Principal Component Particle Swarm Optimization

A Step Towards Topological Swarm Intelligence

Mark S. Voss, Ph.D.
Prediction Engineering
2260 Par Lane, Suite 806
Willoughby Hills, OH 44094
MarkVoss@PredictionEngineering.com

ABSTRACT

Particle Swarm Optimization (PSO) is based on the notion of particles flying through solution space. Each particle is assumed to have n-dimensions that are mapped to the variables of the function that is being evaluated. The standard PSO algorithm updates a particle's position by moving towards the particle's past personal best and the best particle that has been found. This paper introduces the Principal Component Particle Swarm Optimization (PCPSO) procedure. The Principal Component Particle Swarm Optimization procedure flies the particles in two separates spaces at the same time; the traditional n-dimensional x space and a rotated m-dimensional z space where $m \leq n$.

Categories and Subject Descriptors

G.1.6 [**Global Optimization**]

General Terms

Swarm, Algorithms

Keywords

Particle Swarm Optimization, Principal Component Analysis, Topological Vector Space

1. INTRODUCTION

Particle Swarm Optimization (PSO) [1] [2] was introduced as an optimization methodology based on a social psychological metaphor. The standard PSO is based on an analogy with particles flying through solution space whereby the particle locations are mapped from a fixed coordinate system. Tracking the particles from a fixed location is known as the Eulerian point of view. Another possibility is to map the particles from a coordinate system that moves with the swarm. When using this perspective the particles are first referenced by their local coordinate system which is mapped back to a fixed system. This two stage mapping is known as the Lagrangian point of view [3]. This paper introduces one methodology for creating and *flying a dynamic coordinate system with the particles.* Before discussing the specifics of this particular implementation it should be noted that topological psychology is a field that has a long history of being theoretically advanced beyond its numerical implementation [4]. Lewin's field theory is discussed in [5], where the question is asked:

> We can only guess, if Lewin had a computer, how he could have demonstrated his vision in ways that others could really understand...

The following quotes are taken directly from Kurt Lewin's books [6] [7] [8] written between 50 and 70 years ago. From Kurt Lewin's book *Field Theory in Social Sciences – 1951*:

> Obviously, the state of development of psychology is not such that a systematic linking of every construct with any other by a system of quantitative equations can be realized. On the other hand, I am inclined to think that psychology is not far away from a level where a good number of the basic constructs can be linked in a precise manner [6].

Lewin also refers to Thurstone's factor analysis [9] as a useful device for finding relations between the factors that influence an event[7]. In an earlier work Lewin states:

> That there is a direct relationship between the momentary state of the individual and the structure of his psychological environment.
>
> These imperative environmental facts – we shall call them valences [*Aufforderungscharakter*] – determine the direction of behavior.
>
> The valences change with the momentary state of the needs [8].

It is seen that Lewin anticipated much of the mathematics and many of the social–psychological constructs that comprise the Principal Component Particle Swarm Optimization methodology[10]. He surely would have implemented something similar given a modern computer.

2. PRINCIPAL COMPONENT ANALYSIS

The goal of Principal Component Analysis (PCA) is to transform a set of correlated variables x_i into a set of minimally correlated variables z_i. This is accomplished by selecting a set of orthonormal basis vectors u_i [11] that are

Genetic and Evolutionary Computation Conference (GECCO)'05,
June 25-29, 2005, Washington, DC, USA.

used to define the new variables as linear combinations of the original variables. A weighted covariance matrix is used to calculate the orthonormal basis vectors u_i and is defined as:

$$S = \begin{bmatrix} s_1^2 & s_{12} & \cdots & s_{1n} \\ s_{12} & s_2^2 & \cdots & s_{2n} \\ \vdots & \vdots & & \vdots \\ s_{1n} & s_{2n} & \cdots & s_n^2 \end{bmatrix} \tag{1}$$

$$s_{jk} = \frac{\sum_{i=1}^{N} W_i \left(x_{ji} - \bar{x}_j\right)\left(x_{ki} - \bar{x}_k\right)}{n-1} \tag{2}$$

where n is the number of dimensions, N is the number of points defining the covariance matrix, and W_i is the weight given to point i. W_i is defined as a function of the iteration/generation k.

$$W_i = \left(\frac{k}{k_{\max}}\right)^{\lambda} \tag{3}$$

λ allows for non-linear control over the applied weights.

In order to fly the particles in a rotated z space it is necessary to span the z space by a matrix of orthonormal eigenvectors U, where U is implicitly defined as:

$$U'SU = L \tag{4}$$

The matrix U contains the eigenvectors as columns:

$$U = [u_1|u_2|\cdots|u_n] \tag{5}$$

The u_i eigenvectors are orthonormal:

$$\begin{aligned} u_i' u_i &= 1 \\ u_i' u_j &= 0 \end{aligned} \tag{6}$$

Matrix L is a diagonal matrix containing the eigenvalues l_i.

$$L = \begin{bmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ & & \ddots & \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & l_n \end{bmatrix} \tag{7}$$

