \times (x_{\max,d} - x_{\min,d})$

endif

endfor

Step 3: Update the HM as $x_W = x_{new}$ if $f(x_{new}) < f(x_W)$ (minimization objective)

Step 4: If termination criterion is reached, return the best harmony vector found so far; otherwise go to Step 2.

2.3. DMS-PSO-HS

In order to achieve better results on multimodal problems, DMS-PSO (Liang & Suganthan, 2005) is designed in such a way that the particles have a larger diversity by sacrificing the convergence speed of the global PSO. Even after the globally optimal region is found, the particles will not converge rapidly to the globally optimal solution. Hence, maintaining the diversity and obtaining good solutions rapidly at the same time is a challenge which is tackled by integrating an HS phase in the DMS-PSO to obtain the DMS-PSO-HS. A new harmony will be obtained from the temporary HM formed by the current **pbests** in each sub-swarm. We calculate the Euclidean distance between all the **pbests** in the corresponding sub-swarm and the new harmony vector which will replace the nearest **pbest** if the new harmony has a better fitness value. Except the HS phase in each sub-swarm, the original DMS-PSO is retained.

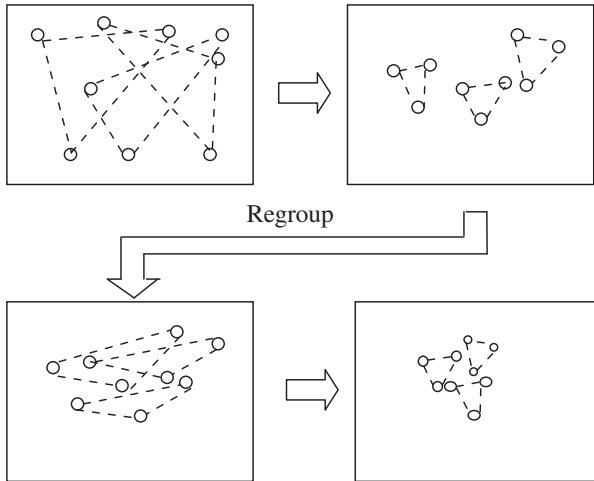


Fig. 2. DMS-PSO's regrouping phase.

In this way, the strong exploration abilities of the original PSO and the exploitation abilities of the HS can be fully exploited. The flow-chart of the proposed DMS-PSO-HS is presented in Fig. 3.

3. Experimental results

3.1. Test problems

In order to comprehensively compare the DMS-PSO-HS with the two original algorithms, namely the DMS-PSO and the HS, 16 diverse benchmark problems are used (Suganthan et al., 2005). Most of them are multimodal and composition test problems. All problems are tested on 10 and 30 dimensions. According to their properties, these problems are divided into four groups: 2 unimodal problems, 6 unrotated multimodal problems, 6 rotated multimodal problems and 2 composition problems. The properties of these functions are presented below.

3.1.1. Group 1: unimodal and simple multimodal problems

(1) Sphere function

$$f_1(x) = \sum_{i=1}^D x_i^2 \quad (6)$$

(2) Rosenbrock's function

$$f_2(x) = \sum_{i=1}^{D-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right) \quad (7)$$

The first problem is the sphere function and is easy to solve. The second problem is the Rosenbrock function. It can be treated as a multimodal problem. It has a narrow valley from the perceived local optima to the global optimum. In the experiments below, we find that the algorithms which perform well on sphere function also perform well on Rosenbrock function.

3.1.2. Group 2: unrotated multimodal problems

In this group, there are six multimodal test functions. Ackley's function has one narrow global optimum basin and many minor local optima. It is probably the easiest problem among the six as its local optima are neither deep nor wide. Griewank's function has a $\prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right)$ component causing linkages among dimensions thereby making it difficult to reach the global optimum. An interesting phenomenon of Griewank's function is that it is more

difficult for lower dimensions than higher dimensions (Liang, Qin, Suganthan, & Baskar, 2006). The Weierstrass function is continuous but differentiable only on a set of points. Rastrigin's function is a complex multimodal problem with a large number of local optima. When attempting to solve Rastrigin's function, algorithms may easily fall into a local optimum. Hence, an algorithm capable of maintaining a larger diversity is likely to yield better results. Non-continuous Rastrigin's function is constructed based on the Rastrigin's function and it has the same number of local optima as the continuous Rastrigin's function. The complexity of Schwefel's function is due to its deep local optima being far from the global optimum. It will be hard to find the global optimum, if many particles fall into one of the deep local optima.

