論 文

Technical Papers

Balancing Exploitation and Exploration in Particle Swarm Optimization: Velocity-based Reinitialization

Kevin J. Binkley — Department of Information and Computer Science, Keio University
kbinkley@soft.ics.keio.ac.jp, http://www.soft.ics.keio.ac.jp/~kbinkley/

Masafumi Hagiwara — (affiliation as previous author)
hagiwara@soft.ics.keio.ac.jp, http://www.soft.ics.keio.ac.jp/~hagiwara/intro-hagi.htm

keywords: particle swarm optimization, velocity-based reinitialization, optimization

Summary

In particle swarm optimization (PSO) algorithms there is a delicate balance to maintain between exploitation (local search) and exploration (global search). When facing multimodal functions, the standard PSO algorithm often converges to a local minimum quickly, missing better opportunities. Methods such as non-global best neighborhoods increase exploration, but at the expense of slowing the convergence of the whole PSO algorithm. In this paper, we propose a new method to extend PSO, velocity-based reinitialization (VBR). VBR is both simple to implement and effective at enhancing many different PSO algorithms from the literature. In VBR-PSO, the velocities of the particles are monitored throughout the evolution, and when the median velocity of the swarm particles has dropped below a threshold, the whole swarm is reinitialized. Through VBR, the problem of premature convergence is alleviated; VBR-PSO focuses on one minimum at a time. In our experiments, we apply VBR to the global-best, local best, and von Neumann neighborhood PSO algorithms. Results are presented using the standard benchmark functions from the PSO literature. VBR enhanced PSO yields improved results on the multimodal benchmark functions for all PSO algorithms investigated in this study.

1. Introduction

Particle swarm optimization (PSO) is an evolutionary optimization algorithm originally introduced by Kennedy and Eberhart [Kennedy 95]. It was observed that in many processes of nature such as birds flocking and fish schooling, the individuals could be simulated by rather simple rules. Originally these rules came from artificial life simulations. It was soon realized that these processes could be used effectively for function optimization.

The standard PSO algorithm employs a population of particles. The particles fly through the n-dimensional domain space of the function to be optimized. As they fly, each particle keeps track of its personal best position (in this study, the best minimum observed value of the function). After some stopping condition, the result of the PSO algorithm is the best minimum observed by the swarm.

In PSO the balancing of exploration (global search) and exploitation (local search) is performed via the choice of PSO algorithm and parameters. The standard PSO parameters are often chosen so that the algorithm eventually converges. The global best PSO algorithm with the Clerc parameter settings [Clerc 99, Eberhart 01b], is simple to implement and provides a good starting balance between exploitation and exploration. We will use this algorithm as a basis for comparison in our study.

For many multimodal problems, the standard global best PSO algorithm suffers from premature convergence. In order to increase the global search capabilities, the standard PSO algorithm has frequently been extended in the literature with neighborhood methods [Kennedy 99, Kennedy 02, Mendes 04]. All common neighborhood methods (other than global best), increase exploration at the expense of slowing or delaying exploitation.

What if we could exploit (local search) and then go back and explore (global search)? It is with this goal in mind that we introduce velocity-based reinitialization (VBR). VBR monitors the velocities of the swarm particles to determine if the swarm has stagnated. If the swarm has stagnated then further func-

tion evaluations are not likely to be fruitful. In order to more effectively apply function evaluations, the current result is saved and the swarm is restarted.

In addition to being straightforward to incorporate into a PSO algorithm, experiments show that VBR-PSO outperforms the standard global best PSO algorithm on all multimodal benchmark functions. Since the neighborhood methods local best (LBEST) [Kennedy 99] and von Neumann (VonNeu) [Kennedy 02] are known to improve the standard PSO algorithm on multimodal function optimization tasks, we also applied VBR to enhance these neighborhood PSO algorithms. The VBR-LBEST and VBR-VonNeu PSO enhanced algorithms also exhibited improvements over their non-enhanced counterparts on the multimodal benchmark problems.

