PSO Facing Non-Separable and Ill-Conditioned Problems

Nikolaus Hansen, Raymond Ros, Nikolas Mauny, Marc Schoenauer, Anne Auger

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

PSO Facing Non-Separable and Ill-Conditioned Problems

Nikolaus Hansen — Raymond Ros — Nikolas Mauny — Marc Schoenauer — Anne Auger

N° 6447

February 2008

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--6447--FR+ENG

PSO Facing Non-Separable and Ill-Conditioned Problems

Nikolaus Hansen*, Raymond Ros, Nikolas Mauny, Marc Schoenauer, Anne Auger

Thème COG — Systèmes cognitifs
Équipe-Projet TAO

Rapport de recherche n° 6447 — February 2008 — 29 pages

Abstract: This report investigates the behavior of particle swarm optimization (PSO) on ill-conditioned functions. We find that PSO performs very well on separable, ill-conditioned functions. If the function is rotated such that it becomes non-separable, the performance declines dramatically. On non-separable, ill-conditioned functions we find the search costs (number of function evaluations) of PSO increasing roughly proportional with the condition number. We never observe premature convergence, but on non-separable, ill-conditioned problems PSO is outperformed by a contemporary evolution strategy by orders of magnitude. The strong dependency of PSO on rotations originates from random events that are only independent within the given coordinate system. We argue that invariance properties, like rotational invariance, are desirable, because they increase the predictive power of performance results.

Key-words: Particle Swarm Optimization, performance assessment, ill-conditioned problems, non-separable problems, invariance

* Raymond Ros is with LRI, Project-team TAO, Université Paris-Sud, 91405 Orsay Cedex, France. All other authors are with INRIA Saclay, Project-team TAO, LRI, Université Paris-Sud, 91405 Orsay Cedex, France. For 1st, 4th and 5th author mail to `forename.name@inria.fr`, for 2nd and 3rd author mail to `forename.name@lri.fr`.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

OEP face à des problèmes non-séparables et mal conditionnés

Résumé : Pas de résumé

Mots-clés : Pas de motclef

1 Introduction

Particle Swarm Optimization (PSO) [14, 22, 23, 4] is a stochastic search algorithm inspired by flock behavior, aiming to minimize a non-linear objective function

$$f : X \subset \mathbb{R}^n \to \mathbb{R} \qquad (1)$$
$$\boldsymbol{x} \mapsto f(\boldsymbol{x}) \ ,$$

where the search space X is typically a convex subset of $\mathbb{R}^n$, while throughout this report X equals to $\mathbb{R}^n$.The typical search space dimensionality n, the number of design variables, ranges between two and a few hundred. Throughout this report the search costs are defined as the number of function evaluations on f. The search objectives are twofold: finding 1) a good solution $\boldsymbol{x}$, where $f(\boldsymbol{x})$ is as small as possible, while 2) the search costs should be as small as possible. Our preferable performance measure will be the expected costs needed to fall below a target function value.

Evolutionary Algorithms are stochastic search algorithms inspired by the principles of biological evolution. Similar to PSO they operate on a set of search points. More than ten years ago Salomon [19] recognized that the typical design of mutation and recombination operators in Genetic Algorithms often leads to an exceptional performance on decomposable problems. While decomposable problems are not considered to be hard, as they are not subject to the curse of dimensionality, a rotation of the problem, such that variables become dependent, often leads to a vast decline in performance. Naturally, for any search algorithm the question arises whether similar holds true. A question pursued for PSO in this article.

1.1 Why Search is Difficult

In order to understand success and failure of search algorithms we need to well understand the difficulties of the objective function f. Understanding f also leads to criteria on how to categorize and design test functions in a useful way. In the following we discuss two function properties that make a problem difficult: ill-conditioning, and, as a prerequisite, non-separability.

1.1.1 Decomposability and Separability

We call an objective function, $f : \boldsymbol{x} \mapsto f(\boldsymbol{x})$, separable with respect to coordinate i, if the optimal value for the i-th coordinate x_i does not dependent on the choice of the remaining coordinates. We call the objective function separable, if it is separable with respect to each coordinate. More formally, let $\boldsymbol{e}_i \in \mathbb{R}^n$ be the i-th unit vector. A function is said separable with respect to coordinate i if, for all $\boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^n$,

$$\arg\min_{\lambda \in \mathbb{R}} f(\boldsymbol{y} - y_i \boldsymbol{e}_i + \lambda \boldsymbol{e}_i) = \arg\min_{\lambda \in \mathbb{R}} f(\boldsymbol{z} - z_i \boldsymbol{e}_i + \lambda \boldsymbol{e}_i) \ .$$

Many well-known test functions are additively decomposable. They can be written as a sum of n one-dimensional functions f_i like $f(\boldsymbol{x}) = \sum_{i=1}^{n} f_i(x_i)$. Additively decomposable functions are separable while separable functions are not necessarily additively decomposable.

