

# Exposing Origin-Seeking Bias in PSO

Christopher K. Monson  
Brigham Young University  
Computer Science Department  
3361 TMCB, Provo, UT 84602  
c@cs.byu.edu

Kevin D. Seppi  
Brigham Young University  
Computer Science Department  
3330 TMCB, Provo, UT 84602  
kseppi@cs.byu.edu

## ABSTRACT

We discuss testing methods for exposing origin-seeking bias in PSO motion algorithms. The strategy of resizing the initialization space, proposed by Gehlhaar and Fogel and made popular in the PSO context by Angeline, is shown to be insufficiently general for revealing an algorithm’s tendency to focus its efforts on regions at or near the origin. An alternative testing method is proposed that reveals problems with PSO motion algorithms that are not visible when merely resizing the initialization space.

## Track Category

Ant Colony Optimization and Swarm Intelligence

## Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—  
*nonlinear programming, unconstrained optimization*

## General Terms

Algorithms

## Keywords

Swarm Intelligence, Optimization, Initialization Bias

## 1. INTRODUCTION

Particle swarms are now well known as an effective and interesting approach to function optimization. The basic algorithm scatters particles in a limited *feasible region* of the function’s domain space, moving them over time in a search for areas of better fitness. Each particle keeps track of a current position  $\mathbf{x}$  and velocity  $\mathbf{v}$ , as well as the most fit location it has ever seen  $\mathbf{p}$ . The best  $\mathbf{p}$  among all particles is denoted  $\mathbf{g}$ . In classical PSO, these data are easily combined:

$$\mathbf{v}_{t+1} = \chi(\mathbf{v}_t + \phi_1 U()(\mathbf{p} - \mathbf{x}_t) + \phi_2 U()(\mathbf{g} - \mathbf{x}_t)) \quad (1)$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1} \quad (2)$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO’05, June 25–29, 2005, Washington, DC, USA.  
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

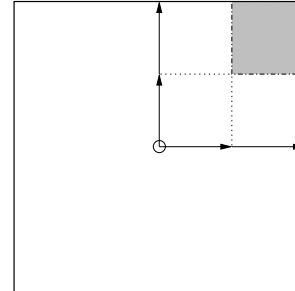


Figure 1: Angeline’s initialization region for PSO

where  $U()$  is a sample from a standard uniform distribution,  $\phi_i$  are usually somewhere near 2, and  $\chi$  represents the addition of a constriction coefficient that serves to control the convergence properties of the algorithm [4].

Throughout the remainder of this paper, it will be assumed that minimization is performed. It will also be implied that  $\mathbf{g}$  refers to the best known position among all  $\mathbf{p}$  in a particle’s neighborhood. While this is a slight departure from the norm, where local neighborhood bests have different notation ( $\mathbf{l}$ ), the two notations are rarely if ever used in the same context. Therefore, it will be understood that where a notion of sociometry is present,  $\mathbf{g}$  is the best known among all of the particles in a particular neighborhood and is therefore particle-dependent.

Many population-based optimization approaches, including PSO, suffer from a notable bias: they tend to perform best when the optimum is located at or near the center of the initialization region, which is often the origin. This is especially true when some kind of averaging operator is used to combine information from different members of the population [1]. In many of the standard benchmark functions, the global optimum is at or very near the origin, making this bias a potential problem when developing and testing a new algorithm. To expose this bias while testing PSO algorithms, Angeline [1] popularized a method previously introduced by Gehlhaar and Fogel [5]. The method, hereafter referred to as “Region Scaling” (RS) explicitly excludes the optimum from the initialization region by initializing particles in a new region whose sides are all 1/4 the length of the original, as shown in Figure 1.

This paper focuses on test methods used to expose origin-seeking bias in PSO algorithms and shows that RS is not always sufficient. Experiments are done using Clerc’s TRIBES [2, 3] with various kinds of particle motion. TRIBES is de-

scribed in some detail, followed by descriptions of the kinds of motion chosen for the experiments. An alternative to RS called “Center Offset” (CO) is proposed as a means of exposing the bias in PSO, and experimental results highlight the contrast between the two approaches. Finally, some discussion of the meaning of the results is presented with accompanying recommendations for testing new algorithms.

## 2. TRIBES

TRIBES is a parameter-free approach to swarm size and sociometry in PSO, at the heart of which is a swarm restructuring algorithm which adapts the number of particles and the topology of particle neighborhoods based on swarm performance [2].

Because it is parameter-free, TRIBES provides a useful way to sidestep the issue of swarm size and sociometry specification, providing an out-of-the box approach that has been shown to work well. Because the parameters of swarm size and sociometry are adapted based on the success of the swarm at a given time, TRIBES zeros in rapidly on settings for those parameters that produce the best performance. In the case of a biased motion algorithm, this feature of TRIBES serves to expose that bias effectively.

