Understanding Particle Swarms through Simplification: A Study of Recombinant PSO

Daniel Bratton
Goldsmiths, University of London
New Cross
London SE14 6NW, UK
dbratton@gmail.com

Tim Blackwell
Goldsmiths, University of London
New Cross
London SE14 6NW, UK
tim.blackwell@gold.ac.uk

ABSTRACT

Simplified forms of the particle swarm algorithm are very beneficial in contributing to understanding of what makes a PSO swarm function in the way that it does. One of these forms, PSO with discrete recombination, is analyzed in depth, demonstrating not just improvements in performance to a standard PSO algorithm, but also significantly different behavior with a reduction in bursting patterns due to the removal of stochastic components from the update equations. This altered behavior accompanied by equal and improved performance leads to conjectures that bursts are not generally efficacious in the optimization process.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search

General Terms

Optimisation

Keywords

Particle swarms, swarm optimisation

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 [7]. Arguably it is one of the more significant 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 hinders 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 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 ideal 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 [7]. In this paper we extend the original work of Pena at al by considering alternative topologies and parameter settings, running comparisons over a more comprehensive test suite and subjecting the model to a burst analysis.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

2. PSO WITH DISCRETE RECOMBINATION

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

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

where d labels components of the position and velocity vectors, $\vec{p}_i$ is the personal best position achieved by i, $\vec{p}_n$ is the best position of informers in i's social neighborhood and $u_{1,2} \sim U(0,1)$ [2].

A recombinant position vector $\vec{r}$ is defined by

$$r_{id} = \eta_d p_{ld} + (1 - \eta_d)p_{rd} \quad (2)$$

where $\eta_d = U\{0,1\}$ and $\vec{p}_{l,r}$ are immediate left and right neighbors of i in a ring topology. Note that separate random numbers η_d are used for each dimension d. This places $\vec{r}_i$ at a corner of the smallest hypercube which has p_l and p_r at its corners.

Pena at al introduced a recombinant version of PSO by replacing either the personal best or the neighborhood best position by the recombinant position [7]. We focus here on the first replacement for reasons of improved performance and the more interesting social aspect. The velocity update for PSO-DR is

$$DR : v_{id}^{t+1} = wv_{id}^{t} + \frac{\phi}{2}(r_{id} - x_{id}^{t}) + \frac{\phi}{2}(p_{nd} - x_{id}^{t}) \quad (3)$$

The authors of [7], in a search for a very efficient implementation, argued for the removal of the random numbers $u_{1,2}$ and parameter settings $\phi = 2$ and $w = 0.5$. The choice of ϕ was based on the observation that $\phi \approx 4.0$ in standard PSO (SPSO), but, since $u_{1,2}$ are uniform in $[0,1]$, the expectation value of $\frac{\phi}{u_{1,2}}$ is 2.0. Furthermore, the multiplication by w can be implemented in hardware by a right shift operation. However efficiency need not concern us here in this study of Eq. 3 and it is one aim of this paper to study PSO-DR for arbitrary parameter values.

Although Equation 3 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 [1]. 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 [1]. The stability condition is

$$\begin{cases} |w| < 1 \\ 0 < \phi < 2(1+w) \end{cases} \tag{4}$$

It is known that decoupled PSO exhibits bursts of outliers [5]. 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:

$$x(t+1) + a(t)x(t) + bx(t-1) = c(t) \tag{5}$$

which is equivalent to SPSO with the identification $a(t) = \frac{\phi}{2}(u_1 + u_2) - w - 1$, $b = w$ and $c(t) = \frac{\phi}{2}(u_1 p_1 + u_2 p_2)$ 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 fall-off as provided by, for example, a Gaussian. A theoretical justification of these power laws and some empirical tests can be found in [1].

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 Levy [8]. In contradistinction to the outliers of these 'bare bones' formulations, the outliers from bursts occur in sequence, and they are 1-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 whilst in outer space. PSO-DR, which is predicted not to have bursts [1], therefore provides a salient comparison.

