In *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature*, G. Rudolph et al. (Eds.), LNCS 5199, pages 889–898, Dortmund, Germany, September 2008.

Theoretical Analysis of Initial Particle Swarm Behavior

Sabine Helwig and Rolf Wanka

Department of Computer Science, University of Erlangen-Nuremberg, Germany
{sabine.helwig,rwanka}@informatik.uni-erlangen.de

Abstract. In this paper, particle trajectories of PSO algorithms in the first iteration are studied. We will prove that many particles leave the search space at the beginning of the optimization process when solving problems with boundary constraints in high-dimensional search spaces. Three different velocity initialization strategies will be investigated, but even initializing velocities to zero cannot prevent this particle swarm explosion. The theoretical analysis gives valuable insight into PSO in high-dimensional bounded spaces, and highlights the importance of bound handling for PSO: As many particles leave the search space in the beginning, bound handling strongly influences particle swarm behavior. Experimental investigations confirm the theoretical results.

1 Introduction

Particle Swarm Optimization (PSO) [1] is a population-based algorithm for global optimization. All population members, from now on called *particles*, explore the n-dimensional search space S of an optimization problem with objective function $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$. Without loss of generality (W.l.o.g.), we will assume minimization problems. Each particle has a *position* $\boldsymbol{x}_{i,t}$, a *velocity* $\boldsymbol{v}_{i,t}$, and a *fitness value* $f(\boldsymbol{x}_{i,t})$, where t is the iteration counter. A position $\boldsymbol{z}_1 \in S$ is called *better* than $\boldsymbol{z}_2 \in S$ iff $f(\boldsymbol{z}_1) < f(\boldsymbol{z}_2)$. The best search space position particle i has visited until iteration t is its *private guide* $\boldsymbol{p}_{i,t}$. To each particle, a subset of all particles is assigned as its neighborhood. The best private guide of all neighbors of particle i is called its *local guide* $\boldsymbol{l}_{i,t}$. In each iteration, position and velocity of each particle i are updated according to the following equations:

$$\boldsymbol{v}_{i,t} = \omega \cdot \boldsymbol{v}_{i,t-1} + c_1 \cdot \boldsymbol{r}_1 \odot (\boldsymbol{p}_{i,t-1} - \boldsymbol{x}_{i,t-1}) + c_2 \cdot \boldsymbol{r}_2 \odot (\boldsymbol{l}_{i,t-1} - \boldsymbol{x}_{i,t-1})$$
$$\boldsymbol{x}_{i,t} = \boldsymbol{x}_{i,t-1} + \boldsymbol{v}_{i,t}$$

where ω, c_1, and c_2 are prespecified parameters, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of random real numbers uniformly chosen between 0 and 1, and independently drawn every time they occur. $\odot$ denotes element-by-element vector multiplication.

In this paper, optimization problems with *boundary constraints* are studied, i.e., $S = [lb_1, ub_1] \times [lb_2, ub_2] \times \ldots \times [lb_n, ub_n]$ is bounded. W.l.o.g., we will assume $S = [-r, r]^n$. As many real-world problems and most benchmark suites have boundary constraints, a lot of strategies to handle them can be found in the literature (e.g., [2–6]). We will consider the following three strategies:

- *Infinity* allows particles to enter invalid space, does not alter positions or velocities, and skips the evaluation step for infeasible particles [6].
- *Absorb* sets invalid particles on the nearest boundary and all affected velocity components are set to zero [7].
- *Random* sets all invalid components of a particle's position vector to a random value, and the velocity is adjusted: $\boldsymbol{v}_{i,t+1} = \boldsymbol{x}_{i,t+1} - \boldsymbol{x}_{i,t}$ [4].

Most theoretical studies of PSO concentrate on the important task of selecting appropriate values for c_1, c_2, and ω [8–10]. Some widely-used standard settings were derived from these analyses, e.g., $c_1 = c_2 = 1.496172, \omega = 0.72984$ [6], or $c_1 = c_2 = 1.193, \omega = 0.721$ [11]. Often, PSO analyses assume 1-dimensional problems as each component of position and velocity vector is updated separately. However, for a deeper understanding of particle swarms in high-dimensional, bounded search spaces, the peculiarities of high-dimensional spaces have to be taken into account. Previous studies have already shown that the "curse of dimensionality", which means that high-dimensional spaces are not intuitive, is an important topic in particle swarm optimization [12]. It was proven that particles are initialized very close to at least one boundary. Moreover, it was shown that, *with overwhelming probability* (*w.o.p.*, for a definition, see Section 2), the best particle leaves the search space in the first iteration when velocities are initialized uniformly at random in $[-r, r]^n$.