The rest of this paper is outlined as follows. Section 2 discusses related research. Section 3 give the standard PSO algorithm. Section 4 discusses VBR enhanced PSO. Experimental results are given in Section 5. Our conclusion and future research are presented in Section 6.

2. Related Research

In velocity-based reinitialization (VBR), the velocity is the indicator of swarm stagnation and the swarm is completely reinitialized. There are, however, other ways of determining stagnation and taking action. Here we discuss some of the approaches seen in the PSO literature.

In [Clerc 99], a "no-hope" criterion based on the spread of particles in space is introduced and applied to determine when to restart the particles. In contrast to VBR, this method places more emphasis on local search in that the particles are restarted around the previous best particle.

Complete swarm reinitialization has been proposed for use with dynamic environments [Eberhart 01a, Hu 02]. In [Hu 02], the environment is monitored (by for example, checking the global best to see if the evaluation changed) and when a change is detected the whole swarm is reinitialized. Since VBR continually reinitializes the particles, it seems natural that some form of VBR will be applicable to dynamic environments. We leave this possible application for future research.

Re-initialization of velocity components was applied in the "self-organizing hierarchical particle swarm optimizer" (HPSO) [Ratnaweera 04]. The HPSO method is interesting in that it keeps the particle's previous velocity fixed at zero (essentially setting $w = 0$). With $w = 0$, the particle will move between the global best and its personal best until its velocity approaches zero. The HPSO algorithm checks the particle's velocity on a component by component basis and reinitializes any component velocity if it reaches zero.

In the Gregarious PSO (G-PSO) [Pasupuleti 06], a social component only version of PSO is proposed (the personal best component of the velocity update equation is not used, $c_1 = 0$). In the G-PSO algorithm, the velocity vector is completely reinitialized when the particle reaches a certain distance, ϵ, from the global best.

In [Clerc 06], stagnation is determined by the number of time steps without an improvement in a particle's local best. In this study, random neighborhoods are initially generated and on stagnation, the neighborhoods are reinitialized.

VBR differs from the approaches observed in the literature in that it offers a unique velocity-based method to determine the need for complete swarm reinitialization.

3. The PSO Algorithm

In the PSO algorithm, the state of each particle is represented by its position $\vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, velocity $\vec{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in})$, and its personal best position $\vec{p}_i = (p_{i1}, p_{i2}, \ldots, p_{in})$, where i is the particle index and n is the number of dimensions of the search space. The state of the whole swarm consists of state of all the particles plus a global best position $\vec{p}_g = (p_{g1}, p_{g2}, \ldots, p_{gn})$.

In the standard global best PSO algorithm, the particles head toward both their own personal best and the best position observed by the whole swarm. This tendency is controlled by the following PSO equations.

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

Where r_1 and r_2 are random numbers between 0 and 1, i is the particle index, and d is the coordinate being updated. Note that the velocity and position vectors are updated on a coordinate by coordinate basis.

The three key parameters to PSO are in the velocity update equation (1). The first component is the momentum component where the inertial constant w

controls how much the particle remembers its previous velocity. The second component is the cognitive component, the acceleration constant c_1 controls the how much the particle heads toward its personal best position. The third component, referred to as the social component, draws the particle toward swarm's best ever position; the acceleration constant c_2 controls this tendency.

There are many possible PSO parameter settings. In this study, we use the standard Clerc settings [Clerc 02, Clerc 99, Eberhart 01b] ($w = 0.729$ and $c1 = c2 = 1.49455$). These settings provide a good balance between exploration and exploitation.

When the standard global best PSO algorithm does not provide enough exploration, neighborhoods methods are often applied [Kennedy 99, Kennedy 02, Mendes 04]. With the neighborhood methods, the particles are only aware of their neighbors as defined by the particular neighborhood method being applied. Information transfer concerning the global best is delayed resulting in more exploration. Implementation is straightforward, the PSO velocity equation (1) is modified so that the global best $\vec{p}_g$, becomes a "neighborhood best" (including oneself).

