Confident but Weakly Informed: Tackling PSO's Momentum Conundrum

Christopher K. Monson
Google, Inc.
25 Massachusetts Ave, 9th Floor
Washington, D.C. 20001–1430
Email: chris@highentropy.com

Kevin D. Seppi
BYU Computer Science Dept.
3361 TMCB
Provo, UT 84602–6576
Email: kseppi@byu.edu

***Abstract*—Particle Swarm Optimization uses noisy historical information to select potentially optimal function samples. Though information-theoretic principles suggest that less noise indicates greater certainty, PSO's momentum term is usually both the least informed and the most deterministic. This dichotomy suggests that while momentum has a profound impact on swarm diversity, it would benefit from a more principled approach. We demonstrate that momentum can be made both more effective and better behaved with informed feedback, and that it may even be completely eliminated with proper application of more straightforward and well-behaved diversity injection strategies.**

I. Introduction

Particle Swarm Optimization (PSO) is a stochastic hill-climbing algorithm especially suitable for use in continuous domains. At its most basic, it initializes several "particles" with random positions and velocities in a limited region within the function's domain. This region is known as the "feasible rectangle", generally assumed to contain the global optimum. Each particle samples the function at its current position, examines the findings of its neighbors, and applies simple rules to reposition itself. This procedure is then repeated [1]. Particles maintain a modest amount of state including velocity, best known position, and best known function value. Additional state requirements can be found in the many published adaptations to the basic algorithm.

An effort has been made to codify a standard variant of PSO to be used as a baseline comparison for ongoing research [2]. In our experience, this "Standard PSO" can be very effective when carefully implemented. It uses a static, minimally-connected, symmetric "ring" topology with a swarm of 20 particles: each has exactly two neighbors. Other topologies are possible[1], including the popular fully-connected "star" topology at the other extreme, but the ring topology is simple and efficient [2].

The basic behavior of a single standard particle is described by the following update equations [1], [6], [2]:

$$\mathbf{v}_{t+1} = \chi\left(\mathbf{v}_t + \phi_p\mathbf{U}\circ(\mathbf{p}_t - \mathbf{x}_t) + \phi_g\mathbf{U}\circ(\mathbf{g}_t - \mathbf{x}_t)\right) \quad (1)$$

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

where $\phi_p = \phi_g = 2.05$ are the "cognitive" and "social" coefficients [7], each $\mathbf{U}$ is a distinct vector whose elements are independently drawn from a standard uniform distribution for each use, and $\circ$ represents element-wise multiplication. The constriction coefficient is typically $\chi \approx 0.72984$ [6], [2].

Particles update their positions based on their own history and that of their neighbors. This is embodied in $\mathbf{p}$, the location of the particle's best historical fitness, and $\mathbf{g}$, the location of the best historical fitness among its neighbors.

A common variant of (1) makes use of an inertia weight: a coefficient ω that controls the momentum term directly in place of the more broadly applied constriction χ [8]:

$$\mathbf{v}_{t+1} = \omega\mathbf{v}_t + \phi_p\mathbf{U}\circ(\mathbf{p}_t - \mathbf{x}_t) + \phi_g\mathbf{U}\circ(\mathbf{g}_t - \mathbf{x}_t) \quad . \quad (3)$$

Though possessing different analysis characteristics, these formulations are trivially reducible to one another, e.g., letting $\omega = 0.72984$ and $\phi_p = \phi_g \approx 1.5$ produces an equation that is equivalent to (1) with $\chi = 0.72984$ and $\phi_p = \phi_g = 2.05$.

The coefficient ω is typically between 0.4 and 1.0, though values close to 1.0 can cause divergence. When constriction values are not used (a departure from Standard PSO), the inertia weight is often decreased linearly during optimization [9].

Occasionally the velocity update is expressed in two parts, making the notation more convenient to manipulate:

$$\mathbf{a}_{t+1} = \phi_p\mathbf{U}\circ(\mathbf{p}_t - \mathbf{x}_t) + \phi_g\mathbf{U}\circ(\mathbf{g}_t - \mathbf{x}_t)$$
$$\mathbf{v}_{t+1} = \omega\mathbf{v}_t + \mathbf{a}_{t+1} \quad .$$

Although the inertia weight formulation can be equivalent to standard PSO, its coefficients are typically less tightly interrelated; it is therefore easy to choose values that cause divergence. Thus, $\mathbf{v}$ is subject to a per-dimension maximum $V_{\max}$, used to clip each element of $\mathbf{v}$ whose absolute value is too large before calculating $\mathbf{x}$. The elements of $V_{\max}$ are often set to the corresponding side lengths of the feasible rectangle.

While simple, the particle update in (3) exhibits complex and effective optimization behavior when applied to a swarm. It has the added benefit of being relatively straightforward to implement, though setting the free parameters properly for a given optimization task can at times be tedious and error-prone. To this end, some effort has been devoted to reducing the sensitivity of PSO to its various parameter settings, generating reasonable behavior on interesting classes of functions while reducing the configuration burden on the optimization practitioner [3], [10], [11].

[1] Indeed, the number of possible topologies grows exponentially with swarm size, and topology studies were popular in PSO publications for several years, including dynamic approaches [3], [4], [5].

Encouraging progress has been made in this regard, but despite many efforts to similarly improve the situation for ω, it continues to be a fairly opaque and sensitive parameter. More importantly, however, and the focus of this work, is that the highly influential momentum term is very much unlike the others when considering information and determinism.

We examine the use of momentum in PSO in light of basic principles of information theory, specifically that determinism is usually directly proportional to confidence, and that highly confident influences should therefore arise from high-quality information. In contrast, PSO's momentum term is typically both more confidently applied and less directly informed than its peers. We explore preliminary results that suggest that PSO can benefit from either better informing momentum or replacing it completely with other diversity enhancements.