Separable functions are not subject to the curse of dimensionality. The global function minimizer can be found by minimizing n one-dimensional objective functions $f_i : \lambda \mapsto f(\boldsymbol{x} + (\lambda - x_i)\boldsymbol{e}_i)$, $i = 1, \ldots, n$, for any given $\boldsymbol{x}$ in $\mathbb{R}^n$. Their difficulty scales linearly with the search space dimension. In contrast, the search space volume increases exponentially fast with the dimension, leading to the notion of curse of dimensionality. In this report we will conduct experiments on separable ("easy") and non-separable ("hard") functions.

1.1.2 Ill-Conditioning

Informally speaking, the term ill-conditioned refers to a situation where different variables, or different directions in search space, show a largely different sensitivity in their contribution to the objective function value.

For a convex-quadratic function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x}$, where $\boldsymbol{H}$ is symmetric positive definite, the problem condition is well-defined by the condition number[1] of the Hessian matrix $\boldsymbol{H}$. The condition number of $\boldsymbol{H}$ is the ratio between its largest and smallest eigenvalue. Figuratively, the eigenvalues correspond to the squared relative lengths of the principal axes of the ellipsoid $\{\boldsymbol{x} \,|\, \boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} = 1\}$. For points located on the principal axes the gradient aligns with the axis direction, and the axis length determines the magnitude of movement that would be needed to achieve a certain change in function value.

More generally, we can call a function ill-conditioned, if for points with similar function values the minimal displacement that produces a certain function value improvement differs by orders of magnitude.

In practice, few domain knowledge is usually sufficient to identify separable problems. Therefore, n-dimensional search problems often exhibit intricate dependencies between their design variables. Ill-conditioned problems also often lead to premature convergence.

1.2 Invariance

Invariance is a fundamental concept in science, well reflected in the following quote attributed to Albert Einstein: *The grand aim of all science is to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms.* Invariance is the mathematical concept associated with this aim. For example, we desire a physical law or biological model to be invariant to environmental parameters, say weekday, temperature, or air humidity. Inclusion of these parameters into the model or the need for controlling them makes the model more complex and/or less general. The more invariance properties a model exhibits, or the fewer dependencies on exogenous parameters the model reveals, the wider is its applicability and the greater is its predictive power.

The same idea holds for invariance properties of search algorithms. In search, invariance properties induce equivalence classes of objective functions, on which the performance of the search algorithm is identical. Consequently, any result observed on a real world problem, or on a test function, does not only hold for this single problem instance, but inevitably generalizes to the complete class of problems induced by the invariance property, thereof the tested problem is an

[1] The condition number is defined here with respect to the euclidean norm.

element. Hence stronger statements on the performance of the search algorithm can be made—a greater number of empirical facts is covered.

The drawback to invariance properties in search is that whenever an invariance property is achieved, some piece of information cannot be exploited anymore. For example, rotational invariance means to abandon exploitation of the orientation of the coordinate system and therefore exploitation of separability. We review important invariance properties of search algorithms.

1.2.1 Invariance under Function Value Transformations

First we consider invariance under certain transformations $T : \mathbb{R} \to \mathbb{R}$ of the objective function value, specifically for the objective function $f(\boldsymbol{x}) = T(g(\boldsymbol{x}))$, for all $g : \mathbb{R}^n \to \mathbb{R}$.

- Invariance to adding a constant to the function value, that is $T(g) = g + a$ for $a \in \mathbb{R}$.
- Invariance under scaling of the function value, that is $T(g) = a \times g$ for $a > 0$.
- Invariance under order preserving transformations of the objective function value, where T is a strictly monotonically increasing function. Invariance under order preserving transformations includes the above listed invariance properties and is much more general.

Because PSO depends only on ranking of function values, it achieves the above listed invariance properties—the sequence of generated search points is independent of T. In particular, PSO is invariant under order preserving transformations of f. We believe that this is a very important feature of PSO and of all comparison-based search algorithms [7].

1.2.2 Invariance under Search Space Transformations

Next, we consider invariance under certain transformations $U : \mathbb{R}^n \to \mathbb{R}^n$ of the search space. Stated are invariances for the objective function $f(\boldsymbol{x}) = g(U(\boldsymbol{x}))$ under certain transformations U, for all $g : \mathbb{R}^n \to \mathbb{R}$. Strictly speaking, invariance under U only holds, if also the initial conditions are chosen appropriately. That is, in PSO, the distribution of the initial swarm and velocities must be chosen accordingly.

- Translation invariance: invariance under $U(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{a}$ for $\boldsymbol{a} \in \mathbb{R}^n$. Translation invariance must be taken for granted in continuous domain search. Being not translational invariant must be interpreted as having a problem independent build-in assumption about the location of the optimal solution. For a search algorithm this seems to be a contradiction in terms. On many test function sets this can lead to exceptional performance, but is entirely artificial. In contrast, the initial solution must be interpreted as a justified, problem dependent assumption about the location of the optimal solution.
- Scale invariance: invariance under $U(\boldsymbol{x}) = \alpha\, \boldsymbol{x}$, where $\alpha > 0$. PSO is scale invariant.

Finally, we have invariance properties where $U(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}$ is a full rank matrix.