The principal components are then defined as:

$$z = U'\left[x - \bar{x}\right] \tag{8}$$

Each z_i space variable (principal component) is formed by a linear combination of the original x space variables and has a mean of zero and a variance l_i. This is written in terms of the eigenvector u_i and the centered vector $(x - \bar{x})$ as:

$$z_{i\in\{1\ldots n\}} = u_i'\left[x - \bar{x}\right] \tag{9}$$

The reverse operation is equally valid. Each particle in the z space can be mapped back to the x space using the equation:

$$x = \bar{x} + Uz \tag{10}$$

A reduced dimension $\tilde{z}$ space ($\sim$ representing a dimensional contraction) can be defined as:

$$\tilde{z} = \tilde{U}'\left[x - \bar{x}\right] \tag{11}$$

where $\tilde{U}$ is defined using the m eigenvectors associated with the largest $m \leq n$ eigenvalues. This can also be written for each variable $\tilde{z}_j$ as:

$$\tilde{z}_{j\in\{1\ldots m\}} = \tilde{u}_j'\left[x - \bar{x}\right] \tag{12}$$

Again, the z space can be mapped back to the x space by the equation:

$$x = \bar{x} + \tilde{U}\tilde{z} + (x - \hat{x}) \tag{13}$$

where $\hat{x}$ is the residual.

3. PARTICLE SWARM OPTIMIZATION

The velocity weighting parameter w_k is defined as:

$$w_k = w_1 - (w_1 - w_2)\left(\frac{k}{k_{\max}}\right) \tag{14}$$

where k is the current generation/iteration and $k_{\max}$ is the maximum number of iterations for the current run. w_1 and w_2 are therefore the starting and stopping values for w_k respectively.

The particle velocity is updated according to:

$$v_{ij}^{k+1} = w_k v_{ij}^k + \varphi_{1_{ij}}\left(x_{ig}^k - \hat{x}_{ij}^k\right) + \varphi_{2_{ij}}\left(x_{il}^k - \hat{x}_{ij}^k\right) \tag{15}$$

where $i{\leq}n$ is the x dimension index, j is the particle index, g designates the best particle, and l designates the personal best particle i. $\varphi_{1_{ij}}$ and $\varphi_{2_{ij}}$ are uniform random numbers between 0 and 2. That is, x_{ig}^k represents the i^{th} dimension of the best particle (solution) found during the last k iterations. The $\wedge$ represents a possible $\tilde{z}$ space contribution as defined by equation 24. The x space particle positions are updated as follows:

$$x_{ij}^{k+1} = v_{ij}^{k+1} + \hat{x}_{ij}^k \tag{16}$$

4. PRINCIPAL COMPONENT PARTICLE SWARM OPTIMIZATION

The z space velocity weighting parameter $\tilde{w}_k$ is defined as:

$$\tilde{w}_{\mathrm{k}} = \tilde{w}_1 - (\tilde{w}_1 - \tilde{w}_2)\left(\frac{k}{k_{\max}}\right) \tag{17}$$

The z space particle velocity is updated according to:

$$\tilde{v}_{ij}^{k+1} = \tilde{w}_k \tilde{v}_{ij}^k + \tilde{\varphi}_{1_{ij}}\left(\tilde{z}_{ig}^k - \tilde{z}_{ij}^k\right) + \tilde{\varphi}_{2_{ij}}\left(\tilde{z}_{il}^k - \tilde{z}_{ij}^k\right) \tag{18}$$

where $i \leq m \leq n$ is the $\tilde{z}$ dimension index, j is the particle index, g designates the best particle, and l designates the personal best particle i. $\tilde{\varphi}_{1_{ij}}$ and $\tilde{\varphi}_{2_{ij}}$ are uniform random numbers between 0 and 2. That is, $\tilde{z}_{ig}^k$ represents the i^{th} *transformed dimension* of the best particle (solution) found during the last k iterations.

If the principal components have been recalculated since the last $\tilde{v}$ calculation then $\tilde{z}$ and $\tilde{v}$ are re-initialized before $\tilde{v}_{ij}^{k+1}$ is updated according to:

$$\tilde{z} = \tilde{U}'\left[x - \bar{x}\right] \tag{19}$$

$$\tilde{v} = \tilde{U}'\left[v\right] \tag{20}$$

where x and v are the current x space particle locations and velocities respectively.

Figure 1 illustrates how the principal axes move with the swarm for a 2-dimensional problem. The smaller dots represent the global best locations found so far at time t_1. The larger dots represent the global best solutions found since time t_1 up to time t_2, where $t_2 > t_1$. Having the coordinate

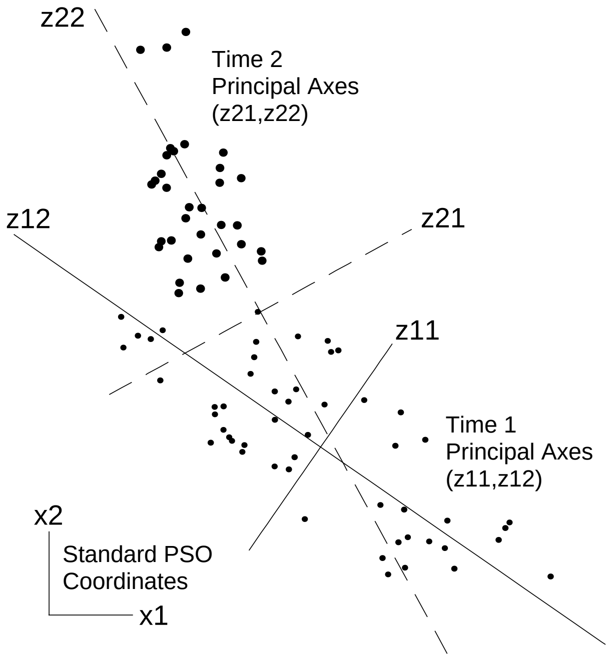


Figure 1: Principal Component Recalculation

system follow the swarm can seen as a form of adaptive Lagrangian swarm coordinates.