II. Related Work

The coefficient ω has long been viewed as a means of controlling swarm exploration and diversity [12], [13], [14]. It is, however, difficult to predict how changing ω will impact swarm behavior on any given function. Numerous attempts have been made to improve the situation [12], [15], [16], and these generally fall into one of several classes:

- Constant inertia,
- Linearly decreasing or increasing inertia,
- Nonlinearly decreasing or increasing inertia,
- Random inertia of various kinds,
- Self-optimizing inertia, and
- All conceivable combinations of the above [16].

For any given problem set or research endeavor, one of these will be the most effective, but there no generally useful ranking [14], [16]; momentum remains somewhat opaque.

A. Bare Variants

An early, classic analysis of particle behavior led to the "Bare Bones" PSO [17], a method that simply samples from a distribution centered on the average of $\mathbf{p}$ and $\mathbf{g}$. This work did not produce the leading method for moving particles in a swarm, nor did it become the standard approach, but it *basically works*, and the idea of it has some interesting consequences. First, it dispenses with velocity (and thus momentum) entirely, directly moving particles to the region most likely to be promising given the information available in $\mathbf{p}$ and $\mathbf{g}$:

$$\mathbf{x}_{t+1} \sim \mathcal{N}\left(\frac{1}{2}(\mathbf{p}+\mathbf{g}), \mathbf{I}\|\mathbf{p}-\mathbf{g}\|_2^2\right) \tag{4}$$

This approach of sampling a position from a Normal distribution is elegant and intuitive, and works surprisingly well for having stripped off so many algorithmic features from the contemporary state of the art, including momentum. Momentum is missing from other notable algorithmic variants, as well, including the spherical uniform update strategy proposed with TRIBES [3], and HPSO-TVAC, where momentum is omitted in favor of alternate forms of diversity injection [13].

PSO can clearly function without momentum, but the large volume of literature following Bare Bones, including standard PSO itself, appears to argue strenuously for its retention. Momentum has an undeniable impact on swarm effectiveness, but it is hard to reason about, hard to control, and from a purely information-theoretic point of view, hard to motivate.

B. Dynamic Tuning

Practically speaking, a lack of grounding in information theory is not in itself a fatal problem. It is, however, not the only problem with momentum. Momentum is also hard to tune. This simple fact may go a long way toward explaining why the simplest inertia weight strategies, though sometimes improved upon by other more complex or even principled methods, remain the most popular: they are easier to understand, more straightforward to implement, and typically employ fewer tuning parameters.

The tedium and uncertainty inherent in tuning evolutionary algorithms, PSO included, has been the subject of many papers. Basic PSO has several free parameters, all of which can have a dramatic impact on performance. Among them are

ϕ_g : The social coefficient,
ϕ_p : The cognitive coefficient,
$V_{\max}$: The maximum speed in each dimension,
ω : The inertia weight,
N : The number of particles in the swarm,
And, of course, the swarm's topology.

To address swarm size and topology, Clerc invented TRIBES [3], a brilliant (if somewhat hard to follow without alternate exposition, e.g., [4]) approach to dynamically adjusting swarm size and topology based on current swarm performance. This method is comparatively nontrivial to implement and never achieved great popularity, but it is very effective and removes the considerations of both swarm size and topology simultaneously: the swarm starts out small and grows or shrinks as needed while adjusting information links between particles based on performance.

PSO, especially when applied to multi-modal functions, is known to benefit from supplemental mechanisms that increase swarm diversity at appropriate times (e.g., SEPSO [18], [19] or ARPSO [20]). Though useful, such mechanisms necessarily increase the number of tuning parameters. The Contracting Radius Increasing Bounce SEPSO (CRIBS) and similar algorithms represent approaches to diversity that make performance less sensitive to these parameterizations by adapting them dynamically, using swarm behavior to inform parameter adaptations over time [10]. This not only improves performance of the underlying mechanisms, but also relieves the practitioner of the need to think carefully about initial settings: the swarm will adapt and quickly recover from poor settings, allowing optimization to proceed succesfully.

In a similar spirit, the Simple Adaptive Cognition (SAC) algorithm dynamically adjusts ϕ_g and ϕ_p over time using swarm behavior feedback, and improves performance of multiple kinds of swarm algorithms on a variety of functions [11].

It seems evident that the concept of informed feedback is a powerful one: if a swarm can adapt its own parameters based on information gained from the task at hand, it stands to reason that it should behave more consistently and robustly. This is, in fact, what is observed when applying feedback-driven parameter adaptation via TRIBES, CRIBS, and SAC. It would be useful to apply this principle to the inertia weight.

III. The Trouble With Momentum

The major ideas in Section II, specifically omitting momentum and/or finding a way to tune it more or less automatically, are more closely related than they first appear.

In (3) we see that each particle is acting on uncertain information: it determines the most likely candidates for good fitness by noisily combining its own history with that of its neighbors, encoded in ϕ_p and ϕ_g, respectively. This added noise represents uncertainty inherent in the sparse information available about the fitness function while simultaneously restricting the next region of exploration to one that is the most informed. In contrast, the momentum term popularly enjoys a deterministic status; it is certainly always influential.

When momentum is *not* deterministic, as in one of the early adaptations that selected it randomly from the interval $[0.5, 1.0)$ [12], [14], [16], the added noise is not usually *informed*. Whatever momentum's level of determinism, the quantity of information on which it is based is generally small.