- Diagonal invariance: invariance under diagonal linear transformations, *i.e.* under a scaling of variables. The matrix $\boldsymbol{A}$ is diagonal. PSO is invariant under diagonal linear transformations.
- Rotational invariance: invariance under angle preserving, *i.e.* rigid linear transformations of the search space (rotation, reflection). That is, $\boldsymbol{A}$ is an orthogonal matrix. Rotational invariance is closely related to decomposability and separability (see above), because, in most cases, a separable function becomes non-separable under rotation. Also any search algorithm can only either exploit separability or be rotational invariant.
- General linear invariance: invariance under any full rank, *i.e.* invertible matrix $\boldsymbol{A}$. This invariance requires invariance under rotation and requires additionally to abandon any invariable, inherent model of isotropy and scales.

Rotational invariance and general linear invariance of PSO depend on the way the update equations are precisely implemented and will be discussed below. We suspect that the benefit of an invariance property is related to the number of free parameters of the related transformation. Consequently orthogonal and general linear invariance must be considered important all together with invariance under order preserving transformations of the objective function value.

Most likely, the initial swarm and velocity distributions cannot be chosen according to a desired search space invariance property in practice. Therefore, an invariant search algorithm must also be adaptive: initial distributions, eventually with poor performance, must evolve within the iteration sequence into "the invariant" distributions, preferably with good performance, rendering the algorithm as independent of the initial distributions as possible. Adaptivity has the additional advantage that changes of the optimal distributions over time, *i.e.* their dependency on the positioning in search space, can be assimilated. While any time-invariable distribution literally exhibits general linear invariance as well, given the initial distribution is chosen appropriately, it is ultimately rather useless. Adaptivity must be regarded as the practical counterpart of the theoretical invariance property.

1.3 Objective of this Report

This article focuses on rotational invariance and ill-conditioned problems. We will review the reason for the coordinate system dependency of PSO and quantitatively answer the following questions.

1. How well does PSO perform on ill-conditioned problems?
2. How strongly does the performance depend on coordinate system rotations?

We will investigate PSO on a small number of carefully chosen test functions, and, in order to assess its performance, compare the results to results obtained from a contemporary Evolution Strategy, the CMA-ES [8].

2 Particle Swarm Optimization (PSO)

The so-called Standard PSO 2006[2] will be considered throughout this report and will be simply referred as S-PSO. We first describe the algorithm using non-standard notations (without velocities) thereby gaining some interesting insights.

S-PSO tracks a number of so-called particles (solution vectors) in a swarm. The default swarm size is $S = 10 + \lfloor 2\sqrt{n} \rfloor$. For each particle, $\boldsymbol{x} \in \mathbb{R}^n$, its previous best position $\boldsymbol{p}$ is recorded. Let $t \in \mathbb{N}$ be the time index and $\boldsymbol{x}^t$ the position of particle $\boldsymbol{x}$ at time t, then we have

$$\boldsymbol{p} \in \{\boldsymbol{x}^t, \boldsymbol{x}^{t-1}, \ldots, \boldsymbol{x}^1\} \tag{2}$$

chosen such that $f(\boldsymbol{p})$ is minimal. Additionally each particle has so-called informants. The particles for which a particle serves as informant are randomly drawn from all particles with replacement in $K = 3$ rounds for each particle anew, if there was no improvement of the overall best particle during the last iteration. Therefore, each particle serves as informant for between zero and K other particles. Additionally a particle informs always itself. Consequently, in the extreme cases, a particle can only have itself as informant, or the complete swarm.[3]

The informants serve to compute the "global best" position $\boldsymbol{g}$, that is the best previous best position of the informants of a particle.[4] If all particles are informants, $\boldsymbol{g}$ is the overall (globally) best position ever visited.

In order to compute, for each particle $\boldsymbol{x}$, the new position, $\boldsymbol{x}^{t+1}$, the four vectors $\boldsymbol{x}^t$, $\boldsymbol{x}^{t-1}$, $\boldsymbol{p}$, and $\boldsymbol{g}$ are used. The new position is coordinate wise a linear combination of these four vectors with coefficients summing to one. At each iteration step t for each particle $\boldsymbol{x}^t = (x_i^t)_{i=1,\ldots,n}$ a new position $\boldsymbol{x}^{t+1}$ is computed. We first consider the stochastic part of this computation which leads to the intermediate position

$$\begin{aligned} x_i^{t+\frac{1}{2}} &= x_i^t + U_i^+ \left(p_i - x_i^t\right) + V_i^+ \left(g_i - x_i^t\right) \\ &= \left(1 - U_i^+ - V_i^+\right) x_i^t + U_i^+ p_i + V_i^+ g_i \\ &= \frac{p_i + g_i}{2} + (p_i - x_i^t)U_i + (g_i - x_i^t)V_i \end{aligned} \tag{3}$$

for each coordinate $i = 1, \ldots, n$, where U_i^+ and V_i^+ are uniformly distributed in $[0, \varphi]$ with $\varphi = \log(2) + \frac{1}{2} \approx 1.19$, and $U_i = U_i^+ - \frac{1}{2}$, $V_i = V_i^+ - \frac{1}{2} \in [-0.5, 0.7]$. The final new position is a deterministic shift of $\boldsymbol{x}^{t+\frac{1}{2}}$ according to

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^{t+\frac{1}{2}} + w\left(\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\right) \tag{4}$$

[2] C code of the "Standard PSO 2006" can be found in `http://www.particleswarm.info/Standard_PSO_2006.c`.