(3) Ackley's function

$$f_1(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e \quad (8)$$

(4) Griewank's function

$$f_2(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (9)$$

(5) Weierstrass function

$$f_3(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} \left[a^k \cos \left(2\pi b^k (x_i + 0.5) \right) \right] \right) - D \sum_{k=0}^{k_{\max}} \left[a^k \cos \left(2\pi b^k \cdot 0.5 \right) \right],$$

$$a = 0.5, \quad b = 3, \quad k_{\max} = 20 \quad (10)$$

(6) Rastrigin's function

$$f_4(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (11)$$

(7) Non-continuous Rastrigin's function

$$f_5(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10),$$

$$y_i = \begin{cases} x_i & |x_i| < 1/2 \\ \text{round}(2x_i)/2 & |x_i| \geq 1/2 \end{cases} \text{ for } i = 1, 2, \dots, D \quad (12)$$

(8) Schwefel's function

$$f_6(x) = 418.9829 \times D - \sum_{i=1}^D x_i \sin(|x_i|^{1/2}) \quad (13)$$

3.1.3. Group 3: rotated multimodal problems

In Group 1, some functions are separable and they can be solved by using D one-dimensional searches where D is the dimensionality of the problem. Hence, in Group 2, we have the corresponding six rotated multimodal problems.

(9) Rotated Ackley's function

$$f_9(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi y_i) \right) + 20 + e, \quad \mathbf{y} = \mathbf{M}^* \mathbf{x} \quad (14)$$