In the following analyses, PSO in high-dimensional search spaces is studied in more depth. We will show that uniform velocity initialization causes not only the best, but all particles to leave the search space, w.o.p. The fact that many particles leave the search space was noted earlier [13, 7], but never proven theoretically.

In order to avoid that too many particles leave the search space at the beginning, other velocity initialization strategies were proposed:

- *Zero* [13]: Particle velocities are initialized to zero.
- *Half-diff* [11]: Let $\boldsymbol{x}_{i,0}$ be the initial position of particle i, and $\boldsymbol{y}_i$ drawn uniformly at random in S. Then, $\boldsymbol{v}_{i,0}$ is set to $\frac{1}{2}(\boldsymbol{y}_i - \boldsymbol{x}_{i,0})$.

In Section 2, we will prove that using zero or half-diff initialization also causes many particles to leave the search space, w.o.p. We will derive some consequences for PSO application afterwards. In Section 3, different velocity initialization strategies and bound handling mechanisms will be studied experimentally on known benchmark problems.

2 Theoretical Results

Particle trajectories in the first iteration will be analyzed for three different velocity initialization strategies. Two main results will be derived:

- When using uniform velocity initialization, all particles leave the search space in the first iteration, w.o.p.

– When using zero or half-diff initialization, all particles which have a better neighbor than themselves leave the search space in the first iteration, w.o.p. For realistic optimization problems and most commonly-used neighborhood topologies, this is the majority of the particles.

The following assumptions will be used: $1 < c_2 \leq 2$, $0 < \omega \leq 1$, c_2 and ω do not depend on n, and particles are initialized uniformly at random in the n-dimensional search space $[-r, r]^n \subset \mathbb{R}^n$. The particles are connected via an arbitrary neighborhood topology, including the fully connected swarm. For each particle i with $\boldsymbol{x}_{i,0} \neq \boldsymbol{l}_{i,0}$, $\boldsymbol{l}_{i,0}$ is distributed uniformly at random in S. Actually, the local guides' positions depend on the optimization problem. However, the above assumption is not too restrictive for higher-dimensional problems, which is confirmed by Examples 1, 3, and 4 (*PSO Exp.*).

Definition: An event $A(n)$ happens *with overwhelming probability (w.o.p.)* with respect to n if there exists a constant $\gamma > 0$ such that $P(A(n)) = 1 - e^{-\Omega(n^\gamma)}$, where Ω belongs to the *big-O notation* for expressing asymptotic behavior. Hence, an overwhelming probability with respect to n rapidly approaches 1 when n increases.

2.1 Uniform Velocity Initialization

Theorem 1. *If velocities are initialized with uniform distribution in the search space, all particles which are initialized such that they have at least one neighbor with better fitness value than themselves (i.e., $\boldsymbol{x}_{i,0} \neq \boldsymbol{l}_{i,0}$) leave the search space, w.o.p.*

Proof. Let particle i be an arbitrary particle satisfying the above assumptions. As $\boldsymbol{p}_{i,0} = \boldsymbol{x}_{i,0}$, its position and velocity in the first iteration evaluate to

$$\begin{aligned} \boldsymbol{v}_{i,1} &= \omega \cdot \boldsymbol{v}_{i,0} + c_2 \cdot \boldsymbol{r}_2 \odot (\boldsymbol{l}_{i,0} - \boldsymbol{x}_{i,0}) \\ \boldsymbol{x}_{i,1} &= \boldsymbol{x}_{i,0} + \boldsymbol{v}_{i,1} = \omega \cdot \boldsymbol{v}_{i,0} + (\boldsymbol{1} - c_2 \cdot \boldsymbol{r}_2) \odot \boldsymbol{x}_{i,0} + c_2 \cdot \boldsymbol{r}_2 \odot \boldsymbol{l}_{i,0} \ . \end{aligned} \tag{1}$$