Clerc’s TRIBES paper defines notions of *tribes* and *informers*. A *tribe* is a data structure that keeps track of the particles that belong to it, representing a fully connected subgraph of the overall swarm topology. The *informers* of a particle are itself, all of the particles in its tribe, and any particles of other tribes to which it is connected. All links are symmetrical.

The algorithm begins with one or more particles in a single tribe. The memory of a particle is extended slightly to include not only  $\mathbf{p}$ , but also the number of times it has changed in succession. If the number of successive changes is greater than 0, then improvement was made during the last position update and the particle is labeled “good”<sup>1</sup>.

The tribes themselves also receive the labels “good” or “bad”, depending on the number of good particles in the tribe. A tribe containing  $T$  particles is itself “good” only if  $U() \leq G/T$ , where  $G$  is the number of good particles in a tribe and  $U()$  is a draw from a standard uniform distribution. Otherwise the tribe is “bad”.

Good tribes, because they are doing well and presumably do not need as many particles, will remove one of their particles. Assuming that  $f$  is the function being minimized, a good tribe containing more than one particle will remove its worst performer, or the particle with the highest  $f(\mathbf{p})$ . When this occurs, any external links to the particle are reassigned to the best performer in the tribe, i.e. the particle with the lowest  $f(\mathbf{p})$ .

If a good tribe contains only one particle, the tribe itself is removed only if its particle’s best external informer has a better  $f(\mathbf{p})$  than itself. In this latter case, all external links to the particle are reassigned to the external informer.

Bad tribes, on the other hand, presumably need more information, so each creates a new particle *outside of its tribe* and forms a link between the new particle and the best particle within the tribe. The set of all new particles created during one restructuring step forms a *new tribe*. Each new

<sup>1</sup>TRIBES also has a notion of “excellent”, assigned to a particle if the number is 2 or higher, but we do not make use of that distinction in this paper.

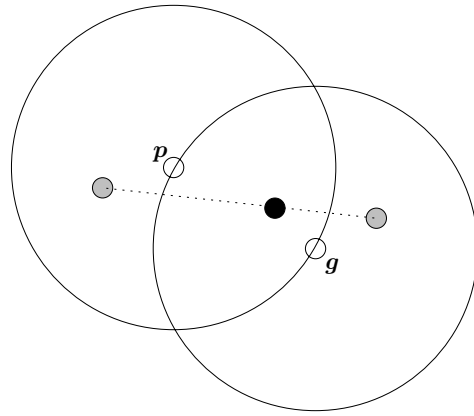


Figure 2: TRIBES Pivot method

particle is generated randomly and uniformly within the initialization space.

Restructuring occurs once at the beginning of the algorithm and then periodically as it progresses. If, after restructuring, the swarm has  $N$  particles and  $L$  information links, then restructuring will occur again after  $L/2$  swarm iterations, or  $NL/2$  function evaluations.

## 3. PSO MOTION ALGORITHMS

Several motion algorithms have been suggested for PSO, so many that they cannot all be discussed here. This section describes algorithms that are representative of some interesting features of existing approaches to PSO motion, and these will be used in this paper’s empirical study of origin-seeking bias.

### 3.1 Pivot

The central motion algorithm introduced in the TRIBES paper is the “Simple Pivot” method. The “Noisy Pivot” method, also introduced in the TRIBES paper, is an extension of Simple Pivot which performs an additional Gaussian sample to generate the final position [2]. We will focus on the Simple Pivot here, referring to it simply as “Pivot”.

The Pivot method, illustrated in Figure 2, generates a new position by taking noisy samples in the neighborhood of  $\mathbf{p}$  and  $\mathbf{g}$ . First, each is taken to be the center of a hypersphere whose radius is  $\|\mathbf{p} - \mathbf{g}\|_2$ . Second, a point is sampled from a uniform distribution within each sphere. Each of these samples is given a mass based on the relative fitness of its corresponding center point (either  $\mathbf{p}$  or  $\mathbf{g}$ ), and the new position is the center of mass of the two sampled points.

Mass may be assigned in a number of ways. One simple approach is to assign mass linearly based on the relative fitness of each particle, thus:

$$x_{t+1} = \frac{f(\mathbf{p})}{f(\mathbf{p}) + f(\mathbf{g})} (\mathbf{g} + U(\|\mathbf{p} - \mathbf{g}\|_2)) + \frac{f(\mathbf{g})}{f(\mathbf{p}) + f(\mathbf{g})} (\mathbf{p} + U(\|\mathbf{p} - \mathbf{g}\|_2)) \quad (3)$$

where  $U(\cdot)$  is a sample from a hyperspherical uniform distribution with the specified radius. This formula assumes that minimization is occurring, so smaller values of  $f$  are favored. Maximization would reverse the positions of the fractional coefficients.

