Research Article

A Simplified Recombinant PSO

Dan Bratton and Tim Blackwell

Department of Computing, Goldsmiths, University of London, New Cross, London SE14 6NW, UK

Correspondence should be addressed to Dan Bratton, d.bratton@gold.ac.uk

Simplified forms of the particle swarm algorithm are very beneficial in contributing to understanding how a particle swarm optimization (PSO) swarm functions. One of these forms, PSO with discrete recombination, is extended and analyzed, demonstrating not just improvements in performance relative to a standard PSO algorithm, but also significantly different behavior, namely, a reduction in bursting patterns due to the removal of stochastic components from the update equations.

1. INTRODUCTION

Originally conceived as a modification to the standard PSO algorithm for use on self-reconfigurable adaptive systems used in on-chip hardware processes, PSO with discrete recombination (PSO-DR) introduces several appealing and effective modifications, resulting in a simpler variant of the original [1]. It is one of the more interesting advances in PSO research over the last few years because these simplifications apparently do not degrade performance yet they remove various issues associated with the stochasticity of the PSO acceleration parameters that hinder theoretical analysis of PSO.

Physical creation of hardware-based optimizers is a substantially more intricate undertaking than software implementations, so fast, simple algorithms are desirable in order to minimize complexity. The comparative straightforwardness of PSO to many other evolutionary optimization algorithms makes it a good choice for this purpose, and further modifications were applied by the authors of [1] in order to simplify it even further and to introduce concepts from recombinant evolutionary techniques. The resulting algorithm, which can be implemented using only addition and subtraction operators and a simple 1-bit random number generator, is well suited for dedicated hardware settings.

Despite this rather specific original design specification, PSO-DR has shown to be a robust optimizer in its own right, equalling or surpassing a more common PSO implementation on a few tested benchmarks [1]. In this paper we extend the original work of Peña et al. by considering alternative topologies and parameter settings, running comparisons over a more comprehensive test suite, deriving simplified variants of the algorithm, and subjecting the model to a burst analysis.

The following section introduces PSO-DR (known here as model 1) as originally defined by Peña et al. and summarizes the burst analysis of [2]. Section 3 describes a series of simplifications to PSO-DR (models 2 and 3) which are introduced in this paper. The motivations for these simplifications are explained. Section 4 presents the results of performance experiments of models 1–3, and for comparative purposes, standard PSO. Following this, the paper proceeds with an empirical investigation of bursting patterns in recombinant PSO. The final section together draws together the experimental results of this paper and advances some ideas for the immediate future of PSO research.

2. PSO WITH DISCRETE RECOMBINATION

The velocity update for particle i in standard PSO (SPSO) in the inertia weight formalism is

$$\text{IW}: v_{id}^{t+1} = w v_{id}^t + \frac{\phi}{2} u_1 \left(p_{id} - x_{id}^t \right) + \frac{\phi}{2} u_2 \left(p_{nd} - x_{id}^t \right), \quad (1)$$

where d labels components of the position and velocity vectors, $d = 1, 2, \ldots, D$, $\vec{p}_i$ is the personal best position achieved by i, $\vec{p}_n$ is the best position of informers in i's social

neighborhood and $u_{1,2} \sim U(0,1)$ [3]. After velocity update, the particle position is adjusted:

$$x_{id}^{t+1} = v_{id}^{t+1} + x_{id}^t. \tag{2}$$

Peña et al. introduced a recombinant version of PSO by replacing either the personal best or the neighborhood best position by the recombinant position [1]. We focus here on the former for reasons of improved performance and the more interesting social aspect. A recombinant position vector $\vec{r}$ is defined by

$$r_{id} = \eta_d p_{ld} + (1 - \eta_d) p_{rd}, \tag{3}$$

where $\eta_d = U\{0,1\}$ and $\vec{p}_{l,r}$ are immediate left and right neighbors of i in a ring topology. While separate random numbers η_d are used for separate dimensions d, a single value is generated for each single dimension and used for both occurrences of η_d in that dimension. This places $\vec{r}_i$ at a corner of the smallest D-dimensional box which has p_l and p_r at its corners.

The authors of [1], in a search for a very efficient implementation, argued for the removal of the random numbers $u_{1,2}$ from (1) and parameter settings $\phi = 2$ and $w = 0.5$. The velocity update for the original form of PSO-DR is

$$\text{DR}: v_{id}^{t+1} = wv_{id}^t + \frac{\phi}{2}(r_{id} - x_{id}^t) + \frac{\phi}{2}(p_{nd} - x_{id}^t). \tag{4}$$

The choice of ϕ was based on the observation that $\phi \approx 4.0$ in standard PSO, but, since $u_{1,2}$ are uniform in $[0,1]$, the expectation value of $\phi u_{1,2}$ is 2.0. Furthermore, the multiplication by $w = 0.5$ can be implemented in hardware by a right shift operation. While optimal efficiency is desirable for hardware implementations, this issue does not concern us to the same degree in this study of (4) and it is one aim of this paper to study PSO-DR for arbitrary parameter values.

Although (4) contains a random element in the recombinant position, the acceleration parameters are constant. In other words, the update rule has additive rather than multiplicative stochasticity [2]. This has two ramifications; first, a stability condition can be computed based on the theory of second order, fixed parameter, difference equations and second, recombinant PSO is predicted not to exhibit particle velocity bursts. The details of these results are to be found in [2]. The stability condition is

$$|w| < 1, \quad 0 < \phi < 2(1 + w). \tag{5}$$

It is known that PSO at stagnation, that is, when no improvements to personal bests are occurring, and the particles effectively decouple, exhibits bursts of outliers [4]. These are temporary excursions of the particle to large distances from the attractors. A burst will typically grow to a maximum and then return through a number of damped oscillations to the region of the attractors. The origin of bursts, and of the concomitant fattening of the tails of the position distribution at stagnation, can be traced to the second-order stochastic difference equation

$$x(t + 1) + a(t)x(t) + bx(t - 1) = c(t) \tag{6}$$

which is equivalent to SPSO with the identification $a(t) = (\phi/2)(u_1 + u_2) - w - 1$, $b = w$, and $c(t) = (\phi/2)(u_1 p_1 + u_2 p_2)$ for fixed attractors $p_{1,2}$. Since $\max(|a|) > 0$, amplification of $x(t)$ can occur through repeated multiplication of $x(t)$ by a despite the second order reduction by multiplication by the constant b. Interestingly, the distribution tail of $|x|$, by virtue of the bursts that become increasingly less probable for increasing size, is fattened compared to an exponential falloff as provided by, for example, a Gaussian. A theoretical justification of these power laws and some empirical tests can be found in [2].