Hence, for fixed $\boldsymbol{r}_2$, the d-th component of $\boldsymbol{x}_{i,1}$, $x_{i,1,d}$, is the sum of three non-identical uniformly distributed, independent random variables. Its density function $f_{x_{i,1,d}}(z)$ can be computed using the formula presented by Bradley and Gupta [14, Theorem 1]. The probability that a particle crosses the boundary in dimension d evaluates to:

$$q_1(r_{2,d}, c_2, \omega) = \int_{-\infty}^{-r} f_{x_{i,1,d}}(z)\mathrm{d}z + \int_{r}^{\infty} f_{x_{i,1,d}}(z)\mathrm{d}z =$$

$$= \begin{cases} \frac{-3\omega^2 + 6c_2 r_{2,d}\omega - 4c_2^2 r_{2,d}^2}{-12\omega(1 - c_2 r_{2,d})} (=: p_1) & \text{if } 0 \leq r_{2,d} < \frac{\omega}{2c_2} \\ \frac{\omega^2}{24(1 - c_2 r_{2,d}) c_2 r_{2,d}} (=: p_2) & \text{if } \frac{\omega}{2c_2} \leq r_{2,d} < \frac{2-\omega}{2c_2} \\ \frac{4c_2^2 r_{2,d}^2 + 6\omega c_2 r_{2,d} - 8c_2 r_{2,d} + 3\omega^2 + 4 - 6\omega}{12\omega c_2 r_{2,d}} (=: p_3) & \text{if } \frac{2-\omega}{2c_2} \leq r_{2,d} < \frac{2+\omega}{2c_2} \\ \frac{24 + \omega^2 + 24c_2^2 r_{2,d}^2 - 48c_2 r_{2,d}}{-24c_2 r_{2,d}(1 - c_2 r_{2,d})} (=: p_4) & \text{if } \frac{2+\omega}{2c_2} \leq r_{2,d} \leq 1 \end{cases} \tag{2}$$

As $r_{2,d}$ is uniformly drawn from $[0,1]$, the probability $p_A(c_2,\omega)$ that a particle violates the boundary in a specific dimension evaluates to:

$$p_A(c_2,\omega) = \int_0^1 q_1(r_{2,d},c_2,\omega)\mathrm{d}r_{2,d} =$$

$$= \begin{cases} (24\omega c_2)^{-1}\cdot(-36\omega+6\omega^2\ln(2)-12\omega^2\ln(2-\omega)+5\omega^2-36\omega\ln(2)+24\omega\ln(2-\omega)+8\ln(2) \\ \quad -16\ln(2-\omega)-3\omega^3\ln(\omega)+2\omega^3\ln(2-\omega)+8\ln(2+\omega)+12\omega\ln(2+\omega)+6\omega^2\ln(2+\omega) \\ \quad -24\omega\ln(c_2)-\omega^3\ln(c_2)+\omega^3\ln(2+\omega)+\omega^3\ln(c_2-1)+24\omega c_2) & \text{if } 2+\omega-2c_2<0 \\ (12\omega c_2)^{-1}\cdot(-10\omega+6\omega^2\ln(2)-6\omega^2\ln(2-\omega)+5\omega^2-12\omega\ln(2) \\ \quad +12\omega\ln(2-\omega)+8\ln(2)-8\ln(2-\omega)-\omega^3\ln(\omega)+\omega^3\ln(2-\omega) \\ \quad +4\ln(c_2)+6-6\omega\ln(c_2)+3\omega^2\ln(c_2)+6\omega c_2+2c_2^2-8c_2) & \text{if } 2+\omega-2c_2\geq 0 \end{cases} \quad (3)$$

Eq. (3) can be used for calculating $p_A(c_2,\omega)$ for specific values of c_2 and ω (see Example 1). In order to prove that particles leave the search space w.o.p., we must show that $p_A(c_2,\omega) > 0$. Therefore, we use the following fact:

$$p_A(c_2,\omega) = \underbrace{\int_0^{\frac{\omega}{2c_2}} p_1\mathrm{d}r_{2,d}}_{l_1(c_2,\omega)\geq 0} + \underbrace{\int_{\frac{\omega}{2c_2}}^{\frac{2-\omega}{2c_2}} p_2\mathrm{d}r_{2,d}}_{l_2(c_2,\omega)\geq 0} + \underbrace{\int_{\frac{2-\omega}{2c_2}}^{\min\{\frac{2+\omega}{2c_2},1\}} p_3\mathrm{d}r_{2,d}}_{l_3(c_2,\omega)\geq 0} + \underbrace{\int_{\min\{\frac{2+\omega}{2c_2},1\}}^{1} p_4\mathrm{d}r_{2,d}}_{l_4(c_2,\omega)\geq 0}$$