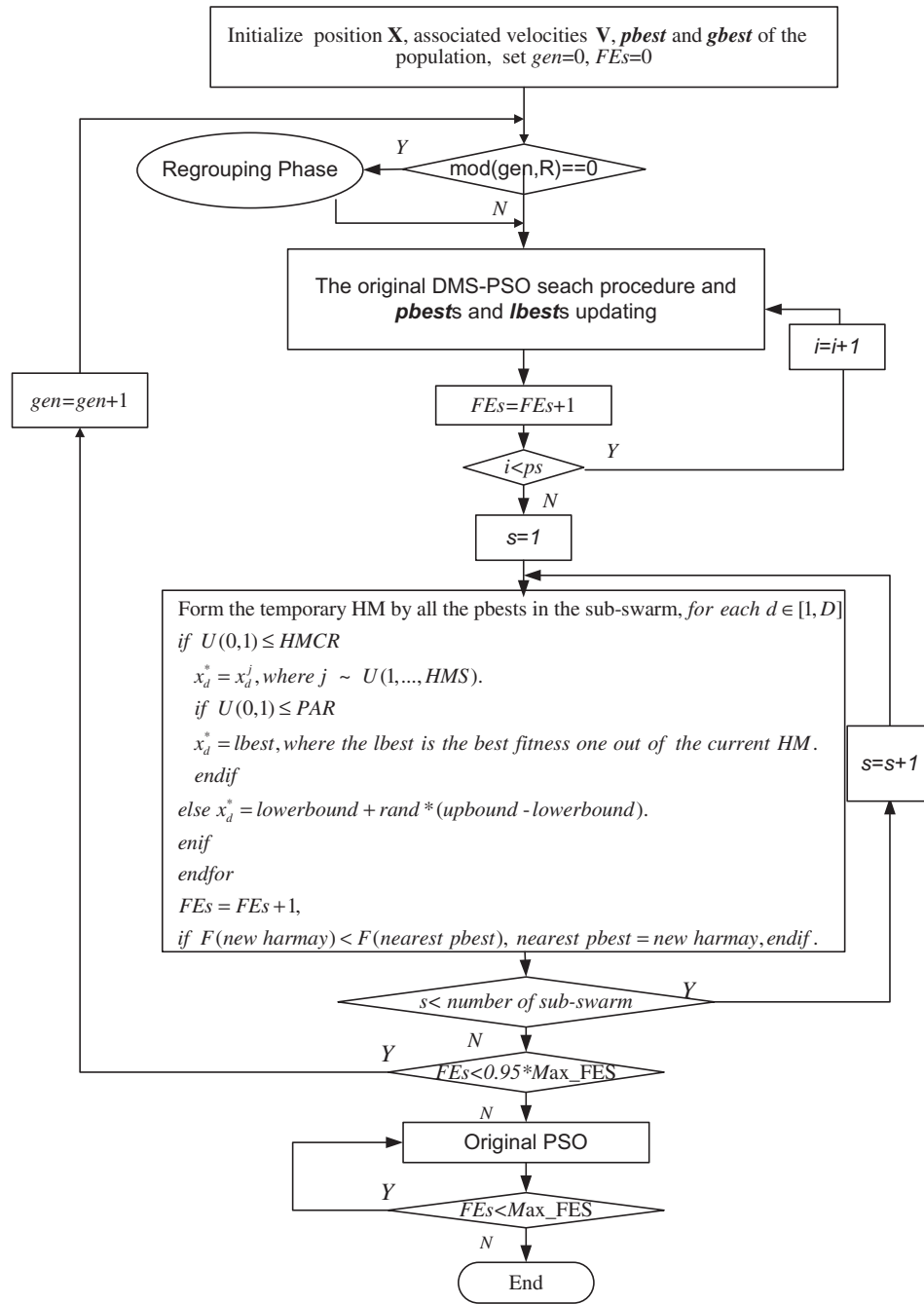


Fig. 3. The flowchart of the DMS-PSO-HS.

(10) Rotated Griewank's function

$$f_{10}(x) = \sum_{i=1}^D \frac{y_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1, \quad \mathbf{y} = \mathbf{M}^* \mathbf{x} \quad (15)$$

(11) Rotated Weierstrass function

$$f_{11}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (y_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)],$$

$$a = 0.5, \quad b = 3, \quad k_{\max} = 20, \quad \mathbf{y} = \mathbf{M}^* \mathbf{x} \quad (16)$$

(12) Rotated Rastrigin's function

$$f_{12}(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10), \quad \mathbf{y} = \mathbf{M}^* \mathbf{x} \quad (17)$$

(13) Rotated Non-continuous Rastrigin's function

$$f_{13}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10)$$

$$z_i = \begin{cases} y_i & |y_i| < 1/2 \\ \text{round}(2y_i)/2 & |y_i| \geq 1/2 \end{cases} \quad \text{for } i = 1, 2, \dots, D, \quad \mathbf{y} = \mathbf{M}^* \mathbf{x} \quad (18)$$

(14) Rotated Schwefel's function

$$f_{14}(x) = 418.9829 \times D - \sum_{i=1}^D z_i$$

$$z_i = \begin{cases} y_i \sin(|y_i|^{1/2}) & \text{if } |y_i| \leq 500 \\ 0.001(|y_i| - 500)^2 & \text{if } |y_i| > 500 \end{cases}, \text{ for } i = 1, 2, \dots, D,$$

$$y = y' + 420.96, \quad y' = M^*(x - 420.96)$$
(19)

In rotated Schwefel's function, in order to keep the global optimum in the search range after rotation, noting that the original global optimum of Schwefel's function is at $[420.96, 420.96, \dots, 420.96]$, $y' = M^*(x - 420.96)$ and $y = y' + 420.96$ are used instead of $y = M^*x$. Since Schwefel's function has better solutions out of the search range $[-500, 500]^D$, when $|y_i| > 500$, $z_i = 0.001(|y_i| - 500)^2$, z_i is set in portion to the square distance between y_i and the bound.

3.1.4. Group 4: composition problems

Composition functions are constructed using some basic benchmark problems to obtain more challenging problems with a randomly located global optimum and several randomly located deep local optima. The Gaussian function is used to combine the simple benchmark functions and blur the function's structures. The composition functions are asymmetrical multimodal problems, with different properties in different search regions. The details on how to construct this class of functions and six composition functions are presented in Liang, Suganthan, and Deb (2005). Two of the six composition functions defined in Liang et al. (2005) are used here. Parameters settings for the following two composition functions:

(15) Composition function 1 (CF1) in Liang et al. (2005):

The f_9 (CF1) are composed using ten sphere functions. The global optimum is easy to find once the global basin is found.