The notation of (3) is well-suited to implementation, but a formulation in terms of random variables will be more helpful here. Consider the following equivalent formulation:

$$\mathbf{P}_{t+1} \sim \mathbf{U}\left[\mathbf{0}, \phi_p(\mathbf{p}_t - \mathbf{x}_t)\right] \tag{5}$$
$$\mathbf{G}_{t+1} \sim \mathbf{U}\left[\mathbf{0}, \phi_g(\mathbf{g}_t - \mathbf{x}_t)\right] \tag{6}$$
$$\mathbf{v}_{t+1} = \omega\mathbf{v}_t + \mathbf{P}_{t+1} + \mathbf{G}_{t+1} \tag{7}$$

Here, $\mathbf{P}$ and $\mathbf{G}$ are sampled from a multivariate uniform distribution over independent variables bounded by a hyper-rectangle. These are then used to produce a new velocity.

The idea of rewriting the equation in terms of distributions can be taken still further, since it is a well-known statistical result that obtaining the sum of two samples from two independent distributions is the same as taking a single sample from the convolution of those distributions. More concretely, adding samples from $\mathbf{P}$ and $\mathbf{G}$ is the same as sampling from a single distribution $\mathbf{C}[\mathbf{a}, \mathbf{b}] \equiv \mathbf{U}[\mathbf{0}, \mathbf{a}] \star \mathbf{U}[\mathbf{0}, \mathbf{b}]$ obtained from convolving the two uniforms:

$$\mathbf{\Theta}_{t+1} \sim \mathbf{C}\left[\phi_p(\mathbf{p}_t - \mathbf{x}_t), \phi_g(\mathbf{g}_t - \mathbf{x}_t)\right] \tag{8}$$
$$\mathbf{v}_{t+1} = \omega\mathbf{v}_t + \mathbf{\Theta}_{t+1} \ . \tag{9}$$

An example of one such convolution is depicted in Figure 1.

Keep in mind that this is exactly the same update as before, merely differently perceived. Figure 1 also lends some intuition to Bare Bones PSO and its use of a Gaussian distribution centered on a point between $\mathbf{p}$ and $\mathbf{g}$; another well-known statistical result indicates that contributing additional uniform distributions to the convolution (as would be the case in, e.g., a "Fully-Informed" particle swarm [21]) would cause it to approach a Gaussian distribution.

The roughly pyramid-shaped distribution $\mathbf{C}$ represents information about promising areas of the fitness function's domain. The $\mathbf{C}$ distribution attempts to capture both this information and the inherent uncertainty in not yet having sampled every possible location.

There are, of course, ways in which this distribution can be justifiably critiqued. For example, the mode of $\mathbf{C}$ is centered on the midpoint between $\phi_p(\mathbf{p} - \mathbf{x})$ and $\phi_g(\mathbf{g} - \mathbf{x})$. When these point in opposite directions, it "compromises" in an overly naive way that lends weight to areas of the space that neither vector indicates. This may explain why the Simple Adaptive Cognition (SAC) method is successful: it gives more weight to the vector that has the most recent update, skewing this distribution toward the most promising information [11]. Even without SAC, however, disagreeing vectors cause the probability to spread out more, sensibly (though somewhat arbitrarily) encoding the uncertainty inherent in disagreement.

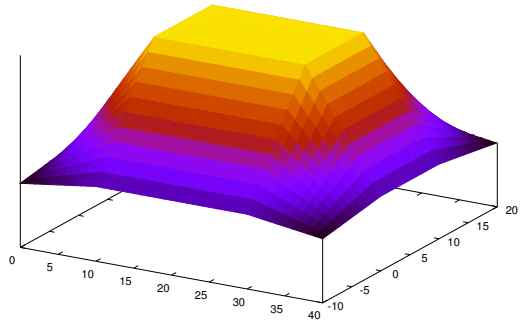

Fig. 1. The convolution of $\mathbf{U}\left[(0,0),(30,20)\right]$ and $\mathbf{U}\left[(0,0),(10,-10)\right]$. By convention, the particle is located at (0, 0) for any such distribution.

But again, the momentum term represents a quandary. It effectively shifts the distribution $\mathbf{C}$ by an amount related to the particle's current velocity, and it does so with a great deal of confidence, *regardless of whether this shift agrees with acquired data or not*. Indeed, it does not arise *directly* from observed data at all. This momentum shift is, in a very real sense, the most influential determiner of the new velocity; far from *informing* $\mathbf{C}$ about potential regions of interest, it simply *moves the distribution somewhere else*.

Of course, to claim that even standard momentum is completely deterministic is a bit of an overstatement, as it inherits uncertainty from the previous velocity calculation and, by extension, from all previous velocity calculations in its history. There is, however, no clear reason that a particle's current velocity should have *more* certainty than newly-acquired data about locations of good fitness, and it is easy to imagine a great many reasons why it should not.

All of this leads to the critical question of what might be done instead. On the one hand, the determinism enjoyed by momentum raises some questions that make adding noise an obvious thing to try, but prior attempts at adding noise have not produced a clear winner. To answer this difficulty, we observe that the problem is not merely certainty, but the *lack of information* from which a more correct level of certainty might be derived. Blind attempts to merely suppress certainty (e.g., through randomization [12]) are therefore every bit as difficult to tune as the original algorithm: the level of desired certainty (and thus the characteristics of any added noise) is not tied to real information and is therefore impossible to guess. To improve the situation, we take inspiration from feedback-based auto-tuning historically applied to other PSO update terms.