We compute $l_2(c_2,\omega) = \frac{\omega^2\cdot(\ln(2-\omega)-\ln(\omega))}{\ln(c_2)}$ and $l_1(c_2,1) = \frac{1+2\ln(2)}{24c_2} > 0$.

If $\omega < 1$, then $l_2(c_2,\omega) > 0$, otherwise $l_1(c_2,\omega) > 0$. Thus, $p_A(c_2,\omega) > 0$, and the probability that a particle leaves the n-dimensional search space is

$$p'_A(c_2,\omega,n) = 1-(1-p_A(c_2,\omega))^n = 1-e^{-\Theta(n)} \ . \quad (4)$$

□

Example 1. Two experiments were conducted for this and subsequent examples:

Conf. Exp.: In order to confirm that the mathematical analysis is correct under the given assumptions, the following experiment was conducted: $\boldsymbol{x}_{i,0}$, $\boldsymbol{v}_{i,0}$, $\boldsymbol{l}_{i,0}$, and $\boldsymbol{r}_2$ were randomly drawn according to the assumptions, and $\boldsymbol{x}_{i,1}$ was calculated according to Eq. (1). The probability for $\boldsymbol{x}_{i,1} \notin S$ was evaluated by performing 10^7 runs per considered problem dimensionality.

PSO Exp.: In order to determine the relevance of the theoretical results for PSO, the following experiment was performed: The PSO is applied on all CEC 2005 benchmarks [15] which are scalable with respect to the search space dimensionality (some problems include matrices which are only given for at most 50 dimensions): f1, f2, f5, f6, f9, f12, f13, f15. Standard settings for the PSO as presented in Section 3 were used, except for that particles are not included in their own neighborhood so that for all particles $\boldsymbol{x}_{i,0} \neq \boldsymbol{l}_{i,0}$ holds. For each benchmark, $10,000$ runs with 50 particles were performed.

The theoretical results were obtained by using Eq. (3) and Eq. (4).

	Theor. result	Conf. Exp.	PSO Exp.
$p'_A(1.496172, 0.72984, 1)$	0.17074	0.17072	0.149587
$p'_A(1.496172, 0.72984, 30)$	0.99636	0.99648	0.99582
$p'_A(1.496172, 0.72984, 100)$	0.9999999926	1	1

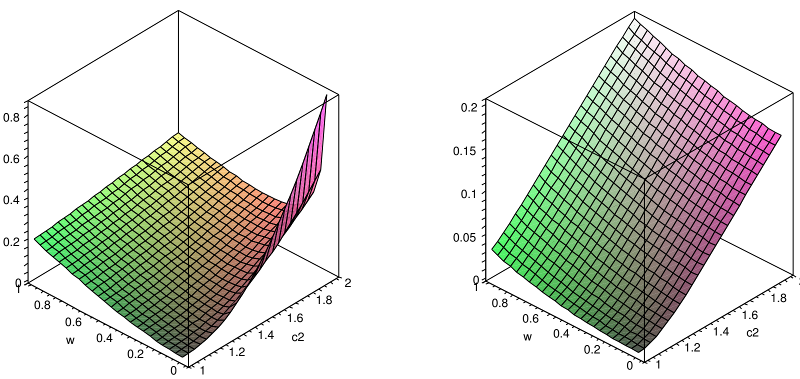

Fig. 1. The probabilities $p_A(c_2, \omega)$ (left) and $p_D(c_2, \omega)$ (right)

The comparison of the theoretical results and *Conf. Exp.* confirms the mathematical analysis. *PSO Exp.* shows that the assumption that local guides $\boldsymbol{l}_{i,0}$ are uniformly distributed at random in S is not too restrictive (see also subsequent examples) for higher-dimensional spaces. Moreover, the example demonstrates that the probability that a particle leaves the search space rapidly approaches 1 when increasing the search space dimensionality.

