Particle Swarm Optimization with \textit{pbest} Crossover

Stephen Chen  
School of Information Technology  
York University  
Toronto, Canada  
sychen@yorku.ca

\textbf{Abstract}—Particle swarm optimization can be viewed as a system with two populations: a population of current positions and a population of personal best attractors. In genetic algorithms, crossover is applied after selection – the goal is to create a new offspring solution using components from the best available solutions. In a particle swarm, the best available solutions are in the population of personal best attractors. Compared to standard particle swarm optimization, a modified version which periodically creates particle positions by crossing the personal best positions can achieve large improvements. These improvements are most consistent on multi-modal search spaces where the new crossover solutions may help the search process escape from local optima.

\textbf{Keywords}—particle swarm optimization; crossover; multi-modal search spaces; exploration; exploitation

I. \textbf{INTRODUCTION}

The development of particle swarm optimization (PSO) includes inspirations from “bird flocking, fish schooling, and swarming theory in particular” [1]. Each particle (e.g. a simulated bird) is attracted to its personal best position and the best position of a neighbouring member of the swarm. Rather than a simple line search between a current position and a best position, the velocity and momentum of each particle encourage a more explorative search path. Overall, particle swarm optimization is a simple and elegant nature-inspired metaphor for heuristic search that has achieved good performance across a broad range of optimization problems [2].

The development of genetic algorithms is inspired by Darwinian evolution and genetic crossover [3]. New solutions are created from components taken from the best solutions in the population. Since only the best solutions are allowed to be part of this “mating pool”, this selection process mimics survival of the fittest, and the resulting process allows better solutions to “evolve” over time. Overall, the genetic algorithm is also a simple and elegant nature-inspired metaphor for heuristic search that has achieved good performance across a broad range of optimization problems [4].

Several attempts have been made to add the advantages of genetic algorithms to the process of particle swarm optimization (e.g. [5][6]). Although these attempts have been mostly successful in terms of improving the performance of PSO, the increased complexity of these systems has limited their popularity. In general, the performance of genetic algorithms is greatly affected by many more parameters and design decisions (e.g. crossover rate, mutation rate, crossover operator, selection mechanism, population size, etc) than particle swarm optimization which has a recommended standard configuration that has been shown to be robust across a broad range of fitness landscapes [7].

The following proposal to leverage the benefits of genetic algorithms in particle swarm optimization is relatively simple – it adds one new parameter and recommends small modifications to two existing PSO parameters. The new proposal also differs significantly from previous attempts (e.g. [5][6]) in that it crosses the personal best positions (\textit{pbest}) instead of the current positions (\textit{x}) of each particle. This change eliminates the need for additional selection mechanisms as the \textit{pbest} positions already represent the best solutions available for mating.

The addition of \textit{pbest} crossover is tested across a broad range of benchmark functions. In general, heuristic search techniques are only needed for multi-modal search spaces since gradient descent tends to be sufficient for unimodal search spaces. Therefore, the primary functions used are set 4 (multi-modal functions with adequate global structure) and set 5 (multi-modal functions with weak global structure) from the Black-Box Optimization Benchmarking (BBOB) functions [8]. The goals of these experiments are to show that \textit{pbest} crossover can be more effective than crossing the current position \textit{x}, and to show how this simple addition can lead to large improvements over standard PSO [7].

This presentation of the potential benefits of \textit{pbest} crossover starts in Section II with a review of key ideas from genetic algorithms and particle swarm optimization. The addition of \textit{pbest} crossover requires a new parameter for frequency (Section III), and it benefits from adjustments to the constriction factor (Section IV) and population size (Section V). Several features of \textit{pbest} crossover help scale the PSO search process to higher dimension problems, and this is shown in Section VI. Despite the relative simplicity of the new modifications, they are shown in Section VII to be highly competitive with other more complex PSO-based systems. The paper then ends with a discussion of the key results in Section VIII and their summary in Section IX.

II. \textbf{BACKGROUND}

Several comparisons between particle swarm optimization and evolutionary computation (EC) approaches have been previously conducted (e.g. [9]). In EC, new solutions are
formed using crossover and mutation. In crossover, information from existing solutions is used to guide the search process, and this is similar to the attraction vectors in PSO which can encourage particles to explore “a region that represents the geometric mean between two promising regions” [9]. The exact points explored in a region can be influenced by mutation in EC which is generally omnidirectional, while the generation of new candidate solutions in PSO is affected by a particle’s velocity which is directional.