Parabola (Sphere): $(-50, 50)$	$f_{\mathbf{c}}(\mathbf{x})$	$= \|\mathbf{x} - \mathbf{c}\|_2^2$
Ackley: $(-5, 5)$	$f_{\mathbf{c}}(\mathbf{x})$	$= 20 + e - 20 \exp\left(\frac{-\|\mathbf{x} - \mathbf{c}\|_2}{5\sqrt{D}}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos 2\pi(x_i - c_i)\right)$
Rastrigin: $(-5.12, 5.12)$	$f_{\mathbf{c}}(\mathbf{x})$	$= \|\mathbf{x} - \mathbf{c}\|_2^2 + 10\sum_{i=1}^{D} 1 - \cos(2\pi(x_i - c_i))$
Rosenbrock: $(-100, 100)$	$f_{\mathbf{c}}(\mathbf{x})$	$= \sum_{i=1}^{D-1} 100\left((x_{i+1} - c_{i+1}) - (x_i - c_i)^2\right)^2 + (x_i - c_i - 1)^2$

TABLE I. BENCHMARK FUNCTIONS, WITH ASSOCIATED PER-DIMENSION INITIALIZATION BOUNDS AND OFFSET VECTOR $\mathbf{c}$.

IV. INFORMING MOMENTUM

Feedback cannot be useful without a connection to the parameter it adapts. An illustrative example of this is the SAC adaptation to PSO. The idea behind SAC is that newer information is better information; all else being equal, a particle favors moving toward locations that have been discovered more recently over those that were discovered long ago:

$$\begin{aligned}\mathbf{v}_{t+1} = \omega\mathbf{v}_t &+ \gamma^{t-t_p}\phi_p\mathbf{U} \circ (\mathbf{p}_t - \mathbf{x}_t) \\ &+ \gamma^{t-t_g}\phi_g\mathbf{U} \circ (\mathbf{g}_t - \mathbf{x}_t) \quad .\end{aligned} \tag{10}$$

Here, the social and cognitive terms are selectively suppressed by adaptation factor $\gamma \approx 0.99$, adjusted by the time since the last local update t_p and the last neighborhood update t_g. As these terms are fundamental to all PSO algorithms, this can be applied nearly universally with good results [11].

This simple adaptation causes PSO to behave more consistently, possibly because in suppressing the extent of one or more uniform distributions, it can also sharpen the peak of the convolution $\mathbf{C}$ (adding informed certainty). Furthermore, it has the convenient property that it reverts to the standard formulation of (3) not only when $\gamma = 1$ *but also* any time an update of the best known locations occurs (allowing it to continue performing as well as standard PSO on "easy" functions like "Parabola", where updates are frequent).

This sort of feedback is easy to justify because it is closely tied to particle behavior: at each step, an assumption is made that favorable regions tend to be close to other favorable regions, and the particle attempts to explore in that vicinity. Furthermore, because updates only occur when a *better* location is discovered, a newer update is likely to be in a more promising area: parameter and feedback are connected.

The principle of informed feedback also applies to diversity-increasing techniques, as exemplified in the Contracting Radius Increasing Bounce SEPSO (CRIBS) algorithm, summarized here in preparation for a discussion of alternative momentum strategies. The basic Spatial Extension PSO (SEPSO) endows particles with volume and causes them to bounce off of one another. CRIBS adapts both particle radius and bounce distance as more collisions occur [10]. A collision is detected for particle i when the following is satisfied:

$$\exists_{j \neq i}. \|\mathbf{x}_i - \mathbf{x}_j\| < (\beta^{b_i} + \beta^{b_j})r \tag{11}$$

typically with $\beta = 0.9$, r one tenth the length of the feasible rectangle's longest diagonal, and b the number of times a particle has experienced a collision. Note that letting $r = 0$ restores standard non-bouncing behavior. Like SAC, this algorithm automatically approaches standard PSO behavior: the radius r is adjusted toward 0 after each collision.

With the detection of each collision, the calculated velocities and positions of the corresponding particles are modified thus before being used to generate a new fitness sample:

$$\mathbf{v}_{t+1} \Leftarrow -\mathbf{v}_{t+1} \tag{12}$$

$$\mathbf{x}_{t+1} \Leftarrow \mathbf{x}_t - \beta^{-b}(\mathbf{x}_{t+1} - \mathbf{x}_t) \quad . \tag{13}$$

Again, this kind of feedback is easy to justify: a particle that is consistently colliding with other particles indicates that there is *consensus* that the region contains something worth sampling. The radius contracts and the bounce distance increases so that increasingly rare bounces are also more exploratory. Furthermore, while particles may begin with large r and bounce frequently, they rapidly become smaller and collide less frequently. This allows the practitioner to be somewhat more cavalier about initial parameter settings.

Ideally, similar kinds of feedback would apply to momentum, allowing it to adjust automatically to the current function. The core question is this: what can be measured that meaningfully connects the current swarm state to the *previous trajectory*? In the case of ϕ_g and ϕ_p, the update age at least provides an obvious and useful proxy for information quality. In the case of CRIBS, each collision means that the level of detail should increase. For ω, there is nothing quite so obvious.

A. Feedback for Momentum

One class of techniques that can be applied to momentum is stochastic learning. The basic premise is to allow momentum to take a random walk, nudging it toward values that correlate with past performance. For example, keeping track of a time-weighted average of ω values for which a new swarm-global best is found, one might favor that average over time, eventually converging to a suitable value for the current task.

Several variations on this theme have been proposed with varying levels of consistency and success [16]. Our own attempts produced mixed results not effective enough to warrant the additional algorithmic and tuning complexity. A software bug, however, led to a discovery: some variants of PSO, particularly those with added diversity such as CRIBS, appear to work well with $\omega = 0$. This is not only consistent with results elsewhere [13], but is generally suggestive that momentum as a means of controlling exploration is not something to be taken for granted, and it may not be needed as often, or even at all.

Armed with the feedback principles and examples above, we now outline new strategies for momentum and demonstrate their behavior on the common benchmarks in Table I. In all cases, Standard PSO is used as the baseline, not only

because it is standard [2], but also because it is effective on a broad class of functions. SAC and CRIBS, when indicated, are implemented as previously described. All graphs show an average of 20 runs.