(16) Composition function 2 (CF5) in Liang et al. (2005):

The f_{10} (CF5) is composed using ten different benchmark functions: two rotated Rastrigin's functions, two rotated Weierstrass functions, two rotated Griewank's functions, two rotated Ackley's functions and two sphere functions. The CF5 is more complex than CF1 since even after the global basin is found, the global optimum is not easy to locate.

The global optimum x^* , the corresponding fitness value $f(x^*)$, the search ranges $[X_{\min}, X_{\max}]$ and the initialization range of each function are given in Table 1. Biased initializations are used for the functions whose global optimum is at the centre of the search range.

3.2. Parameter settings

The parameters are set as in the general PSO variants (Eberhart & Kennedy, 1995; Kennedy & Eberhart, 1995; Liang & Suganthan, 2005): $\omega = 0.729$, $c_1 = c_2 = 1.49445$, $R = 10$. Vmax restricts particles' velocities and is equal to 20% of the search range. To solve these problems, the number of sub-swarms is set at 10 which is also the same setting as in the original DMS-PSO (Liang & Suganthan, 2005). To tune the remaining parameters, six selected test functions are used to investigate the impact of them. They are 10 dimensional test functions: f_3 Ackley's function, f_4 Griewank's function, f_5 Weierstrass function, f_9 Rotated Ackley's function, f_{10} Rotated Griewank's function, f_{11} Rotated Weierstrass function. Experiments were conducted on these six 10 Dimensional test functions and the mean values of 30 runs are presented.

3.2.1. Sub-population PS

For the sub-population size of each sub-swarm, the results of investigation on the selected test problems are show in Table 2. In this table, the mean values of six problems with different parameter settings are given. Based on the comparison of the results, the best setting is 5 particles for each sub-swarm. This is also the setting for the HS population size. Hence, in DMS-PSO-HS the population size is 50 as there are 10 sub-swarms.

3.2.2. Regrouping period R

For the regrouping period R , it should not be very small because we need to allow enough number of iterations for each sub-swarm to search. It should not also be too large because function evaluations will be wasted when the sub-swarm could not further improve. The Table 3 presents the results of tuning R . Based on the results, the best value for R is 5.

With respect to the HS in the DMS-PSO-HS and the original HS, the effects of the two main parameters of the HS namely $HMCR$ and HMS were investigated by many researchers (Geem, 2009; Omran & Mahdavi, 2008). Based on those observations, a large value for $HMCR$ (i.e. larger than 0.95) is generally used except for problems with a very low dimensionality for which a small value of $HMCR$ is recommended. A small value for HMS seems to be a good choice in general. Dynamically adjusting the value of PAR is also recommended. Hence, in this paper, the parameters of the HS are set as: $HMCR = 0.98$, $bw_{\min} = 0.0001$, $bw_{\max} = 0.05^*(UB-LB)$, $PAR_{\min} = 0.01$, $PAR_{\max} = 0.99$, the PAR and the bandwidth bw are updated using the following expressions:

$$PAR(t) = PAR_{\min} + \frac{(PAR_{\max} - PAR_{\min})}{Max_gen} \times t \quad (20)$$

Table 1

The global optimum, search ranges and initialization ranges of the problems.

f	x^*	$f(x^*)$	Search range	Initialization range
f_1	$[0, 0, \dots, 0]$	0	$[-100, 100]^D$	$[-100, 50]^D$
f_2	$[1, 1, \dots, 1]$	0	$[-2.048, 2.048]^D$	$[-2.048, 2.048]^D$
f_3	$[0, 0, \dots, 0]$	0	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
f_4	$[0, 0, \dots, 0]$	0	$[-600, 600]^D$	$[600, 200]^D$
f_5	$[0, 0, \dots, 0]$	0	$[-0.5, 0.5]^D$	$[-0.5, 0.2]^D$
f_6	$[0, 0, \dots, 0]$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_7	$[0, 0, \dots, 0]$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_8	$[420.96, 420.96, \dots, 420.96]$	0	$[-500, 500]^D$	$[-500, 500]^D$
f_9	$[0, 0, \dots, 0]$	0	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
f_{10}	$[0, 0, \dots, 0]$	0	$[-600, 600]^D$	$[600, 200]^D$
f_{11}	$[0, 0, \dots, 0]$	0	$[-0.5, 0.5]^D$	$[-0.5, 0.2]^D$
f_{12}	$[0, 0, \dots, 0]$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_{13}	$[0, 0, \dots, 0]$	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_{14}	$[420.96, 420.96, \dots, 420.96]$	0	$[-500, 500]^D$	$[-500, 500]^D$
f_{15}	Predefined rand number distributed in the search range	0	$[-500, 500]^D$	$[-5, 5]^D$
f_{16}	Predefined rand number distributed in the search range	0	$[-500, 500]^D$	$[-5, 5]^D$

Table 2

Parameter tuning of sub-population size (SP).