PSO bursts differ from the random outliers generated by PSO models which replace velocity by sampling from a distribution with fat tails such as a Richer and Blackwell [5]. In contradistinction to the outliers of these "bare bones" formulations [6], the outliers from bursts occur in sequence, and they are one dimensional. Bursting will therefore produce periods of rectilinear motion where the particle will have a large velocity parallel to a coordinate axis. Furthermore, large bursts may take the particle outside the search space. Although this will not incur any penalty in lost function evaluations if particles that exit the feasible bounds of the problem are not evaluated, as is the common approach to this situation, they are not contributing to the search while in outer space. PSO-DR, which is predicted not to have bursts [2], therefore provides a salient comparison.

3.   SIMPLIFYING RECOMBINANT PSO

This section details the two new recombinant models that are being proposed in this paper. To begin, an investigation into PSO-DR reveals more interesting properties of the formulation. Performance plots for a sweep through parameter space to find an optimal balance between the inertia weight coefficient w and the ϕ coefficients show that while the optimal region is spread across the parameter space, it also intersects the axis for the w term (see Figure 1 for results on selected functions from Table 1). This demonstrates that the system is able to obtain good optimal results even at $w = 0.0$ and there is no inertia term in the velocity update equations.

Model 2 PSO-DR sets $w = 0$, with a velocity update,

$$\text{DR2}: v_{id}^{t+1} = \frac{\phi}{2}(r_{id} - x_{id}^t) + \frac{\phi}{2}(p_{nd} - x_{id}^t). \tag{7}$$

Velocity now serves as a dummy variable in the update equations (1) and (2) and model 2 can be represented as a single, velocity-free rule

$$\text{DR2}: x_{id}^{t+1} = x_{id}^t + \frac{\phi}{2}(r_{id} - x_{id}^t) + \frac{\phi}{2}(p_{nd} - x_{id}^t). \tag{8}$$

At this point, the two ϕ terms were detached and another sweep through parameter space to find an optimal combination of the recombinant component via its coefficient ϕ_1 and the neighborhood best component via its coefficient ϕ_2 was performed. Surprisingly, results again showed that the optimal region intersects an axis, this time for the neighborhood term $(p_{gd} - x_{id}^t)$ (see Figure 2 for selected results).
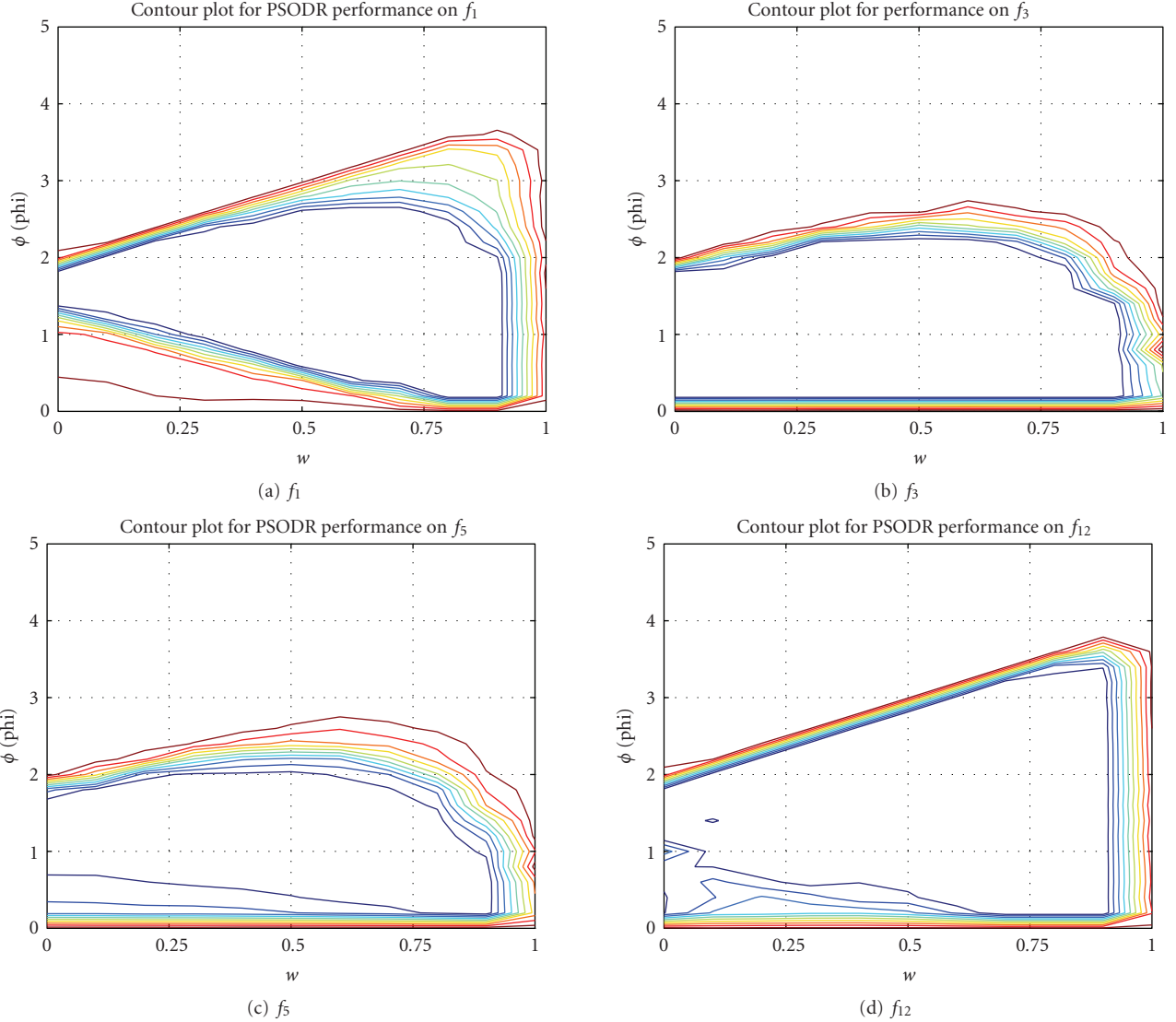
FIGURE 1: Optimal regions for w versus ϕ in PSO-DR model 1, found empirically through 30 runs for each combination of parameters $w = 0.0, \ldots, 1.0, \phi = 0.0, \ldots, 5.0$ at a granularity of 0.1. Each contour line represents a 10% improvement in performance with the region within the innermost line representing the best performing 10% of possible combinations of w and ϕ.