A key feature in the development of \( p_{best} \) crossover is the concept of elitism. Although it is claimed that “PSO is the only evolutionary algorithm that does not incorporate survival of the fittest” [9], personal best information serves a highly analogous role. In particular, early GAs had generational replacement schemes [3][10]. Similar to natural systems, all members of a population are eventually replaced – even if it is the best solution. Most EC techniques (e.g. differential evolution [11]) are now implemented with elitism – the best solution always survives, and it is always active in guiding the search process.

In a natural swarm, it is plausible that a bird might know its personal best position. However, it is more likely to be affected by a neighbour’s current position and velocity than its neighbour’s personal best position. This attraction to a neighbour’s personal best position has similar effects to elitism in EC – information about personal best positions is always maintained in the swarm memory, and this information is always active in guiding the search process.

Attraction to personal best positions encourages exploitation and the particle’s momentum encourages exploration. As a particle explores away from its personal best attractor, this attraction “force” eventually draws the particle back to the area around the personal best. However, similar to a comet which spends a large amount of time very far away from its attracting star, a particle’s search trajectory can spend large amounts of time very far from its personal best position. This search process can be made more exploitative by periodically resetting the particle back to its personal best position [12]. Starting from “particle swarm optimization with resets”, it becomes very easy to add \( p_{best} \) crossover.

### III. Periodic Crossover

Crossover is a widely used mechanism in heuristic search (e.g. [3][11][13]). However, before the application of crossover, selection is used – crossover creates a new solution in a region of the search space defined by two promising solutions. The storage of \( p_{best} \) positions is effectively selection, so crossover can be applied to these \( p_{best} \) positions at the same time that a PSO with resets [12] returns each particle back to its personal best position. Compared to standard PSO, the resulting system is both more exploitative (because of the resets) and more explorative (because of the crossover).

The following experiments build from the published source code of a PSO benchmark implemented by El-Abd and Kamel [14]. This benchmark implementation is for a gbest swarm using a star topology, so it requires a slight modification to become an lbest swarm with a ring topology. After this modification, the benchmark becomes an implementation of standard PSO [7] in which each dimension \( d \) of a particle’s velocity \( v \) is updated for the next iteration \( i+1 \) by

\[
v_{i+1,d} = \chi(v_{i,d} + c_1 \varepsilon_1(p_{best,d} - x_{i,d}) + c_2 \varepsilon_2(l_{best,d} - x_{i,d}))
\]

where \( \chi \) is the constriction factor, \( c_1 \) and \( c_2 \) are weights which vary the contributions of personal best and local best attractors, \( \varepsilon_1 \) and \( \varepsilon_2 \) are independent uniform random numbers in the range \([0,1]\), \( x \) is the position of the particle, \( p_{best} \) is the best position found by the current particle, and \( l_{best} \) is the best position found by any particle communicating with the current particle (i.e. two neighbours in an LBest ring topology). Key parameters in [14] include \( \chi = 0.792 \), \( \chi * c_1 = \chi * c_2 = 1.4944 \), i.e. \( c_1 = c_2 \approx 1.887 \), and \( p = 40 \) particles.

To add periodic crossover, the standard position and velocity updates (1) are replaced instead by \( p_{best} \) crossover every \( crossiter \) iterations. (See Algorithm 1) This crossover step sets the position of every particle to the midpoint of its \( p_{best} \) and a random \( p_{best} \) drawn without replacement from the population of \( p_{best} \) attractors. In this crossover scheme, it is possible for a solution to be mated with itself which causes the particle to be reset to its \( p_{best} \) position. Regardless, only the current position \( x \) of each particle is affected – \( p_{best} \) and \( v \) are not changed during this time step (unless the new position \( x \) is better than its \( p_{best} \) position).