### 3.2 PSOGauss

The second type of motion proposed in conjunction with TRIBES, but not given a name, is based on constricted PSO with Gaussian noise [2]. We will refer to this motion as PSOGauss:

$$\mathbf{v}_{t+1} = \chi \left( \mathbf{v}_t + \mathbf{G} \left( \mathbf{p} - \mathbf{x}_t, \frac{1}{4} \mathbf{I} \|\mathbf{p} - \mathbf{x}_t\|_2^2 \right) + \mathbf{G} \left( \mathbf{g} - \mathbf{x}_t, \frac{1}{4} \mathbf{I} \|\mathbf{g} - \mathbf{x}_t\|_2^2 \right) \right) \quad (4)$$

where  $\mathbf{I}$  is the identity matrix,  $\chi \approx 0.71441$ , and  $\mathbf{G}(\cdot, \cdot)$  is a sample from a Gaussian distribution parameterized by the supplied mean and covariance matrix.

This approach is similar to constricted PSO and different from the others in this section because it uses velocity instead of computing a position directly.

### 3.3 BareBones

The BareBones motion algorithm is probably the simplest PSO algorithm proposed to date, but it is very successful at optimization [6]. The motion equation is given here:

$$\mathbf{x}_{t+1} = \mathbf{G} \left( \frac{1}{2}(\mathbf{p} + \mathbf{g}), \mathbf{I} \|\mathbf{p} - \mathbf{g}\|_2^2 \right) \quad (5)$$

where  $\mathbf{G}(\cdot, \cdot)$  is a sample from a Gaussian distribution parameterized on a mean and covariance matrix.

This approach was developed after noting that the distribution of samples between  $\mathbf{p}$  and  $\mathbf{g}$  was distinctly Gaussian at each time step in classical PSO and was an effort to cull out any useless properties of the traditional motion equations.

This particular method was not the only one proposed in Kennedy’s Bare Bones paper [6], but it is the simplest and is very effective.

## 4. EXPERIMENTS

To test for origin seeking behavior, the following benchmark functions were used:

**Sphere:**

$$f(\mathbf{x}) = \sum_{i=1}^D x_i^2 \quad R = (-50, 50)^D \quad (6)$$

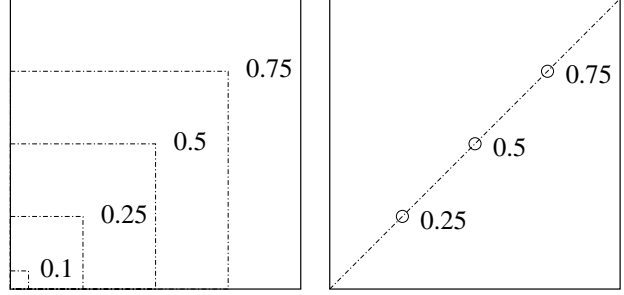
**Rastrigin:**

$$f(\mathbf{x}) = \sum_{i=1}^D x_i^2 + 10 - 10 \cos(2\pi x_i) \quad R = (-5.12, 5.12)^D \quad (7)$$

**Rosenbrock:**

$$f(\mathbf{x}) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \quad R = (-100, 100)^D \quad (8)$$

Sphere is unimodal and symmetric, Rastrigin is highly multimodal and symmetric, and Rosenbrock is multimodal and asymmetric. These functions are representative of the essential characteristics of a number of popular benchmarks.



(a) Region Scaling (RS)

(b) Center Offset (CO)

**Figure 3: Different methods of exposing origin-seeking bias**

For each function and type of motion, experiments were performed using “Region Scaling” (RS) and “Center Offset” (CO). In the first (RS), several different initialization regions were chosen, each formed by taking a fraction of the feasible rectangle in each dimension as shown in Figure 3(a)<sup>2</sup>. In the second (CO), the center of each function was moved to a different location of space, as shown in Figure 3(b), leaving the initialization region in its original location. For example, Sphere would become

$$f(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^D (x_i - c_i)^2$$

where  $\mathbf{c}$  is the location of the new center, calculated using the feasible region and the numbers shown in the figure (0.5 leaves the center unchanged). This can only be applied to functions whose support extends outside of the feasible region, as is the case with all of the benchmarks used here. CO values outside of the range  $[0, 1]$  are valid and indicate that the center has moved beyond the boundaries of the feasible region along the line shown.

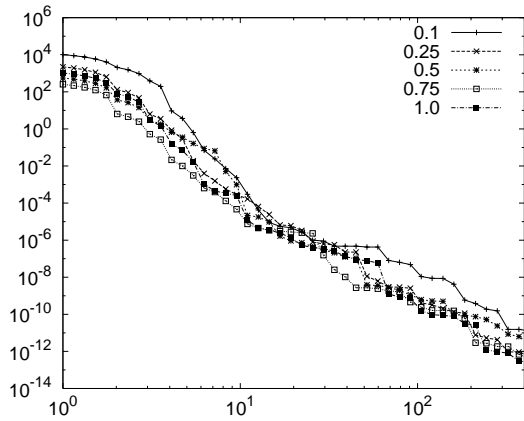
## 5. RESULTS

The results of all of the experiments are shown in Figures 4, 5, and 6. The  $x$ -axis of each graph shows one tic per 50 function evaluations, and the  $y$ -axis is the best fitness obtained among all particles. Because the focus is minimization, lower values are better. All results are averaged over 30 runs and plotted on a log-log scale.