The standard global best neighborhood is the fully connected graph. Information about the global best is known to all particles immediately. Of all the possible neighborhoods, this neighborhood places the most emphasis on exploitation. Another commonly used neighborhood is the local best neighborhood. Here the particles are placed in the circle and each particle has only two neighbors. Information about the global best will travel around the circle slowly allowing for more exploration before the swarm finally converges. A more connected, but not fully connected neighborhood is the von Neumann neighborhood. In this neighborhood, the particles are placed on a grid and the neighborhood is the particle and its four nearest neighbors. It has been reported that the von Neumann neighborhood performed best overall on a standard set of test functions [Kennedy 02].

The pseudocode for the standard global best PSO algorithm is presented in Figure 1.

```
Initialize population randomly
Do for each step in time until termination condition
  For i = 1 to population size
    For d = 1 to num dimensions
      r1 = rand() // random number in (0,1)
      r2 = rand()
      Update velocity using velocity equation (1)
      Update position using position equation (2)
    End For
    If particle i's position is better than
                  its personal best then
      update particle i's personal best
    End if
  End For
  Update the swarm global best with the best
                      of all particle personal bests
End Do
Return the swarm global best
```

Fig. 1 Pseudocode for the standard PSO algorithm.

4. Velocity-based Reinitialization Particle Swarm Optimization

We have observed many particle swarm evolutions using the standard PSO settings. One common characteristic is that the particles often find a local minimum and due to constriction continue to get closer and closer to this minimum, always improving the global best position slightly. This is fine if the minimum is a global minimum. However, on multimodal functions, there are almost always many other better minimums. We propose that rather than the slight improvements offered by continuing local search, it is often better to make note of the result found and reinitialize the swarm completely, performing global search again.

The problem left is when to restart the search. The width of the search area as determined by either the particles' position (or personal best) spread is a possibility [Clerc 99]. However, we observed that often a few particles out of the whole population would be spread out and the rest would be "wasting" valuable function evaluations while waiting for the remaining particles to converge to the swarm's global best. In many cases, these few remaining particles might never get lucky enough to land near enough to the swarm's global best, improve their personal best and converge.

With these observations, we developed velocity-based reinitialization (VBR). With VBR, we estimate when most of the swarm's function evaluations are not being applied effectively and restart the swarm. In VBR, the median velocity of the swarm particles is monitored at each time step during the evolution. When the median velocity drops below a predetermined threshold, α, we declare the swarm stagnant and initiate a restart. We define stagnation very loosely as "further function evaluations are not likely to be very productive". Specifically, stagnation is determined by the following steps.

(1) calculate the Euclidean norm of each velocity

vector, $n_i = ||\vec{v}_i|| = \sqrt{\vec{v}_i \bullet \vec{v}_i}$.

(2) sort the norms.

(3) compare the median of the velocity vector norms to the stagnation threshold, α. When the median velocity vector norm has dropped below α, the swarm is considered stagnate.

A more general method than the median is to use the n^{th} norm, where the first norm is the smallest. The n^{th} norm can be used to adjust the amount of reinitialization. In some preliminary experiments, we found that VBR was not too sensitive which norm was used. We settled on the median for this study. However, we note that the first norm does not work as well since there will be cases where one particle's velocity is very small but the swarm is still productive and reinitialization would be premature. It is actually natural for the global best particle's velocity get small quickly since its personal best is the global best. If a new global best is not found, the particle will lose momentum as it homes in on the global best (for the usual case where $w < 1$). The average of the velocity vector norms can also be considered. However, the median was chosen to be able to declare a swarm stagnate when the majority of the particles have stagnated.

The modifications to the standard PSO algorithm given in Section 3 are quite small and simple to implement. With VBR, a list of all global bests is kept, and at the beginning of each time step a check for stagnation is performed. If stagnation is detected the swarm's global best is saved and the swarm is reinitialized. When a predetermined termination condition is reached, the best of the union of the current swarm's global best and the global bests stored in the global best list is returned as the optimal value found. The pseudocode for the VBR enhanced PSO algorithm is given in Figure 2.

```
Initialize population randomly
Set global best list to empty
Do for each step in time until termination condition
  If swarm has stagnated then
    Add swarm's global best to global best list
    Reinitialize swarm
  else // perform normal swarm movements
    For i = 1 to population size
      For d = 1 to num dimensions
        r1 = rand() // random number in (0,1)
        r2 = rand()
        Update velocity using velocity equation (1)
        Update position using position equation (2)
      End For
      If particle i's position is better than
                    its personal best then
        update particle i's personal best
      End if
    End For
    Update the swarm global best with the best
                    of all particle personal bests
  End if
End Do
Add swarm's global best to global best list
Return the best of the global best list
```

Fig. 2 Pseudocode for the VBR enhanced PSO algorithm.