The following section describes experiments to compare PSO-DR to standard PSO over a large set of benchmarks at Pena et al's suggested parameter settings. Performance of PSO-DR for other parameter settings is also investigated. Section 4 examines bursting and its relevance to performance.

3. PERFORMANCE EXPERIMENTS

Algorithms were tested over a series of 14 benchmark functions chosen for their variety, shown in Table 1. 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 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 [3]. To avoid initializing the entire swarm directly within a local minima, as could be possible with F12-F14 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 [6].

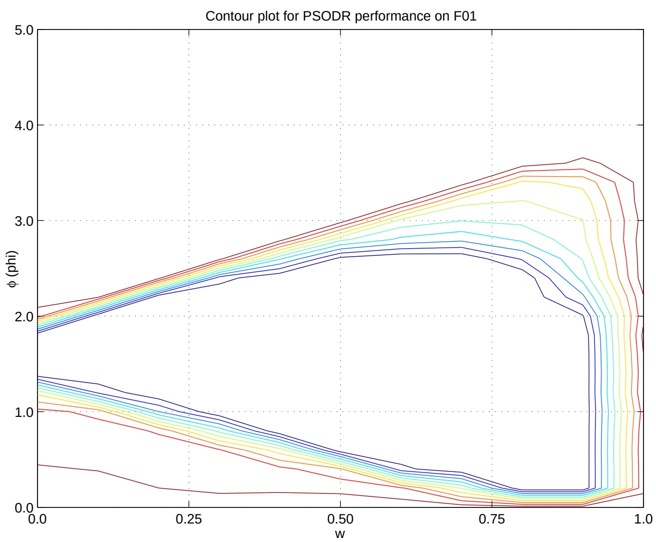
This investigation tested PSO-DR using both a global (as used in the originally proposed algorithm) and a local ring topology for selecting the neighborhood operator p_n. The parameter settings were Pena et al's, so the velocity update has the simple form

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

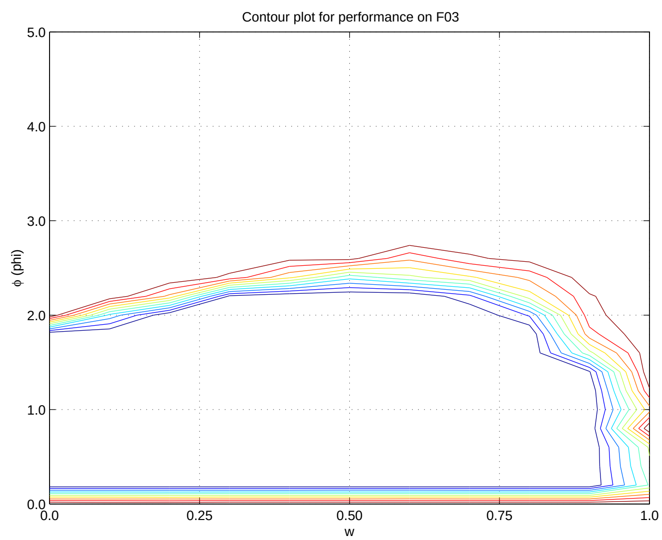
For comparison, results are presented for the Standard PSO algorithm (SPSO), which operates using the constricted velocity update equation with $\phi = 4.1, \chi = 0.72984$ and with 50 particles [2]. PSO-DR tests were carried out using 50 particles as well. Algorithm performance was measured as the minimum 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 2. In cases where a value was $< 10^{-15}$ it was rounded to 0.0 in order to accommodate reproduction using programming languages that may not include floating point precision at smaller values.