The frequency of crossover as represented by \( crossiter \) is the only new parameter introduced in the proposed implementation of \( p_{best} \) crossover. A large value for this parameter will naturally reduce the effects of the new modification. Small values for this parameter will increase both the use of crossover and the effects of resets. In the limit of \( crossiter = 1 \), the system will behave like a simple and extremely greedy genetic algorithm. In between the two limits of a simple GA and standard PSO, the \( crossiter \) parameter helps to define the balance between swarm-based search trajectories and a more focused search among the \( p_{best} \) positions.

The need for balance between exploration and exploitation is most evident in multi-modal search spaces. In unimodal search spaces, highly exploitative techniques like hill climbing can be very effective at finding the (single) global optimum. For multi-modal search spaces, some exploration is required. However, a purely explorative technique (e.g. random search in

---

**Algorithm 1** Adding \( p_{best} \) crossover to PSO

```plaintext
for iter = 1 : maxiterations
    if mod(iter,crossiter) == 0
        parents = randperm(popsize);
        for i = 1 : popsize
            x(i,:) = (pbest(i,:)+pbest(parents(i,:)))/2;
        end
    else
        w = w*w + c1*rand(popsize,DIM).*pbest + c2*rand(popsize,DIM).*lbest;
        % remainder of velocity update from [14]
```
the limit) will not be able to conduct a finer search around the most promising regions. An effective search technique for multi-modal search spaces therefore requires explorative capabilities to find the most promising regions of the search space and exploitative capabilities to find the best solutions (e.g. the local optima) in these regions.

The following analysis of particle swarm optimization with pbest crossover focuses on two sets from the Black-Box Optimization Benchmarking (BBOB) functions [8]: set 4, multi-modal functions with adequate global structure, and set 5, multi-modal functions with weak global structure. See Table I for more information on the BBOB functions. To be consistent with previous results (e.g. [15][16][17]), the following experiments perform 25 independent trials on each function (5 trials on each of the first 5 instances) with a fixed limit of 5000*D function evaluations. Initial experiments start with D = 20 dimensions.

The first set of experiments measures the effect of different values for the crossiter parameter. The results for the previously described benchmark implementation of standard PSO were first reported in [16]. Compared to these results, Table II reports the difference (%-diff = (a-b)/a) between the mean performance for standard PSO across 25 independent trials for each function (a) and the mean performance for PSO with pbest crossover (b). A positive %-diff indicates that the addition of crossover has led to an improvement.

These improvements are then averaged over the 10 functions (mean) to pick a single value of crossiter = 100 which produces the best overall performance. Naturally, there are some functions which can benefit from a more explorative (higher) value for crossiter (e.g. BBOB fn 19 and 20) and a more exploitative (lower) value for crossiter (e.g. BBOB fn 17 and 18). The improvement achieved with periodic crossover of the current positions (replace pbest with x in Algorithm 1) is shown for reference.

The addition of pbest crossover leads to meaningful improvements compared to standard PSO on the tested multi-modal functions, but they are not that much larger than the improvements that can be achieved by crossing the current position x. However, crossover of the current positions x seems to work best with a larger value of crossiter = 200 – a value which may allow the current positions to converge and become more similar to their pbest positions at the time of crossover. This perceived need for greater convergence in x-based crossover may limit the advantages of the following parameter adjustments – first, an increase to the rate of convergence (see Section IV) and then the leveraging of this increased convergence through the use of a larger swarm size (see Section V) – which are able to improve the performance of PSO with pbest crossover.

### IV. ADJUSTING THE CONSTRICTION FACTOR

There are several functions (e.g. BBOB fn 15) in which performance improves as the value of crossiter decreases, and then the performance degrades considerably with values of crossiter that become too small. One possible explanation of this effect is that the reduced number of iterations between resets (and crossover) interferes with the ability of the swarm to fully converge to a local optimum. An alternative possibility is that the dislocations to particle positions caused by crossover can lead to larger attraction vectors (and thus velocities) which will again reduce the swarm’s overall rate of convergence. For both reasons, the following set of experiments was conducted to determine if PSO with pbest crossover should use a smaller constriction factor $\chi$. 