SP	3 particles	4 particles	5 particles	6 particles	7 particles
f_3	2.7844e–15	2.9425e–15	2.5847e–15	2.8451e–15	5.2118e–15
f_4	2.8958e–03	2.8417e–03	1.2389e–03	3.8145e–03	1.6541e–02
f_5	0	0	0	0	0
f_9	2.9606e–15	3.2518e–15	3.3159e–15	5.8110e–15	6.8887e–15
f_{10}	4.1842e–02	3.9254e–02	1.9966e–02	7.1814e–02	9.4799e–02
f_{11}	5.9698e–00	0	0	3.0029e–00	4.5151e–00

Table 3

Parameter tuning of the regrouping period (R).

R	3 iterations	4 iterations	5 iterations	6 iterations	7 iterations
f_3	3.1974e–15	3.1974e–15	2.5847e–15	3.3159e–15	3.3159e–15
f_4	1.2145e–02	1.1412e–02	1.2389e–03	1.3376e–02	1.3702e–02
f_5	0	0	0	0	0
f_9	3.3159e–15	3.1974e–15	3.3159e–15	3.1974e–15	3.1974e–15
f_{10}	2.2805e–02	2.1011e–02	1.9966e–02	1.8709e–02	2.1990e–02
f_{11}	4.0333e–00	4.0957e–00	0	4.0000e+00	3.6000e–00

$$bw(t) = bw_{\max} e^{\left(\frac{\ln \left(\frac{bw_{\min}}{bw_{\max}} \right)}{-\frac{Max_gen}{t}} \times t \right)} \quad (21)$$

where the $PAR(t)$ is the pitch adjusting rate for generation t , PAR_{\min} is the minimum adjusting rate, PAR_{\max} is the maximum adjusting rate and t is the generation number; $bw(t)$ is the bandwidth for generation t , bw_{\min} is the minimum bandwidth and bw_{\max} is the maximum bandwidth, Max_gen is total number of generations.

3.3. Experimental study

For each problem, the DMS-PSO-HS, the DMS-PSO and the HS are run 30 times. The maximum function evaluations Max_FEs are set at 100,000 for 10-D, and 200,000 for 30-D. The computer

system is Windows XP (SP1) with Pentium (R) 4 3.00 GHz CPU, 2 GB RAM running the Matlab 7.1. For each function, we present the Mean of the 30 runs in Tables 4 and 5. The best mean value achieved for each test problem is shown in bold. In order to determine whether the results obtained by the DMS-PSO-HS are statistically different from the results generated by the DMS-PSO and the HS, the h values obtained by the t -tests are presented in Tables 4 and 5. An h value of 1 indicates that the performance of the DMS-PSO-HS is statistically superior with 95% certainty, whereas an h value of 0 implies that the performances are not statistically different.

From Tables 4 and 5, we can easily find that the DMS-PSO-HS always achieved better results on all four multimodal groups. According to the results of t -tests, DMS-PSO-HS significantly improves the results on almost all the test problems especially on the difficult multimodal problems in groups 2, 3 and 4. On the two composition functions with randomly distributed local and global optima, DMS-PSO-HS performs much better on both 10 and 30 dimensional versions.