This allows a further simplification to the update equation (4), down to PSO-DR model 3:

$$\text{DR3} : x_{id}^{t+1} = x_{id}^t + \phi(r_{id} - x_{id}^t) \qquad (9)$$

which is clearly a substantial reduction of the original PSO-DR equation. This PSO variant, if it proves to be viable, would raise a couple of interesting questions. To what extent is velocity a necessary component, or is it a relic of the biological origins of PSO [6]? Secondly, how important is the neighborhood component drawn from the single best neighbor? The optimization process of Model 3 is entirely driven by the recombinant component; this idea is reminiscent of fully informed particle swarms (FIPS) [7], where the entire neighborhood influences particle behavior. However, whereas FIPS allows every neighbor to influence a particle's behavior in every dimension, Model 3 allows only a single randomly chosen neighbor to fully influence the particle in each dimension. This gives the particle an updated position that is a combination of the best positions of all of its neighbors throughout all dimensions.

The following section presents evidence that PSO-DR3 is a viable alternative to standard PSO by reporting on performance results for all three models of PSO-DR over a number of commonly used test functions.

4. PERFORMANCE EXPERIMENTS

Algorithms were tested over a series of 14 benchmark functions chosen for their variety, shown in Tables 1 and 2. Functions $f_1 - f_3$ are unimodal functions with a single minimum, $f_4 - f_9$ are complex high-dimensional multimodal problems, each containing many local minima and a single

TABLE 1: Benchmark function equations.

Equation
$f_1 = \sum_{i=1}^{D} x_i^2$
$f_2 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$
$f_3 = \sum_{i=1}^{D-1} \left\{ 100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right\}$

$$f_4 = -\sum_{i=1}^{D} x_i \sin\left(\sqrt{x_i}\right)$$

$$f_5 = \sum_{i=1}^{D} \left\{ x_i^2 - 10\cos(2\pi x_i) + 10 \right\}$$

$$f_6 = -20\exp\left\{ -0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2} \right\} - \exp\left\{ \frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i) \right\} + 20 + e$$

$$f_7 = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$f_8 = \frac{\pi}{D}\left\{ 10\sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2\left\{ 1 + 10\sin^2(\pi y_{i+1}) \right\} + (y_D - 1)^2 \right\}$$
$$+ \sum_{i=1}^{D} \mu(x_i, 10, 100, 4)$$
$$y_i = 1 + \frac{1}{4}(x_i + 1)$$
$$\mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$$

$$f_9 = 0.1\left\{ \sin^2(3\pi x_i) + \sum_{i=1}^{D-1} (x_i - 1)^2\left\{ 1 + \sin^2(3\pi x_{i+1}) \right\} + (x_D - 1)^2 \right.$$
$$\left. \times \left\{ 1 + \sin^2(2\pi x_D) \right\} \right\} + \sum_{i=1}^{D} \mu(x_i, 5, 100, 4)$$

Equation
$f_{10} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$
$f_{11} = \left\{ 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right\}$ $\times \left\{ 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right\}$
$f_{12} = -\sum_{i=1}^{5} \left\{ \sum_{j=1}^{4} (x_j - a_{ij})^2 + c_i \right\}^{-1}$
$f_{13} = -\sum_{i=1}^{7} \left\{ \sum_{j=1}^{4} (x_j - a_{ij})^2 + c_i \right\}^{-1}$
$f_{14} = -\sum_{i=1}^{10} \left\{ \sum_{j=1}^{4} (x_j - a_{ij})^2 + c_i \right\}^{-1}$

global optimum, and $f_{10} - f_{14}$ are lower-dimensional multimodal problems with few local minima and a single global optimum apart from f_{10}, which is symmetric about the origin with two global optima.