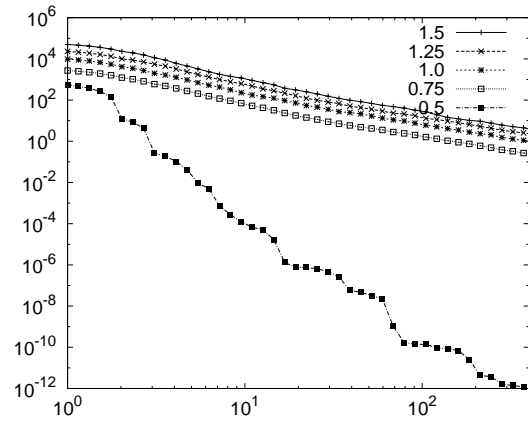
Figure 4 shows the results of the experiments using the Pivot method. Using RS the Pivot method appears to perform equally well in all cases, easily overcoming the difficulties imposed by a smaller initialization region. When CO is applied, however, the bias becomes evident. Pivot only performs well when the global minimum is located at or near the origin.

Figure 5 displays results for PSOGauss. In this case, the origin seeking bias is more subtle. The key is to look for natural grouping of results near the end of a run. With

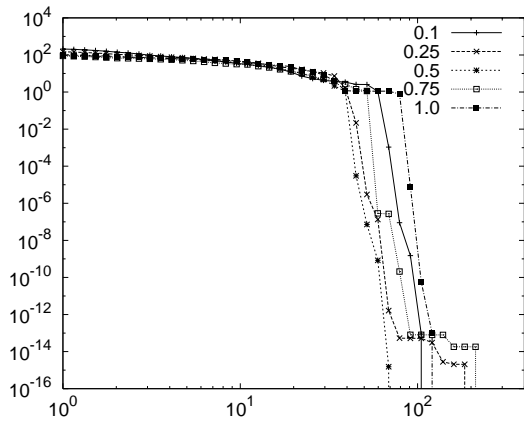
<sup>2</sup>This is somewhat different from Angeline’s approach, since the region is chosen from the opposite corner. While this does not affect the symmetric Sphere or Rastrigin functions, it tends to initialize particles in a more challenging part of Rosenbrock’s domain.



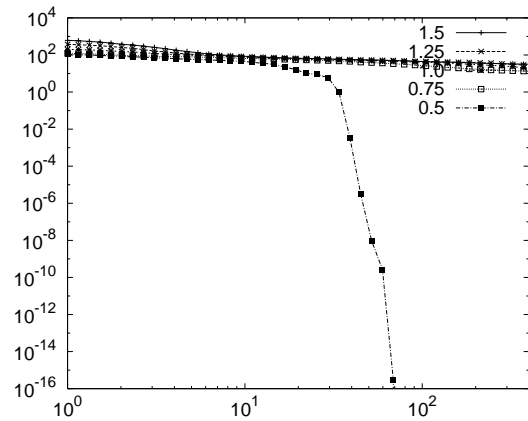
(a) Sphere iterations (RS)



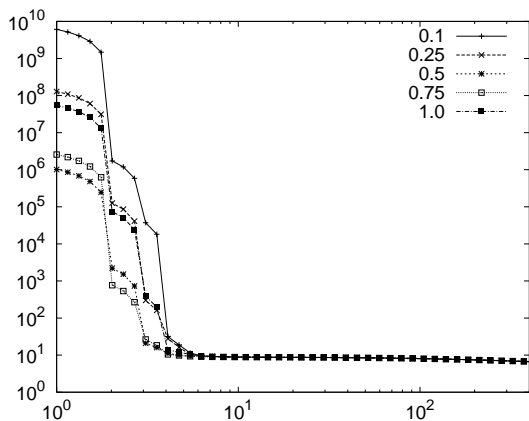
(b) Sphere iterations (CO)



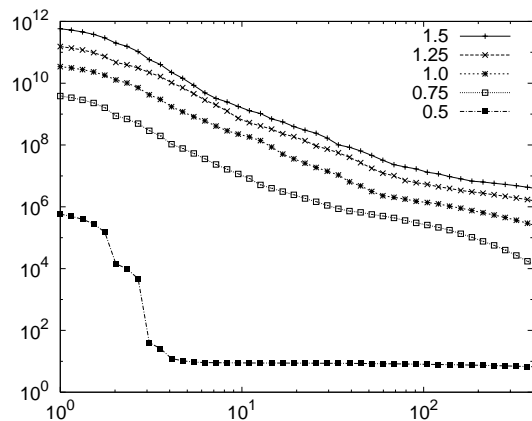
(c) Rastrigin iterations (RS)



