Selection Strategies for Initial Positions and Initial Velocities in Multi-optima Particle Swarms

Stephen Chen
School of Information Technology
York University
4700 Keele Street, Toronto, ON
(416) 736-2100 x30526
sychen@yorku.ca

James Montgomery
Faculty of ICT
Swinburne University
of Technology, Hawthorn, VIC
+61 3 9214 5735
jmontgomery@swin.edu.au

ABSTRACT
Standard particle swarm optimization cannot guarantee convergence to the global optimum in multi-modal search spaces, so multiple swarms can be useful. The multiple swarms all need initial positions and initial velocities for their particles. Several simple strategies to select initial positions and initial velocities are presented. A series of experiments isolates the effects of these selected initial positions and velocities compared to random initial positions and velocities. A first set of experiments shows how locust swarms benefit from "scouting" for initial positions and the use of initial velocities that "launch away" from the previous optimum. A second set of experiments show that the performance of WoSP (Waves of Swarm Particles) can be improved by using new search strategies to select the initial positions and initial velocities for the particles in its sub-swarms.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*

General Terms
Algorithms

Keywords
Particle swarm optimization, coarse search-greedy search, exploration-exploitation, locust swarms, WoSP, multi-swarm system

1. INTRODUCTION
Particle swarm optimization (PSO) [10] mimics the flocking of birds and the schooling of fish to perform a greedy search around the best found location. This "global best" location acts as a position-based attractor which draws the other particles in the swarm to search around it. The convergent nature of each particle's search trajectory encourages the swarm to perform a highly exploitive search around the global best location.

A key challenge in the implementation of heuristic search techniques is the balancing of exploration and exploitation. One way to promote more exploration in particle swarms is to use a ring topology (i.e. LBest) instead of a star topology (i.e. GBest) [1]. With less communication among the particles, there is less convergence towards the global best location (i.e. exploitation), and more opportunity for (groups of) particles to explore areas around different local best locations. However, the cost of increased exploration can be greater computational effort/reduced computational efficiency.

One general framework that can reduce the conflict between exploration and exploitation is the multi-optima particle swarm – a multi-swarm system where each swarm converges to a (different) local optimum. The underlying sub-swarms focus more on exploitation, and a new mechanism is added between each sub-swarm to focus more on exploration. As opposed to a compromise which may weaken both the exploitive and explorative mechanisms of a search process, using two distinct phases can potentially increase the efficiency of the exploitive components. Multiple executions of these more concentrated, exploitive searches may then lead to a more efficient and effective overall algorithm. The use of this two-phase process adds three new design decisions – multi-optima particle swarms must specify the number of swarms to use and/or the stopping criteria for each of the sub-swarms; have a strategy to select the initial positions of the particles in the sub-swarms; and have a selection strategy for the initial velocities of these particles.

The first multi-swarm system studied in this paper is locust swarms [3]. In locust swarms, the overall strategy is based on a "devour and move on" strategy. Addressing the three new design issues, locust swarms use a fixed number of sub-swarms which converge more rapidly than standard PSO – each of these sub-swarms "devours" a relatively small region of the search space to find a local optimum. For these sub-swarms, the initial positions are selected by "scouts" [4]. If the scout position is fitter/better than the previous optimum, it is possible that the scout particle has found an underlying gradient in the search space. Thus, the initial velocity of a locust/particle in locust swarms is on a line from the previous optimum towards the scout location.

Locust swarms obtain both positional and directional information from the scouts to initialize the positions and velocities of the particles in each new sub-swarm. Variations of locust swarms that use only positional information or only directional information are created to isolate the effects of these two independent design decisions. The goal of these experiments is to demonstrate how the selection of both initial positions and initial velocities can

each have important effects on the performance of multi-optima particle swarms.

The second multi-swarm system studied in this paper is WoSP (Waves of Swarm Particles) [9]. WoSP uses a short-range force which (under the effects of discrete time updates) prevents complete convergence. Specifically, as two particles become closer together (e.g. as they near convergence), the magnitude of this force becomes larger. Based on discrete updates, the magnitude of this force on any particle within a certain radius will be sufficient to cause the two particles to radically overshoot from their previously convergent trajectories.

Starting a new wave/sub-swarm after such a divergence, WoSP addresses the new design issues by focusing primarily on the stopping criteria – the subsequent number of swarms is based on how many expulsions occur during the total number of allowed function evaluations. For the two remaining design issues, WoSP does not explicitly select initial positions or initial velocities for its new waves/sub-swarms. When two particles become sufficiently close such that the effects of the short-range force lead to divergence instead of convergence, a new wave is created using initial positions and initial velocities that are inherited relics from the previous wave. With respect to features of the search space, these values are effectively random.

New experiments are designed to first show that the performance of WoSP with these inherited values is essentially the same as it is with randomly selected initial positions and initial velocities. A modified version of WoSP is then developed in which the creation of a new wave leads to a search procedure that selects the initial positions and initial velocities for the particles in the new wave. The goal of these experiments is to further analyze the effects of initial positions and initial velocities on the operation and performance of a vastly different multi-optima particle swarm.