Where Standard PSO is indicated, the adaptive and diversity constants that govern SAC and CRIBS, respectively, are disabled with $\gamma = 1.0$ and $r = 0.0$. The baseline constriction settings are $\phi_p = \phi_g = 1.5$ and $\omega = 0.72984$. Where SAC is indicated, $\gamma = 0.99$, and where CRIBS is indicated, r is one-tenth the longest diagonal of the feasible rectangle.

B. Informed Truncation of Momentum

Zero momentum can be effective [17], but is still less effective *in general* than Standard PSO (unless additional diversity mechanisms are employed, as we will see later). Perhaps, then, the balance between exploration and exploitation is not found in the relative strengths of the momentum term and its peers, but in a compromise bewteen zero momentum and non-zero momentum. Viewed in that way, is it possible to predict *when* momentum is useful and cut it off selectively when it is not? To answer this question, we must first discover what information can appropriately be used to make such a determination.

The most immediate and salient information available to a particle at any time is the location of the best known values in its own history and among its neighbor(s). The "Random Momentum Truncation" algorithm that follows is an outgrowth of the following reasoning: if momentum is taking a particle away from the location that would *otherwise be chosen without it* (high-probability regions of $\mathbf{C}$), then it should generally be cut off. If, on the other hand, a particle's momentum more or less *agrees* with the information at hand, then it should be allowed to have some influence. Coarsely, momentum is only assumed to be useful when it points toward $\mathbf{p}$ and $\mathbf{g}$. In keeping with the stochastic nature of PSO, truncation is applied randomly, utilizing a Bernoulli trial where the probability of truncation τ is taken from the strength of disagreement, thus[2]:

$$\tau_{t+1} = \min\{0, -\hat{\mathbf{a}}_{t+1} \cdot \hat{\mathbf{v}}_t\} \ . \tag{14}$$

If the trial with probability τ_{t+1} is successful, $\omega = 0$ for the calculation of $\mathbf{v}_{t+1}$, otherwise it assumes its normal value. Thus, ω can be truncated to zero if $\mathbf{v}_t$ is not pointing toward $\mathbf{a}_{t+1}$, the sample from $\mathbf{C}$. The stronger the evidence against momentum, the more likely that it will be truncated. Note that, in contrast with earlier methods that merely add noise to momentum, this approach adds *asymmetric adaptive noise*, based on what the information at hand indicates is appropriate.

The result of this operation is shown in Figure 2, where the lines with the "-R" suffix indicate the application of random momentum truncation as described above. It is noteworthy that with the exception of Rastrigin, none of the applied methods are *failing*; they all find the global optimum. In the case of Rastrigin, however, the methods exhibit distinct behavior, but none of the approaches are *succeeding*. Interestingly, when looking at those functions where PSO succeeds, the gains reported in the published description of SAC are still present by the end of a session, but the effect of momentum truncation is even more pronounced.

[2] $\hat{\mathbf{a}} = \mathbf{a}/\|\mathbf{a}\|_2$, the unit-length vector pointing in the same direction as $\mathbf{a}$.

Unsurprisingly, swarm behavior on Rosenbrock is less consistent and on Rastrigin it suffers from stagnation. As both of these are multi-modal functions that are typically prone to premature convergence, an informed diversity-enhancing method like CRIBS is prescribed.

Adding swarm diversity produces a new set of interesting results shown in Figure 3. Here the SAC lines have been removed, as the effects are not significant enough compared to those of random truncation to warrant the added visual noise. Not unexpectedly, the convergence of the swarm on some of the easier functions has slowed, but the effect of this is mitigated where random truncation is employed; it appears to have retained more of its ability to seek minima in spite of the artificial injection of diversity that comes from particle bouncing. Apparently two diversity injection mechanisms working at full capacity (momentum + CRIBS) is one too many, and suppression of momentum is helping convergence. Progress on Rosenbrock has also slowed, but again much less with random truncation than without.

Also unsurprising is the fact that Rastrigin responds well to the extra injection of diversity. Importantly, in all cases random truncation represents a measurable improvement over the best results achieved and in the speed of obtaining them. Any gains on Rastrigin are modest at best, but the addition of randomly truncated momentum is certainly not harmful.

The use of random truncation is also interesting for its tuning properties. PSO momentum has a significant impact on swarm exploration, but it is hard to adjust it to affect exploration *directly and predictably*. With that in mind, consider the following table showing the frequency with which a particle's normalized velocity $\mathbf{v}_t$ has a positive dot product with the normalized $\mathbf{a}_{t+1}$:

	Standard	Standard-R
Parabola	0.005	0.017
Ackley	0.005	0.021
Rastrigin	0.002	0.008
Rosenbrock	0.005	0.017

Besides the fact that agreement is uncommon in general, it is notable that random truncation significantly increases its frequency, providing a potentially more direct tuning methodology for swarm exploration than can be achieved with the inertia weight alone. For example, one might choose to *directly and predictably* increase exploration by triggering truncation when the dot product falls below -0.1 instead of 0.

C. No Momentum

Removing momentum altogether in a diversity-enhanced environment (Standard PSO with CRIBS and SAC) is an interesting exercise that is not without precedent [13], though such results have not necessarily received deserved attention. This may be due to the fact that, without the diversity enhancements, zero momentum fares so poorly as to not merit further experimentation. When used in conjunction with diversity injection, however, some intriguing results emerge. Consider Figure 4. Here we see these four variations of PSO:

- **Standard-R**: The Standard PSO algorithm from before, with CRIBS, SAC, and Random Truncation applied.