The z space particle positions are updated as follows:

$$\tilde{z}_{ij}^{k+1} = \tilde{v}_{ij}^{k+1} + \tilde{z}_{ij}^{k} \tag{21}$$

The updated z space locations are mapped back to x space using the partial orthonormal basis $\tilde{U}$, where $\tilde{U}$ contains the first m eigenvectors of U corresponding to the m largest eigenvalues. When m equals the dimension of the x space, $m = n$ resulting in $\tilde{U} = U$.

$$\tilde{x}_{ij}^{k+1} = \bar{x}_{ij}^{k+1} + \tilde{U}\left[\tilde{z}_{ij}^{k+1}\right] \tag{22}$$

The fraction of the $\tilde{z}$ space flight that is included is defined by the parameter β:

$$\beta_k = \alpha_1 - (\alpha_1 - \alpha_2)\left(\frac{k}{k_{\max}}\right) \tag{23}$$

where α_1 and α_2 are the starting and stopping values for β_k respectively.

The z and x space flight components are combined using the β parameter to define the new x space particle position:

$$\hat{x}_{ij}^{k+1} = \beta_k\left(\tilde{x}_{ij}^{k+1}\right) + (1-\beta_k)\left(x_{ij}^{k+1}\right) \tag{24}$$

When $\beta_k = 0$:

$$\hat{x}_{ij}^{k+1} = x_{ij}^{k+1} \tag{25}$$

That is:

$$(\beta_k \rightarrow 0) \Rightarrow (PCPSO \rightarrow PSO) \tag{26}$$

4.1 PCPSO algorithm

1. Initialization
   (a) Initialize swarm
   (b) Initialize covariance matrix (1-3)
   (c) Calculate principal components (4)
   (d) Map particles to Z space (8-12)
2. Fly Particles
   (a) Fly particles in X space (14-16)
   (b) Fly particles in Z space (17-21)
      i. Map new Z locations to X space (22)
3. Form weighted average of 2.a and 2.b.i (23-24)
4. New personal best
   (a) Update Pbest
   (b) Pbest covariance updating:
      i. Update covariance matrix using the new weighted Pbest location (1-3)
      ii. Recalculate principal components (4)
      iii. Map current X space velocities and locations to Z space (19-20)
5. New global best
   (a) Update Gbest
   (b) Gbest covariance updating:
      i. Update covariance matrix using the new weighted Gbest location (1-3)
      ii. Recalculate principal components (4)
      iii. Map current X space velocities and locations to Z space (19-20)
6. Fly Particles Again

Equation numbers in parentheses.

5. COMPARISON STUDY

A study comparing the PCPSO algorithm with previously published results [12] [13] [14] [15] was performed using 30 dimensional Sphere, Griewank, generalized Rosebrock and Rastrigin functions (f_1, f_2, f_3 and f_4) respectively.

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{27}$$

$$f_2(x) = \frac{1}{4000}\sum_{i=1}^{n-1}(x_i - 100)^2 - \prod_{i=1}^{n} Cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1 \tag{28}$$

$$f_3(x) = \sum_{i=1}^{n-1} 100\left(x_{i+1} - x_i^2\right)^2 - (x_i - 1)^2 \tag{29}$$

$$f_4(x) = \sum_{i=1}^{n}\left(x_i^2 - 10Cos\left(2\pi x_i\right) + 10\right) \tag{30}$$

30 Dimensions, 20 Particles, 2000 Iterations, 100 Runs		
PSO: $w_1 = .7$, $w_2 = .4$, $\phi \in \{0 \ldots 2\}$ PCPSO: $\tilde{w}_1 = .7$, $\tilde{w}_2 = .4$, $\tilde{\phi} \in \{0 \ldots 2\}$ PCPSO: $\beta = .3$, $m = n = 30$		
f_1 – Sphere		
Initial Range $\{50, 100\}$	Fitness	σ
$PSO^{\dagger}$	7.06E-13	2.00E-12
$PCPSO^{\dagger}$ ($\lambda = .5$)	1.00E-13	5.31E-14
$PCPSO^{\dagger}$ ($\lambda = 1.$)	1.51E-14	9.99E-15
PSO[12]	6.29E-13	7.64E-14
$Hybrid$[12]	0.012	6.3E-4
f_2 – Griewank ($V_{max} = 600.$)		
Initial Range $\{300, 600\}$	Fitness	σ
$PSO^{\dagger}$	0.0120	0.0197
$PCPSO^{\dagger}$ ($\lambda = .5$)	0.0076	0.0109
$PCPSO^{\dagger}$ ($\lambda = 1.$)	0.0055	0.0116
$PSO^{\ddagger}$	0.0167	0.0127
$PCPSO^{\ddagger}$ ($\lambda = .5$)	0.0075	0.0106
$PCPSO^{\ddagger}$ ($\lambda = 1.$)	0.0073	0.0118
PSO[13]	0.0182	NA
PSO[12]	0.0150	0.0024
$Hybrid$[12]	0.0991	0.0011
$FPSO$ [14]	0.0216	NA
$HPSO_1$[15]	0.0157	NA
f_3 – Rosenbrock ($V_{max} = 30.$)		
Initial Range $\{15, 30\}$	Fitness	σ
$PSO^{\dagger}$	276	518
$PCPSO^{\dagger}$ ($\lambda = .5$)	126	180
$PCPSO^{\dagger}$ ($\lambda = 1.$)	142	224
$PSO^{\ddagger}$	115	198
$PCPSO^{\ddagger}$ ($\lambda = .5$)	121	155
$PCPSO^{\ddagger}$ ($\lambda = 1.$)	85	144
PSO[13]	316	NA
PSO[12]	154	25
$Hybrid$[12]	187	23
$FPSO$ [14]	184	NA
$HPSO_2$[15]	128	NA
f_3 – Rastrigin ($V_{max} = 5.12$)		
Initial Range $\{2.56, 5.12\}$	Fitness	σ
$PSO^{\dagger}$	76	20
$PCPSO^{\dagger}$ ($\lambda = .5$)	155	54
$PCPSO^{\dagger}$ ($\lambda = 1.$)	154	47
$PSO^{\ddagger}$	47	13
$PCPSO^{\ddagger}$ ($\lambda = .5$)	113	61
$PCPSO^{\ddagger}$ ($\lambda = 1.$)	123	61
PSO[13]	47	NA
PSO[12]	47	1.3
$Hybrid$[12]	27.8	0.81
$FPSO$ [14]	48.5	NA
$HPSO_1$[15]	35.6	NA
†: (this study): $V_{max} = \infty$ ‡: (this study): V_{max} = specified limit		