The probability $p_A(c_2, \omega)$ that a particle which is not its own local guide violates a specific boundary in the first iteration is shown in Fig. 1. It approaches zero for $c_2 \to 1$ and $\omega \to 0$. However, choosing such small values for c_2 and ω prevents exploration, and can therefore not be recommended.

Theorem 2. *If velocities are initialized with uniform distribution in* $[-r, r]^n$*, each particle* i *with* $\boldsymbol{x}_{i,0} = \boldsymbol{l}_{i,0}$ *leaves the* n*-dimensional search space in on average* $\frac{\omega}{4} \cdot n$ *dimensions in the first iteration.*

Proof. As $\boldsymbol{p}_{i,0} = \boldsymbol{l}_{i,0} = \boldsymbol{x}_{i,0}$, particle i's position $\boldsymbol{x}_{i,1}$ and velocity $\boldsymbol{v}_{i,1}$ in the first iteration evaluate to $\boldsymbol{v}_{i,1} = \omega \cdot \boldsymbol{v}_{i,0}$ and $\boldsymbol{x}_{i,1} = \boldsymbol{x}_{i,0} + \boldsymbol{v}_{i,1} = \boldsymbol{x}_{i,0} + \omega \cdot \boldsymbol{v}_{i,0}$. Hence, the d-th component of $\boldsymbol{x}_{i,1}$, $x_{i,1,d}$, is the sum of two independent, uniformly distributed random variables. Its density function $f_{x_{i,1,d}}$ is trapezoidal and can be determined by convolution:

$$f_{x_{i,1,d}}(z) = \begin{cases} \frac{1}{4\omega r^2} z + \frac{1}{4r} + \frac{1}{4\omega r} & \text{for } -r - \omega r < z \le \omega r - r \\ \frac{1}{2r} & \text{for } \omega r - r < z < r - \omega r \\ -\frac{1}{4\omega r^2} z + \frac{1}{4r} + \frac{1}{4\omega r} & \text{for } -\omega r + r \le z < r + \omega r \\ 0 & \text{otherwise} \end{cases}$$

Thus, the probability $p_B(\omega)$ that particle i exceeds the search space boundary in dimension d is $p_B(\omega) = \int_{-r-\omega r}^{-r} f_{x_{i,1,d}}(z) \mathrm{d}z + \int_{r}^{r+\omega r} f_{x_{i,1,d}}(z) \mathrm{d}z = \frac{\omega}{4}$. □

Corollary 1. *Each particle satisfying the assumptions of Theorem 2 leaves the n-dimensional search space in the first iteration, w.o.p.*

Proof. The probability $p'_B(\omega, n)$ that a particle which satisfies the given assumptions leaves the search space evalutes to

$$p'_B(\omega, n) = 1 - (1 - \omega/4)^n = 1 - e^{-\Theta(n)} \ .$$

□

Example 2. We evaluate $p'_B(0.72984, 30) = 0.99763$, $p'_B(0.72984, 100) = 0.99999$, which shows that $p'_B(\omega, n)$ rapidly approaches 1 when increasing n.

From Theorem 1 and Corollary 1 follows that, w.o.p., all particles leave the search space in the first iteration, when initializing velocities uniformly at random in S.

2.2 Zero and Half-Diff Velocity Initialization

We will now show that using zero or half-diff initialization cannot avoid that many particles leave the search space in the first iteration, either.

Theorem 3. *If velocities are initialized to zero, each particle i with $\boldsymbol{x}_{i,0} \neq \boldsymbol{l}_{i,0}$ leaves the search space in the first iteration, w.o.p.*

Proof. Let particle i be an arbitrary particle satisfying the above condition. Its position and velocity in the first iteration are given by

$$\begin{aligned} \boldsymbol{v}_{i,1} &= \omega \cdot \boldsymbol{v}_{i,0} + c_1 \cdot \boldsymbol{r}_1 \odot (\boldsymbol{p}_{i,0} - \boldsymbol{x}_{i,0}) + c_2 \cdot \boldsymbol{r}_2 \odot (\boldsymbol{l}_{i,0} - \boldsymbol{x}_{i,0}) = \\ &= c_2 \cdot \boldsymbol{r}_2 \odot (\boldsymbol{l}_{i,0} - \boldsymbol{x}_{i,0}) \\ \boldsymbol{x}_{i,1} &= \boldsymbol{x}_{i,0} + \boldsymbol{v}_{i,1} = (\boldsymbol{1} - c_2 \cdot \boldsymbol{r}_2) \odot \boldsymbol{x}_{i,0} + c_2 \cdot \boldsymbol{r}_2 \odot \boldsymbol{l}_{i,0} \ . \end{aligned}$$