It is obvious that well-selected initial positions can improve search performance. Thus, the isolation of the effects of initial velocities is the key goal of this paper. The development of supporting results begins with an introduction of locust swarms in Section 2 and baseline results for locust swarms on the Black Box Optimization Benchmark (BBOB) problem set [7] in Section 3. In Section 4, variations of locust swarms are developed to isolate the effects of the initial positions and initial velocities selected by scouting and to show comparisons with randomly selected initial positions and initial velocities. This analysis is extended to WoSP which is introduced in Section 5. A new selection technique for initial positions and initial velocities is presented in Section 6 and their combination with WoSP follows in Section 7. The discussion in Section 8 highlights contrasting selection techniques for initial positions and initial velocities used by other PSO researchers. Finally, a brief summary of the paper's results is provided in Section 9.

2. LOCUST SWARMS

Locust swarms were developed based on a "devour and move on" strategy. After the particles/locusts have spent sufficient time "devouring" an area (to the point that further improvements are unlikely), there comes a time to "move on" to new (ideally unexplored) regions of the search space. Locust swarms use periodic restarts, so the number of swarms is selected *a priori*, and the swarm parameters must be selected to ensure adequate convergence in the allotted time/function evaluations.

The benchmark and embedded PSO for the current experiments is a constricted LBest version (i.e. standard PSO [1]) developed from the published source code for the constricted GBest version by El-Abd and Kamel [6] – code was added to convert the star topology into a ring topology. In a constricted PSO, each dimension d of a particle's velocity v is updated for the next iteration i+1 by

$$v_{i+1,d} = \chi\left(v_{i,d} + c_1\varepsilon_1\left(pbest_{i,d} - x_{i,d}\right) + c_2\varepsilon_2\left(gbest_{i,d} - x_{i,d}\right)\right) \quad (1)$$

where χ is the constriction factor, c_1 and c_2 are weights which vary the contributions of attractions towards personal best and global best locations, ε_1 and ε_2 are independent uniform random numbers in the range of [0,1], x is the location of the particle, *pbest* is the best position found by the current particle, and *gbest* is the best position found by any particle communicating with the current particle (i.e. all particles in the GBest star topology or two neighbouring particles in an LBest ring topology).

The specific weights used in [6] are $\chi = 0.792$ and $\chi^* c_1 = \chi^* c_2 = 1.4944$, i.e. $c_1 = c_2 \approx 1.887$. The published implementation uses $p = 40$ particles, so the use of 5000 times the number of dimensions D as the limit for function evaluations ($FE = 5000 * D$) leads to $n = 125 * D$ iterations per particle in the swarm. Results for the original constricted GBest version of this benchmark PSO on the BBOB problem set are reported in [5][6], and results for the constricted LBest version are presented in Section 3.

To build a multi-optima particle swarm on top of this constricted LBest implementation, the number of function evaluations in each embedded swarm is reduced. The comparison of the effectiveness of multi-swarm systems with standard PSO is made using a constant number of function evaluations (i.e. $FE = 5000 * D$). Previous work with locust swarms showed that 1000 function evaluations per swarm (5 particles * 200 iterations) can lead to good overall results [5]. To benefit from the increased explorative capabilities of the constricted LBest implementation, this has been increased to 1500 function evaluations with $p = 10$ particles per swarm. Previous implementations also balanced the explorative nature of the scouts with the exploitive nature of the swarms (i.e. the number of scouts $S = p * n$), but the LBest implementation is more explorative than previously used GBest versions. Thus, the number of scouts S is left at 1000 for 2500 total function evaluations per optimum/swarm. Given the overall limit of $FE = 5000 * D$ function evaluations, the experiments in Sections 3 and 4 use $N = 2 * D$ for the total number of optima/swarms.

With fewer iterations per particle (i.e. n = 150), the rate of convergence is increased by multiplying the previous constriction factor by 0.9, i.e. $\chi = 0.7128$ is used in the sub-swarms. Having decided on $N = 2 * D$ swarms and a fixed number of function evaluations per swarm, the remaining design decisions are to select the initial positions and the initial velocities for the particles in each of the $N - 1$ new swarms. (Note: the initial positions and initial velocities for the first swarm are all uniform random numbers drawn from the range of the search space – see (2) and (3) respectively.)

$$x_{o,d} = \varepsilon(upperbound_d - lowerbound_d) + lowerbound_d \quad (2)$$

$$v_{o,d} = \varepsilon(upperbound_d - lowerbound_d) + lowerbound_d \quad (3)$$

ε is a uniform random number in the range of [0,1] and the upper and lower bound for all dimensions of all BBOB functions are +5 and –5 respectively.

The obvious goal for the selection of initial positions is to do better than random initial positions. The simplest search strategy is random search, and it has been shown that a (greedy) search technique (like PSO) can perform better when it is started from points selected by random search as opposed to purely random points [4]. Thus, the $p = 10$ points used as the initial particle positions are the p best of S = 1000 (random) scout points.