Table 1: PSPSO Comparison Study

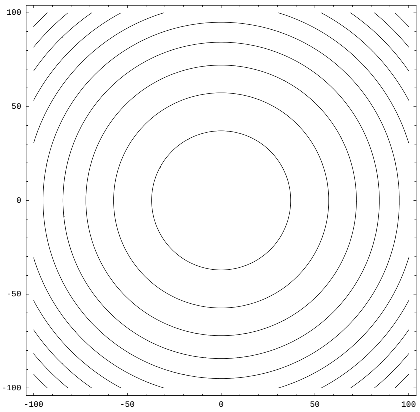


Figure 2: Rastrigin Contour Plot $\{x, y\} \in \{-100, 100\}$

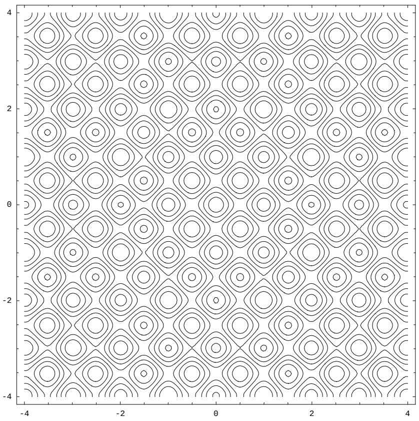


Figure 3: Rastrigin Contour Plot $\{x, y\} \in \{-4, 4\}$

The settings for the PSO and PCPSO are shown at the top of table 1. The averages and standard deviations for this study were calculated using 100 runs. $PSO^{\ddagger}$ and $PCPSO^{\ddagger}$ represent runs that enforced a V_{max} velocity limit as specified in table 1. The PSO algorithm did not employ a constriction factor χ [16]. The PCPSO was run with two different covariance matrix weighting exponents ($\lambda = .5$ and $\lambda = 1.$) as described in equation 3. The PCPSO used a full set of principal components ($m = n = 30$). On each iteration the z space particle location is combined with the x space location using a constant beta factor ($\beta = .3$). The values of λ and β were not optimized. The reported values for the PCPSO are reasonable when compared with the previous studies. The PCPSO solutions were less than the previously published results for the Sphere, Griewank and Rosenbrock functions without the inclusion of a maximum velocity(V_{max}). The results improved with the inclusion of a maximum velocity.

The PCPSO did not perform as well on the Rastrigin function (f_4). Figures 2 and 3 demonstrate the topology of the Rastrigin function. From a distance the Rastrigin function appears as a type of Sphere function. This is illustrated by the contour plot in Figure 2. As the swarm approaches the bottom of the bowl the Rastrigin takes on a dimple topology

Config	Gen	Rotated Space Contribution	Principal Components
1	500	$\beta = .0$	NA
2	1000	$\beta = .0$	NA
3	500	$\beta \in \{.3 \rightarrow .0\}$	$m = \frac{n}{10} = 10$
4	1000	$\beta \in \{.3 \rightarrow .0\}$	$m = \frac{n}{10} = 10$
5	500	$\beta = .3$	$m = n = 100$
6	1000	$\beta = .3$	$m = n = 100$

Table 2: 100 Dimension, 30 Particle – Configurations

Config	f_1	f_2	f_3	$f_4^{\#1}$	$f_4^{\#2}$
1	88364	591	6.8E+8	1334	2.39E5
2	22521	165	2.8E+8	968	2.35E5
3	2305	13.8	1.6E+5	1142	2519
4	18.5	1.09	3.3E+3	1024	953
5	2074	1.43	1.0E+4	1076	1282
6	13.1	.0211	937	1136	840
2^*	620	6.43	7.9E+5	473	2031
6^*	3.9E-5	.0078	887	638	798
Config*: V_{max} = as defined in Table 1					
$\left(\frac{2^*}{6^*}\right)$	**1.6E7**	**824**	**891**	**.71**	**2.6**
$\left(\frac{2^*}{6}\right)$	**47**	**305**	**843**	**.42**	**2.4**

Table 3: 100 Dimension, 30 Particle – Study Results

as shown in Figure 3. The dimples are lined up with the standard Rastrigin coordinates at regular intervals.