[3] The probability for a particle to only have itself as informant is $\left(\frac{S-1}{S}\right)^{K(S-1)}$, where S is the swarm size. The probability to have all particles as informant is $\left(1 - \left(\frac{S-1}{S}\right)^K\right)^{S-1} \leq \left(\frac{K}{S}\right)^{S-1}$.

[4] Remark that $f(\boldsymbol{g}) \leq f(\boldsymbol{p})$ and $Pr(\boldsymbol{g} = \boldsymbol{p}) > \frac{1}{S}$. Also $f(\boldsymbol{g})$ can increase, *i.e.* $\boldsymbol{g}$ can become worse from one iteration step to the next, because the informants can change from one generation to the next. In contrast $f(\boldsymbol{p})$ is non-increasing.

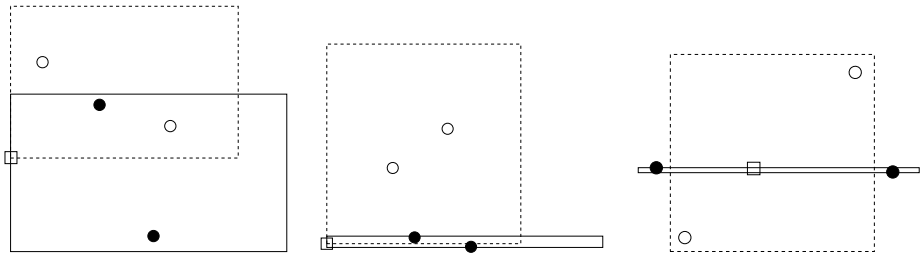

Figure 1: Domain of non-zero density for the intermediate particle position $\boldsymbol{x}^{t+\frac{1}{2}} = \boldsymbol{x}^{t+1} - w(\boldsymbol{x}^t - \boldsymbol{x}^{t-1})$, in three cases (solid rectangles), and after a 45 ° rotation around □ (dashed rectangles). The symbols indicate: □ = particle $\boldsymbol{x}^t$, ● = $\boldsymbol{p}$ and $\boldsymbol{g}$ (interchangeably), ○ = $\boldsymbol{p}$ and $\boldsymbol{g}$ in the rotated case.

where the inertia weight w equals to $\frac{1}{2\log(2)} \approx 0.72$. Equation 4 implements a momentum by repeating the position change of the last iteration step in a lossy way. The entire step taken by the particle is also called velocity

$$\begin{aligned}
\boldsymbol{v}^{t+1} &= \boldsymbol{x}^{t+1} - \boldsymbol{x}^t \\
&= \boldsymbol{x}^{t+\frac{1}{2}} + w\left(\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\right) - \boldsymbol{x}^t \\
&= w\left(\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\right) + (\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{x}^t) \\
&= w\,\boldsymbol{v}^t + (\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{x}^t) \ . \qquad (5)
\end{aligned}$$

2.1 Interpretation

The typical interpretation of Equation (3) is that the particle is pulled toward the best positions $\boldsymbol{g}$ and $\boldsymbol{p}$ as it can be implied from the first line of Equation (3). Because φ is greater than one, in fact the particle can be pulled beyond the best positions.

A second interpretation is that Equation (3) resembles a recombination operator of real coded genetic algorithms [17]. The new position results from a linear combination of x_i^t, p_i, and g_i, where the coefficients, different in each coordinate i, sum to one. This idea suggests itself from the second line of Equation (3). Realizing that the expected value for the coefficient in front of x_i^t is smaller than zero makes this interpretation less attractive. When including Equation (4) into this view point, the four values x_i^{t-1}, x_i^t, p_i, and g_i are linearly combined with a negative coefficient for x_i^{t-1}.

A third interpretation of Equation (3), emphasized in the last line, is that the new position is sampled around the average of $\boldsymbol{g}$ and $\boldsymbol{p}$. A similar, even more general reformulation was analyzed in [4]. According to our third interpretation $\boldsymbol{g}$ and $\boldsymbol{p}$ are averaged, and $\boldsymbol{x}^t$ primarily effects the perturbation width (step-size) around this average, according to its coordinate-wise distance to $\boldsymbol{g}$ and $\boldsymbol{p}$. The width is chosen such that the particle $\boldsymbol{x}^t$ itself is not outside the sampling domain. We reckon, from a simple combinatorial reasoning, that $\boldsymbol{x}^t$ is located on the boundary of the sample domain with a large probability, namely larger than $1 - \frac{1}{2}^n$. **Figure 1** illustrates this view point. Because $\varphi \approx 1.19 > 1$ the sample width is about a factor of $\frac{\varphi - 0.5}{0.5} \approx 1.4$ times larger opposite to $\boldsymbol{x}^t$ than toward $\boldsymbol{x}^t$.