Unlike a completely random restart (where the scouts/initial positions are randomly distributed throughout the entire search space – see (2)), locust swarms "move on" from the previous optimum. Scout points are created from the previous optimum by using (4) for *DIMr* randomly selected dimensions and (5) for the remaining $D - DIMr$ dimensions. Previous work in [5] has shown that reducing the number of dimensions being actively searched has an important effect on the performance of locust swarms. In the following experiments, *DIMr* for each scout is drawn randomly with a uniform distribution from 1 to 10 inclusive. Compared to the fixed value of *DIMr* = 5 used in [5], the randomly drawn values lead to much better performance on the separable and unimodal functions at the cost of slightly lower performance on some of the non-separable, multi-modal functions.

$$scout_d^j = optimum_d^{j-1} + delta_d \quad (4)$$

$$scout_d^j = optimum_d^{j-1} \quad (5)$$

The scouts for swarm j of N are distributed around the previous optimum (i.e. $optimum^{j-1}$) by adding a *delta* (6) to *DIMr* terms. In (6), *gap* = 0.01 and *spacing* = 0.3 are used for the experiments in Sections 3 and 4, and *randn*() is a normally distributed random number with mean = 0 and standard deviation = 1. The performance of locust swarms is actually quite sensitive to these parameters with the best performance occurring when their values are matched to the spacing of local optima (or other gradient features) in the search space. For unimodal problems, *gap* = 0 works best, but preliminary parameter tuning discovered that a *gap* of 0.01 was sufficient for most of the multi-modal problems without degrading the performance on the other problems too severely.

$$delta_d = \pm range_d * (gap + abs(randn() * spacing)) \quad (6)$$

Better scouts are likely located in more promising areas of the search space, and global structure could lead to even more promising areas in the direction of the best scouts – see Figure 1. The primary component of the initial velocity in (7) attempts to exploit any such underlying gradients and/or global structure that may exist in a search space. As shown in Figure 1, the initial velocity will "launch" the locust away from the previous optimum and towards these promising new areas of the search space. Note: a "noise" term of 1% of (3) is also added to the initial velocity to improve local search capabilities.

$$v_0^j = x_0^j - optimum^{j-1} \quad (7)$$

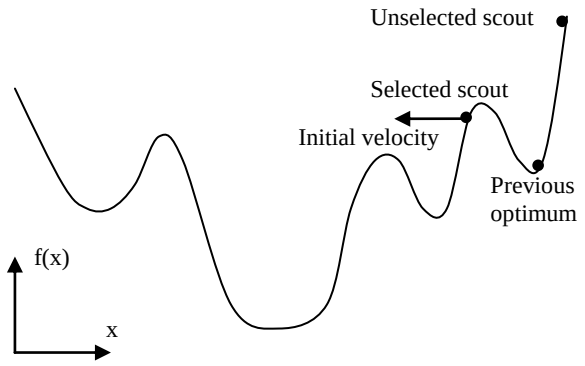

Figure 1. Example of directed initial velocity

3. RESULTS FOR LOCUST SWARMS

The experiments on locust swarms have been performed using the Black-Box Optimization Benchmarking (BBOB) functions [7]. The BBOB problems are broken into five sets – (1) separable functions, (2) functions with low or moderate conditioning, (3) unimodal functions with high conditioning, (4) multi-modal functions with adequate global structure, and (5) multi-modal functions with weak global structure. In Table 1, some key attributes of the functions (fn) are indicated – i.e. whether or not they are separable (s), unimodal (u), or have (adequate) global structure (gs).

The BBOB results are reported as expected running times to certain levels of performance. However, the benchmark PSO implementation [6] has a limit of $FE = 10{,}000 * D$ – the constriction factor effects a convergence rate, and additional computational effort after the expected time to convergence is most likely wasted. Although expected running times are a useful

Table 1. BBOB Functions

Set	fn	Function Name	Attribute		
			s	u	gs
1	1	Sphere	X	X	X
	2	Ellipsoidal, original	X	X	X
	3	Rastrigin	X		X
	4	Büche-Rastrigin	X		X
	5	Linear Slope	X	X	
2	6	Attractive Sector		X	
	7	Step Ellipsoidal			X
	8	Rosenbrock, original			
	9	Rosenbrock, rotated			
3	10	Ellipsoidal, rotated		X	X
	11	Discus		X	X
	12	Bent Cigar		X	
	13	Sharp Ridge		X	
	14	Different Powers		X	
4	15	Rastrigin, rotated			X
	16	Weierstrass			X
	17	Schaffers F7			X
	18	Schaffers F7, moderately ill-conditioned			X
	19	Composite Griewank-Rosenbrock F8F2			X
5	20	Schwefel			
	21	Gallagher's Gaussian 101-me Peaks			
	22	Gallagher's Gaussian 21-hi Peaks			
	23	Katsuura			
	24	Lunacek bi-Rastrigin			