For the first function in group 1, as well as the relatively easier multimodal functions f_3 , f_6 and f_7 with 10 dimensions in group 2, all the three approaches can solve them if we apply the $1e-4$ as the 'value to reach' success criterion, while the HS performs worst among them. On the more difficult test problems f_2 , f_4 , f_5 , f_8 in group 2 with 10 dimensions, HS is almost yielding the worst performance except the f_8 on which HS performs better than the DMS-PSO, while DMS-PSO-HS still performs superior on the two groups. According to the results of the rotated functions with 10 dimensions in group 3, the DMS-PSO-HS performs much better than the original HS. Since the original HS cannot solve these difficult test problems, the proposed Hybrid algorithm takes the advantage of the DMS-PSO and performs slightly better than the DMS-PSO. On the two most difficult test problems in group 4, the DMS-PSO-HS performs significantly better. On the 30 dimensional test functions, we could observe that the performance of the DMS-PSO deteriorates on functions f_6 , f_7 and f_8 in group 2 when compared with the 10 dimensional

Table 4

Results of 10-D test problems.

Algorithms	Func			
	Group 1 f_1	Group 1 f_2	Group 2 f_3	Group 2 f_4
DMS-PSO	5.6941e–100	1.4986e–00	8.9514e–14	1.6456e–03
HS	2.5709e–09	3.0398e–00	1.8476e–05	6.5027e–02
DMS-PSO-HS	1.9541e–102	9.4626e–02	2.5847e–15	1.2389e–03
h	1	1	1	1
	Group 2 f_5	Group 2 f_6	Group 2 f_7	Group 2 f_8
DMS-PSO	0	0	0	1.7384e+02
HS	2.8941e–01	1.0918e–09	1.4175e–09	9.4118e–09
DMS-PSO-HS	0	0	0	0
h	0	0	0	1
	Group 3 f_9	Group 3 f_{10}	Group 3 f_{11}	Group 3 f_{12}
DMS-PSO	3.3159e–15	1.9966e–02	0	3.2289e–00
HS	9.4486e–01	1.5422e–01	3.4432e–00	2.0198e+01
DMS-PSO-HS	3.3190e–15	1.9038e–02	0	3.0512e–00
h	0	1	0	1
	Group 3 f_{13}	Group 3 f_{14}	Group 4 f_{15}	Group 4 f_{16}
DMS-PSO	4.3225e–00	3.6628e+02	1.3338e+01	1.6485e+01
HS	1.2533e+01	1.0648e+03	1.7333e+02	2.5391e+02
DMS-PSO-HS	4.1333e–00	3.5394e+02	6.6667e–00	1.0518e+01
h	1	1	1	1

Table 5

Results of 30-D test problems.

Algorithms	Func			
	Group 1 f_1	Group 1 f_2	Group 2 f_3	Group 2 f_4
DMS-PSO	1.2658e–66	1.9828e+01	6.5178e–14	0
HS	9.6429e–08	5.1963e+01	7.7705e–05	2.8834e–02
DMS-PSO-HS	3.6514e–78	1.7023e+01	3.2145e–15	0
h	1	1	1	0
	Group 2 f_5	Group 2 f_6	Group 2 f_7	Group 2 f_8
DMS-PSO	0	1.4014e+01	1.8754e+01	1.7282e+03
HS	7.6028e–01	5.4821e–08	5.1302e–08	3.0365e–01
DMS-PSO-HS	0	0	0	9.9496e–02
h	0	1	1	1
	Group 3 f_9	Group 3 f_{10}	Group 3 f_{11}	Group 3 f_{12}
DMS-PSO	4.8554e–15	4.8307e–04	1.1058e–02	2.8101e+01
HS	1.3461e–00	2.3159e–02	1.1471e+01	7.9255e+01
DMS-PSO-HS	4.3817e–15	4.3147e–04	1.1086e–02	1.9269e+01
h	1	1	0	1
	Group 3 f_{13}	Group 3 f_{14}	Group 4 f_{15}	Group 4 f_{16}
DMS-PSO	3.2051e+01	3.2048e+03	1.3333e+01	1.9840e+01
HS	5.1067e+01	3.1524e+03	1.5667e+02	1.5861e+02
DMS-PSO-HS	1.7967e+01	1.9558e+03	2.4542e–31	4.4920e–00
h	1	1	1	1

cases. However, the HS performs robustly on these functions even when the number of dimensions is increased from 10 to 30, which are much better than the original DMS-PSO. As the hybrid algorithm, the DMS-PSO-HS can yield statistically superior performance on most functions, especially on f_6 , f_7 , f_8 , f_{15} and f_{16} . Hence, the DMS-PSO-HS can fully exert its comprehensive search ability benefited from the HS and DMS-PSO when solving hard optimization problems.