### TABLE I

<table>
<thead>
<tr>
<th>Set</th>
<th>fn</th>
<th>Function Name</th>
<th>Attribute</th>
<th>s</th>
<th>u</th>
<th>gs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Sphere</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>Ellipsoidal, original</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>Rastrigin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>Büche-Rastrigin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>Linear Slope</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>Attractive Sector</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>Step Ellipsoidal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>Rosenbrock, original</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>Rosenbrock, rotated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>Ellipsoidal, rotated</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>Discus</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>Bent Cigar</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>Sharp Ridge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>Different Powers</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>Rastrigin, rotated</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>Weierstrass</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>17</td>
<td>Schaffers F7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td>Schaffers F7, moderately ill-conditioned</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>Composite Griewank-Rosenbrock F8 F2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>Schwefel</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>Gallagher’s Gaussian 101-me Peaks</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>Gallagher’s Gaussian 21-hi Peaks</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>Katsuura</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>Lunacek bi-Rastrigin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Names and selected attributes of the 24 functions in the BBOB problem set – separable (s), unimodal (u), global structure (gs).

### TABLE II

<table>
<thead>
<tr>
<th>BBOB fn</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>24.3%</td>
<td>45.9%</td>
<td>26.8%</td>
<td>21.2%</td>
<td>14.2%</td>
</tr>
<tr>
<td>16</td>
<td>-10.7%</td>
<td>16.6%</td>
<td>15.1%</td>
<td>1.2%</td>
<td>5.4%</td>
</tr>
<tr>
<td>17</td>
<td>87.3%</td>
<td>75.2%</td>
<td>56.0%</td>
<td>35.7%</td>
<td>5.9%</td>
</tr>
<tr>
<td>18</td>
<td>69.7%</td>
<td>52.3%</td>
<td>40.3%</td>
<td>13.5%</td>
<td>-1.4%</td>
</tr>
<tr>
<td>19</td>
<td>-10.8%</td>
<td>9.6%</td>
<td>28.8%</td>
<td>24.0%</td>
<td>9.1%</td>
</tr>
</tbody>
</table>

| 20      | -27.7% | -0.1% | 6.6% | 8.9% | -0.5% |
| 21      | -19.7% | 19.0% | -6.5% | 7.8% | 16.9% |
| 22      | -0.1% | -7.7% | -13.9% | 17.9% | -45.4% |
| 23      | -36.0% | 13.5% | 43.8% | 32.7% | 20.4% |
| 24      | 8.3% | 31.6% | 29.7% | 15.9% | 11.4% |