It is this type of topology where a rotated coordinate system does not have any beneficial effect and actually decreases the probability of a particle landing on the global optimum. In this case there is no linkage[17] to learn.

6. PCPSO – HIGH DIMENSIONAL STUDY

A primary motive for developing the PCPSO algorithm was to increase the convergence of the PSO on high dimensional problems. A study was performed using 100 dimensional Sphere, Griewank, generalized Rosebrock and Rastrigin functions (f_1, f_2, f_3 and f_4) respectively. The six configurations investigated are shown in table 2. The other settings (with the exception of β) are the same as those given in table 1. Configurations 1 and 2 represent the standard PSO. Special cases of configurations 2 and 6 (2^* and 6^*) investigate the inclusion of a maximum velocity (V_{max}).

When the number of principal components is less than the dimensionality of the problem ($m < n$) the rotated space will not be able to find an exact answer. One solution to this is to reduce the rotated space contribution with increasing iterations as in configurations 3 and 4. The motivation for this approach is to use the principal component information (via combined particle movement) early in the run (large β) to accelerate convergence, while transitioning to a standard PSO at the end of the run (small β). Convergence is enhanced by using the principal component algorithm during early generations to identify particle gradient information.

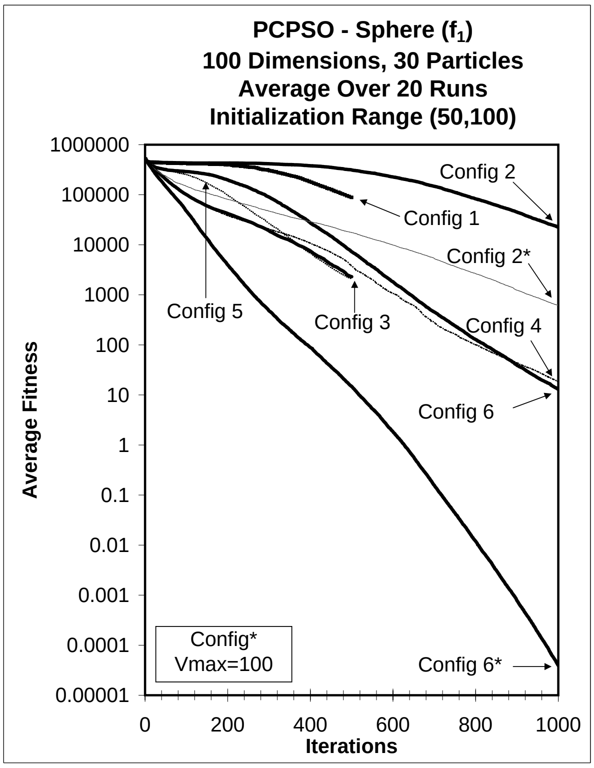


Figure 4: Sphere Study (f_1)

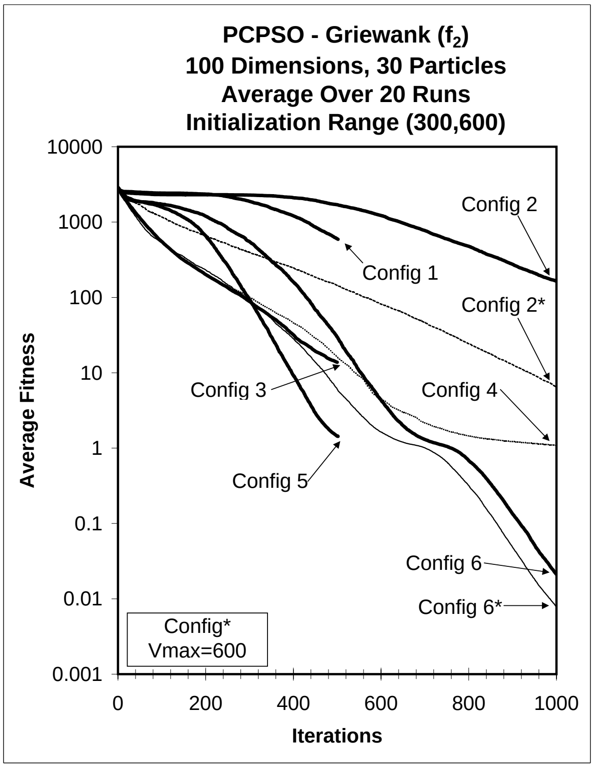
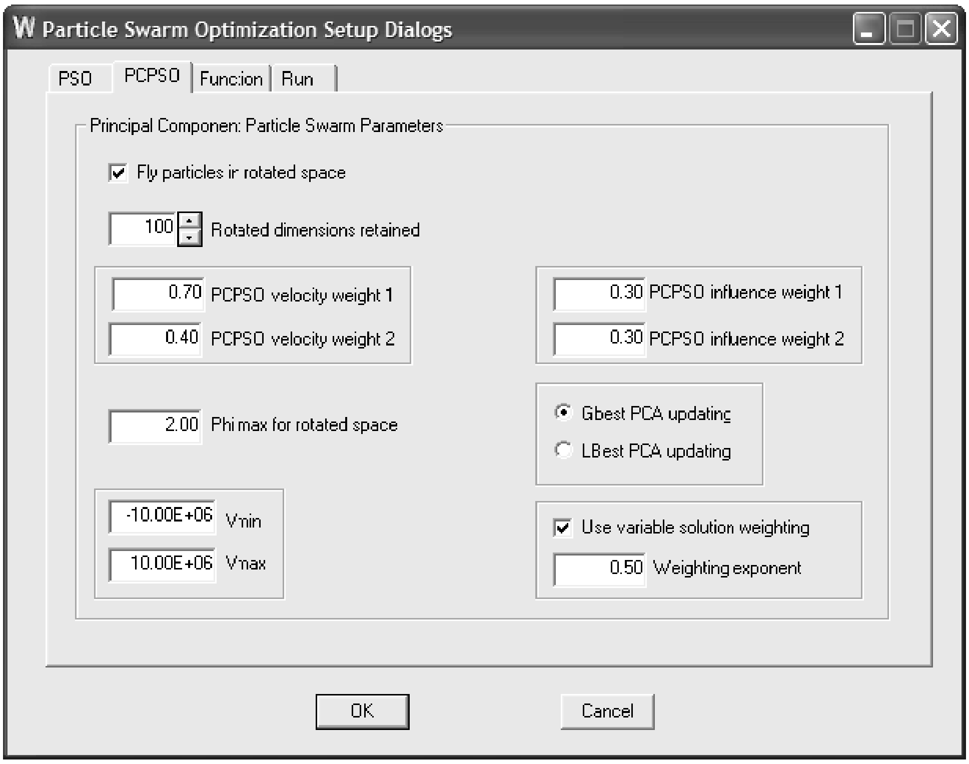