For fixed $\boldsymbol{r}_2$, $x_{i,1,d}$ is the sum of two non-identical uniformly distributed random variables, and therefore trapezoidally distributed. If $r_{2,d} < \frac{1}{c_2}$, particle i does not violate the boundary in dimension d. Otherwise, the density function $f_{x_{i,1,d}}$ of $x_{i,1,d}$ can be computed (omitted due to space constraints), and the probability $q_2(r_{2,d}, c_2)$ that particle i crosses the search space boundary in dimension d is

$$q_2(r_{2,d}, c_2) = \begin{cases} \int_{-\infty}^{-r} f_{x_{i,1,d}}(z)\mathrm{d}z + \int_{r}^{\infty} f_{x_{i,1,d}}(z)\mathrm{d}z = 1 - \frac{1}{c_2 r_{2,d}} & \text{if } r_{2,d} > \frac{1}{c_2} \\ 0 & \text{otherwise} \end{cases}$$

As $r_{2,d}$ is uniformly distributed between 0 and 1, we finally determine the probability $p_C(c_2)$ that a particle violates the boundary in a specific dimension to

$$p_C(c_2) = \int_0^1 q_2(r_{2,d}, c_2)\mathrm{d}r_{2,d} = \frac{-1 - \ln(c_2) + c_2}{c_2} \ . \tag{5}$$

From $c_2 > 1$, $p_C > 0$ follows. Thus, the probability $p'_C(c_2)$ that particle i leaves the n-dimensional search space evaluates to

$$p'_C(c_2, n) = 1 - (1 - p_C(c_2))^n = 1 - e^{-\Theta(n)} \ . \tag{6}$$

□

Example 3. For the theoretical results, Eq. (5) and Eq. (6) were used.

	Theor. result	Conf. Exp.	PSO Exp.
$p'_C(1.496172, 1)$	0.06233	0.06223	0.04548
$p'_C(1.496172, 30)$	0.85497	0.85574	0.83358
$p'_C(1.496172, 100)$	0.988	0.998	0.99801

The example confirms the theoretically derived formulas for $p_C(c_2)$ and $p'_C(c_2, n)$, and the relevance of the theoretical results for high-dimensional PSO application.

Conjecture 1. If the particles' velocities are initialized according to the *half-diff* strategy, each particle i with $\boldsymbol{x}_{i,0} \neq \boldsymbol{l}_{i,0}$ leaves the search space, w.o.p.

Similar to the proof for Theorem 1, the probability $p_D(c_2, \omega)$ that a particle leaves the search space in dimension d can be evaluated to

$$p_D(c_2, \omega) = \int_0^1 \left(\int_{-\infty}^{-r} f_{x_{i,1,d}}(z)\mathrm{d}z + \int_r^{\infty} f_{x_{i,1,d}}(z)\mathrm{d}z \right) \mathrm{d}r_{2,d} =$$

$$= (12\omega c_2(\omega-2))^{-1} \cdot (32\omega - 22\omega^2 + 3\omega^3 - 16\ln(2) + 24\omega\ln(2) + 16\ln(2-\omega) - 24\omega\ln(2-\omega) + 24\ln(c_2)\omega - 12\omega^2\ln(2) + 12\omega^2\ln(2-\omega) - 12\ln(c_2)\omega^2 + 2\omega^3\ln(2) - 2\omega^3\ln(2-\omega) + 2\ln(c_2)\omega^3 + 2\omega^3\ln(\omega) - 24\omega c_2 + 12\omega^2 c_2 - 2\omega^3\ln(\omega - 2 + 2c_2)) \quad (7)$$

and is plotted in Fig. 1 (right). The probability that a particle leaves the n-dimensional search space in the first iteration is

$$p'_D(c_2, \omega, n) = 1 - (1 - p_D(c_2, \omega))^n \quad (8)$$

which is overwhelming if $p_D(c_2, \omega) > 0$. Fig. 1 shows that there is strong evidence that $p_D(c_2, \omega) > 0$, at least for commonly used values for c_2 and ω, e.g., $c_2 > 1.1$ and $\omega > 0.3$.