Tests were performed on the PSO-DR algorithm to determine the relationship between the w and ϕ variables - although PSO-DR was designed to remove these variables from the algorithm, this was done based on the restrictions of hardware implementations. We faced no such restrictions here, so an examination of these influences is beneficial to better understanding. Contour plots of the performance landscape are shown in Figure 1, with improved performance levels indicated inside of the contours. This tuning was done solely for these tests, however; the variables were set to default values of $w = 0.5$ and $\phi = 2.0$ for the other investigations that are detailed here. The contour plots confirm these values to be within the optimal ranges of all four of the tested problems, though barely so on f_5. Finding ideal values for these variables across all problems will be explored in future work, but it is interesting to note that for values of ϕ approx 1.6, the optimal range includes an inertia weight setting of $w = 0.0$ for all four sample benchmarks. This has implications for the velocity term that warrant considerable attention, but are currently beyond the scope of this paper.

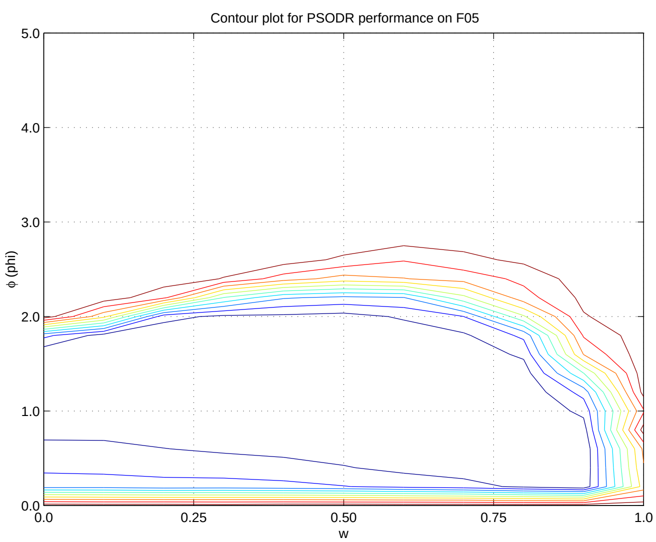
Performance results in Table 2 for PSO-DR vs SPSO clearly indicate that it is an extremely competitive variant, especially on highly complex problems such as f_5 (Rastrigin). Statistical tests were performed on these results to determine the significance of the performance differences between the two 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 [4]. This procedure involves inversely ranking observations



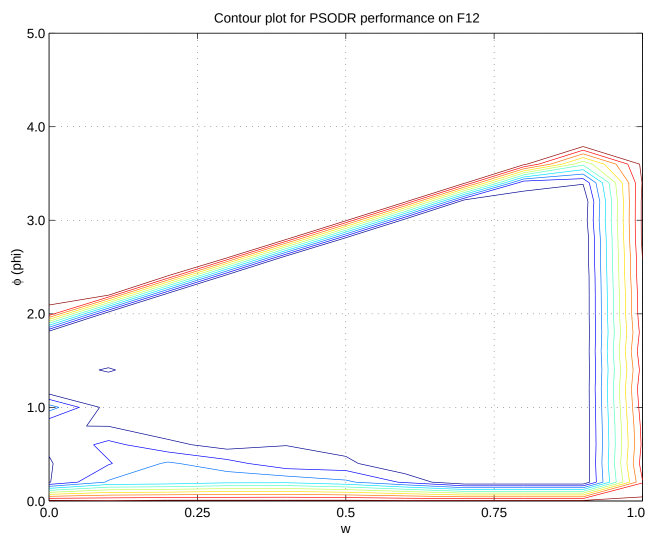
(a) f_1 (Sphere/Parabola)



(b) f_3 (Generalized Rosenbrock)



(c) f_5 (Generalized Rastrigin)



(d) f_{12} (Shekel 5)

Figure 1: Contour plots of PSO-DR performance for multiple combinations of *w* and ϕ.