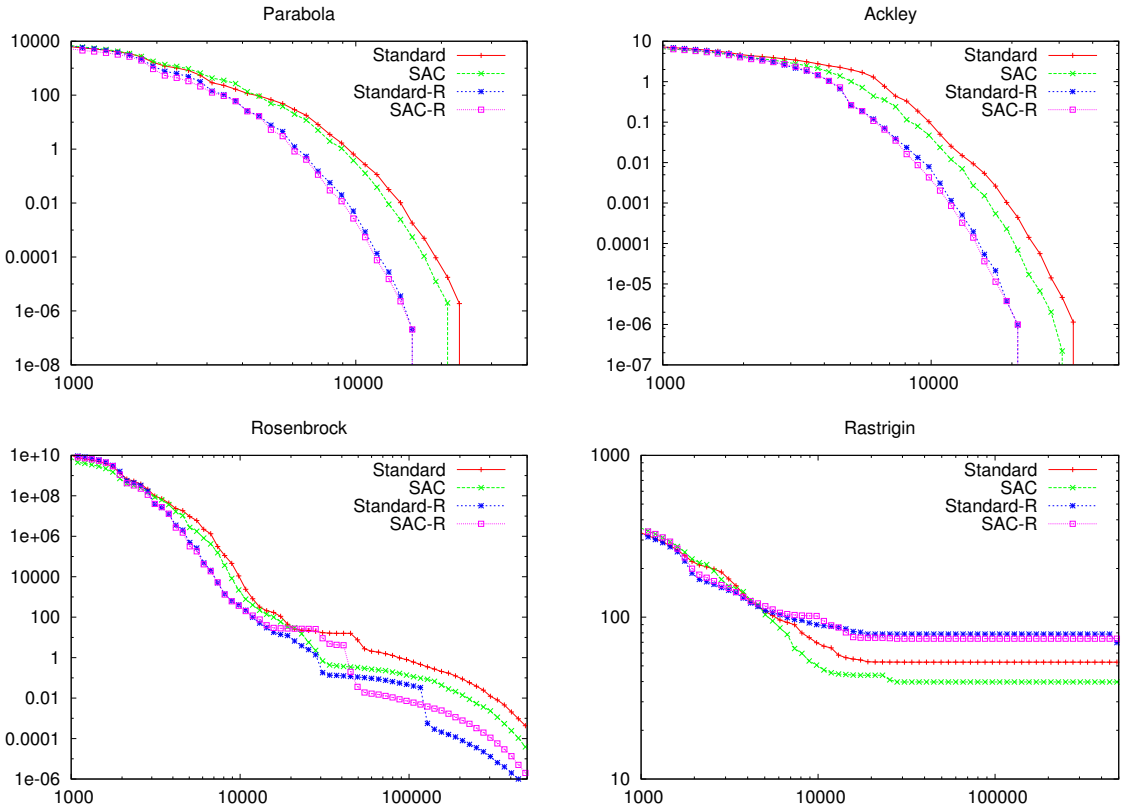

Fig. 2. Random truncation compared with standard PSO, with and without SAC applied.

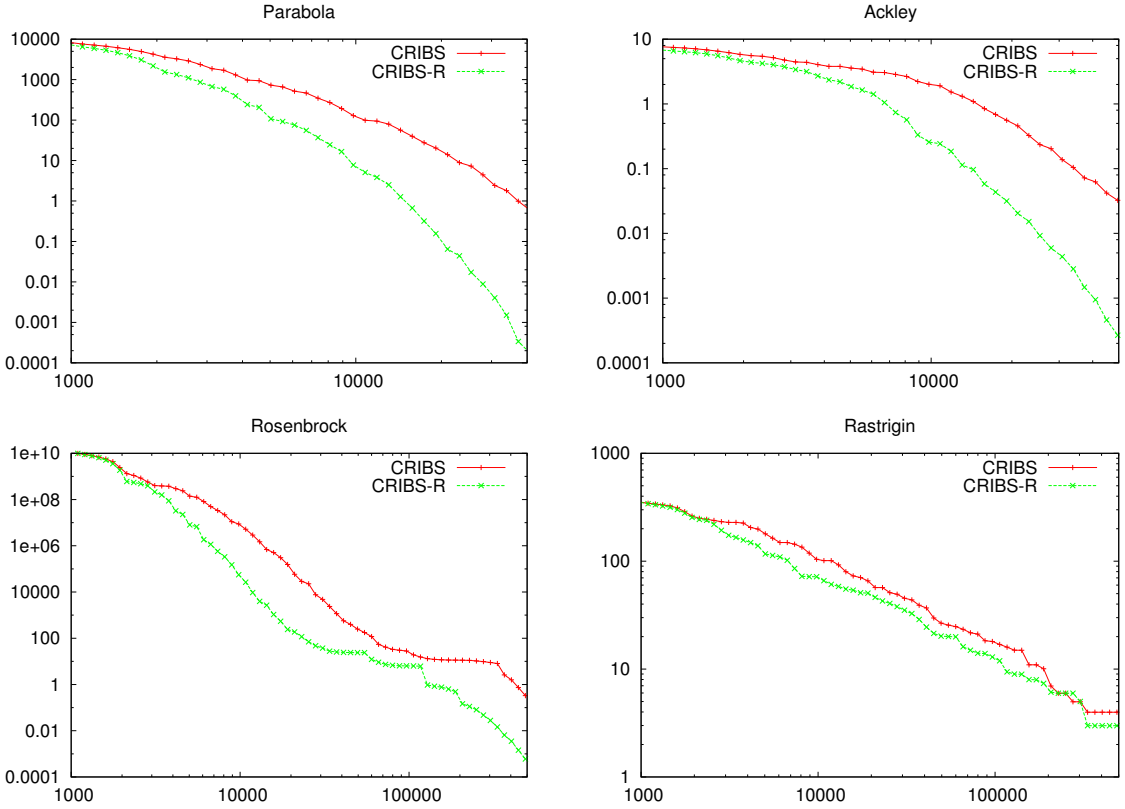

Fig. 3. CRIBS (adaptive diversity injection) with and without random truncation.