5. Experiments

In this section, the standard global best PSO algorithm and the neighborhood local best and von Neumann PSO algorithms are compared to their respective VBR enhanced versions. First we discuss the common experimental settings and then we give the experimental results.

5·1 Common Experimental Settings

To compare the VBR enhanced PSO algorithms to the standard PSO algorithms, five standard benchmark functions commonly found in the PSO literature are employed. Besides the benchmark functions, there are many problem settings to be determined such as the population size, the number of generations to run (we will use the terms "time period" and "generation" interchangeably), the search space size, etc. In efforts to keep these settings as standard as possible, we follow the simulation settings in [Ratnaweera 04] whenever possible.

The benchmark functions are shown in Table 1. The GO Test Problems web site [Hedar 07] has graphs and program code useful for visualizing these functions. The simulation difficulty is adjusted by increasing the number of dimensions of the functions. We performed simulations in 10, 20 and 30 dimensions for all functions except Schaffer's f6 function which is two dimensional. The simulations were stopped when a predetermined stopping criterion was reached or after a maximum number of generations were run. The stopping criterion was set to 0.01 for all benchmark functions except Schaffer's f6 function, where it was set to 0.00001. For Schaffer's f6 function the maximum number of generations was set at 1000. For the sphere function the maximum number of generations was set at 1000, 2000, and 3000 respectively, for the 10, 20 and 30 dimension simulations. For the remaining functions the maximum number of generations was set at 3000, 4000, and 5000 respectively, for the 10, 20 and 30 dimension simulations.

Table 1 Benchmark functions.

Name	Function
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigrin	$f_3(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$
Griewank	$f_4(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$
Schaffer's f6	$f_5(x) = 0.5 + \frac{(\sin\sqrt{x_1^2+x_2^2})^2 - 0.5}{(1.0+0.001(x_1^2+x_2^2))^2}$

To keep the particles from straying too far from the valid search range, the particle positions are restricted to a predetermined search range. There is also no need for the particle velocity to ever exceed the size of the search range. The velocities are normally restricted to some fraction of the search range [Eberhart 00, Eberhart 01b]. Being generous, we restrict the velocity to one half the size of the search range. The search ranges and velocity maximums are shown in Table 2. If a component of a particle's position exceeds the search range, then the component is randomly placed back in the search range, and the corresponding velocity component is set to 0. If the absolute value of a component of a particle's velocity exceeds the velocity maximum, it is set to the velocity maximum.

Since the common benchmarks have minimums near the origin, uniform random initialization of the particles would result in the initial positions of the particles surrounding the minimum. In real problems it is highly unlikely the initial placement of the particles will have the benefit of surrounding the global minimum. To avoid this bias, the particle positions are initialized asymmetrically as was done in [Angeline 98, Ratnaweera 04]. The asymmetric initialization ranges are shown in Table 2.

The particle initial velocities are initialized randomly between plus and minus the maximum allowed velocity given in Table 2. All simulations were run 50 times with a population size of 40 particles.

5·2 Results

First, we show that velocity-based reinitialization can be used to enhance the standard global best PSO algorithm, yielding improved results for all multimodal benchmark functions. Second, we look at the sensitivity of VBR to the threshold parameter α. We find that VBR is not too sensitive to the threshold parameter setting and give guidelines for determining appropriate settings. And third, since the neighborhood methods such as LBEST and von Neumann also are known to improve PSO performance on multimodal problems, VBR is applied to enhance these neighborhood PSO algorithms. We find that VBR also improves the performance of these neighborhood PSO algorithms on multimodal functions.