Table 2. Locust Swarms vs. Standard PSO – D = 20

fn	Locust Swarms		Standard PSO		%-diff	p-value
	mean	std dev	mean	std dev		
1	7.50e-6	5.20e-6	0.00e+0	0.00e+0	- ∞	0.0%
2	5.55e-3	6.38e-3	0.00e+0	0.00e+0	- ∞	0.0%
3	9.59e+0	3.30e+0	2.56e+1	4.99e+0	62.5%	0.0%
4	1.44e+1	3.53e+0	3.23e+1	8.55e+0	55.6%	0.0%
5	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.0%	
6	4.15e-1	2.42e-1	8.53e-1	8.89e-1	51.3%	3.1%
7	2.72e+0	9.57e-1	7.04e+0	2.68e+0	61.4%	0.0%
8	1.29e+1	4.01e+0	1.22e+1	3.67e+0	-5.4%	53.4%
9	1.58e+1	1.97e+0	1.55e+1	2.24e+0	-1.8%	66.8%
10	3.30e+3	1.66e+3	6.85e+3	3.39e+3	51.8%	0.0%
11	2.84e+1	6.39e+0	6.54e+1	1.71e+1	56.6%	0.0%
12	1.79e+1	1.57e+1	1.53e+0	4.23e+0	-1.0e3%	0.0%
13	3.49e+0	3.43e+0	1.50e+0	1.99e+0	-132.9%	3.6%
14	3.35e-3	8.77e-4	1.34e-3	2.66e-4	-150.5%	0.0%
15	3.05e+1	7.87e+0	6.05e+1	1.46e+1	49.6%	0.0%
16	2.91e+0	1.13e+0	5.37e+0	1.53e+0	45.8%	0.0%
17	1.81e-1	7.24e-2	6.61e-1	2.64e-1	72.6%	0.0%
18	1.00e+0	4.00e-1	2.87e+0	1.28e+0	65.1%	0.0%
19	3.16e+0	4.34e-1	3.61e+0	4.32e-1	12.6%	0.0%
20	6.95e-1	1.62e-1	1.14e+0	1.38e-1	38.7%	0.0%
21	2.85e+0	3.00e+0	1.41e+0	1.21e+0	-101.7%	5.1%
22	4.92e+0	5.43e+0	1.69e+0	1.51e+0	-190.2%	1.1%
23	6.43e-1	2.69e-1	1.33e+0	2.49e-1	51.7%	0.0%
24	7.23e+1	1.57e+1	1.13e+2	1.12e+1	35.8%	0.0%

Table 3. Locust Swarms vs. Standard PSO – D = 200

fn	Locust Swarms		Standard PSO		%-diff	p-value
	mean	std dev	mean	std dev		
1	4.90e-2	1.54e-2	1.49e-7	1.03e-7	-3.2e6%	0.0%
2	4.72e+1	1.39e+1	1.52e+0	7.57e+0	-3.0e3%	0.0%
3	2.46e+2	1.88e+1	1.10e+3	1.13e+2	77.7%	0.0%
4	3.19e+2	2.97e+1	1.64e+3	2.04e+2	80.5%	0.0%
5	0.00e+0	0.00e+0	4.91e+1	3.43e+1	100.0%	0.0%
6	3.26e+1	6.19e+0	1.95e+3	1.73e+2	98.3%	0.0%
7	1.43e+2	2.51e+1	1.34e+3	1.87e+2	89.3%	0.0%
8	3.22e+2	5.18e+1	1.00e+3	7.97e+2	67.8%	0.0%
9	2.25e+2	2.64e+1	7.08e+2	1.79e+2	68.2%	0.0%
10	5.84e+4	1.05e+4	8.36e+5	1.67e+5	93.0%	0.0%
11	1.53e+2	1.31e+1	5.54e+2	3.29e+1	72.3%	0.0%
12	4.40e+4	1.40e+4	3.09e+4	1.54e+5	-42.4%	67.3%
13	5.07e+1	7.49e+0	2.55e+2	1.77e+2	80.1%	0.0%
14	6.02e-2	6.37e-3	2.74e-1	4.03e-1	78.1%	1.4%
15	1.11e+3	1.19e+2	2.37e+3	3.50e+2	53.2%	0.0%
16	7.68e+0	9.47e-1	3.18e+1	2.50e+0	75.9%	0.0%
17	5.28e+0	7.76e-1	6.82e+0	8.66e-1	22.6%	0.0%
18	1.67e+1	2.46e+0	2.60e+1	3.52e+0	35.9%	0.0%
19	9.17e+0	2.83e-1	1.29e+1	7.77e-1	28.6%	0.0%
20	1.39e+0	8.28e-2	1.81e+2	6.76e+2	99.2%	19.6%
21	2.75e+0	2.39e+0	2.56e+0	1.68e+0	-7.4%	71.8%
22	4.11e+0	4.16e+0	4.01e+0	4.17e+0	-2.6%	91.1%
23	1.80e+0	1.68e-1	3.07e+0	1.65e-1	41.3%	0.0%
24	1.80e+3	7.76e+1	2.96e+3	1.24e+2	39.3%	0.0%

measure for multi-optima search techniques, the current focus is on analyzing the improved efficiencies that multi-swarm techniques can have over standard PSO. With additional computational resources, multi-swarm techniques should be easier to scale since more optima can be explored without requiring more complicated adjustments to swarm parameters (e.g. the constriction factor χ). Using fixed computational resources in the following experiments, the choice of $FE = 5000 * D$ provides some consistency with other results (e.g. [2][5][16][17]).

On the BBOB functions, the "devour and move on" strategy for locust swarms was designed to provide a specific advantage on multi-modal functions [3] (e.g. BBOB fn 15-24). Dimension reductions in the scouts (which allow lower-dimensional searches) should also provide locust swarms with an advantage on separable functions [5] (i.e. BBOB fn 1-5). The aim of using all the functions in the standardized BBOB problem set is to show the robust improvements available with locust swarms, and to allay any concerns that only functions with specific features advantageous to locust swarms were selected.

Previous implementations of locust swarms [2][3][5] have been based on GBest sub-swarms. Compared to the GBest benchmark [6], the superior performance of locust swarms demonstrates that the initialization of the new swarms with the help of scouts does indeed provide some advantages. However, these advantages were not enough to overcome the superior performance of an LBest PSO (i.e. standard PSO [1]). Thus, the current implementation of locust swarms is based on LBest sub-swarms. The following results again help to show that there are performance advantages available to a multi-optima particle swarm when scouts are used to select the initial positions and initial velocities of each new sub-swarm.