Equation (3) depends on rigid transformations (rotations) of the search space, because the random numbers are sampled independently for each coordinate. Equation (4) is not coordinate system dependent and exhibits general linear invariance.

The effect of a search space rotation on Equation (3) is illustrated in **Fig. 1**. The effect can be dramatic. The distribution becomes narrow only if $\boldsymbol{g}$, $\boldsymbol{p}$, and $\boldsymbol{x}^t$ are aligned with a coordinate axis. Consequently, only if the swarm is aligned with a coordinate axis it can become arbitrarily narrow in the iteration sequence and resemble a high condition number. For other search space directions a narrow swarm shape cannot be realized. The illustration suggests to expect a significant dependency on whether an ill-conditioned problem is aligned with the coordinate system or rotated. This difference will later be empirically quantified.

2.2 Preserving Rotational Invariance

Rotational invariance and general linear invariance depend on a subtlety in Equation (3). If the random variable instances for U_i and V_i are chosen identically for all i (so-called linear update rule) the algorithm becomes invariant under linear transformations of the search space. In this case the trajectory of each particle tends to collapse into a "long narrow plane", reminiscent of a line search, leading to unfavorable performance [16, 24]. This behavior is in accordance with our general experience: search algorithms that exhibit general linear invariance generally tend to degenerate into low-dimensional subspaces. This phenomenon might be denoted as invariance-diversity dilemma. Evolutionary algorithms sometimes use a large population size to combat this degeneration implying large search costs. In contrast, rotationally invariant algorithms do not necessarily tend to degenerate. For example, rotational invariance is preserved when an isotropic perturbation is applied which moreover prevents effectively the degeneration into low-dimensional subspaces.

Wilke *et al.* [25] propose a modified linear update that preserves rotational invariance, in their terminology frame invariance, without the loss of diversity in the swarm. Despite that their *"goal is not to propose yet another competitive and/or superior PSO variant, but merely to illustrate that formulations that are both diverse and invariant do exist"*[25], the modification outperforms the original algorithm on a convex-quadratic function with moderate condition number by a factor of about three, and on a variant of the Rosenbrock function (where no such factor can be concluded from the presented data). In order to maintain diversity, rotation matrices with small angles are applied to the difference vectors. The angle parameter determines the mean width of the rotation and controls the compromise between diversity preservation (for larger values) versus direction preservation (for smaller values). Unfortunately, general linear invariance is not preserved. Therefore, we conjecture that the "amount of elongation" that a swarm can exhibit will be limited, depending on the angle parameter chosen.

3 Methods and Test Functions

3.1 Test Functions

We use a small set of well-established benchmark functions, given in Table 1. All

Table 1: Test functions and coordinate-wise initialization intervals, where $\boldsymbol{y} := \boldsymbol{B}\boldsymbol{x}$ and $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$ implements an angle-preserving, linear transformation, *i.e.* $\boldsymbol{B}$ is orthogonal

Name	Function	α
Ellipsoid	$f_{\text{elli}}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha^{\frac{i-1}{n-1}} y_i^2$	$[1, 10^{10}]$
Rosenbrock	$f_{\text{Rosen}}(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(\alpha \left(y_i^2 - y_{i+1}\right)^2 + \left(y_i - 1\right)^2\right)$	$[1, 10^{8}]$
Diff-Powers	$f_{\text{diffpow}}(\boldsymbol{x}) = \sum_{i=1}^{n} y_i^{2+\alpha\frac{i-1}{n-1}}$	$[0, 10]$
Rastrigin	$f_{\text{Rastrigin}}(\boldsymbol{x}) = 10\,n + \sum_{i=1}^{n} \left(y_i^2 - 10\cos(2\pi y_i)\right)$	

Name	Initialization range	Target function value
Ellipsoid	$[-20, 80]^n$	10^{-9}
Rosenbrock	$[-20, 80]^n$	10^{-9}
Diff-Powers	$[-20, 80]^n$	10^{-14}
Rastrigin	$[-20, 80]^n$	10^{-9}

functions are tested in their original axis-parallel version (i.e. $\boldsymbol{B}$ is the identity and $\boldsymbol{y} = \boldsymbol{x}$), and in rotated versions, where $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]\,\boldsymbol{x}$. The orthogonal matrix $\boldsymbol{B}$ is chosen such that each $\boldsymbol{b}_i$ is uniformly distributed on the unit hypersphere surface [11], fixed for each run. We shall now discuss each function in turn.

The ellipsoid is a convex-quadratic function. The parameter α is the condition number of the Hessian matrix which will be varied between 1 and 10^{10} in our experiments. For $\boldsymbol{B} = \boldsymbol{I}$, the Hessian matrix is diagonal. If $\alpha = 1$ the ellipsoid is the isotropic sphere function.