Figure 5: Griewank Study (f_2)



Principal Componen: Particle Swarm Parameters

☑ Fly particles ir rotated space

100 Rotated dimensions retained

0.70 PCPSO velocity weight 1

0.40 PCPSO velocity weight 2

0.30 PCPSO influence weight 1

0.30 PCPSO influence weight 2

2.00 Phi max for rotated space

● Gbest PCA updatinc

○ LBest PCA updating

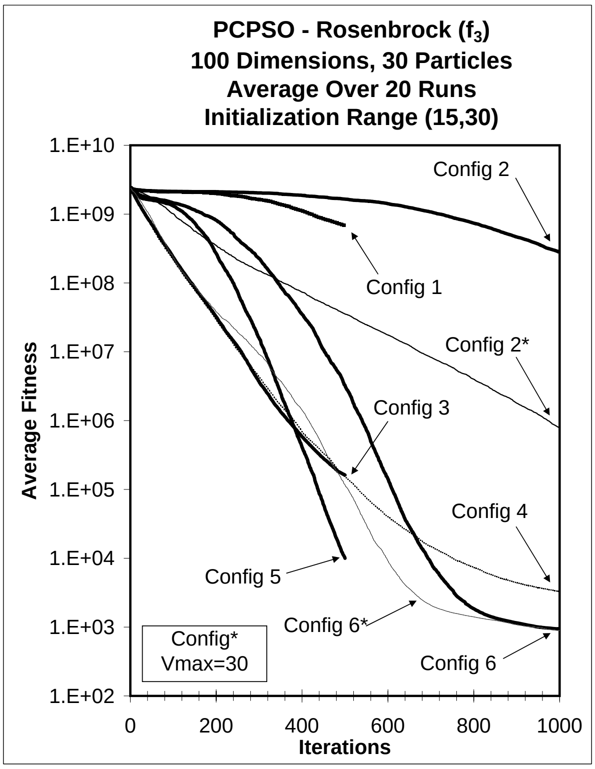
-10.00E+06 Vnin

10.00E+06 Vnax

☑ Use variable solution weighting

0.50 Weighting exponent

Figure 10: PCPSO parameter dialog



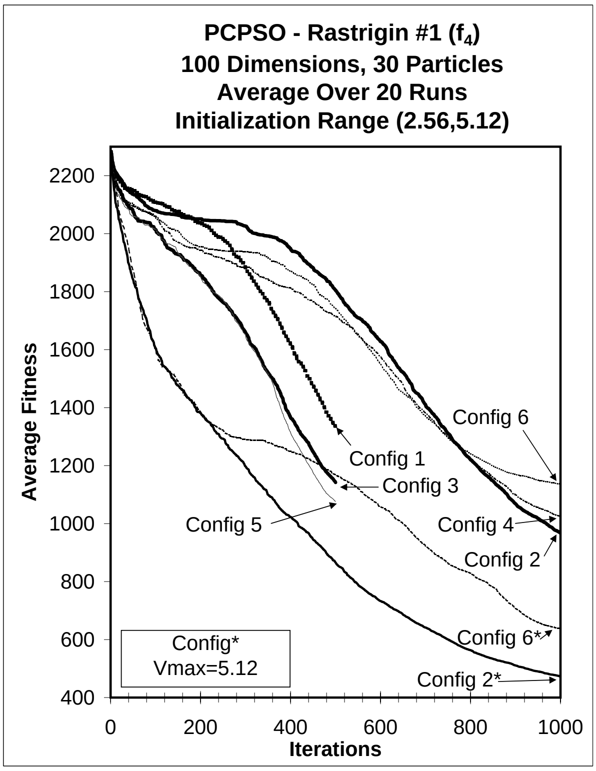


Figure 6: Rosenbrock Study (f_3)