The current experiments compare standard PSO [1] and the implementation of locust swarms as described in Section 2 by using the BBOB problem set with D = 20 dimensions (see Table 2) and D = 200 dimensions (see Table 3). The presented results are the mean errors from optimum and standard deviations (std dev) for five trials on each of the first five instances of each BBOB function (i.e. 25 independent trials for each function). The relative improvement (%-diff) achieved with locust swarms versus standard PSO is shown, and so is the p-value of a t-test to help measure the significance of these differences in performance.

The results in Tables 2 and 3 show that the relative performance of locust swarms as compared to standard PSO improves greatly with increasing dimensionality. As discussed in [5], this result is primarily caused by the ability of locust swarms to use lower dimensional searches, and these benefits exist even in non-separable search spaces. In addition to this overall trend, it can be seen that locust swarms have trouble finding exact (local) optima (e.g. BBOB fn 1 and 2 in Table 2) because each sub-swarm does not converge to the same extent as a single large swarm. For D = 20 dimensions (see Table 2), locust swarms have some relative performance weaknesses on the unimodal functions (BBOB fn 10-14) but have significant advantages as intended on the multi-modal functions with adequate global structure (BBOB fn 15-19).

4. ISOLATION OF THE EFFECTS OF NEW POSITIONS AND NEW VELOCITIES

It is claimed that the performance of locust swarms benefits from the selection of both initial positions and initial velocities during the scouting phase. To isolate each of these benefits, modified versions of locust swarms have been created – the "positions only" version uses the scout positions (4)-(6) with random velocities (3) and the "velocities only" version uses the scout velocities (7) on a set of particles initially located at the previous optimum. A final version uses random velocities (3) with particles at the previous optimum. All sub-swarm parameters for this

Table 4. Effects of initial positions and velocities – $D = 20$

fn	Locust Swarms	Positions Only	Velocities Only	Random
1	1.0	13.2	0.0	0.0
2	1.0	15.5	0.0	16043.5
3	1.0	1.3	0.8	3.6
4	1.0	1.2	1.0	4.3
5	1.0	1.0	1.0	1.0
6	1.0	2.3	0.5	9.0
7	1.0	1.0	4.9	5.3
8	1.0	1.1	0.6	1.4
9	1.0	1.2	0.9	1.1
10	1.0	1.3	0.8	0.9
11	1.0	1.1	1.2	1.7
12	1.0	4.5	0.3	0.2
13	1.0	1.8	2.4	2.8
14	1.0	1.8	0.1	0.2
15	1.0	1.2	2.1	2.5
16	1.0	1.0	2.9	2.8
17	1.0	1.2	11.0	11.7
18	1.0	1.3	6.2	7.2
19	1.0	1.0	0.6	0.7
20	1.0	1.4	1.8	2.1
21	1.0	1.4	1.0	1.5
22	1.0	0.5	0.6	0.8
23	1.0	1.0	1.5	1.5
24	1.0	1.3	1.0	1.1

Table 5. Effects of initial positions and velocities – $D = 200$

fn	Locust Swarms	Positions Only	Velocities Only	Random
1	1.0	6.1	0.0	23.2
2	1.0	3.0	4.9	888.9
3	1.0	1.3	0.9	5.2
4	1.0	1.3	1.1	6.1
5	1.0	1.0	1.0	∞
6	1.0	5.0	6.6	13.9
7	1.0	1.1	13.9	15.1
8	1.0	2.6	1.2	17.6
9	1.0	1.9	1.1	1.1
10	1.0	2.2	0.9	1.7
11	1.0	1.3	0.5	1.4
12	1.0	6.4	0.0	111.6
13	1.0	2.6	0.0	3.5
14	1.0	2.6	0.0	17.9
15	1.0	1.3	3.0	3.1
16	1.0	1.4	4.9	5.2
17	1.0	0.9	1.4	1.4
18	1.0	1.0	1.7	1.6
19	1.0	1.0	1.9	2.1
20	1.0	1.4	1.2	1.2
21	1.0	1.0	1.0	1.4
22	1.0	0.8	0.9	1.1
23	1.0	1.1	1.4	1.4
24	1.0	1.2	1.3	1.4

"random" version are held constant which causes this version to use 40% fewer function evaluations (i.e. no scouting) – these results isolate the potential benefits of specifically selected initial positions and initial velocities without pursuing a full analysis of the costs associated with the involved selection strategies.

The results in Tables 4 and 5 show the ratio of the mean error for each version relative to the mean error for the base implementation of locust swarms (see Tables 2 and 3). The most consistent trend in the data is the advantage of scout-based initial velocities compared to random initial velocities. Specifically, the versions in column one and two both use scout-based initial positions, so they only differ by the use of scout-based initial velocities in the first column and random initial velocities in the second column. Similarly, the versions in column three and four both use the previous optimum as the initial position, so they only differ by the use of scout-based initial velocities in the third column and random initial velocities in the fourth column. In each of these column pairs, the value in the left column is almost always lower than the value in the right column – the specific selection of initial velocities (in isolation) has led to performance improvements in these otherwise identical implementations of multi-optima particle swarms. Table 6 shows the p-values for t-tests in these column-to-column comparisons – the bold values indicate that the use of selected initial velocities has led to a significant performance advantage compared to random initial velocities (i.e. $p < 5.0\%$). The scout-based selection of initial velocities is particularly effective on separable functions (BBOB fn 1-5) and unimodal functions (BBOB fn 10-14) for the higher dimensional problems.