| mean     | 8.4% | 25.6% | 22.7% | 17.9% | 3.6% |

$\chi$-based crossover (top) works best with crossiter = 100 and x-based crossover (bottom) works best with crossiter = 200, but the performance difference between these two options for crossover is quite small.
multi-modal functions, the best overall results occur when the effect is related to the shape (e.g. narrowness) of the local (e.g. BBOB fn 19, 23, and 24) have noticeable benefits from values (i.e. 0.95, 0.90, 0.85, 0.80, and 0.75). Several functions χ constriction factor crossover every = 100 iterations and a reduction to the crossiter performance of PSO with this system to require more constriction to properly converge. that the dislocations to the current positions would also cause trend which is a minor divergence from the crossover x matches expectations for the base/benchmark PSO results. The capacity of the swarm, the observed decrease in performance smaller constriction factor) will decrease the explorative the benchmark PSO (base). Since more constriction (i.e. a factor were also performed for crossiter -based crossover results (with = 200) also follow this TABLE III EFFECTS OF REDUCED CONSTRICTION FACTORS fn constriction factor reduction 0.95 0.90 0.85 0.80 0.75 15 53.3% 41.9% 40.7% 37.0% 32.6% 16 21.7% 18.8% -3.2% -14.1% -22.3% 17 60.9% 48.7% 29.0% -3.6% -44.5% 18 45.2% 45.8% 19.1% -0.6% -14.6% 19 28.7% 43.9% 48.0% 44.6% 49.3% 20 -5.2% 1.4% -13.9% -13.0% -18.8% 21 16.5% -21.4% 20.9% 34.6% 38.9% 22 -1.1% 7.2% -4.1% -12.2% 2.7% 23 41.6% 49.8% 46.4% 43.4% 44.6% 24 45.7% 52.6% 52.4% 58.5% 51.8% mean 30.7% 28.9% 23.5% 17.5% 12.0% 15 41.2% 36.4% 35.5% 25.6% 16.7% 16 10.6% -5.7% -14.8% -26.1% -45.7% 17 37.1% 9.8% -4.3% -29.4% -48.9% 18 24.2% -2.3% -14.6% -18.9% -18.4% 19 39.1% 34.9% 38.4% 41.1% 41.6% 20 -0.8% -7.8% -17.4% -15.0% -20.8% 21 -23.2% -11.9% 3.8% 31.9% 13.5% 22 3.9% 4.8% -34.4% -31.0% -2.6% 23 38.2% 41.0% 39.6% 36.1% 35.8% 24 37.7% 45.3% 48.2% 49.6% 47.8% x mean 20.8% 14.4% 8.0% 6.0% 1.9% base -3.4% -9.6% -4.7% -10.8% -16.5% Only pbest crossover (top) with crossover = 100 receives a benefit from a constriction factor reduction (of 0.95). The results in Table III again report the % -diff between the performance of PSO with pbest crossover and standard PSO. The total modification between these two variations is pbest crossover every crossover = 100 iterations and a reduction to the constriction factor χ produced by multiplying it with the listed values (i.e. 0.95, 0.90, 0.85, 0.80, and 0.75). Several functions (e.g. BBOB fn 19, 23, and 24) have noticeable benefits from smaller constriction factors, and it is hypothesized that this effect is related to the shape (e.g. narrowness) of the local optima wells in the fitness landscape. However, across all 10 multi-modal functions, the best overall results occur when the constriction factor is reduced by five percent (i.e. χ = 0.792 * 0.95 = 0.752). Similar experiments on the effects of a reduced constriction factor were also performed for x-based crossover (x mean) and the benchmark PSO (base). Since more constriction (i.e. a smaller constriction factor) will decrease the explorative capacity of the swarm, the observed decrease in performance matches expectations for the base/benchmark PSO results. The x-based crossover results (with crossover = 200) also follow this trend which is a minor divergence from the pbest crossover results (i.e. no benefit at any reduced levels). It was expected that the dislocations to the current positions would also cause this system to require more constriction to properly converge. V. FINDING THE BEST POPULATION SIZE The effect of a ring topology is that communication from one particle to another particle (e.g. on the other side of the ring) can be greatly delayed. This delayed communication slows the rate of convergence. The effect of crossover between random pairs of particles is to increase the rate of communication and thus the rate of convergence. With a higher rate of convergence, a swarm with more particles should be able to converge in the allotted number of function evaluations, and more particles should also promote a more diverse exploration of a multi-modal search space. The results in Table IV have the same format as Tables II and III except that the varying parameter is the population size p and the base parameters for pbest crossover are crossover = 100 and χ = 0.752 (i.e. 0.95 times the benchmark value). Crossover of the current positions x uses crossover = 200, and the constriction factor of χ = 0.792 is used for both x-based crossover (x mean) and the benchmark PSO (base). The tested population sizes are p = 30, 50, 60, 80, and 100. For all three systems, p = 30 leads to worse results and p = 50, 60, 80, and 100 lead to better results. As expected, it is a general benefit to have more particles when exploring a multi-modal search space. However, the best population size is different for all three systems. Since more particles mean fewer iterations (i.e. less time to converge), a system with features that increase swarm convergence (e.g. more constriction, resets, and more communication among particles as facilitated by crossover) should be able to use more particles effectively. For its best set of parameters, pbest crossover is 40% better on average than the original configuration of standard PSO and 10% better on average than the best implementation of crossover applied to the current positions x. VI. HIGH DIMENSION SEARCH SPACES The complexity of an optimization problem (e.g. the number of local optima) can increase exponentially with dimensionality, so search techniques that are very effective on low-dimension problems may not be efficient enough to handle problems in higher dimensions. One reason why dimensionality may negatively affect the performance of
The benefits of crossover tend to improve with increasing dimensions.\footnote{The proposed explanation for the performance improvements from which to escape, the ability to restimulate particle motions both of the original solutions, and there are no local optima solutions on a unimodal function is unlikely to be better than \( \frac{1}{10^6} \) except that the function evaluations grow with \( \chi = 0.792 \).}

Comparing the results for \( D = 20 \) and \( D = 200 \), the performance improvements with crossover are more or less consistent on the multi-modal functions (e.g. BBOB fn 15-24). However, there are exceptionally large changes in the performance as dimensionality increases on the unimodal functions (e.g. BBOB fn 10-14). Since the mid-point of two solutions on a unimodal function is unlikely to be better than both of the original solutions, and there are no local optima from which to escape, the ability to restimulate particle motions is the proposed explanation for the performance improvements of crossover on unimodal functions in higher dimensions.