(d) Rastrigin iterations (CO)

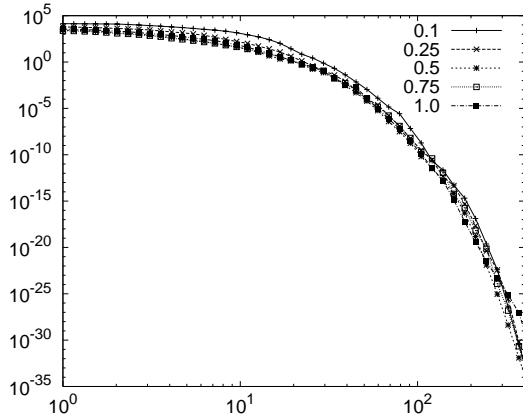


(e) Rosenbrock iterations (RS)

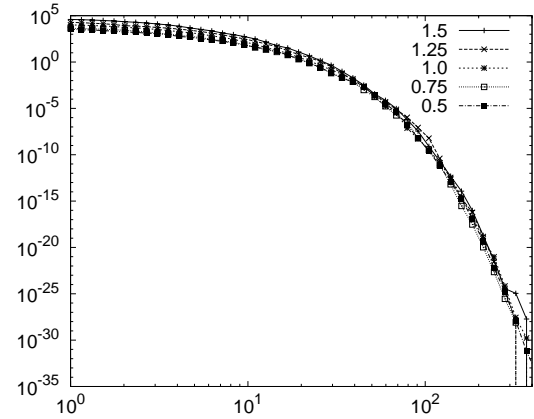


(f) Rosenbrock iterations (CO)

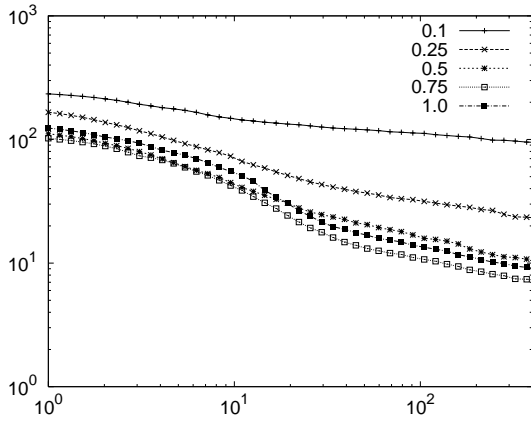
Figure 4: Pivot performance under Region Scaling and Center Offset



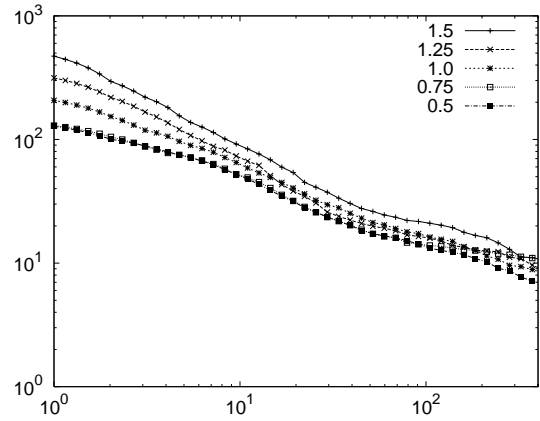
(a) Sphere iterations (RS)



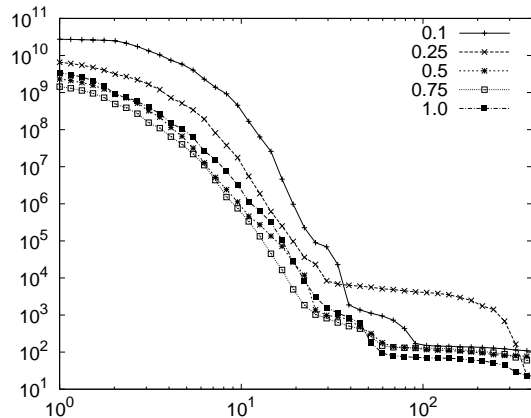
(b) Sphere iterations (CO)



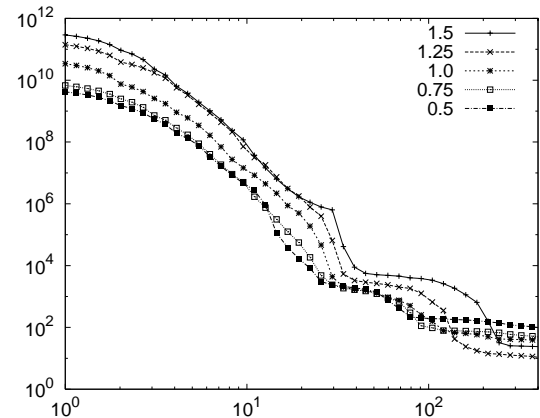
(c) Rastrigin iterations (RS)