Figure 7: Rastrigin Study $f_4^{\#1}$

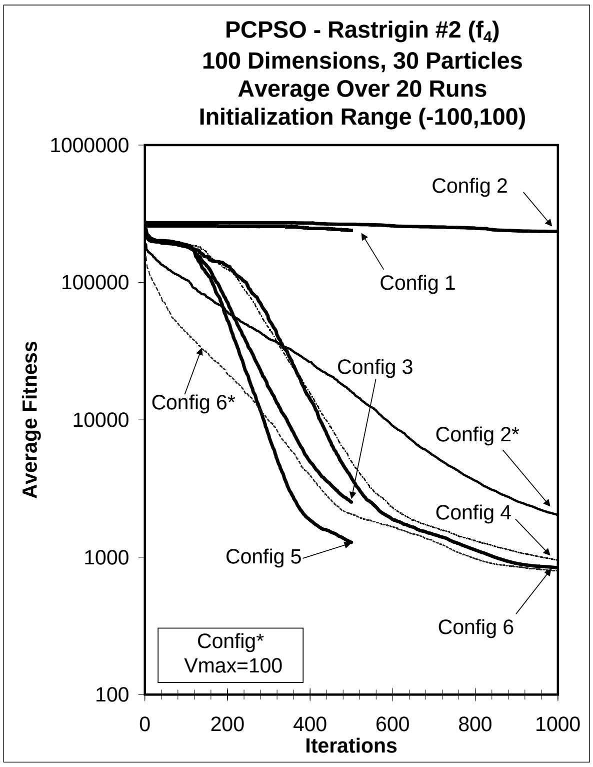


Figure 8: Rastrigin Study $f_4^{\#2}$

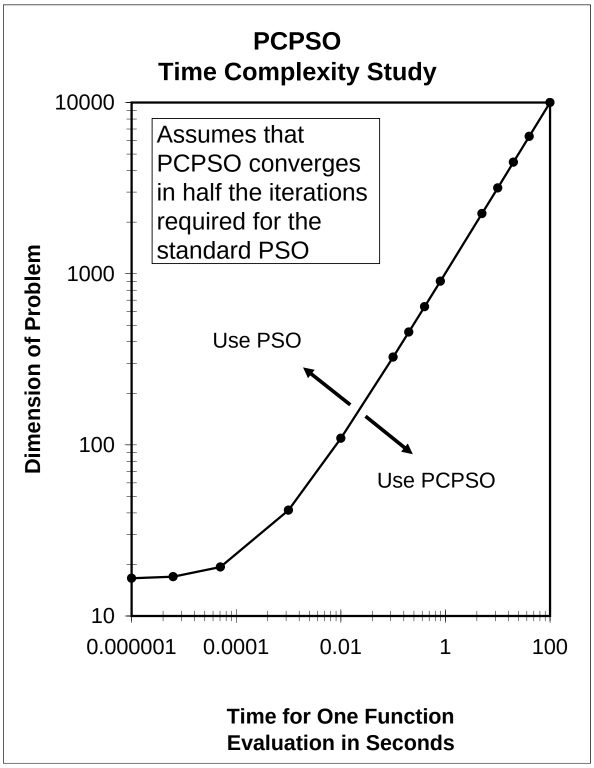


Figure 9: PCPSO and PSO time complexity

In configurations 5 and 6, the full set of principal components is used throughout the run. That is, the dimension of the rotated space is maintained at $m = n = 100$. The particle flight from the rotated space and the traditional space were combined throughout the run ($\beta = .3$). An option for associating a generationally dependent weight with each update to the covariance matrix was included in the PCPSO algorithm (configurations 3-6 used equation 3 with $\lambda = .5$). The current generation is divided by the max generation and this fraction is raised to a weighting exponent. Higher weights tend to attract the eigenvectors that fill out the z space; which has the effect of making the current direction the particles are flying in more important than past directions. *With respect to the particle swarm social psychological metaphor, this has the effect of making current ideas more important than past ideas.*

Configurations 3-6 used global updating for the covariance matrix. The developed program also implements local updating (figure 10, LBest PCA updating) of the covariance matrix resulting in each particle having input into the *rotation of the life space*[18]. A GUI program was developed for implementing the PCPSO algorithm. Figure 10 shows the settings for the PSPSO parameters for configuration's 5 and 6.

6.1 PCPSO – High Dimensional Results

The results of the high dimensional study are given in table 3. Figures 4, 5 and 6 illustrate the improved convergence characteristics of the PCPSO algorithm on the Sphere, Griewank and Rosenbrock functions respectively. Figure 7 illustrates the problem of initializing the PCPSO at the bottom of the Rastrigin bowl as discussed in the previous section. When the PCPSO algorithm was initialized in a larger Rastrigin space, as in Figure 8, it again converged faster than the standard PSO.

The last two rows of table 3 demonstrate that the PCPSO is less dependent on the selection of a maximum velocity (V_{max}). That is, the convergence characteristics of configurations 6 ($V_{max} = \infty$) and 6* were very similar (see figures 4, 5, 6 and 8). PCPSO configuration 6* outperformed the PSO configuration 2* on all but the Rastrigin $f_4^{\#1}$ as shown in figure 7.

The time complexity study that follows assumes that the PCPSO converges at least twice as fast as the PSO on many functional spaces. The results of the high dimensional study support this hypothesis.