Table 2 Search range, maximum velocity, and asymmetric initialization range.

Function	Search Range	Max. Vel.	Init. Range
$f_1(x)$	$(-100, 100)^n$	100	$(50, 100)^n$
$f_2(x)$	$(-100, 100)^n$	100	$(50, 100)^n$
$f_3(x)$	$(-10, 10)^n$	10	$(2.56, 5.12)^n$
$f_4(x)$	$(-600, 600)^n$	600	$(300, 600)^n$
$f_5(x)$	$(-100, 100)^n$	100	$(15, 30)^n$

§ 1 Applying VBR to enhance the standard global best PSO algorithm

We found that VBR strongly improves the results of the standard global best PSO for all multimodal problems. Table 3 and Table 4 give the results comparing the standard global PSO algorithm to the VBR-PSO algorithm. From Table 3 it is observed that the stopping criterion was reached much more frequently on the multimodal Griewank functions and the Schaffer's f6 function. For the multimodal Rastrigrin function, the stopping criterion was mostly not achieved. Looking at the Rastrigrin function results in Table 4, it is clear that in terms of average minimization result the VBR-PSO consistently yields improved results. In terms of average minimization result achieved, Table 4, the VBR enhanced standard global best PSO yields improved results on all multimodal functions in the test suite. Overall, both in terms of the number of stopping criterion reached and the average minimization result achieved, VBR has clearly improved the results of the standard global best PSO algorithms for all multimodal functions.

For the unimodal Sphere function, both algorithms reach the goal equally often, the results being identical as no reinitializations occur. A similar case occurs with the unimodal Rosenbrock function in 10 dimensions, the VBR-PSO algorithm yields the exact same results as standard global best PSO for $\alpha <= 0.001$. The stagnation threshold was never reached and VBR-PSO gracefully degenerated to the standard PSO algorithm. For the Rosenbrock function in higher dimensions, the results are slightly affected by reinitializations. This affect can be removed by

Table 3 A comparison of the PSO algorithms in terms of the number of stopping criterion reached out of 50 trials. The functions $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_4(x)$, and $f_5(x)$ represent respectively the Sphere, Rosenbrock, Rastrigrin, Griewank, and Schaffer's f6 functions. The upper number is the number of successes and the lower number is the average time period the successes occurred. The standard global PSO results are shown in column STD. The next three columns show the result of applying VBR to standard global best PSO for various thresholds: $\alpha = 0.01$, $\alpha = 0.001$, and $\alpha = 0.0001$. The results of the LBEST and von Neumann neighborhood PSO algorithms are respectively shown in the LBEST and VonNeu columns. The corresponding VBR enhanced versions of the LBEST and von Neumann are shown in respectively in the VBR-L and VBR-N columns (using $\alpha = 0.001$). VBR has clearly improved the results of the standard global best PSO for all multimodal functions.

Func.	Dim	STD	α0.01	α0.001	α0.0001	LBEST	VBR-L	VonNeu	VBR-V
$f_1(x)$	10	50	50	50	50	50	50	50	50
		(107)	(107)	(107)	(107)	(194)	(194)	(155)	(155)
	20	50	50	50	50	50	50	50	50
		(195)	(195)	(195)	(195)	(407)	(407)	(314)	(314)
	30	50	50	50	50	50	50	50	50
		(316)	(316)	(316)	(316)	(628)	(628)	(477)	(477)
$f_2(x)$	10	15	18	15	15	1	3	4	2
		(2576)	(2671)	(2576)	(2576)	(652)	(743)	(2375)	(888)
	20	6	0	5	6	3	0	1	0
		(3227)	—	(3422)	(3227)	(2679)	—	(3153)	—
	30	1	0	0	2	1	1	1	1
		(4054)	—	—	(4244)	(4182)	(3024)	(2495)	(2061)
$f_3(x)$	10	0	1	0	1	2	0	2	1
		—	(2814)	—	(2633)	(2854)	—	(1414)	(2654)
	20	0	0	0	0	0	0	0	0
		—	—	—	—	—	—	—	—
	30	0	0	0	0	0	0	0	0
		—	—	—	—	—	—	—	—
$f_4(x)$	10	1	3	6	2	7	17	2	14
		(188)	(536)	(1010)	(1728)	(617)	(1401)	(412)	(1459)
	20	15	50	49	47	45	50	37	50
		(227)	(860)	(1135)	(969)	(592)	(762)	(613)	(770)
	30	26	50	50	50	47	50	39	50
		(353)	(1045)	(945)	(1045)	(776)	(785)	(679)	(891)
$f_5(x)$	2	32	45	40	36	37	41	48	49
		(193)	(234)	(226)	(304)	(334)	(412)	(322)	(316)