Particles were initialized using the *region scaling* technique where initialization takes place in an area of the search space known not to contain the global optimum [8]. To avoid initializing the entire swarm directly within a local minimum, as could be possible with $f_{12} - f_{14}$ if initialization takes place in the bottom quarter of the search space in each dimension (as is common), an area of initialization composed of the randomly chosen top or bottom quarter of each dimension was defined, into which all particles were placed with uniform distribution. This method ensures that the swarm will not be initialized within the same area for every optimization run, but will still be confined to an area at most 0.25^D of the search space, making the chance of initialization directly on or near the global optimum extremely unlikely. In instances where the global optimum was located at the center of the search space (i.e., $f_1, f_2, f_5 - f_7$), the function was *shifted* by a random vector with maximum magnitude of a tenth of the size of the search space in each dimension for each run to remove any chance of a centrist bias [9].
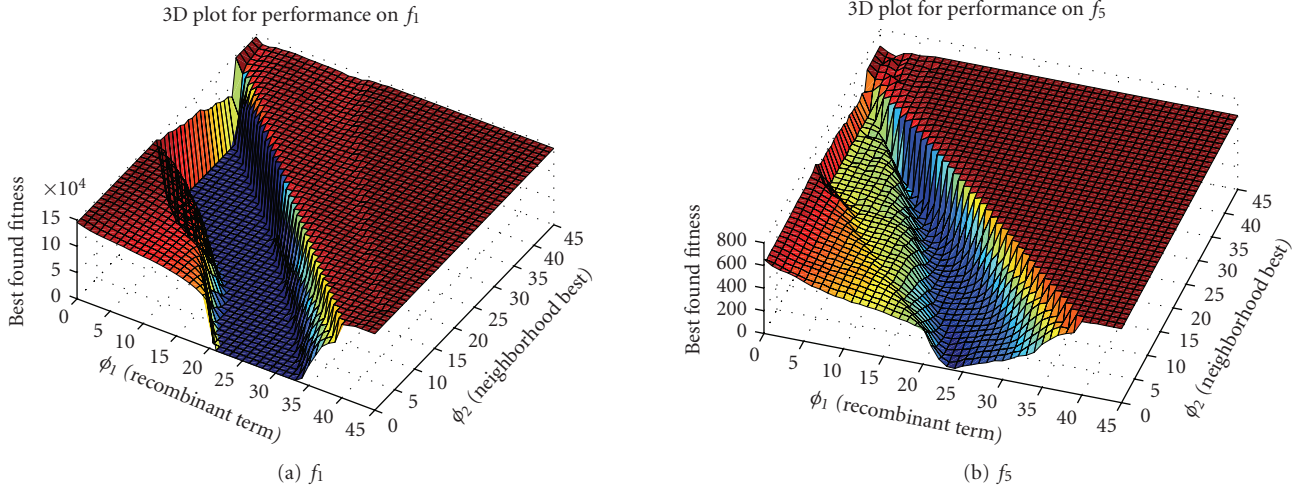
(a) f_1

(b) f_5

FIGURE 2: Performance plots for ϕ_1 and ϕ_2 in PSO-DR model 2, found empirically through 30 runs for each combination of parameters $\phi_1 = 0.0, \ldots, 4.5, \phi_2 = 0.0, \ldots, 4.5$ at granularity 0.1.

TABLE 2: Benchmark function details.

Function	Name	D	Feasible bounds
f_1	Sphere/parabola	30	$(-100, 100)^D$
f_2	Schwefel 1.2	30	$(-100, 100)^D$
f_3	Generalized Rosenbrock	30	$(-30, 30)^D$
f_4	Generalized Schwefel 2.6	30	$(-500, 500)^D$
f_5	Generalized Rastrigin	30	$(-5.12, 5.12)^D$
f_6	Ackley	30	$(-32, 32)^D$
f_7	Generalized Griewank	30	$(-600, 600)^D$
f_8	Penalized function P8	30	$(-50, 50)^D$
f_9	Penalized function P16	30	$(-50, 50)^D$
f_{10}	Six-hump camel-back	2	$(-5, 5)^D$
f_{11}	Goldstein-price	2	$(-2, 2)^D$
f_{12}	Shekel 5	4	$(0, 10)^D$
f_{13}	Shekel 7	4	$(0, 10)^D$
f_{14}	Shekel 10	4	$(0, 10)^D$

This investigation tested PSO-DR model 1 using both global (as used in the originally proposed algorithm) and local ring topologies for selecting the neighborhood operator p_n. The parameter settings were Pena's, giving a velocity update with the form

$$v_{id}^{t+1} = 0.5 v_{id}^t + (r_{id} - x_{id}^t) + (p_{nd} - x_{id}^t). \quad (10)$$

Results shown for PSO-DR model 2 use the value $\phi \approx 1.6$, while those for PSO-DR model 3 use $\phi \approx 1.2$. These values were empirically determined to be optimal for these algorithms; an analytical determination is the subject of current research. Results for both models 2 and 3 are shown for runs using a ring topology, which showed superior performance in testing.

For comparison, results are presented for a standard PSO algorithm (SPSO), which operates using the constricted velocity update equation

$$v_{t+1} = \chi \left(v_t + \frac{\phi}{2} u_1 (p_i - x_t) + \frac{\phi}{2} u_2 (p_g - x_t) \right) \quad (11)$$

with $\phi = 4.1, \chi = 0.72984$ and with 50 particles [3]. All PSO-DR model tests were carried out using 50 particles as well. Algorithm performance was measured as the minimum error $|f(x) - f(x^\star)|$ found over the trial where $f(x^\star)$ is the fitness at the global optimum for the problem. Results were averaged over 30 independent trials, and are displayed, with standard error, in Table 3. Values less than 10^{-15} have been rounded to 0.0.

Performance results in Table 3 for all models of PSO-DR versus SPSO clearly indicate that it is a competitive variant, especially on highly complex problems such as f_5 (Rastrigin). Statistical tests were performed on these results to determine the significance of the performance differences between the

TABLE 3: Mean error after 30 trials of 300 000 evaluations. Necessary function evaluations are shown where 0.0 error was attained.