Example 4. For the theoretical results, Eq. (7) and Eq. (8) were used. Again, the theoretical results are confirmed:

	Theor. result	Conf. Exp.	PSO Exp.
$p'_D(1.496172, 0.72984, 1)$	0.094572	0.094661	0.077998
$p'_D(1.496172, 0.72984, 30)$	0.949229	0.950656	0.941712
$p'_D(1.496172, 0.72984, 100)$	0.999952	0.999954	0.999935

2.3 Consequences for PSO Application

The theoretical analysis showed that none of the three investigated velocity initialization strategies can avoid that many particles leave high-dimensional search spaces as early as in the first iteration. Even initializing velocities to zero cannot prevent particle explosion. For PSO application, there exist several strategies to deal with this observation:

In order to avoid that particles leave the search space, bound handling strategies which keep the particles inside the search space, such as *Absorb* or *Random*, can be applied. Many other strategies exist in the literature [2–5]. From the methods studied in the experimental analysis in Section 3, *Absorb* performed best. However, hybrid methods as proposed by Clerc [2] seem to be promising.

When using *Infinity* bound handling, which is a commonly used strategy [6], many particles mainly explore invalid space at the beginning of the optimization process. In our experiments (see Section 3), *Infinity* was significantly outperformed by *Random* and *Absorb* on almost all 100-dimensional problems. Velocity clamping can help particles to reenter the search space.

More general approaches for constraint handling, e.g., the use of penalty functions, can also be applied to deal with the initial particle explosion.

3 Experimental Results

The following experimental analysis studies the impact of velocity initialization and bound handling in particle swarm optimization. For all experiments, the PSO with standard parameter settings [6] (50 particles, $c_1 = c_2 = 1.496172, \omega = 0.72984$) was applied on four widely-used benchmarks, Sphere, Rosenbrock, Rastrigin, and Griewank (function descriptions and particle initialization ranges see, e.g., [6]), and on the CEC 2005 benchmarks f1, f2, f5, f6, f9, f12, f13, f15 [15]. When solving a CEC benchmark, particles were initialized uniformly at random in S. The swarm is connected via the *von Neumann* topology, a two-dimensional grid with wrap-around edges [16]. A particle is included in its own neighborhood.

The PSO terminated after $100,000$ function evaluations, and each configuration was repeated 100 times. In order to compare the performance of two algorithms A and B, the one-sided Wilcoxon rank sum test was used with null-hypothesis $H_0 : F_A(z) = F_B(z)$, and the one-sided alternative $H_1 : F_A(z) < F_B(z)$ (where $F_X(z)$ is the distribution of the results of algorithm X). Statistical significance was evaluated on a significance level of 0.01.

3.1 Velocity Initialization

In the theoretical analysis, three different velocity initialization strategies were studied. Uniform initialization causes all particles to leave the search space, w.o.p, whereas zero initialization slows down initial exploration. Hence, although none of the strategies can avoid that many particles leave high-dimensional search spaces, w.o.p., half-diff initialization seems to have fewest drawbacks.

In order to confirm this assumption, experiments were conducted on all 100-dimensional benchmarks. The following tables summarize the one-sided Wilcoxon rank sum test. For each algorithmic combination (A, B) the matrices show how often A significantly outperformed B. E.g., entry "1" in the first table shows that uniform significantly outperformed half-diff on one benchmark (out of 12).

Absorb bound handling

	1	2	3
uniform (1)	0	2	1
zero (2)	2	0	0
half-diff (3)	4	4	0

Random bound handling

	1	2	3
uniform (1)	0	0	0
zero (2)	0	0	0
half-diff (3)	2	2	0

Table 1. Comparison of bound handling strategies. The tables show summaries of one-sided Wilcoxon rank sum tests with significance level 0.01. $\mathcal{S}$ is the set of all benchmarks. *Example:* Entry {f5, f9} in the first table shows that *Absorb* performed significantly better than *Infinity* on f5 and f9.