Table 4 A comparison of the PSO algorithms in terms of the average minimization result achieved out of 50 trails. The functions $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_4(x)$, and $f_5(x)$ represent respectively the Sphere, Rosenbrock, Rastrigrin, Griewank, and Schaffer's f6 functions. The upper number is the average result achieved and the lower number is the standard deviation of the results. When the stopping criterion was reached in all trials, the result is labeled with the term 'goal'. The standard global PSO results are shown in column STD. The next three columns show the result of applying VBR to standard global best PSO for various thresholds: $\alpha = 0.01$, $\alpha = 0.001$, and $\alpha = 0.0001$. The results of the LBEST and von Neumann neighborhood PSO algorithms are respectively shown in the LBEST and VonNeu columns. The corresponding VBR enhanced versions of the LBEST and von Neumann are shown in respectively in the VBR-L and VBR-N columns (using $\alpha = 0.001$). VBR has clearly improved the results of the standard global best PSO for all multimodal functions. VBR has also improved the results of the LBEST and von Neumann neighborhood PSO algorithms for all multimodal functions.

Func.	Dim	STD	α0.01	α0.001	α0.0001	LBEST	VBR-L	VonNeu	VBR-V
$f_1(x)$	10	goal	goal	goal	goal	goal	goal	goal	goal
		(—)	(—)	(—)	(—)	(—)	(—)	(—)	(—)
	20	goal	goal	goal	goal	goal	goal	goal	goal
		(—)	(—)	(—)	(—)	(—)	(—)	(—)	(—)
	30	goal	goal	goal	goal	goal	goal	goal	goal
		(—)	(—)	(—)	(—)	(—)	(—)	(—)	(—)
$f_2(x)$	10	5.09	1.26	5.09	5.09	1.63	1.01	3.16	3.49
		(16.69)	(1.81)	(16.69)	(16.69)	(2.84)	(1.35)	(11.04)	(6.54)
	20	8.63	11.57	9.16	8.63	8.09	8.09	13.76	17.56
		(21.31)	(26.39)	(21.51)	(21.31)	(11.84)	(13.45)	(23.21)	(31.99)
	30	15.27	16.29	20.94	14.67	25.94	19.19	33.29	24.53
		(26.19)	(18.52)	(30.32)	(27.51)	(32.01)	(28.59)	(48.08)	(30.74)
$f_3(x)$	10	6.23	2.6277	2.75	2.81	5.39	4.74	3.85	2.72
		(3.38)	(0.9486)	(1.01)	(1.07)	(2.74)	(1.22)	(2.11)	(1.12)
	20	37.11	18.76	18.41	19.58	35.65	29.80	23.81	17.86
		(14.52)	(4.82)	(4.34)	(4.54)	(8.62)	(7.17)	(8.70)	(4.45)
	30	82.48	47.66	54.45	53.16	89.10	73.84	58.39	47.20
		(21.66)	(10.29)	(10.91)	(9.75)	(14.58)	(13.46)	(14.92)	(7.99)
$f_4(x)$	10	0.0672	0.0331	0.0343	0.0343	0.0345	0.0176	0.0371	0.0193
		(0.0268)	(0.0139)	(0.0156)	(0.0136)	(0.0195)	(0.0098)	(0.0164)	(0.0099)
	20	0.0276	goal	0.0096	0.0096	0.0101	goal	0.0121	goal
		(0.0216)	(—)	(0.0007)	(0.0013)	(0.0034)	(—)	(0.0071)	(—)
	30	0.0258	goal	goal	goal	0.0097	goal	0.0118	goal
		(0.0279)	(—)	(—)	(—)	(0.0015)	(—)	(0.0058)	(—)
$f_5(x)$	2	0.0035	0.0008	0.0019	0.0026	0.0019	0.0008	0.0004	0.0002
		(0.0047)	(0.0026)	(0.0037)	(0.0042)	(0.0038)	(0.0026)	(0.0019)	(0.0014)