	SPSO Ring	SPSO Global	PSO-DR M1 Ring	PSO-DR M1 Global	PSO-DR M2	PSO-DR M3
f_1	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**
	97063 ± 377	46897 ± 421	59322 ± 125	33290 ± 170	60063 ± 41	75810 ± 322
f_2	0.12 ± 0.01	**0.0 ± 0.0**	0.01 ± 0.002	**0.0 ± 0.0**	3.7E-7 ± 7.5E-8	5.14 ± 1.27
	—	297800 ± 928	—	168852 ± 1205	—	—
f_3	6.18 ± 1.07	8.37 ± 2.26	16.79 ± 0.49	0.80 ± 0.29	34.57 ± 5.46	18.64 ± 4.45
	—	—	—	—	—	—
f_4	3385 ± 40	3522 ± 32	2697 ± 36	3754 ± 48	2418 ± 27	1830 ± 46
	—	—	—	—	—	—
f_5	163.50 ± 5.64	140.16 ± 5.87	44.64 ± 2.71	115.51 ± 7.03	35.21 ± 2.13	9.88 ± 0.86
	—	—	—	—	—	—
f_6	18.28 ± 0.85	12.93 ± 1.59	0.68 ± 0.67	18.51 ± 0.90	**0.0 ± 0.0**	**0.0 ± 0.0**
	—	—	—	—	287220 ± 2105	248160 ± 1945
f_7	**0.0 ± 0.0**	0.019 ± 0.004	**0.0 ± 0.0**	0.008 ± 0.002	**0.0 ± 0.0**	**0.0 ± 0.0**
	110616 ± 3320	—	101526 ± 9227	—	81226 ± 6560	70348 ± 2954
f_8	0.004 ± 0.003	0.15 ± 0.05	**0.0 ± 0.0**	0.05 ± 0.02	**0.0 ± 0.0**	**0.0 ± 0.0**
	—	—	61370 ± 249	—	85101 ± 581	95810 ± 655
f_9	**0.0 ± 0.0**	0.003 ± 0.001	**0.0 ± 0.0**	0.002 ± 0.0007	**0.0 ± 0.0**	**0.0 ± 0.0**
	106163 ± 537	—	61793 ± 221	—	86031 ± 377	92416 ± 437
f_{10}	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**
	9348 ± 190	11808 ± 445	44577 ± 7608	40015 ± 4483	6103 ± 104	5918 ± 75
f_{11}	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**
	8258 ± 104	7080 ± 108	4772 ± 47	3968 ± 27	6720 ± 66	6846 ± 87
f_{12}	0.59 ± 0.33	4.61 ± 0.54	0.17 ± 0.17	4.34 ± 0.59	**0.0 ± 0.0**	**0.0 ± 0.0**
	—	—	—	—	59958 ± 43	16229 ± 855
f_{13}	1.09 ± 0.45	4.40 ± 0.60	**0.0 ± 0.0**	2.55 ± 0.62	8.1E-11 ± 1.8E-11	**0.0 ± 0.0**
	—	—	13433 ± 249	—	—	14012 ± 764
f_{14}	0.96 ± 0.45	3.24 ± 0.66	**0.0 ± 0.0**	3.13 ± 0.66	6.6E-11 ± 1.7E-11	**0.0 ± 0.0**
	—	—	12760 ± 1386	—	—	121031004 ±

two algorithms. To avoid the problem of the probabilistic nature of t-tests potentially affecting results when conducting multiple significance tests, a modified Bonferroni procedure was applied to values of α for successive tests [10]. This procedure involves inversely ranking observations by ascending values of p, then setting

$$\alpha' = \frac{\alpha}{\text{inverse rank}}. \tag{12}$$

Results for these statistical tests on PSO-DR model 3 and SPSO are shown in Table 4 and confirm that the performance is significantly improved on 3 of the 14 tested functions, equivalent for 10 functions, and worsened for 1 function for PSO-DR model 3 versus SPSO with ring topology. Perhaps the most impressive improvement comes for f_5 (Rastrigin), a notoriously difficult multimodal problem that PSO algorithms perform poorly on some problems in high dimensionality.

Due to the high number of function evaluations that were performed to obtain these results relative to previous work (where only 30 k–60 k function evaluations might be performed), selected convergence plots are shown in Figure 3. These show that the standard PSO obtains superior results at the very start of the optimization process, up to 5000 function evaluations for the highest observed value (Figure 3(b)). After the point at which this occurs, PSO-DR model 3 surpasses the standard algorithm in performance, and maintains this advantage to the end of the 300 k function evaluations on 7 of the 14 tested problems ($f_4 - f_6, f_8, f_{12} - f_{14}$). On problems for which both algorithms attained equal error levels of 0.0 ($f_1, f_7, f_9 - f_{11}$), the point at which this occurs, that is, when SPSO "catches up" to PSO-DR model 3, can be observed in Table 3. On average, SPSO took 25% more function evaluations to attain the optimum than PSO-DR model 3 on these problems. Finally, for the two problems on which SPSO outperformed PSO-DR model 3 (f_2, f_3), the same early performance is seen with PSO-DR model 3 surpassing SPSO in performance early in the optimization process; in these cases, SPSO eventually repasses the other algorithm by 50 k function evaluations.

TABLE 4: Significance for SPSO versus PSO-DR model 3 with ring topologies.