The Rosenbrock function has its global minimum at $\boldsymbol{x} = [1, 1, \ldots, 1]$ and, for large enough α and n, one local minimum close to $\boldsymbol{x} = [-1, 1, \ldots, 1]$, see also [21]. The probability to end up in the local optimum is to our experience clearly smaller than 50%. In the Rosenbrock function the parameter α tunes the width of the bent ridge that guides to the global optimum. In the classical Rosenbrock function α equals 100. For smaller α the ridge becomes wider and the function becomes less difficult to solve. We vary α between one and 10^8.

The Diff-Powers function takes the variables to different powers. The function cannot be solved by applying a constant scaling between variables. The sensitivity differences between the variables increase with decreasing distance to the optimum: the closer to the optimum, the more difficult it gets to approach it further. The parameter α, varied between 0 and 10, determines the largest exponent and was originally set to 10 [11].

The Rastrigin function is highly multi-modal. While we are mainly interested in investigating the effect of ill-conditioning, the Rastrigin function serves to find a possible trade off between the ability to effectively conduct local search on ill-conditioned functions versus the ability to effectively search globally in a highly multi-modal topography.

For $\boldsymbol{B} = \boldsymbol{I}$ all functions but the Rosenbrock function are separable. Otherwise only the ellipsoid function for $\alpha = 1$ and the Diff-Powers function for $\alpha = 0$ remain separable.

3.2 CMA-ES

In order to assess its performance we compare S-PSO to the well-established Covariance Matrix Adaptation Evolution Strategy, CMA-ES [10, 11, 8]. The CMA-ES is shown to perform well on ill-conditioned non-separable problems [11, 9, 15] as well as on multi-modal problems [8, 15]. The CMA-ES represents the state-of-the-art in continuous domain evolutionary computation as it performed superior in a competition on 25 test functions conducted for the Congress on Evolutionary Computation CEC 2005.[5] It was competitive on the subset of all unimodal functions and on the subset of all multi-modal functions. Only on separable functions it was significantly outperformed by other competitors.

In the CMA-ES, at iteration step t, new individuals $\boldsymbol{x} \in \mathbb{R}^n$ are generated by sampling a multi-variate normal distribution,

$$\boldsymbol{x} = \boldsymbol{m}^t + \sigma^t \times \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{C^t}\big) \quad , \tag{6}$$

where $\mathcal{N}\big(\boldsymbol{0}, \boldsymbol{C^t}\big)$ is a normal distribution with mean $\boldsymbol{0}$ and $n \times n$ covariance matrix $\boldsymbol{C^t}$, and $\sigma^t > 0$ is a scaling parameter, the step-size. All individuals obey the same distribution with mean $\boldsymbol{m}^t$. After $\lambda > 1$ individuals have been sampled, evaluated on f, and sorted according to their objective function values, the distribution parameters $\boldsymbol{m}^t$, σ^t, and $\boldsymbol{C^t}$ are updated for a new iteration step using the sorted population.

Unsurprisingly CMA-ES and PSO share common concepts. They both are set-based (the iteration is based on a set of solutions, rather than on a single solution point), stochastic search procedures. More interestingly though both algorithms are essentially based on a momentum equation of virtually identical nature. The update of the velocity vector $\boldsymbol{v}^t$ with discount factor $w < 1$ and stochastic "input" $\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{x}^t$, as developed in Equation (5), reads

$$\boldsymbol{v}^{t+1} = w\, \boldsymbol{v}^t + (\boldsymbol{x}^{t+\frac{1}{2}} - \boldsymbol{x}^t) \quad ,$$

where $\boldsymbol{x}^{t+\frac{1}{2}}$ was interpreted as the perturbed average of the best positions $\boldsymbol{p}$ and $\boldsymbol{g}$. This equation has a close conceptual counterpart in the CMA-ES, where the notion of an evolution path, or cumulation, has been used [11]. An evolution path accumulates the movements of the mean population, driven by selection to better positions, just as the velocity accumulates movements of a single particle toward a disturbed average good position. The update of the evolution path is conducted with a discount factor $\eta \approx n/(n+4)$. For a single-parent population, where $\boldsymbol{m}^t$ equals to the best individual of the former iteration step, the update reads exemplary

$$\boldsymbol{p}^{t+1} = \eta\, \boldsymbol{p}^t + \frac{\sqrt{1-\eta^2}}{\sigma} (\boldsymbol{m}^{t+1} - \boldsymbol{m}^t) \quad . \tag{7}$$

The stochastic "input" to Equation (7) is the movement of the population mean (toward better solutions), adequately normalized. The choice for the discount

[5] `http://www.bionik.tu-berlin.de/user/niko/cec2005.html`

factor corresponds to a backward time horizon of $\frac{1}{1-\eta} \approx \frac{n}{4} + 1$ iterations, where the recommended range is between $\sqrt{\frac{n}{2}}$ and n, see [11]. Finally, the evolution path directs new mutations by updating covariance matrix $\boldsymbol{C}^{\boldsymbol{t}}$ and step-size σ^t. Here arises an interesting difference in the usage of $\boldsymbol{p}^{t+1}$, compared to the usage of velocity vectors $\boldsymbol{v}^{t+1}$ in PSO. The evolution path is used point symmetrically, in that $\boldsymbol{p}^{t+1}$ and $-\boldsymbol{p}^{t+1}$ are equivalent. New mutations in both directions are, on purpose, equally amplified and $\boldsymbol{p}^{t+1}$ in itself has no influence on the future mean position. A complete description of the CMA-ES algorithm and its default parameter settings, as used in this article, can be found in [8].