decreasing α. However, we propose that most real-life optimization problems that a PSO algorithm will be facing are likely to be multimodal, and that a sacrifice in unimodal function optimization capability is acceptable given the improvements offered in the multimodal problem domain.

§ 2 Determining the appropriate threshold, α, for VBR

Referring to Table 3 and Table 4 we observe that VBR is not too sensitive to the threshold parameter α. In these tables, α is varied by a factor of 100 and VBR-PSO exhibits improvements over the standard global best PSO algorithm achieved for all settings.

We hypothesis that what is important in the VBR-PSO algorithm is that there is a "give up, record the global best, and reinitialize" point during the optimization. The VBR threshold setting, α, determines when this occurs and if it is set to reasonably conservative values, the VBR-PSO algorithm yields improved results over the standard PSO algorithm. Indeed, as $\alpha \to 0$ reinitialization becomes rare and VBR-PSO approaches PSO.

However, setting α too conservatively will not yield the benefits of VBR-PSO. We offer the following guidelines for setting α. For a particular PSO application, choose α based on how close it is desired to get to the actual local minimum of the problem. If one-half of the swarm particle's velocity vectors fit in an n-dimensional ball of radius α then VBR will consider the swarm stagnate. Keep in mind that for most real-life problems there will be many local minimums. PSO will be finding one of them, generally not the global minimum (if it exists). Rather than get close to a particular local minimum, it might actually be better to reinitialize and find a better location further away from a different local minimum.

§ 3 Applying VBR to enhance the LBEST and von Neumann neighborhood PSO algorithms

We now show that VBR can also benefit the local best and von Neumann neighborhood PSO algorithms that are known to perform better than the standard global best PSO algorithm for multimodal problems. In terms of number of stopping criterion reached, Table 3 indicates that the VBR enhanced algorithms show clearly improved results on the multimodal Griewank function. This is also confirmed by a look at the average minimization result given in Table 4. On the Rastrigrin function in 10 dimensions, results appear to be worse, but a look at the average minimization result in Table 4 shows that VBR achieved a better minimum on average. In 20 and 30 dimensions too, the VBR enhanced PSO always achieved a better average result, the spread getting wider as the dimensionality increases. We conclude that the VBR enhanced PSO performed better on the Rastrigrin function overall. Overall, we conclude that VBR has improved the results for all multimodal functions.

On the unimodal Sphere function, all algorithms performed equally well. Results on the unimodal Rosenbrock function are mixed. The non-enhanced PSO algorithms achieved the stopping criterion more often according to Table 3. But the VBR-enhanced PSO algorithms frequently achieved better results in terms of average minimization result. As most applications that PSO algorithms are facing will be multimodal, we believe that VBR-enhanced PSO offers strong overall benefits.

6. Conclusion

In this study, we have proposed a new method, velocity-based reinitialization (VBR), to enhance the search capabilities of PSO algorithms. VBR is different from other reinitialization methods in that it uses a novel median-based method to determine when reinitialization should occur.

VBR helps PSO algorithms achieve a good balance between exploitation (local search) and exploration (global search). With VBR the algorithm can first exploit, and then through reinitialization resume exploration.

VBR is simple to implement and has only one threshold parameter. Our experiments showed that VBR was not too sensitive to the threshold setting. The family of VBR enhanced PSO algorithms also gracefully approaches the standard PSO algorithm as the threshold setting approaches zero.