The best overall results occur with the base version of locust swarms in which both scout-based initial positions and scout-based initial velocities are used. This is consistent with the goals of the initial parameter selection which attempted to avoid optimizing the parameters for one (set of) function(s) at the cost of another. However, it should be noted that using the previous optimum instead of the scout-based positions in the velocities only version has many advantages on the unimodal functions (e.g. BBOB fn 10-14). Conversely, the lack of scout-based initial positions in the same velocities only version has many disadvantages on the multi-modal functions (e.g. BBOB fn 15-19). Overall, the importance of initial positions and initial velocities will vary in different fitness landscapes.

Table 6. Significance of initial velocities

fn	$D = 20$		$D = 200$	
	Columns 1-2	Columns 3-4	Columns 1-2	Columns 3-4
1	**0.0%**	73.7%	**0.0%**	**0.5%**
2	**0.0%**	8.2%	**0.0%**	**0.0%**
3	**0.4%**	**0.0%**	**0.0%**	**0.0%**
4	**0.3%**	**0.0%**	**0.0%**	**0.0%**
5				32.7%
6	**0.6%**	**4.8%**	**0.0%**	**0.0%**
7	86.9%	54.4%	5.6%	13.5%
8	5.5%	**3.5%**	**0.0%**	7.3%
9	30.5%	22.5%	**0.0%**	75.1%
10	5.6%	36.3%	**0.0%**	**0.0%**
11	9.1%	**0.3%**	**0.0%**	**0.0%**
12	**0.0%**	54.0%	**0.0%**	**2.0%**
13	**0.4%**	65.0%	**0.0%**	**0.0%**
14	**0.0%**	**0.0%**	**0.0%**	**0.2%**
15	**0.2%**	7.1%	**0.0%**	29.1%
16	96.8%	49.4%	**0.0%**	**1.1%**
17	6.5%	60.1%	15.3%	10.3%
18	**1.6%**	12.6%	71.1%	0.6%
19	55.5%	60.1%	**0.0%**	5.9%
20	**0.0%**	**0.3%**	**0.0%**	96.7%
21	29.8%	37.5%	96.5%	30.1%
22	2.5%	15.6%	39.0%	32.8%
23	98.3%	68.2%	**0.5%**	96.6%
24	**0.0%**	10.4%	**0.0%**	11.0%

5. WOSP

The effects of selecting initial positions and initial velocities will also depend on the particular multi-optima particle swarm. WoSP (Waves of Swarm Particles) is a multi-optima particle swarm that adds a short range force to prevent complete convergence [9]. As particles converge, this short range force (under the effects of discrete time updates) causes them to accelerate greatly and subsequently diverge instead. After divergence, a new wave is created which is not attracted to the *pbest* and *gbest* locations of any previous wave. Specifically, WoSP starts with the same velocity update equation as standard PSO in (1)

$$v_{i+1,d}^{n} = \chi\left(v_{i,d}^{n} + c_1\varepsilon_1\left(pbest_{i,d}^{n} - x_{i,d}^{n}\right) + c_2\varepsilon_2\left(gbest_{i,d}^{n} - x_{i,d}^{n}\right)\right) \quad (8)$$

except that particles are now grouped into waves numbered by n. The short range force adds another velocity component:

$$v_{ij} = \frac{K}{d_{ij}^{p}} \quad (9)$$

where d_{ij} is the distance between particles i and j, K is a constant, and p is power factor that ensures that v_{ij} is small except for very small d_{ij}.

Whenever the total velocity of a particle is above a certain threshold, the new location for that particle will be so far away from the previous "promotion point" that it is considered to be in a new wave. Since this total velocity specifies a limit on the distance to which particles can converge (i.e. within this radius, the velocity component due to (9) alone will be sufficient to promote the particles into a new wave), WoSP typically requires local optimization of the promotion points to find exact optima. Further, the efficiency and effectiveness of WoSP benefit greatly if this "radius of convergence" is well matched to the size of the local optima wells in the search space. In general, the performance of WoSP can be quite sensitive to the short range force and other parameters related to wave promotion.

6. IMPROVED SCOUTING FOR WOSP

Upon promotion, the particle in the new wave keeps its existing position and velocity. The largest influence on these values will be from the short range force which is a function of the relative position between the current particle and the other particle that it interacted with. Essentially, with respect to information from the search space or previously found local optima, these values are effectively random. Further, based on the promotion criteria, the position will be quite far away from the promotion point and the magnitude of the velocity will be quite large. In terms of the three specified design criteria for multi-optima particle swarms, WoSP focuses primarily on a stopping condition for the sub-swarms, and it uses effectively random initial positions and velocities for its new sub-swarms.

The results of Sections 3 and 4 suggest that the performance of WoSP can be improved by better selection of the initial positions and initial velocities that it uses in its sub-swarms. A procedure similar to that used by locust swarms in (6) could be used, but the inability of locust swarms to "scale" the search process [12] to find the exact optimum on simple problems like sphere (BBOB fn 1) encourages the introduction of a search procedure that can be greedier and more adaptive.