- **Standard-Zero**: The same algorithm, but with $\omega = 0$.
- **Zero**: The same as Standard-Zero, but with pre-constriction values $\phi_g = \phi_p = 2.05$, set higher to aid in exploration.
- **Zero-Neg**: The same as Zero, but with a simple twist on the way that $\mathbf{U}$ is used, described below.

In these graphs, we use "Standard-R" as the baseline, and compare it with various ways of removing momentum from a swarm. Let us ignore "Zero-Neg" for the moment.

The behavior of omitting momentum on the Parabola function is not surprising. It is convex and therefore most exploration effort is wasted: there is a perfect inverse correlation between distance to the optimum and fitness. Though Ackley is not convex, once the neighborhood of the optimum is discovered it exhibits this behavior as well. Zero momentum apparently works reasonably well on easy (or the easy part of) functions, consistent with results published elsewhere [13].

In the case of Rosenbrock, removing momentum altogether appears to harm the swarm's efficiency, but not in all cases, and not definitively; the swarm continues to progress and successfully finds the region of the global minimum. It is noteworthy that the zero-momentum variant with the larger values for ϕ_p and ϕ_g ("Zero") makes progress where the other ("Standard-Zero") does not. Rastrigin responds similarly well to the larger social and cognitive coefficients. This agrees with intuition: larger social and cognitive coefficients cause the corresponding $\mathbf{C}$ distribution to cover more space, increasing exploration. It would thus appear that the cognitive and social terms might be used to *directly compensate for the loss of momentum*. This suggestive result motivates "Zero-Neg".

The "Zero-Neg" approach uses the same update equations and parameters as "Zero", but instead of drawing each of the random vector elements from $\mathbf{U}[0, 1]$ as in (1), it draws them instead from $\mathbf{U}[-0.15, 1]$. This expands $\mathbf{C}$ to include areas "behind" the particle, allowing it to occasionally move opposite the more informed area. There is no momentum present, but it works well. It would seem that the exploration typically provided by momentum can indeed be obtained in a completely different way.

It bears repeating that these last results are with a PSO variant that includes both cognitive adaptation and artificial diversity injection. However, the methods employed are self-tuning, and have been demonstrated to be minimally invasive where they do not help [11], [10]. The omission of momentum certainly reduces swarm diversity [13], but additional diversity-increasing methods are necessary on harder functions *anyway*, and the idea that the responsibility for the remaining effects of momentum might be successfully turned over to more directly-informed terms is very attractive. If we cannot find consistently motivated and effective ways to use momentum to control swarm diversity and exploration, why not eliminate it?

V. Conclusions and Future Work

PSO's momentum term has some suboptimal characteristics as a means of controlling swarm exploration, born out by its history and the persistent lack of consensus on what it does and how to use it. Furthermore, it is unique among the terms in the PSO update equations, simultaneously in the significance of its influence and in the paucity of its motivating information.

The application of informed random truncation of momentum is a promising idea that is simple to implement and understand as well as improving performance when stagnation is not otherwise an issue. Perhaps even more interesting, however, is that this kind of informed feedback can also provide practitioners with a more direct means of tuning momentum-based exploration, e.g., through adjustment of sensitivity to vector disagreement. The idea has been briefly introduced in this work, and further study is warranted.

The potential for an adaptive feedback mechanism like random truncation to reduce algorithmic sensitivity to the inertia weight is also important. Successfully finding the best values for ω has historically been a difficult task, with many available options and few predictive clues. Effective application of momentum is thus something of a black art, and the strategies for dealing with it are as varied as their results on various classes of fitness function. PSO seems to respond well to information-oriented approaches to self-tuning for many of its parameters, including swarm size and topology, cognitive and social coefficients, and diversity injection. Now it can also be seen to respond well to a more informed, automatically adjusted momentum, though there are many variations left to attempt and there is clearly room for improvement on those reported here.

When PSO is viewed as a stochastic sample-based algorithm whose informed distribution is capriciously relocated by momentum, it becomes appealing to simply remove the term. Doing so may be achievable by reassigning the responsibility for exploration and diversity to other more directly-informed (and thus responsive to automatic tuning) mechanisms. Among these are older methods such as CRIBS and SAC, as well as newer methods introduced here, such as the introduction of negative uniform bounds for the cognitive and social terms. The behavior of these approaches is encouraging and suggests that there is more to be uncovered.

There are many potential consequences of a definitive removal of momentum, should such an objective be achieved. Mathematical analysis of PSO behavior would undoubtedly be simplified due to the loss of a momentum term, allowing statistical tools to be more directly brought to bear; after all, at that point the core algorithm would be a relatively straightforward sample from a distribution (like Bare Bones), while still retaining its useful and emergent neighborhood-based complexity (rather *unlike* Bare Bones). Furthermore, the question of divergence would practically disappear, since velocities would no longer be able to grow without bound unless constricted or capped by something like $V_{\max}$. All of these ideas would be interesting to explore in future work.

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *International Conference on Neural Networks IV*. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.

[2] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2007)*, Honolulu, HI, 2007, pp. 120–127.

[3] M. Clerc, "TRIBES - un exemple d'optimisation par essaim particulaire sans paramètres de contrôle," in *Optimisation par Essaim Particulaire (OEP 2003)*, Paris, France, 2003.

