Chapter 2
Biases in Particle Swarm Optimization

William M. Spears
Swarmotics LLC, USA

Derek T. Green
University of Arizona, USA

Diana F. Spears
Swarmotics LLC, USA

ABSTRACT
The most common versions of particle swarm optimization (PSO) algorithms are rotationally variant. It has also been pointed out that PSO algorithms can concentrate particles along paths parallel to the coordinate axes. In this paper, the authors explicitly connect these two observations by showing that the rotational variance is related to the concentration along lines parallel to the coordinate axes. Based on this explicit connection, the authors create fitness functions that are easy or hard for PSO to solve, depending on the rotation of the function.

INTRODUCTION
The popularity and variety of Particle Swarm Optimization (PSO) algorithms have continued to grow at a rapid rate since the initial PSO algorithm was introduced in 1995 (Eberhart et al., 2001; Shi & Eberhart, 2008; Shi & Eberhart, 2009). Recently, great strides have been made in understanding the theoretical underpinnings of the basic PSO algorithm (Poli, 2009; Trelea, 2003; van den Bergh & Engelbrecht, 2006). However, there are still some behaviors exhibited by PSO that require further examination. For example, although it has also been pointed out that when running the traditional PSO algorithm, “most movement steps occurred parallel to one of the coordinate axes” (Janson & Middendorf, 2007), this behavior is not well explained theoretically. In this paper, we examine this behavior and provide a theoretical explanation for why it occurs. Based on this explanation, we also show fitness landscapes in which the performance of PSO depends heavily on the rotation of the fitness
function. Through these observations we hope to help users of PSO-based algorithms to better understand the effects of the biases inherent in PSO on their own particular problems. In this paper, a “bias” is an implicit algorithmic preference; for a general discussion of biases, see Mitchell (1997).

The PSO Algorithm

The basic PSO algorithm (Kennedy & Eberhart, 1995) is usually described as follows. A swarm consists of \(N\) particles. Each particle \(i\) has a position at time \(t\) denoted by \(\overrightarrow{X}_i(t) = (X_{i,1}(t), \ldots, X_{i,D}(t))\) where \(\overrightarrow{X}_i(t)\) is a \(D\) -dimensional vector. Each particle \(i\) has a velocity \(\overrightarrow{V}_i(t) = (V_{i,1}(t), \ldots, V_{i,D}(t))\), which is also a \(D\) -dimensional vector. The equations of motion are generally given as:

\[
\overrightarrow{X}_i(t + 1) = \overrightarrow{X}_i(t) + \overrightarrow{V}_i(t + 1)
\]

\[
\overrightarrow{V}_i(t + 1) = \omega \overrightarrow{V}_i(t) + c_1 \overrightarrow{r}_1 (\overrightarrow{P}_i - \overrightarrow{X}_i(t)) + c_2 \overrightarrow{r}_2 (\overrightarrow{G} - \overrightarrow{X}_i(t))
\]

\(\overrightarrow{P}_i\) is the “personal best” position, or the position of the best fitness ever encountered by particle \(i\). \(\overrightarrow{G}\) is the “global best” position ever found by all of the particles or, alternatively, the best position ever seen within a neighborhood of particles. In this paper, we will assume that all particles are neighbors. The best positions are updated when particles find positions with better fitness. The \(\omega\) term, an “inertial coefficient” from 0 to 1, was introduced in (Shi & Eberhart, 1998). The “learning rates” \(c_1\) and \(c_2\) are non-negative constants. Very often these are both set to 2.0. Finally, \(r_1\) and \(r_2\) are random numbers generated in the range \([0,1]\).

Looking again at the above equations, we point out an ambiguity that unfortunately continues to propagate throughout the literature. The ambiguity arises in the interpretation of the random numbers. In many papers, it is not made clear when the random numbers are calculated. The random variables \(r_1\) and \(r_2\) may be interpreted as scalars or vectors. Figure 1 shows the two most common implementations seen in the literature. In both versions, \(U(0,1)\) is a uniform random generator in the range of \([0,1]\).

Version 1 is rotationally invariant, while Version 2 is not. James Kennedy, one of the creators of PSO, indicates that the second of the two versions is preferred -- because it is considered to be more explorative (Kennedy, 2007). Hence, the notation of Poli (2009) is preferred:

\[
\overrightarrow{V}_i(t + 1) = \omega \overrightarrow{V}_i(t) + c_1 \overrightarrow{r}_1 (\overrightarrow{P}_i - \overrightarrow{X}_i(t)) + c_2 \overrightarrow{r}_2 (\overrightarrow{G} - \overrightarrow{X}_i(t))
\]

where \(\odot\) represents component-wise multiplication.

In this paper, we will show that updating the random numbers in the preferred way is the cause of the biased behavior. In other words, the rotational variance and coordinate axes bias of Version 2 are related. It is not our intention to suggest that PSO should not be used due to the bias, but rather that users of PSO should be aware of the bias, its cause and how it might affect their particular needs.

The motivation for this view of bias stems from the original No Free Lunch Theorems, where “an algorithm’s average performance is determined by how aligned it is with” the optimization problems being considered (Wolpert & Macready, 1997). The geometric interpretation is to consider both the algorithm and the problems as vectors. If they are aligned, the dot product is zero, and the algorithm is well matched to the problem. In other words, the main objective of this paper is to assist the PSO practitioner in achieving an algorithm/problem alignment. Here, the geometric interpretation of alignment is almost literal -- we show that the performance of PSO can depend to a very large degree on the rotation of fitness landscapes that contain linear features, such as troughs or ridges.