The CMA-ES reveals all invariance properties discussed in Sect. 1.2 where the initial values of $\boldsymbol{m}$ and $\boldsymbol{C}$ must be adjusted accordingly. The population size λ is the only strategy internal parameter that needs to be set depending on the objective function. Yet a default choice and a logical variation procedure are available, namely the increase from its default value by a constant factor. Trying small populations first is logical as they come along with low search costs. Therefore, an automated restart mechanism with increasing population size [2] renders CMA-ES quasi parameter free.

3.3 Experimental Setup

3.3.1 Algorithms and Parameters

For Particle Swarm Optimization the Standard PSO 2006 C-code[6], "*validated by [...] James Kennedy and Maurice Clerc*" was translated into Scilab. Also for CMA-ES Scilab-Code was used. All default parameters were applied including swarm size and population size accordingly leading to similar values for S-PSO and CMA-ES: for search space dimensions $n = 10; 20; 40$ the swarm size was $16; 18; 22$ and the population size was $10; 12; 15$. The default termination criteria were adjusted as described below. No automatic restarts were conducted in order to facilitate the interpretation of the results. Only on the Rastrigin function the swarm size for S-PSO and the population sizes for CMA-ES were varied between 10 and 1000.

3.3.2 Initial Conditions

The initial swarm for S-PSO and the initial distribution mean for CMA-ES were sampled uniformly distributed in the intervals given in Table 1 for both rotated and non-rotated functions. The initial velocities, according to Equation (5), were sampled for each particle as half of its way to another uniformly sampled solution point. The initial σ for CMA-ES was 1/3 of the interval width.

3.3.3 Termination Criteria

A run was terminated when the target function value according to Table 1 was reached or the maximum number of function evaluations 10^7 was exceeded. In order to reduce CPU-time consumption, the CMA-ES was additionally terminated when the population had converged on the Rosenbrock and the Rastrigin function. While this is an disadvantage in principle, the probability that it has affected the outcome of any of the presented results is negligible. For S-PSO

[6] `http://www.particleswarm.info/Standard_PSO_2006.c`

this approached proved to be infeasible, because the swarm did not regularly converge to a single point.

3.3.4 Experiments

In each experiment 21 trials were conducted and for each trial on a rotated function a new basis $\boldsymbol{B}$ was chosen. The same bases were used for both S-PSO and CMA-ES.

3.3.5 Performance Measures

We consider a trial, or run, to be successful, if the target function value is reached before 10^7 function evaluations are exceeded. The success rate is the ratio of successful trials in an experiment of usually 21 trials. The so-called success performance measures the number of function evaluations needed to reach the target function value, taking into account successful and unsuccessful runs. An estimator for the success performance $\widehat{\mathrm{SP1}}$, as used in [8], analyzed in [1], and also denoted as Q-measure in [6], is computed as the average number of function evaluations for the successful trials, divided by the success rate. The $\widehat{\mathrm{SP1}}$ is an estimate for the expected number of function evaluations to reach the target function value (with probability one), when the algorithm is applied repeatedly, given that termination of unsuccessful runs can be accomplished such that the expected run length of unsuccessful runs equals the average run length of successful runs [1].

3.3.6 Statistical Procedures

When we state observing a difference between two results in the following, we have always asserted its statistical significance. Tests for statistical significance were conducted using either the non-parametric Mann-Whitney rank sum test, or Pearson's χ^2 test for the binomial success probabilities. For the latter the function `chisq.test` from the free software environment R was used and the p-value was simulated. Statistical significance is assumed for $p < 0.01$. In order to derive a dispersion measure for $\widehat{\mathrm{SP1}}$, bootstrapping was used [5]. The empirical bootstrap distribution was sampled 10^4 times. Statistical significance was reasoned when the 5%-ile of one distribution was larger than the 95%-ile of the other.

4 Empirical Results

4.1 Ellipsoid Function

Figure 2 depicts the evolution of the median function value of 21 runs for condition numbers 1, 10, and 100 for S-PSO and CMA-ES. In all trials the target function value was reached. For condition number $\alpha = 1$ the rotated and the non-rotated function are identical. With increasing condition number the results for S-PSO on the rotated versus the non-rotated function become different. For a condition number of 100 S-PSO is already about four times slower on the rotated function than in the non-rotated separable function.