Table 1: Benchmark Functions

Equation	Name	D	Feasible Bounds
$f_1 = \sum_{i=1}^{D} x_i^2$	Sphere/Parabola	30	$(-100, 100)^D$
$f_2 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	Schwefel 1.2	30	$(-100, 100)^D$
$f_3 = \sum_{i=1}^{D-1} \left\{100 \left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right\}$	Generalized Rosenbrock	30	$(-30, 30)^D$
$f_4 = -\sum_{i=1}^{D} x_i \sin\left(\sqrt{x_i}\right)$	Generalized Schwefel 2.6	30	$(-500, 500)^D$
$f_5 = \sum_{i=1}^{D} \left\{x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right\}$	Generalized Rastrigin	30	$(-5.12, 5.12)^D$
$f_6 = -20\exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right\} - \exp\left\{\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right\} + 20 + e$	Ackley	30	$(-32, 32)^D$
$f_7 = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Generalized Griewank	30	$(-600, 600)^D$
$f_8 = \frac{\pi}{D}\left\{10\sin^2\left(\pi y_i\right) + \sum_{i=1}^{D-1}\left(y_i - 1\right)^2\left\{1 + 10\sin^2\left(\pi y_{i+1}\right)\right\} + \left(y_D - 1\right)^2\right\}$	Penalized Function P8	30	$(-50, 50)^D$
$+\sum_{i=1}^{D}\mu\left(x_i, 10, 100, 4\right)$			
$y_i = 1 + \frac{1}{4}\left(x_i + 1\right)$			
$\mu\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k\left(-x_i - a\right)^m & x_i < -a \end{cases}$			
$f_9 = 0.1\left\{\sin^2\left(3\pi x_i\right) + \sum_{i=1}^{D-1}\left(x_i - 1\right)^2\left\{1 + \sin^2\left(3\pi x_{i+1}\right)\right\} + \left(x_D - 1\right)^2 \times\right.$	Penalized Function P16	30	$(-50, 50)^D$
$\left.\left\{1 + \sin^2\left(2\pi x_D\right)\right\}\right\} + \sum_{i=1}^{D}\mu\left(x_i, 5, 100, 4\right)$			
$f_{10} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	Six-hump Camel-back	2	$(-5, 5)^D$
$f_{11} = \left\{1 + \left(x_1 + x_2 + 1\right)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right\} \times$	Goldstein-Price	2	$(-2, 2)^D$
$\left\{30 + \left(2x_1 - 3x_2\right)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right\}$			
$f_{12} = -\sum_{i=1}^{5}\left\{\sum_{j=1}^{4}\left(x_j - a_{ij}\right)^2 + c_i\right\}^{-1}$	Shekel 5	4	$(0, 10)^D$
$f_{13} = -\sum_{i=1}^{7}\left\{\sum_{j=1}^{4}\left(x_j - a_{ij}\right)^2 + c_i\right\}^{-1}$	Shekel 7	4	$(0, 10)^D$
$f_{14} = -\sum_{i=1}^{10}\left\{\sum_{j=1}^{4}\left(x_j - a_{ij}\right)^2 + c_i\right\}^{-1}$	Shekel 10	4	$(0, 10)^D$

by ascending values of p, then setting:

$$\alpha' = \frac{\alpha}{inverse\ rank} \tag{6}$$

Results for these statistical tests are shown in Table 3 and confirm that the results are significantly improved on 4 of the 14 tested functions, equal for 9 functions, and worsened for 1 function for PSO-DR vs SPSO with ring topology. These tests were limited to the ring topology due to its exclusive use in all other comparison tests performed. Perhaps the most impressive improvement comes for f_5 (Rastrigin), a notoriously difficult multimodal problem that PSO algorithms often perform poorly on in high dimensionality.

4. EXAMINATION OF BURSTING

Additional tests were performed to measure particle speed on both SPSO and PSO-DR with ring topologies, using a normalised measure

$$z = \frac{|\vec{v}|}{|\vec{p}_1 - \vec{p}_2|} \tag{7}$$

where $\vec{p}_1 = \vec{p}_n$ for the particle being measured, $\vec{p}_2 = \vec{p}_i$ under SPSO, and $\vec{p}_2 = \vec{r}_i$ (the recombinant term) under PSO-DR. Percentile plots of particle velocity at updates to p_i for both algorithms are shown in Figure 2.