In Differential Evolution (DE) [14], a solution x is created by applying a difference vector to a base solution

$$x = x_1 + F\left(x_2 - x_3\right) \quad (10)$$

where x_1, x_2, and x_3 are unique solutions drawn from the population, and F is a scaling factor. To the extent that scouts in locust swarms are tasked with finding exploitable gradients in the search space, difference vectors which are essentially slopes between two points may also be a promising way to identify exploitable gradients. Building from (10), the following search procedure is used

$$searchlocation^{n} = \varepsilon\left(optimum^{a} - optimum^{b}\right) + optimum^{c} \quad (11)$$

where a and b are randomly selected without replacement from $n/2$ to $n-2$ such that $optimum^{a}$ has a better fitness than $optimum^{b}$. Setting $c = n-1$ causes "scouting" to occur around the most recently found optimum/promotion point. Note: normal WoSP occurs for $n < 20$ – the initial waves are more explorative and the later waves are more exploitive. The difference vector is scaled by ε, a uniform random number drawn from a range of [0.5,1]. Similar to (7), the slope of the difference vector may represent an underlying gradient in the search space. Thus, the initial velocity uses this direction.

$$v_0^{n} = \varepsilon\left(optimum^{a} - optimum^{b}\right) \quad (12)$$

Preliminary experiments with locust swarms suggest that these "DE-based" scouts are more effective than the scouting procedure described in Section 2. The purpose of the following experiments is to study the potential effects of adding new selection techniques for initial positions and initial velocities on the performance of other multi-swarm systems – i.e. WoSP in this case.

7. RESULTS FOR WOSP

The BBOB problem set is not available in the programming language of the original implementation of WoSP. Further, the performance of WoSP is highly sensitive to its parameters, so it was desirable to work on only the original functions for which properly selected parameters were already available. Of these available functions, the Rastrigin function is the only one in common with the BBOB problem set. The parameters in (9) for WoSP on this function are K = 1.675 and p = 3.5.

This original version of WoSP is compared against several modified versions (all versions are allowed $5000*D = 100{,}000$ function evaluations). In Table 7, the mean and standard deviation (std dev) for 25 independent trials of each version are reported. The p-value of a t-test helps to show the likelihood that the given version has substantially similar performance to the base version of WoSP. The first modification replaces the initial position and initial velocity of each wave with a randomly selected initial position and initial velocity. The performance of this "random"

Table 7. Effects of initial positions and velocities – WoSP

Version	mean	std dev	t-test
WoSP	91.6	9.3	-
Random	91.8	11.5	90.6%
Position and Velocity	61.0	13.4	0.0%
Position only	73.1	19.1	0.0%
Velocity only	103.3	11.7	0.0%

version of WoSP is similar to the performance of the original version of WoSP – a result that supports the previous claim that WoSP focuses only on the first of the three proposed design criteria for multi-optima particle swarms (i.e. specifying the stopping criteria for each sub-swarm) and that it uses effectively random initial positions and random initial velocities.

The second modification adds the search/scouting procedure presented in Section 6. Upon the termination of a wave (numbered $n > 20$), 250 scouts are created according to (11), and the fittest of these scouts is used to set the initial position and initial velocity for the particle in the new wave. Variations of this version involve using only the initial position or the initial velocity from (11) or (12) and matching that with the initial position or initial velocity from WoSP. Among these variations, the best results occur with specifically selected initial positions and initial velocities. The selection of initial positions also helps, but the selection of initial velocities alone does not help. (In locust swarms, the "velocities only" version also tended to be less effective on the multi-modal functions BBOB fn 15-19 – see Table 4.)

8. DISCUSSION

In general, it is well known that initial positions can affect the performance of heuristic search techniques. Since many techniques display a central bias (i.e. they can more easily find a globally optimal solution that is near the centroid of their initial positions), it has become standard for benchmark problem sets to shift the location of the global optimum away from the center of the search space (e.g. [7][15][16]). Further, to help demonstrate that the performance of PSO is independent from its initial positions, the experiments with standard PSO explicitly ensured that "All swarms were randomly initialized in an area equal to one quarter of the feasible search space in every dimension that was guaranteed not to contain the optimal solution." [1] However, if the goal is to achieve the best overall performance, there is no reason to place such negative restrictions on the initial positions of a swarm.

In multi-swarm systems, the obvious initial positions for subsequent swarms are (around) the convergence point of the previous swarm(s) (e.g. [3][9][11]). In many search spaces, information from previously found local optima can help search techniques (e.g. [13]) find better optima. Although this basic information from previous local optima is used by other multi-swarm systems (e.g. WoSP [9] and DMS-PSO [11]), the results in Section 4 with locust swarms [3] show that these initial positions can be improved further by adding a simple search procedure. This result has been successfully transferred to WoSP as shown in Section 7.

For initial velocities, it is typical to use random values (e.g. [1][9][11]). However, recent work with bounds handling has shown that search space biases and negative performance effects can result with bounds handling techniques that use random velocities [8]. Since bound handling techniques which change a particle's position and/or its velocity are similar to a re-initialization, these results offer some insights into how multi-optima particle swarms might also behave. In particular, if there can be a negative effect from randomly selected initial velocities, it is reasonable to assume that there could be a positive effect from actively selected initial velocities. The results in Section 4 with locust swarms demonstrate this positive effect is indeed possible. The results in Section 7 with WoSP show that these positive effects can be achieved in other multi-swarm systems.