(d) Rastrigin iterations (CO)

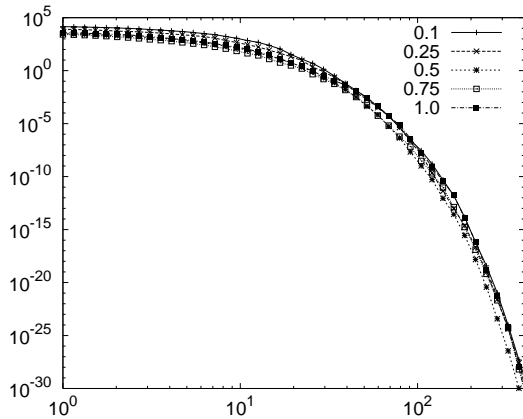


(e) Rosenbrock iterations (RS)

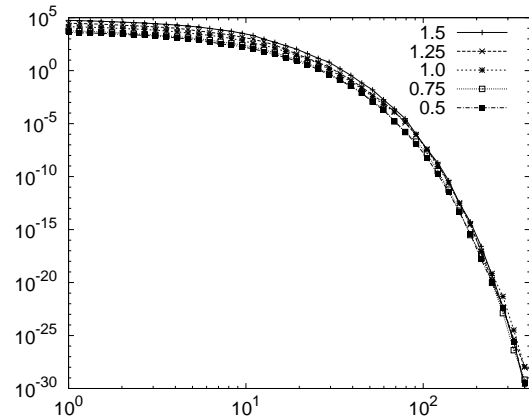


(f) Rosenbrock iterations (CO)

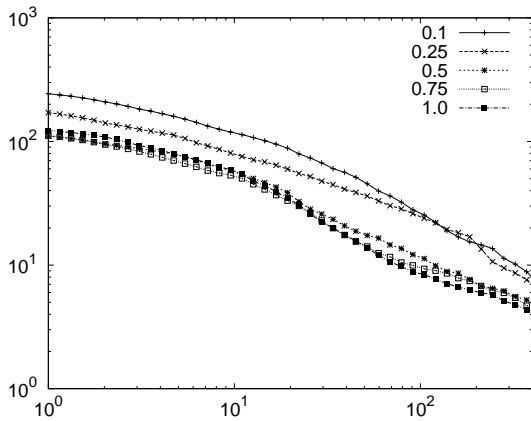
Figure 5: PSOGauss performance under Region Scaling and Center Offset



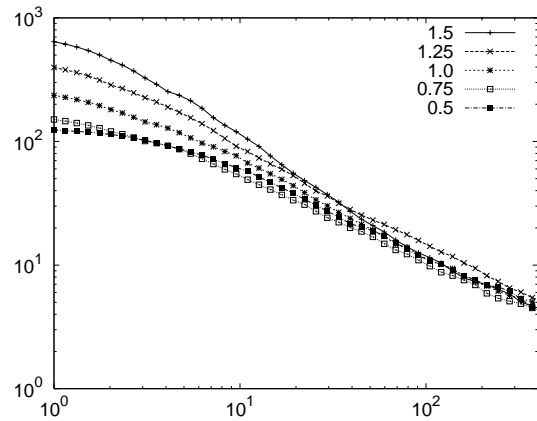
(a) Sphere iterations (RS)



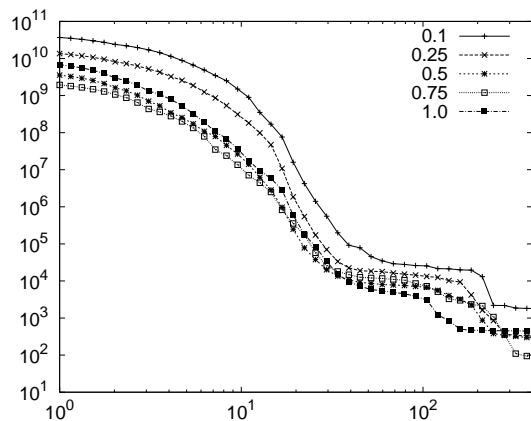
(b) Sphere iterations (CO)



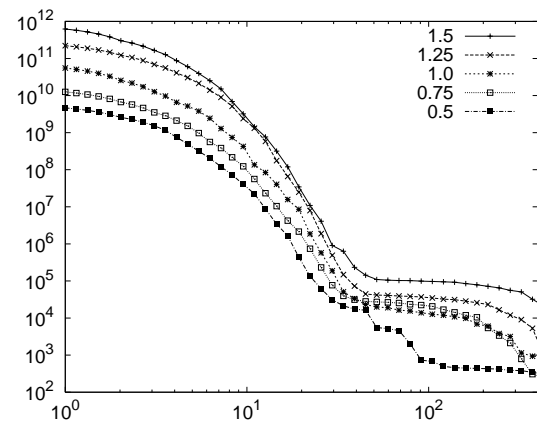
(c) Rastrigin iterations (RS)



(d) Rastrigin iterations (CO)



(e) Rosenbrock iterations (RS)



(f) Rosenbrock iterations (CO)

Figure 6: BareBones performance under Region Scaling and Center Offset

Rastrigin and RS, some clustering occurs among the regions that include the global minimum in Figure 5(c), but it does not show up when using CO in Figure 5(d). Rosenbrock and Sphere show no significant clustering in either case.