7. TIME COMPLEXITY

A parameter study was conducted to illustrate the time complexity of the PCPSO algorithm. Equations 31 and 32 were developed using the Griewank function, 30 particles and n dimensions; on a 3.2GHz Pentium-4 with 1GB RAM.

$$t_{PSO} = (iter \times t_{prob}) + .06 + .002n \tag{31}$$

$$t_{PCPSO} = (iter \times t_{prob}) + .132 - .005n + .0005n^2 \tag{32}$$

The ($iter \times t_{prob}$) term was added after the equation fit assuming that:

$$t_{prob} \gg t_{Griwank} \tag{33}$$

It is seen that the time complexity of the PCPSO algorithm using G_{best} covariance updating is $O(n^2)$. This is less than the time complexity of the covariance matrix and eigenvector calculations (which are $O(p \cdot n^2)$ and $O(n^3)$ respectively) since the eigenvectors and covariance matrix are only updated for each new G_{best}. An incremental covariance matrix algorithm was employed, whereby a new weighted G_{best} is incorporated without requiring a rebuild of the entire covariance matrix. For the PCPSO algorithm to be competitive with the standard PSO it must be able to find solutions in fewer iterations than the standard PSO. Figure 5 illustrates a case where the PCPSO is converging faster than the standard PSO. Setting two times equation 31 equal to equation 32 (with $iter = 500$) and solving for n:

$$2(t_{PSO}) = t_{PCPSO} \tag{34}$$

$$n = 1000\left(.009+\sqrt{.000057 + t_{prob}}\right) \tag{35}$$

Figure 9 is a log–log plot of equation 35. It is interesting to note that, for problems having less than a certain number of dimensions, the PCPSO is a better choice for all function evaluation times. Since many engineering problems (with 1000 variables or less) require more than a second to evaluate, Figure 9 demonstrates that the PCPSO could be an economical alternative for a large number of engineering optimization problems. Figure 5 demonstrates that for high dimensional problems only $\left(\frac{1}{10}\right)^{th}$ of the eigenvectors may need to be calculated for a substantial boost in convergence.

8. CONCLUSION

The Principal Component Particle Swarm Optimization procedure performed well on a set of standard 30-dimensional test functions. The increased performance of the PCPSO on higher dimensional problems was demonstrated to be significant on three of the four test functions. A time complexity study was developed to provide guidance for the practical implementation of the PCPSO algorithm.

Since hybrid PSO algorithms can be implemented within the PCPSO framework without loss of generality the PCPSO should not be viewed as a competing algorithm, but rather a symbiotic algorithm that can be employed to accelerate convergence for high dimensional particle swarm optimization problems. The high dimensional and complexity studies demonstrate that the PCPSO is a promising algorithm for reducing the time complexity for some high dimensional engineering problems.

From a social psychological perspective the PCPSO algorithm is grouping certain ideas together based on past experience. This grouping of ideas can be seen as a form of linkage-learning [17] or dynamic probabilistic building block discovery [19].

The mathematics utilized for the PCPSO were limited to the real vector space $\Re^n$ which is a subset of the set of topological vector spaces. The PCPSO is a step towards a topological swarm intelligence that has the potential to utilize a much richer topological formulation.

A GUI program for verifying the results in this paper will be provided upon request.

9. REFERENCES

[1] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, (Perth, Australia, IEEE Service Center, Piscataway, NJ), pp. 1942–1948, 1995.

[2] R. Eberhart and J. Kennedy, “New optimizer using particle swarm theory,” in *Proceedings of the 1995 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[3] A. S. Lodge, *Body tensor fields in continuum mechanics, with applications to polymer rheology.* Academic Press, 1974.

[4] K. Lewin, *Principles of Topological Psychology.* McGraw–Hill.

[5] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, ch. Humans – Actual, Imagined, and Implied, pp. 200–202. San Francisco: Morgan Kaufmann, CA, 2001.

[6] K. Lewin, *Field Theory in Social Science*, ch. Constructs in Field Theory, p. 37. Harper & Brothers, 1951.

[7] K. Lewin, *Field Theory in Social Science*, ch. Defining the Field at a Given Time, p. 44. Harper & Brothers, 1951.

[8] K. Lewin, *A Dynamical Theory of Personality*, ch. Environmental Forces, pp. 76–77. McGraw–Hill, 1935.

[9] L. L. Thurstone, *Multivariate Statistical Methods: Within–Groups Covariation*, ch. Multiple Factor Analysis – Reprinted from Psychol. Rev., 38, 406–427 (1931), pp. 144–165. John Wiley & Sons, 1975.

[10] M. S. Voss, “Principal component particle swarm optimization,” in *Proceedings of the 2005 IEEE Swarm Intelligence Symposium – Accepted for publication*, (The Westin Pasadena, Pasadena, California), IEEE Press, 8-10 June 2005.

[11] J. E. Jackson, *A User’s Guide to Principal Components.* John Wiley & Sons, 1991.

[12] M. Løvbjerg, T. K. Rasmussen, and T. Krink, “Hybrid particle swarm optimiser with breeding and subpopulations,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, Morgan Kaufmann, 2001.

[13] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence.* San Francisco: Morgan Kaufmann, CA, 2001.

[14] Y. Shi and R. C. Eberhart, “Fuzzy adaptive particle swarm optimization,” in *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pp. 101–106, IEEE Press, 27-30 May 2001.

[15] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, “Hybird particle swarm optimizer with mass extinction,” in *Int. Conf. on Communication, Circuits and Systems (ICCCAS)*, pp. 1170–1173, 2002.

[16] M. Clerc, “The swarm and the queen: Towards a determinstic and adaptive particle swarm optimization,” in *Proceedings of the 1999 Congress of Evolutionary Computation*, IEEE Press, 1999.

[17] D. Thierens and D. Goldberg, “Mixing in genetic algorithms,” in *Fifth International Conference on Genetic Algorithms*, 1993.

[18] K. Lewin, *Field Theory in Social Science.* Harper & Brothers, 1951.

[19] J. H. Holland, *Hidden Order.* Adison-Wesley, 1996.