One of the best multi-swarm systems is DMS-PSO [11] which achieved superior results [18] in the recent Large Scale Global Optimization contest [15]. DMS-PSO does not explicitly select initial velocities for each of its sub-swarms. Instead, like WoSP, the initial velocities for the new sub-swarms are inherited relics from the previous swarm(s). Although the idea of adding a selection mechanism for initial velocities to DMS-PSO is appealing, the implementation difficulties encountered with WoSP give pause for concern. Even though the "inherited" initial values were effectively random in performance (see Table 7), their values were still critical to the overall performance of WoSP. In particular, insufficiently large magnitudes for initial velocities would lead to rapid convergence and an explosion in the number of waves.

Although the results in Section 4 show that performance gains can be achieved through improved selection of initial positions and initial velocities, it can be difficult to add these features to existing multi-swarm systems. Therefore, these design suggestions are probably best directed towards future developers of multi-swarm systems. Further, for multi-optima particle swarms that use a fixed number of sub-swarms (e.g. [3][11]), the establishment of guidelines for sub-swarm parameters would also be useful – e.g. the recommended constriction factor for χ in standard PSO [1] is likely too large for sub-swarms which must converge in a much smaller number of iterations.

9. SUMMARY

The development of multi-optima particle swarms involves three additional design decisions: the number of sub-swarms to use and/or the stopping criteria for each sub-swarm, the selection of initial positions for the particles in each of the new sub-swarms, and the selection of initial velocities for all of these particles. Since most particle swarm implementations use the default of random initial velocities, the last design criterion is particularly overlooked. Experimental results isolate and demonstrate how the specific selection of initial positions and initial velocities can both improve the performance of multi-optima particle swarms. The presented strategies to select initial velocities are quite simple, so the current results are best viewed as the identification of a promising area for further research.

10. ACKNOWLEDGMENTS

The authors thank Tim Hendtlass for access to the WoSP source code.

11. REFERENCES

[1] Bratton, D., and Kennedy, J. Defining a Standard for Particle Swarm Optimization. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*. IEEE Press. Piscataway, NJ, 2007, 120-127.

[2] Chen, S. An Analysis of Locust Swarms on Large Scale Global Optimization Problems. In K. Korb, M. Randall, and T. Hendtlass, (Eds.) *Lecture Notes in Computer Science, Vol. 5865. Proceedings of Fourth Australian Conference on*

Artificial Life. Springer-Verlag, Berlin/Heidelberg, 2009, 211–220.

[3] Chen, S. Locust Swarms – A New Multi-Optima Search Technique. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 2009, 1745–1752.

[4] Chen, S., Miura, K., and Razzaqi, S. Analyzing the Role of "Smart" Start Points in Coarse Search-Greedy Search. In M. Randall, H. Abbass, & J. Wiles, (Eds.) *Lecture Notes in Computer Science, Vol. 4828. Proceedings of Third Australian Conference on Artificial Life*. Springer-Verlag, Berlin/Heidelberg, 2007, 13–24.

[5] Chen, S., and Noa Vargas, Y. Improving the Performance of Particle Swarms through Dimension Reductions – A Case Study with Locust Swarms. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 2010, 2950–2957.

[6] El-Abd, M., and Kamel, M. S. Black-Box Optimization Benchmarking for Noiseless Function Testbed using Particle Swarm Optimization. In *Proceedings of the 2009 Genetic and Evolutionary Computation Conference*. ACM, New York, NY, 2009, 2269–2273.

[7] Hansen, N., Finck, S., Ros, R., and Auger, A. *Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions*. INRIA Technical Report RR-6829, 2009.

[8] Helwig, S., Branke, J., and Mostaghim, S. *Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization.* Technical report, Institute AIFB, Karlsruhe Institute of Technology, Number 3010, 2010.

[9] Hendtlass, T. WoSP: A Multi-Optima Particle Swarm Algorithm. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 2005, 727–734.

[10] Kennedy, J., and Eberhart, R. Particle Swarm Optimization. In *Proceedings of the IEEE International Conference on Neural Networks*. IEEE Press, Piscataway, NJ, 1995, 1942-1948.

[11] Liang, J. J., and Suganthan, P. N. Dynamic Multi-Swarm Particle Swarm Optimizer. In *Proceedings of the 2005 IEEE Swarm Intelligence Symposium*. IEEE Press, Piscataway, NJ, 2005, 124–129.

[12] Montgomery, J. Differential Evolution: Difference Vectors and Movement in Solution Space. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 2009, 2833–2840.

[13] Norman, M. G., and Moscato, P. A Competitive and Cooperative Approach to Complex Combinatorial Search. In *Proceedings of the 20th Informatics and Operations Research Meeting*. 1991, 3.15–3.29.

[14] Storn, R., and Price, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11, 341–359.

[15] Tang, K., Li, X., Suganthan, P. N., Yang, Z., and Weise, T. *Benchmark Functions for the CEC'2010 Special Session and Competition on Large Scale Global Optimization*. Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, 2009.

[16] Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y. P., Chen, C. M., and Yang, Z. *Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization*. Technical Report, 2007.

[17] Zhao, S. Z., Liang, J. J., Suganthan, P. N., and Tasgetiren, M. F. Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 2008, 3845–3852.

[18] Zhao, S. Z., Suganthan, P. N., and Das, D. Dynamic Multi-Swarm Particle Swarm Optimizer with Sub-regional Harmony Search. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation*. IEEE Press, Piscataway, NJ, 2010, 1983–1990.