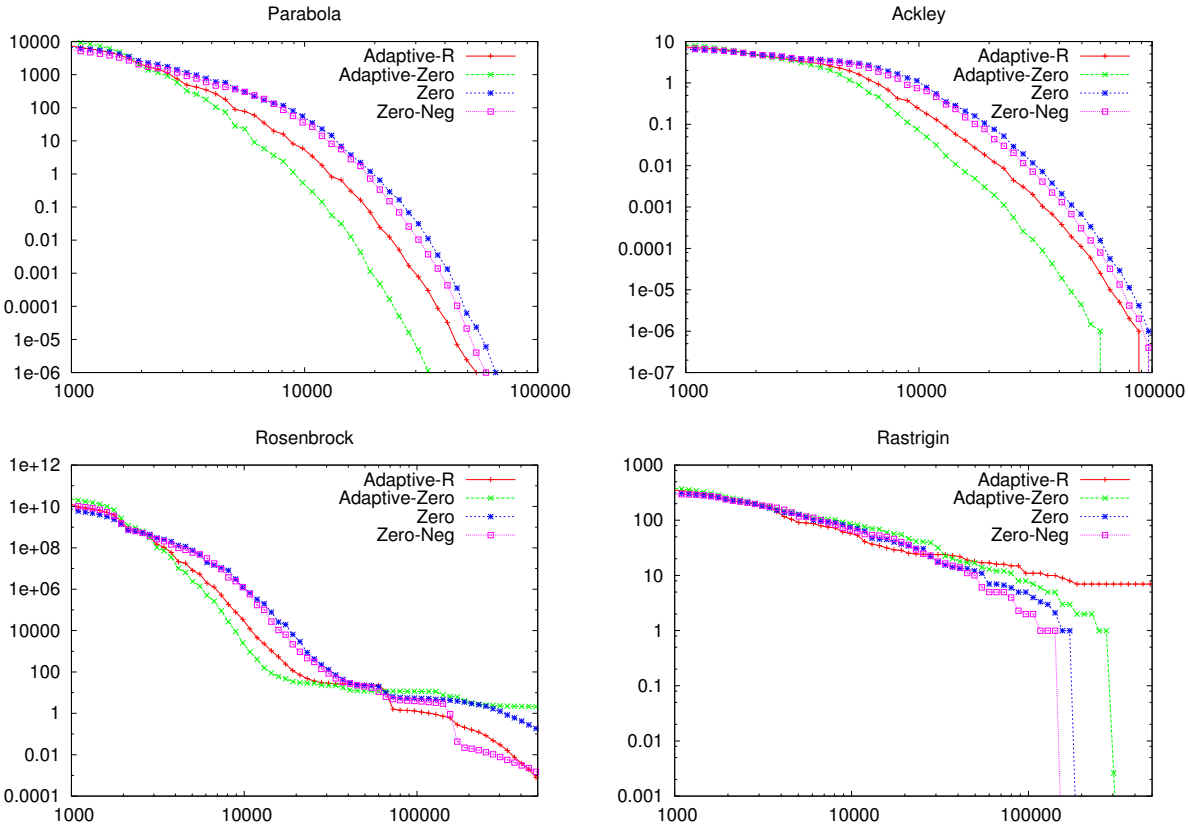

Fig. 4. Comparison of Standard-CRIBS-SAC with zero-momentum variants, with and without negative cognition.

[4] C. K. Monson and K. D. Seppi, "Exposing origin-seeking bias in PSO," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005)*, vol. 1, Washington, D.C., 2005, pp. 241–248.

[5] M. Richards and D. Ventura, "Choosing a starting configuration for particle swarm optimization," in *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IJCNN 2004)*, vol. 3. Piscataway, NJ, USA: IEEE Press, 2004, pp. 2309–2312.

[6] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, February 2002.

[7] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.

[8] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998)*, Piscataway, New Jersey, 1998.

[9] ——, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming*, New York, 1998, pp. 591–600.

[10] C. K. Monson and K. D. Seppi, "Adaptive diversity in PSO," in *Proceedings of the 8th Annual conference on Genetic and Evolutionary Computation (GECCO 2006)*. Seattle, Washington: ACM, 2006, pp. 59–66.

[11] C. K. Monson, "Simple adaptive cognition for PSO," in *Proceedings of the Congress on Evolutionary Computation (CEC 2011)*. New Orleans, Louisiana: IEEE, 2011, pp. 1657–1664.

[12] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001)*, Seoul, Korea, 2001.

[13] A. Ratnaweera, S. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, June 2004.

[14] M. A. M. de Oca, T. Stützle, M. Birattari, and M. Dorigo, "Frankenstein's PSO: A composite particle swarm optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1120–1132, October 2009.

[15] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle warm optimization with time variant inertia and acceleration coefficients," *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, November 2007.

[16] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 2011, pp. 633–640. [Online]. Available: www.softcomputing.net/nabic11_7.pdf

[17] J. Kennedy, "Probability and dynamics in the particle swarm," in *Proceedings of the Congress on Evolutionary Computation (CEC 2004)*, vol. 1, June 2004, pp. 340–347.

[18] T. Krink, J. S. Vestertroem, and J. Riget, "Particle swarm optimisation with spatial particle extension," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002)*, Honolulu, Hawaii, 2002.

[19] J. Zhang, Y. Tan, and X. He, "Concentric spatial extension based particle swarm optimization inspired by brood sorting in ant colonies," in *Swarm Intelligence Symposium (SIS 2009)*. Nashville, Tennessee: IEEE, 2009, pp. 9–15.

[20] J. Riget and J. S. Vesterstrøm, "A diversity-guided particle swarm optimizer — the ARPSO," Department of Computer Science, University of Aarhus, Tech. Rep. 2002-02, 2002.

[21] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, June 2004.

[22] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997. [Online]. Available: citeseer.ist.psu.edu/wolpert96no.html