Examination of the percentile plots shows that p_i is updated by particles with higher velocities only very rarely for both algorithms. However, it is important to note that speed for this particular experiment will include particles with very high velocities in only a few dimensions and low velocities in other dimensions, as well as particles with moderately high velocities in most or all dimensions. In order to investigate bursting behavior a further 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 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 3, where the velocity of a single particle in a 10-dimensional problem is shown. On the plot of the multi-dimensional velocity of the SPSO particle, it can be seen that velocity in a 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 3, 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

Table 2: Mean fitness after 30 trials of 300000 evaluations

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7
SPSO Ring	0.0±0.0	0.12±0.01	6.18±1.07	3385±40	163.50±5.64	18.28±0.85	0.0±0.0
func evals	97063±377	-	-	-	-	-	110616±3320
PSO-DR Ring	0.0±0.0	0.01±0.002	16.79±0.49	2697±36	44.64±2.71	0.68±0.67	0.0±0.0
	59322±125	-	-	-	-	-	101526±9227
SPSO Global	0.0±0.0	0.0±0.0	8.37±2.26	3522±32	140.16±5.87	12.93±1.59	0.019±0.004
	46897±421	297645±848	-	-	-	-	-
PSO-DR Global	0.0±0.0	0.0±0.0	0.80±0.29	3754±48	115.51±7.03	18.51±0.90	0.008±0.002
	33290±170	168852±1205	-	-	-	-	-
Algorithm	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
SPSO Ring	0.0035±0.0034	0.0±0.0	0.0±0.0	0.0±0.0	0.59±0.33	1.09±0.45	0.96±0.45
func evals	-	106163±537	9348±190	8258±104	-	-	-
PSO-DR Ring	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.17±0.17	0.0±0.0	0.0±0.0
	61370±249	61793±221	44577±7608	4772±47	-	13433±249	12760±1386
SPSO Global	0.15±0.05	0.003±0.0009	0.0±0.0	0.0±0.0	4.61±0.54	4.40±0.60	3.24±0.66
	-	-	11808±445	7080±108	-	-	-
PSO-DR Global	0.05±0.02	0.002±0.0007	0.0±0.0	0.0±0.0	4.34±0.59	2.55±0.62	3.13±0.66
	-	-	40015±4483	3968±27	-	-	-

Table 3: Significance for SPSO vs PSO-DR with ring topologies

Func	p-value	Inverse rank	α'	Significant
f_4	0	14	0.003571	Yes
f_5	0	13	0.003846	Yes
f_6	0	12	0.004167	Yes
f_3	2.02e-12	11	0.004545	Yes
f_2	1.31e-10	10	0.005	Yes
f_{13}	0.021	9	0.005556	No
f_{14}	0.0414	8	0.00625	No
f_8	0.3215	7	0.007143	No
f_{12}	0.2663	6	0.008333	No
f_1	1	5	0.01	No
f_7	1	4	0.0125	No
f_9	1	3	0.016667	No
f_{10}	1	2	0.025	No
f_{11}	1	1	0.05	No

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 indicate that it is a highly improbable feature of DR dynamics.

Although the explanation for these rare events cannot lie on multiplicative stochasticity, PSO-DR will have a resonant frequency so attractor movement at this frequency, or at a multiple of this frequency could drive a particle away from the attractors. This mechanism is not possible in SPSO which does not possess a natural frequency due to the stochastic acceleration parameters.