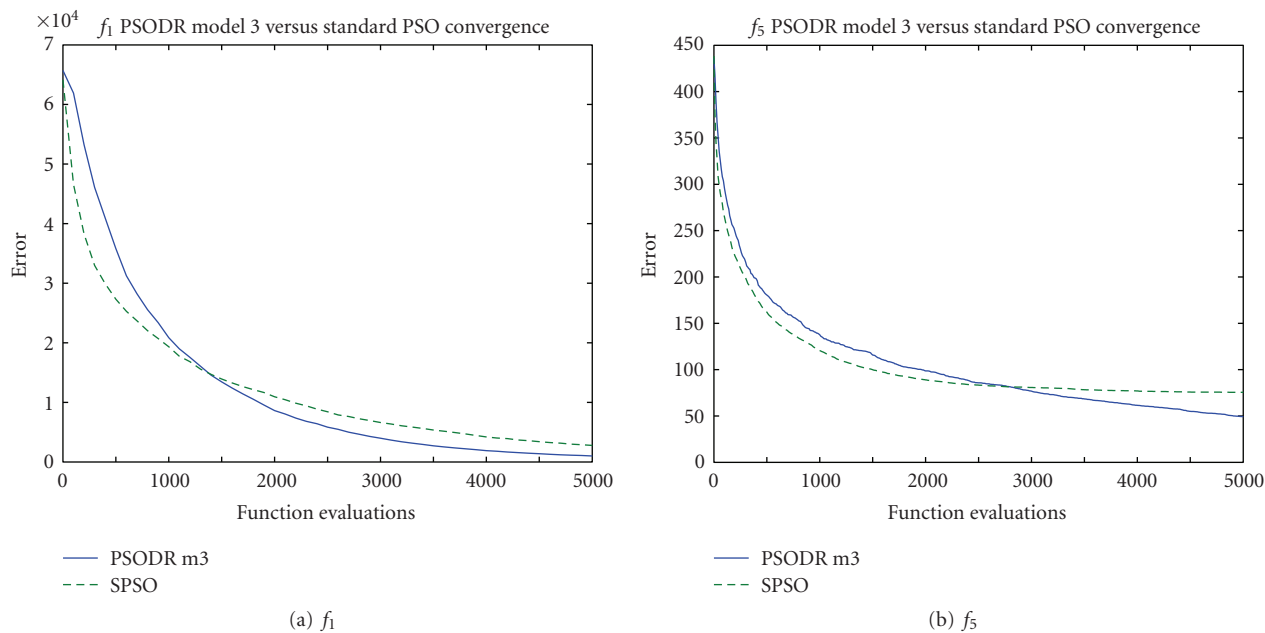| Function	p-value	Inverse rank	α'	Significant
f_4	0	14	0.003 571	Yes
f_5	0	13	0.003 846	Yes
f_6	0	12	0.004 167	Yes
f_2	2.11e-11	10	0.005	Yes
f_3	0.0086	11	0.004 545	No
f_{13}	0.02	9	0.005 556	No
f_{14}	0.04	8	0.00 625	No
f_{12}	0.2663	6	0.08	No
f_8	0.3215	7	0.007 143	No
f_1	1	5	0.01	No
f_7	1	4	0.0125	No
f_9	1	3	0.016 667	No
f_{10}	1	2	0.025	No
f_{11}	1	1	0.05	No

(a) f_1

(b) f_5

FIGURE 3: Convergence plots for SPSO and PSO-DR model 3 early in the optimization process.

A potential explanation for this behavior lies in the diversity of the swarms at this point in the optimization process. Figure 4 shows the mean Euclidean distance between particles for the corresponding convergence plots of Figure 3. It should be noted that uniform initialization was used in the trials used to generate these plots; relative performance between the algorithms was unaffected, and initializing particle positions uniformly throughout the search space removes an unrelated phenomenon in subspace initialization wherein the swarm expands greatly beyond the relatively small initialization region at the start of the optimization process to explore the search space. Expansion is common in the first few iterations using uniform ini-

tialization as well, but this is inherent to the swarm behavior and influenced only by the size of the entire search space.

As can be seen in the plots of Figure 4, neither swarm type begins converging immediately following initialization but rather they maintain their diversity or expand slightly. On a comparative basis, the standard PSO swarm expands substantially more than the PSO-DR model 3 swarm; for example Figure 4(c) shows that after the first 100 function evaluations, the mean distance between particles in the standard PSO swarm increases from 23 to 31.5, while the PSO-DR swarm diversity increases only from 23 to 24.5. Similar disparities were observed for all other tested problems.
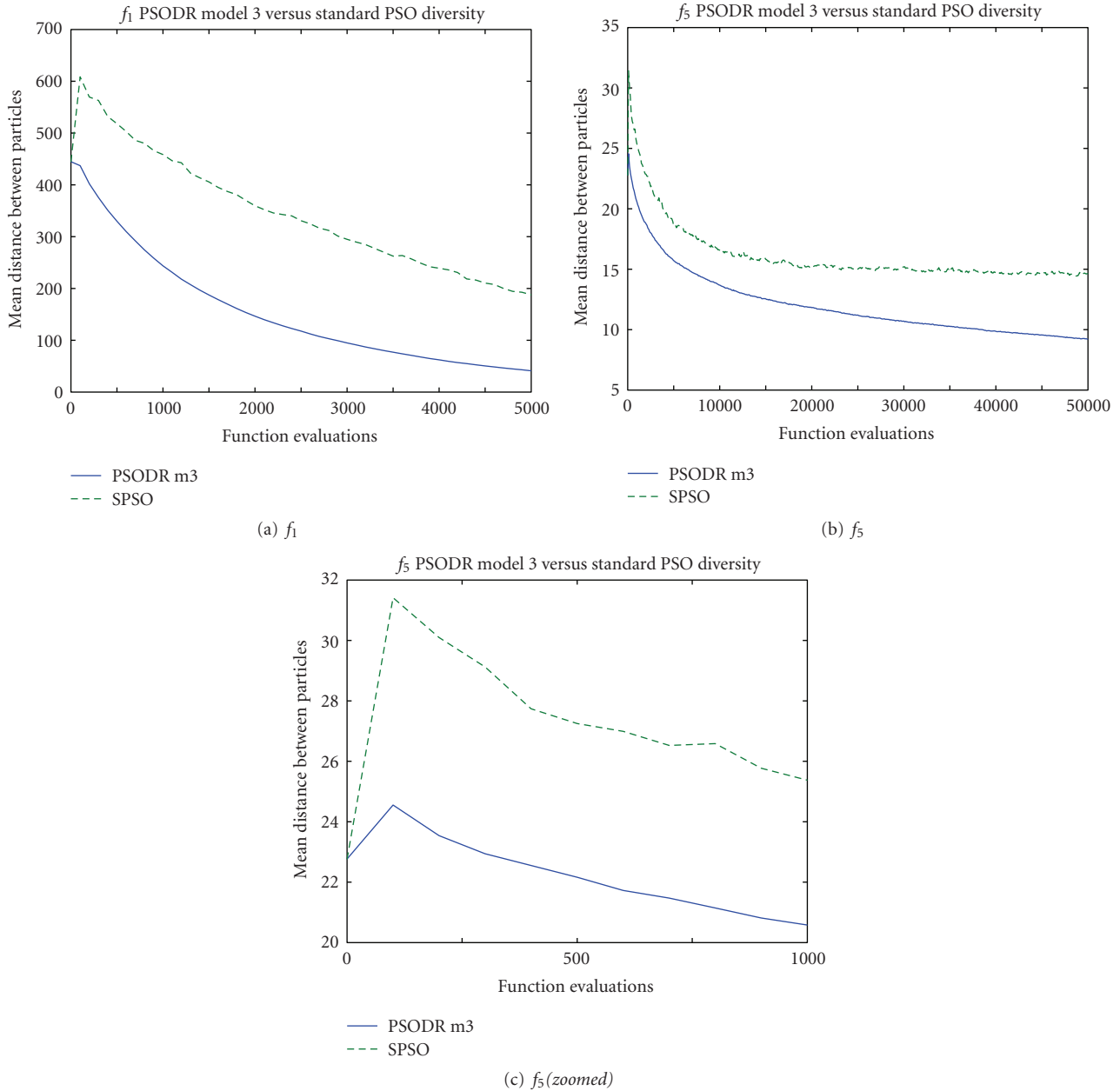
(a) f_1