Half-diff velocity initialization, 30 dimensions

	Absorb	Infinity	Random
Absorb	–	{f5, f9}	{f1, f2, f5, f6, f9, f12, f15}
Infinity	{f2, f12, f15}	–	{f1, f2, f5, f6, f12, f15}
Random	{Rastrigin}	{Rastrigin, Griewank}	–

Half-diff velocity initialization, 100 dimensions

	Absorb	Infinity	Random
Absorb	–	$\mathcal{S}\setminus${Sph., Rosenbr.}	{f1, f2, f5, f6, f9, f12, f15}
Infinity	–	–	–
Random	{Sph., Rosenbr., Rastr.}	$\mathcal{S}\setminus${Sph.}	–

Half-diff velocity initialization provides slightly better results than the other two strategies, and can therefore be recommended for PSO application.

3.2 Bound Handling

The theoretical analysis showed that many particles leave the search space at the beginning of the optimization process. To each of these particles, the bound handling procedure is applied, and therefore, bound handling strongly influences the particle swarm behavior, at least in the early steps of the algorithm. In order to check experimentally whether the bound handling method actually strongly influences particle swarm performance and whether the effect is stronger when more search space dimensions are involved, three bound handling strategies (*Absorb*, *Random*, and *Infinity*) were investigated on 30- and 100-dimensional optimization problems. Half-diff velocity initialization was used. The results are shown in Table 1, and confirm significant performance differences, especially for the 100-dimensional benchmarks.

4 Conclusion

Particle trajectories during the first iteration were investigated theoretically for three widely-used velocity initialization strategies. It was proven that many particles leave the search space as early as in the first iteration. To be more precise: Uniform velocity initialization causes all particles to leave the search space with overwhelming probability (w.o.p.) with respect to the search space dimensionality n. In order to reduce the number of particles leaving the search space, other velocity initialization strategies, among them zero [13] and half-diff [11] initialization, were proposed. However, this study showed that still many particles leave the search space, w.o.p. Examples demonstrated that this probability rapidly approaches 1 if the search space dimensionality is increased. Consequences for PSO application and strategies to deal with this observation were derived.

The presented analysis highlights the importance of bound handling for PSO: As the bound handling procedure is applied to many particles at the beginning of the optimization process, it strongly influences particle swarm behavior. The experimental study confirms significant performance differences when varying the bound handling method.

References

1. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (2001)
2. Clerc, M.: Confinements and Biases in Particle Swarm Optimization. http://clerc.maurice.free.fr/pso/ (2006)
3. Alvarez-Benitez, J.E., Everson, R.M., Fieldsend, J.E.: A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts. In: Evolutionlary Multi-Criterion Optimization. (2005) 459–473
4. Zhang, W.J., Xie, X.F., Bi, D.C.: Handling Boundary Constraints for Numerical Optimization by Particle Swarm Flying in Periodic Search Space. In: Proceedings of the IEEE Congress on Evol. Computation. Volume 2. (2004) 2307–2311
5. Robinson, J., Rahmat-Samii, Y.: Particle Swarm Optimization in Electromagnetics. IEEE Transactions on Antennas and Propagation **52**(2) (2004) 397–407
6. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In: Proc. IEEE Swarm Intelligence Symp. (2007) 120–127
7. Clerc, M.: Particle Swarm Optimization. ISTE Ltd (2006)
8. Clerc, M., Kennedy, J.: The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation **6**(1) (2002) 58–73
9. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters **85**(6) (2003) 317–325
10. Jiang, M., Luo, Y.P., Yang, S.Y.: Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Information Processing Letters **102**(1) (2007) 8–16
11. Clerc, M., et al.: Standard PSO 2007. http://www.particleswarm.info (standard_pso_2007.c) (2007)
12. Helwig, S., Wanka, R.: Particle Swarm Optimization in High-Dimensional Bounded Search Spaces. In: Proc. IEEE Swarm Intelligence Symp. (2007) 198–205
13. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley and Sons Ltd (2005)
14. Bradley, D., Gupta, R.: On the Distribution of the Sum of n Non-Identically Distributed Uniform Random Variables. Annals of the Institute of Statistical Mathematics **54**(3) (2002) 689–700
15. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. KanGAL Report 2005005, Nanyang Technological University, Singapore (2005)
16. Kennedy, J., Mendes, R.: Population Structure and Particle Swarm Performance. In: Proceedings of the IEEE Congress on Evol. Computation. (2002) 1671–1676