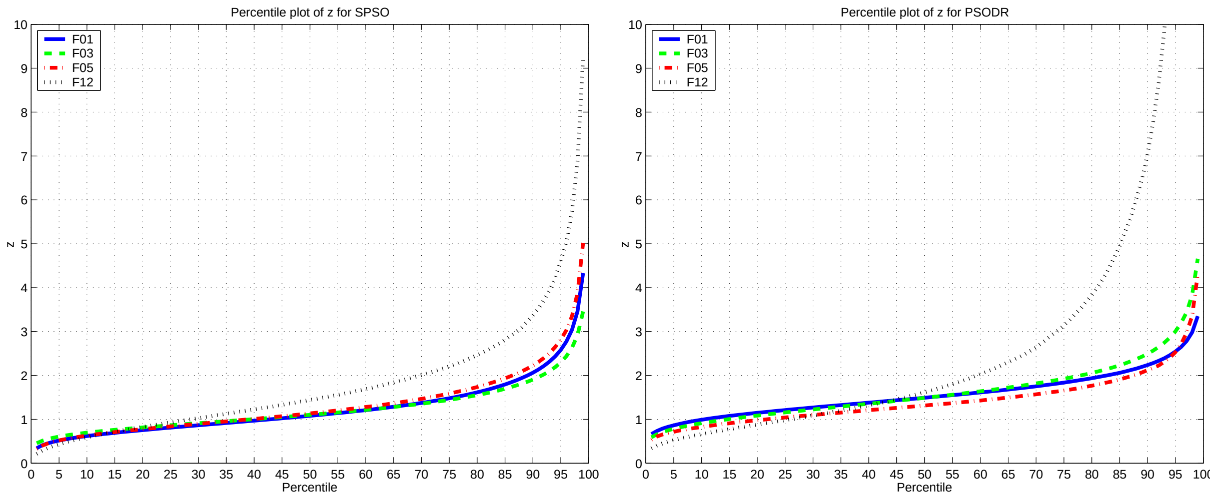
Analysis performed on statistics of several functions shows that particle updates involving bursts are far less effective than more common non-bursting updates. For example, results showed that for SPSO on f_5 with $\lambda = 100$, on average 20.1% of *all* particle updates involve an improvement to 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 4.

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, e.g. 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 $\lambda = 100$).

5. CONCLUSIONS

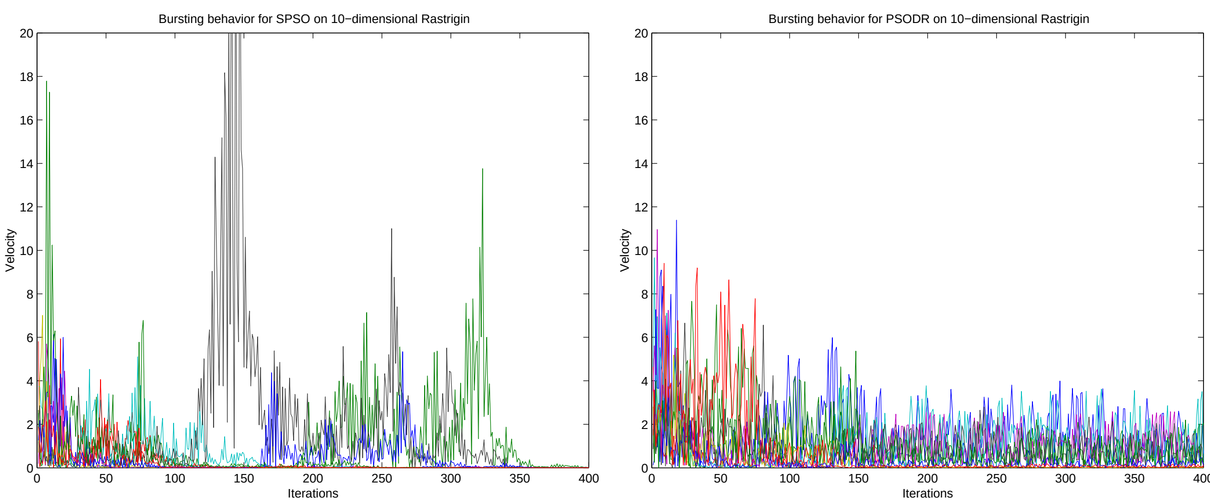
Simplification of the standard PSO algorithm is an important step toward understanding how and why it is an effective optimizer. By removing components of the algorithm and seeing how this affects performance, we are granted insight into what those components contribute to overall particle and swarm behavior. There is still much to be done before questions about what exactly makes PSO behave in the way that it does can be completely answered, and it is expected that the next decade of PSO research will be focus on understanding the basic algorithm that powers both the standard and the numerous variant implementations.