(b) f_5

(c) f_5 (zoomed)

FIGURE 4: Diversity plots for SPSO and PSO-DR model 3 early in the optimization process.

It is reasonable to gather from these results that the higher swarm diversity for the standard PSO algorithm early in the optimization process demonstrates a wider spread of particle dispersion, and hence an improved probability of finding and starting to explore the basin of attraction for global or good local optima. PSO-DR model 3 expands very little, if at all, early in the optimization process, resulting in delayed acquisition of optimal regions of the search space.

5. EXAMINATION OF BURSTING

Bursts in the velocities of particles are commonly observed using the standard PSO algorithm. These are generated by means of the multiplicative stochasticity of the algorithm [2]. In order to investigate bursting behavior in PSO-DR and SPSO an empirical measure was devised.

This bursting measure was implemented to highlight when a particle had a velocity in a single dimension that was considerably higher than the next highest dimensional velocity. Bursting patterns of behavior were detected by reporting that every time particle velocity in a single dimension was a set amount λ times higher than velocity in the next highest dimension. Bursting behavior is demonstrated in Figure 6, where the velocity of a single particle in a 10-dimensional problem is shown. On the plot of the multidimensional velocity of the SPSO particle, it can be seen that velocity in a
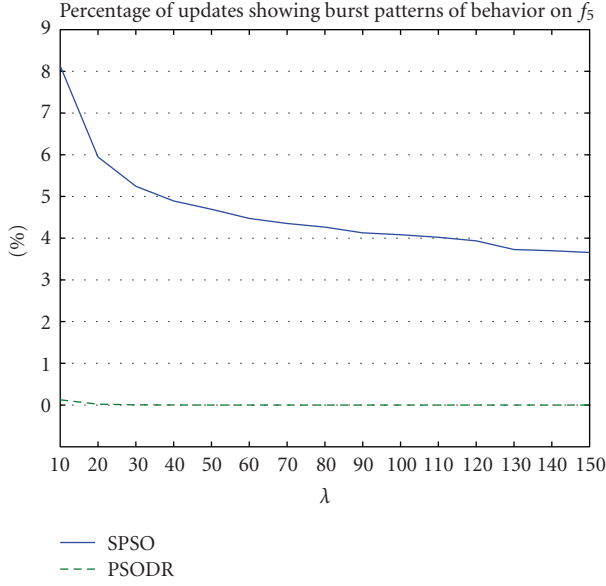
Figure 5: Frequency of updates showing burst behavior for values of λ.

(a) SPSO

(b) PSO-DR

Figure 6: Representative particle velocities for SPSO and PSO-DR on 10D Rastrigin.

single dimension increases suddenly and dramatically while remaining relatively level and low in all other dimensions. This is an example of a velocity burst. While the figure shows velocity for a single particle on a single run, examination of velocity plots for hundreds of particles over dozens of runs confirmed this to be representative of general particle behavior.

Velocity for a PSO-DR particle is also shown in Figure 6, and demonstrates the absence of bursts. Similarly to the SPSO plot, examination of a large number of plots confirmed this to be representative of general behavior for PSO-DR.

Examination of these empirical analyses show that PSO-DR clearly does not contain bursting behavior on the scale of SPSO while demonstrating equal or superior performance on 13 of the 14 benchmark functions, leading to the hypothesis that bursts are not, in fact, integral to the successful operation of particle swarm algorithms. The fact that a very few bursts do occur with PSO-DR indicates that it is a highly improbable feature of DR dynamics.

Analysis performed on statistics of several functions shows that particle updates involving bursts are far less effective than more common nonbursting updates. For example, results showed that for SPSO on f_5 with $\lambda = 100$, on average 20.1% of *all* particle, updates involve an improvement to the particle's best found position p_i, whereas only 1.8% of updates involving bursts result in an improvement to p_i. Likewise, on average 0.9% of all particle, updates improve the best found swarm position g, as opposed to only 0.01% for bursting particles. Burst frequencies for values of λ from 10 to 150 are shown in Figure 5.