VBR is effective at enhancing many different PSO algorithms. In the experiments conducted in this study, we applied VBR to three PSO algorithms: the standard global best, the local best neighborhood, and the von Neumann neighborhood. The VBR enhanced PSO algorithms were all shown to improve the optimization results for all multimodal functions.

In future research, we are looking into applying the VBR idea to PSO in dynamic environments. We expect the velocity-based stagnation criterion will be beneficial PSO algorithms facing changing environments.

♢ References ♢

[Angeline 98] Angeline, P., Inc, N., and Vestal, N.: Using selection to improve particle swarm optimization, *Evolutionary Computation Proceedings of the 1998 IEEE World Congress on Computational Intelligence*, pp. 84–89 (1998)

[Clerc 99] Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol. 3, (1999)

[Clerc 02] Clerc, M. and Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, pp. 58–73 (2002)

[Clerc 06] Clerc, M.: Stagnation analysis in particle swarm optimisation or what happens when nothing happens, Technical Report CSM-460, Department of Computer Science, University of Essex (2006)

[Eberhart 00] Eberhart, R. and Shi, Y.: Comparing inertia weights and constriction factors in particle swarm optimization, *Proceedings of the 2000 Congress on Evolutionary Computation*, Vol. 1, (2000)

[Eberhart 01a] Eberhart, R. and Shi, Y.: Tracking and optimizing dynamic systems with particle swarms, *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 1, (2001)

[Eberhart 01b] Eberhart, and Shi, Y.: Particle swarm optimization: developments, applications and resources, *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 1, pp. 81–86 (2001)

[Hedar 07] Hedar, A. H.: GO Test Problems - Test Functions for Unconstrained Global Optimizations, Website (2007), `http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/%TestGO_files/Page364.htm`

[Hu 02] Hu, X. and Eberhart, R.: Adaptive particle swarm optimization: detection and response to dynamic systems, *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 2, (2002)

[Kennedy 95] Kennedy, J. and Eberhart, R.: Particle swarm optimization, *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 4, (1995)

[Kennedy 99] Kennedy, J.: Small worlds and mega-minds: effects of neighborhood topology onparticle swarm performance, *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol. 3, (1999)

[Kennedy 02] Kennedy, J. and Mendes, R.: Population structure and particle swarm performance, *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol. 2, (2002)

[Mendes 04] Mendes, R., Kennedy, J., and Neves, J.: The fully informed particle swarm: simpler, maybe better, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 204–210 (2004)

[Pasupuleti 06] Pasupuleti, S. and Battiti, R.: The gregarious particle swarm optimizer (G-PSO), *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 67–74 (2006)

[Ratnaweera 04] Ratnaweera, A., Halgamuge, S., and Watson, H.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 240–255 (2004)

〔担当委員：伊藤 公人〕

Received July 20, 2007.

Author's Profile

Binkley, Kevin J.

Kevin J. Binkley received a B.A. in Applied Math and a B.S. in Chemistry from the University of California Berkeley in 1988 and an M.S. in Computer Science from Stanford University in 1991. He is now pursuing a doctorate at Keio University. His current research interests include artificial neural networks and evolutionary computation. He is a member of IEICE and IEEE.

Hagiwara, Masafumi (Member)

Masafumi Hagiwara received his B.E., M.E. and Ph.D degrees in electrical engineering from Keio University, Yokohama, Japan, in 1982, 1984 and 1987, respectively. Since 1987 he has been with Keio University, where he is now a Professor. From 1991 to 1993, he was a visiting scholar at Stanford University. He received the Niwa Memorial Award, Shinohara Memorial Young Engineer Award, IEEE Consumer Electronics Society Chester Sall Award, Ando Memorial Award, Author Award from the Japan Society of Fuzzy Theory and Systems (SOFT), Technical Award and Paper Award from Japan Society of Kansei Engineering in 1986, 1987, 1990, 1994, 1996, 2003, and 2004, respectively. His research interests include neural networks, fuzzy systems, evolutionary computation and kansei engineering. Dr. Hagiwara is a member of IEICE, IPSJ, SOFT, IEE of Japan, Japan Society of Kansei Engineering, JNNS and IEEE (Senior member).