### VII. Comparisons with Multi-Swarm Systems

Multi-swarm systems (e.g. Waves of Swarm Particles (WoSP) \cite{19}, locust swarms \cite{20}, and Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO) \cite{21}) attempt to improve the balance between exploration and exploitation by having the (smaller) individual swarms focus more on exploitation (e.g. converging quickly to a local optimum) and using separate search/coordination techniques which focus more on exploration. These techniques introduce many extra parameters and design considerations (e.g. \cite{16}\cite{22}\cite{23}), but it appears that the much simpler modifications involved with \( p\text{best} \) crossover can be highly competitive with them.

The performance of locust swarms is generally better than WoSP due to its specific selection of initial positions and initial velocities \cite{16}, and locust swarms appear to perform better than DMS-PSO before hybridizations and local search are added \cite{22}. In Table VI, the results for PSO with \( p\text{best} \) crossover and locust swarms (from \cite{16}) are compared on the full BBOB problem set for \( D = 20 \) and \( D = 200 \) dimensions.

Due to several exaggerated percentage changes (i.e. BBOB fn 1 and 2), the mean performance becomes meaningless, so it is not reported. Nonetheless, it can still be seen that the overall performance with \( p\text{best} \) crossover is better than locust swarms. The primary advantages of multi-swarm systems are the ability to explore multiple optima in multi-modal search spaces and the ability to restimulate stalled search components in unimodal search spaces \cite{15}. It appears that crossover can similarly provide both of these advantages in a much simpler overall system.

### VIII. Discussion

The recommended implementation of PSO with \( p\text{best} \) crossover is also much simpler than previous methods of adding crossover to PSO (e.g. \cite{5}\cite{6}). For example, mid-point crossover is simpler than linear-range crossover, and the current recommendation leaves the velocities unchanged (like a PSO with resets \cite{12}) as opposed to also crossing them. Unreported experiments with these variations showed...
negligible changes in performance, so these modifications are omitted from the final system for the sake of simplicity.

One variation which did show an improvement was the use of directed initial velocities [16]. Since the pbest position can converge near a local optimum very quickly, the process of resets and pbest crossover is similar to starting a new sub-swarm in a multi-swarm system. It has been shown in multi-swarm systems that launching the particle away from the previous position (on a line through the new initial position) can lead to improved performance [16][20], so this was tried with pbest crossover. However, since the resulting improvement was relatively small, this modification was also omitted from the final system for the sake of simplicity.

Furthering the comparison with multi-swarm systems, PSO with pbest crossover and DMS-PSO [21] have some interesting similarities. In DMS-PSO, the swarm is divided into parallel sub-swarms which periodically regroup. This regrouping encourages exploration among the (local) optima discovered by the previous sub-swarms. The crossing of pbest positions also encourages exploration among the (local) optima discovered by each particle. The advantages of pbest crossover are that the periodic sampling can be faster (since pbest positions converge faster than the actual particles [24]) and the population of pbest positions can have more diversity than a population divided into n sub-populations that subsequently converge to n < p pbest positions.

Convergence occurs in PSO when a particle with zero speed has the same position as all of its pbest attractors. The current use of pbest crossover reduces convergence by moving the position of a particle away from its pbest attractors. Other methods to reduce convergence include disallowing zero velocities [18], moving the pbest attractors [24][25], or preventing the co-location of the pbest attractors [17][26]. A full comparison of these alternate techniques is an area for further research.

IX. SUMMARY

The addition of crossover to standard PSO can be simple and highly effective. The presented model randomly crosses all of the particles in the swarm every crossover iterations. Performance improvements can be achieved by crossing either current positions x or the pbest positions. The best overall results occur with pbest crossover as this system benefits from being both more exploitive (e.g. more constriction and resets) and more explorative (e.g. more particles).

REFERENCES