It is also interesting to note that far fewer *total* updates result in an improved p_i or g for PSO-DR when compared to SPSO, for example, results showed that 20.1% of all updates improve p_i for SPSO compared with 0.64% for PSO-DR, and
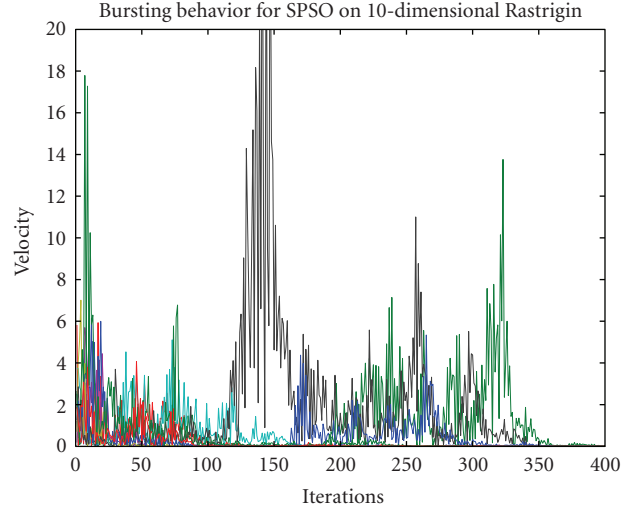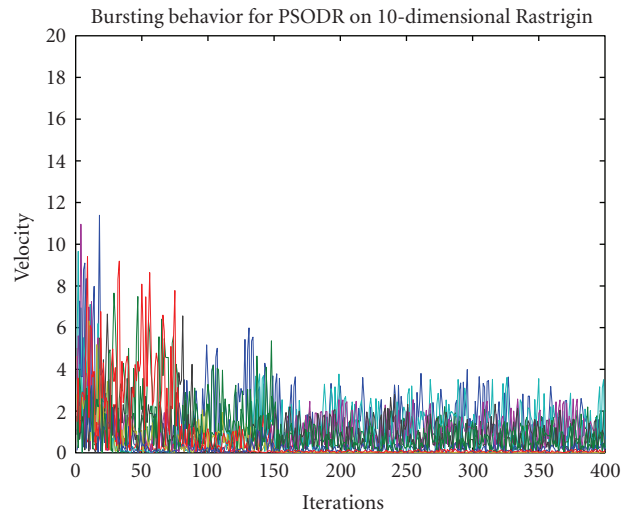
0.91% improve g for SPSO compared with 0.02% for PSO-DR on f_5 for $\lambda = 100$.

6. CONCLUSIONS

Simplification of the standard PSO algorithm is an important step toward understanding how and why it is an effective optimizer. By removing components of the algorithm and seeing how this affects performance, we are granted insight into what those components contribute to overall particle and swarm behaviors.

In particular, this paper has proposed a very simple PSO

$$\text{DR3}: x_{id}^{t+1} = x_{id}^t + \phi\left(r_{id} - x_{id}^t\right) \tag{13}$$

which offers competitive performance to standard PSO, but removes multiplicative randomness, inertia, and the personal memory term p_i from the position update.

There is still much to be done before questions concerning PSO behavior can be completely answered, and it is expected that the next decade of PSO research will be focused on understanding the basic algorithm that powers both the standard implementation and its variants.

In that light, the PSO-DR variant is important not only because of its improved performance on several benchmark functions, but also because its simplified state allows us to examine what happens to the standard algorithm when pieces are modified or removed. Based on the results presented here, it can be argued that large bursts are not generally beneficial or integral to PSO performance, and may possibly be detrimental. Although the presence of particle outliers is demonstrably important for swarm optimization (as demonstrated in bare bones analysis, [6]), bursts, which are sequences of extreme particle positions, occurring along an axis and reaching outside the search space, remain a special feature of velocity-based swarms. This work, which compares standard PSO to a burst-free but comparable optimizer suggests that bursts are disadvantageous in general. (However, in the coincidence that the objective function has a rectangular symmetry aligned with the axes, then bursting may actually be fortuitous.)

Further, the replacement of the direct personal influence operator p_i from SPSO with the recombinant term r_i derived from its neighborhood in PSO-DR strengthens the case for PSO being mostly reliant on social interaction as opposed to personal experience. This is further supported by the effectiveness of PSO-DR model 3, which lacks a cognitive term altogether. The social behavior occurring inside of a swarm is still a wide-open area in the field, and will hopefully constitute a great deal of the future research devoted to the development of a better understanding of this deceptively simple optimizer.

Another property of PSO-DR resides in attractor jiggling that takes place even at stagnation (no updates to any p_i) since r_i is never fixed. This jiggling will work against convergence and could propel the swarm onwards. This, and other matters concerning the nature of recombination within PSO, will be the subject of further study.

ACKNOWLEDGMENTS

REFERENCES

[1] J. Peña, A. Upegui, and E. Sanchez, "Particle swarm optimization with discrete recombination: an online optimizer for evolvable hardware," in *Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems (AHS '06)*, pp. 163–170, Istanbul, Turkey, June 2006.

[2] T. Blackwell and D. Bratton, "Origin of bursts," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, pp. 2613–2620, London, UK, July 2007.

[3] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of IEEE Swarm Intelligence Symposium (SIS '07)*, pp. 120–127, Honolulu, Hawaii, USA, April 2007.

[4] J. Kennedy, "Probability and dynamics in the particle swarm," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 1, pp. 340–347, IEEE Press, Portland, Ore, USA, June 2004.

[5] T. J. Richer and T. Blackwell, "The Lévy particle swarm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '06)*, pp. 808–815, IEEE Press, Vancouver, BC, Canada, July 2006.

[6] J. Kennedy, "Bare bones particle swarms," in *Proceedings of IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 80–87, Indianapolis, Ind, USA, April 2003.

[7] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.

[8] D. K. Gehlhaar and D. B. Fogel, "Tuning evolutionary programming for conformationally flexible molecular docking," in *Proceedings of the 5th Annual Conference on Evolutionary Programming (EP '96)*, L. Fogel, P. Angeline, and T. Back, Eds., pp. 419–429, MIT Press, San Diego, Calif, USA, February 1996.

[9] C. K. Monson and K. D. Seppi, "Exposing origin-seeking bias in PSO," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO '05)*, pp. 241–248, Washington, DC, USA, June 2005.

[10] J. Jaccard and C. K. Wan, *LISREL Approaches to Interaction Effects in Multiple Regression*, vol. 114 of *Quantitative Applications in the Social Sciences*, Sage Publications, Thousand Oaks, Calif, USA, 1996.