By analyzing the results on 10-D and 30-D problems, one can conclude that the DMS-PSO-HS benefits from both the DMS-PSO and the HS algorithms by integrating the faster convergent speed of the HS as well as the stronger exploration ability of the DMS-PSO to tackle diverse problems in four groups. Therefore, it performs significantly better than the two constituent approaches. Furthermore, most results are markedly improved by the proposed DMS-PSO-HS. The DMS-PSO-HS can perform robustly with respect to scaling of the dimensions, rotation and composition of difficult multimodal test problems.

4. Conclusions

This paper proposes a hybridization of dynamic multi-swarm particle swarm optimizer (DMS-PSO) with the harmony search (DMS-PSO-HS). Using the original configuration of the DMS-PSO, we periodically generate the new harmonies based on the current $pbests$ in each sub-swarm after PSOs' positions have been updated. The nearest $pbest$ is replaced by the new harmony if the new harmony vector has better fitness. The DMS-PSO-HS attempts to take merits of the PSO and the HS in order to avoid all particles getting trapped in inferior local optimal regions. The DMS-PSO-HS enables the particles to have more diverse exemplars to learn from as we frequently regroup the sub-swarms and also form new harmonies to search in a larger search space. From the analysis of the experimental results, we observe that the proposed DMS-PSO-HS makes good use of the information in past solutions more effectively to generate better quality solutions frequently when compared to the DMS-PSO and the HS. Based on the results of the three approaches on the test problems belonging to four classes, we can conclude that the DMS-PSO-HS significantly improves the performances of the HS and the DMS-PSO on most multimodal and unimodal problems. The novel configuration of DMS-PSO-HS does not introduce additional complex operations beyond the original DMS-PSO and HS. In fact, the DMS-PSO-HS eliminates the parameters in the original HS, which normally need to be adjusted according to the properties of the test problems. In addition, the DMS-PSO-HS is simple and easy to implement.

Acknowledgement

Authors acknowledge the financial support offered by the A*Star (Agency for Science, Technology and Research, Singapore) under the grant #052 101 0020.

References

- Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Applications*, 6, 76–90.
- Clerc, M., & Kennedy, J. (2002). The particle swarm – explosion stability and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73.
- Davidon, W. C. (1959). Variable metric method for minimization. A.E.C. Research and Development Report, ANL-5990.
- Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micromachine and human science*, Nagoya, Japan (pp. 39–43).
- Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer Journal*, 13, 317–322.
- Fletcher, R., & Powell, M. J. D. (1963). A rapidly convergent descent method for minimization. *Computer Journal*, 6, 163–168.
- Geem, Z. W. (2009). Particle-swarm harmony search for water network design. *Engineering Optimization*, 41, 297–311.
- Goldfarb, D. (1970). A family of variable metric updates derived by variational means. *Mathematics of Computing*, 24, 23–26.
- Kennedy, J. (1999). Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of IEEE congress on evolutionary computation (CEC 1999)*, Piscataway, NJ (pp. 1931–1938).
- Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, Piscataway, NJ (pp. 1942–1948).
- Lee, K., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 194, 3902–3933.
- Liang, J. J., & Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer. In *Proceedings of IEEE international swarm intelligence symposium* (pp. 124–129).
- Liang, J. J., Suganthan, P. N., & Deb, K. (2005). Novel composition test functions for numerical global optimization. In *Proceedings of Swarm Intelligence Symposium*.
- Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3), 281–295.
- Omran, M. G. H., & Mahdavi, M. (2008). Global-best harmony search. *Applied Mathematics and Computation*, 198(2), 643–656.
- Shanno, D. F. (1970). Conditioning of Quasi-Newton methods for function minimization. *Mathematics of Computing*, 24, 647–656.
- Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. -P., & Auger, A., et al. (2005). *Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization*, Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report #2005005, IIT Kanpur, India. <<http://www.ntu.edu.sg/home/EPNSugan>>.
- van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*(3), 225–239.
- Young, M. (1989). *The Technical writers handbook*. Mill Valley, CA.: University Science.
- Zielinski, K., & Laur, R. (2007). Stopping criteria for a constrained single-objective particle swarm optimization algorithm. *Informatica*, 0350-5596, 31(1), 51–59.