In Figure 6 the results are shown for BareBones. Similar to PSOGauss, clustering is observed with Rastrigin and RS in Figure 6(c) but not with CO in Figure 6(d). On Rosenbrock, however, clustering is definitely observed (note the log-log scales) both under RS in Figure 6(e) and CO in Figure 6(f). The clustering observed is much more striking under CO, with some counterintuitive results under RS. Again, no such clustering is observed when tested on Sphere.

## 6. DISCUSSION

### 6.1 Exposure Methods

The behavior of Pivot in Figure 4 suggests that a strong argument can be made for using CO to test for origin-seeking bias; it succeeded where RS failed. Additionally, when looking at results for Rosenbrock among all motion algorithms, anywhere that RS exposed a bias, CO did as well. In that sense, CO appears to be no worse than RS, and in the case of Pivot it is vastly better for discovering bias.

The Rastrigin case is somewhat different, where any bias shown on that function only occurred under RS. While this may say more about Rastrigin than any of the algorithms used to optimize it, it does expose weakness in both PSOGauss and BareBones when dealing with such a highly regular and multimodal function.

The types of bias exposed by these two approaches are different. RS exposes a bias toward the *center of the initialization region*, while CO exposes a bias *toward the absolute origin*. To further verify this idea, other experiments were performed that moved the center of the function and the initialization region by the same amount simultaneously. This simply performed a coordinate shift for the entire problem, something that would not be expected to cause difficulty for any of the algorithms here. The results for Pivot, however, were nearly identical to those shown here when using CO exclusively, indicating that there is indeed a bias toward the absolute origin in that algorithm.

It is possible to combine both the RS technique and the CO technique into a single experiment, shifting the coordinate system and then shrinking the initialization region. This approach can sometimes expose both kinds of bias at once, suggesting that if only one experiment is to be done, RS and CO should be combined. Otherwise, it is best to do each separately in order to expose the various potential algorithmic weaknesses.

### 6.2 TRIBES Behavior

The use of TRIBES as the basis for swarm size and sociometry, while not an arbitrary choice, merits further discussion. It was mentioned previously that TRIBES was chosen because it tends to find the right combination of sociometry and swarm size for effectively exposing the bias in a motion algorithm. It does this because it adapts swarm characteristics based on performance.

The biased behavior does not only show up when using TRIBES, however. Figure 7 gives results for BareBones on various fixed-size fully connected swarms. The experiments are performed under CO. It is especially clear in Figures 7(a) and 7(b) that BareBones displays origin-seeking

bias on Rosenbrock. It is therefore possible to find the behavior using a fixed swarm size and a specific sociometry, but it can be difficult to find the right combination by hand. More particles implies initially more diverse function samples and increases the likelihood of finding a good area to explore at the beginning of the run, making it difficult to see any bias that may exist. Fewer particles have little available information and therefore nearly *always* get stuck quickly, making it difficult to make any convincing statements about observed bias. TRIBES seems to get it just right.

That TRIBES is good at exposing the bias actually makes a very positive statement about the algorithm in general. The exposure occurs because it is using just enough particles and just enough connections between them to get the best results possible. In other words, when an algorithm is center seeking, it *exploits* that fact because it finds a good combination of swarm size and sociometry for that algorithm. It makes sense to test new algorithms using TRIBES because strange origin-seeking behavior may otherwise be masked by lucky choices of fixed swarm size and sociometry.

### 6.3 Benchmark Behavior

Rastrigin is somewhat unique among the benchmarks here in that it exposes a bias only under RS. The others expose it either in both cases or only in the CO case. Why does this happen with Rastrigin? The function is highly multimodal with very deep local minima spread out on a *regular grid*. In order for particles to find the global minimum, they must jump over or out of these local minima in a reliable way. If particles manage to acquire the correct speed, they tend to jump quickly from one minimum to the next since a straight line will pass through many evenly-spaced local minima. A larger initialization space facilitates the discovery of an appropriate velocity while a smaller one tends to generate particles that get stuck because of small initial velocities.

Rosenbrock has some counterintuitive behavior. When applying RS, the bias appears to be reversed in some cases. This may be due to its asymmetric properties; too many particles in a misleading area of the space (with strange local minima) can cause the swarm to converge too quickly to a challenging part of the domain. As the initialization region is made smaller but still includes the global minimum, fewer particles start out in misleading areas.