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 bursts



(a) SPSO (b) PSO-DR

Figure 2: Percentile plots of update frequency to p_i with particle velocity measure z on benchmark function f_3



(a) SPSO (b) PSO-DR

Figure 3: Representative particle velocities for SPSO and PSO-DR on 10D Rastrigin

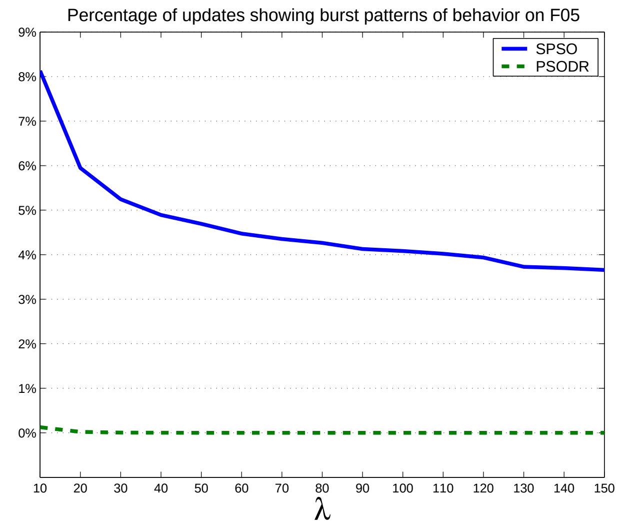


Figure 4: Frequency of updates showing burst behavior for values of λ

are not generally beneficial or integral to PSO performance, and may possibly be detrimental. This may be because bursting happens along an axis and so differs from the exploratory outliers of bare bones particles which sample the surrounding region symmetrically. (However, in the coincidence that the objective function has a rectangular symmetry aligned with the axes, then bursting may actually be fortuitous.) Furthermore, lost particles in large bursts cannot contribute to the search. Because of the association of bursting behavior with the random acceleration parameter in SPSO [1], we can see that this part of the algorithm may not be necessary to produce the desired optimization behavior.

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. 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.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge the support of EPSRC XPS project (GR/T11234/01). The authors also wish to thank J. Kennedy.

7. REFERENCES

[1] T. M. Blackwell and D. Bratton. Origin of bursts. In *Submitted to Particles swarms: the second decade, workshop, GECCO 2007*, 2007.

[2] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization. In *IEEE swarm intelligence symposium*, 2007.

[3] D. K. Gehlhaar and D. B. Fogel. *Tuning evolutionary programming*. 1996.

[4] J. Jaccard and C. Wan. *LISREL approaches to interaction effects in multiple regression*. Sage Publications, Thousand Oaks, CA, 1996.

[5] J. Kennedy. Probability and dynamics in the particle swarm. In *IEEE congress on evolutionary computation*, pages 340–347, 2004.

[6] C. Monson and K. Seppi. Exposing origin-seeking bias in pso. In *Proceedings of the 2005 Genetic and Evolutionary Computation Conference*, pages 241–248, 2005.

[7] J. Pena, A. Upegui, and E. Sanchez. Particle swarm optimization with discrete recombination: an online optimizer for evolvable hardware. In *First NASA/ESA conference on adaptive hardware and systems, AHS '06*, pages 163–170, 2007.

[8] T. Richer and T. Blackwell. The levy particle swarm. In *IEEE congress on evolutionary computation*, pages 3150–3157, 2006.