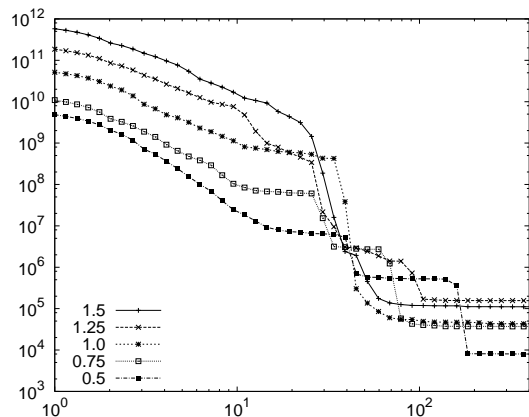
These properties do not discredit these benchmarks as indicators of bias, but rather highlight some of the unique issues that they expose. It is a good idea to use multiple different benchmarks when looking for bias.

### 6.4 Motion Algorithms

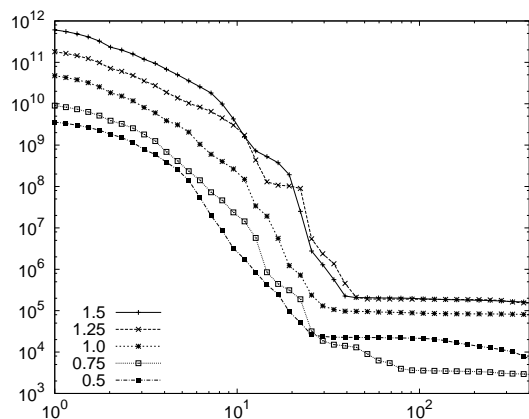
Pivot is undeniably biased, but what of the others? Between PSOGauss and BareBones, PSOGauss appears to display the least bias, since it works well on Rosenbrock no matter what is done to it. BareBones, on the other hand, appears to show significant bias on Rosenbrock. Both show a small amount of bias on Rastrigin when the region size is altered.

If they must be ranked, then, it appears that PSOGauss is the least biased, followed by the slightly more biased BareBones, finally followed by the extremely biased Pivot.

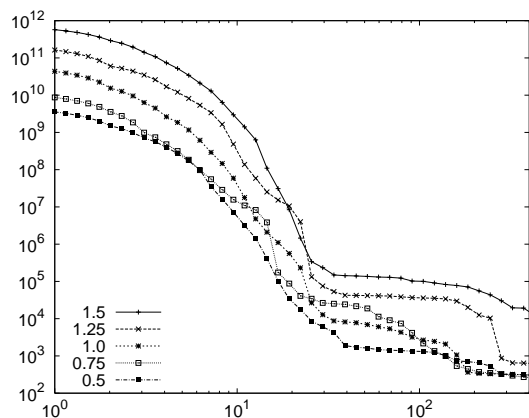
What it is about Pivot and BareBones that makes them biased is not obvious from the results. They are different from PSOGauss in one very fundamental way, however: they



(a) Rosenbrock iterations – 6 Particles



(b) Rosenbrock iterations – 8 Particles



(c) Rosenbrock iterations – 10 Particles

**Figure 7: BareBones average fitness using CO, the star sociometry, and various fixed swarm sizes**

update positions directly while PSOGauss updates velocities. This one difference may be enough to account for a center seeking bias, though that idea has not yet been fully explored. The reasons behind the extreme nature of Pivot's bias also merit further exploration.

## 7. CONCLUSION

Region Scaling (RS), a popular method of testing for bias in PSO, is effective but not always sufficient for detecting origin-seeking behavior. In fact, on motion like the Pivot method, it fails to expose any bias whatsoever. Center Offset (CO), on the other hand, catches cases that are not otherwise visible, making it an essential testing tool for any new PSO algorithm.

Additionally, it was found that TRIBES provides a useful framework for testing different kinds of PSO motion, given that it tends to exploit the best behavior of a motion algorithm. This, in combination with CO and RS is a very effective method of testing for origin-seeking bias.

## 8. REFERENCES

- [1] P. J. Angeline. Using selection to improve particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998)*, Anchorage, Alaska, USA, 1998.
- [2] M. Clerc. TRIBES - un exemple d'optimisation par essaim particulare sans paramètres de contrôle. In *Optimisation par Essaim Particulaire (OEP 2003)*, Paris, France, 2003.
- [3] M. Clerc. Math stuff about PSO. Online at <http://clerc.maurice.free.fr/ps/>, 2005. Repository of papers and source code for PSO algorithms including TRIBES.
- [4] M. Clerc and J. Kennedy. The particle swarm: Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, February 2002.
- [5] D. K. Gehlhaar and D. B. Fogel. Tuning evolutionary programming for conformationally flexible molecular docking. In *Evolutionary Programming*, pages 419–429, 1996.
- [6] J. Kennedy. Bare bones particle swarms. In *Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003)*, pages 80–87, Indianapolis, Indiana, 2003.