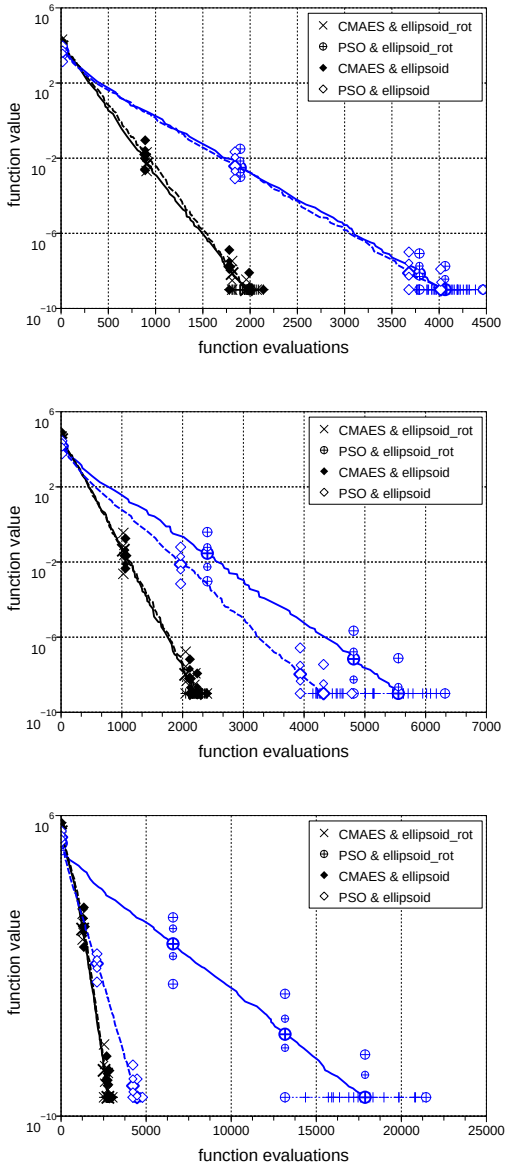

Figure 2: Ellipsoid function in 10-D: time evolution of the median function value with a function condition number α of 1, 10, and 100, from top to bottom. Small symbols indicate the 25% and 75%-ile, large symbols indicate smallest and largest value from 21 trials. Solid lines: rotated function, dashed lines: non-rotated (separable) function. S-PSO: $\oplus, \diamond$; CMA-ES: $\times, \blacklozenge$

Summarized performance results on the Ellipsoid function for all dimensions and condition numbers are shown in **Fig. 3**, where $\widehat{\mathrm{SP1}}$ is plotted versus the condition number α. Besides for some of the right most displayed points on the rotated function all runs reached the target function value of 10^{-9}. Therefore

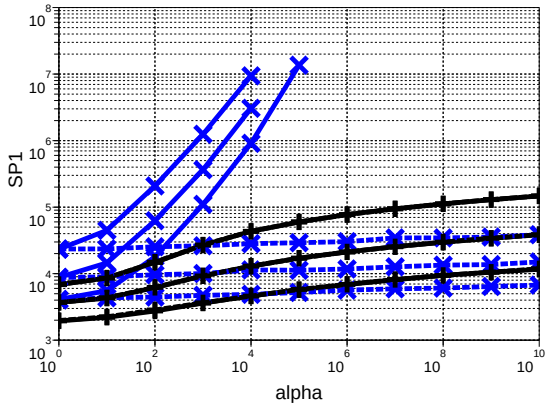

Figure 3: Ellipsoid function: $\widehat{\text{SP1}}$ to reach $f_{\text{target}} = 10^{-9}$ on the rotated (solid) and the non-rotated (dashed) function in dimensions $10, 20, 40$ versus condition number α. S-PSO: $\times$, CMA-ES: $+$. Up to the condition number of 10^3 the success rates are 100%. For condition numbers larger than 10^4 S-PSO regularly exceeds the maximum number of function evaluations of 10^7.

$\widehat{\text{SP1}}$ equals to the average number of function evaluations to reach the target function value. The dependency of S-PSO on the rotation of the ellipsoid is dramatic. In the non-rotated case S-PSO works very well for all tested condition numbers and outperforms CMA-ES for large condition numbers by a factor of up to four. In the rotated case the performance degrades fast with increasing condition number. For $\alpha = 10^4$ S-PSO is already more than a hundred times slower than on the non-rotated Ellipsoid function. The rotation leads to a failure to reach the target function value before 10^7 function evaluations for condition numbers larger than 10^5 in dimension 10 and for condition numbers larger than 10^4 in dimensions 20 and 40. For condition numbers larger than 100 $\widehat{\text{SP1}}$ becomes roughly proportional to α, that means the necessary number of function evaluations increase linearly with an increasing condition number. For CMA-ES the $\widehat{\text{SP1}}$ increases at most with $\alpha^{0.25}$, and, for large α, roughly with $\alpha^{0.1}$.

4.2 Rosenbrock Function

The Rosenbrock function already exhibits dependencies between parameters in its original non-rotated version, where the condition parameter α is set to 100. In **Fig. 4** the time evolution of the median function value is shown in 10-D for the rotated and the original functions. Surprisingly also on the Rosenbrock function the rotation has a remarkable effect on the performance of S-PSO when the optimum needs to be approached (function values smaller than about four). On the rotated version S-PSO is finally slower by a factor of about ten. We investigate the effect of a varying parameter α. Table 2 shows the rates of successful runs for all experiments conducted on the Rosenbrock function. For small values of α the success rates are generally large. In 10 and 20-D the rotation becomes visible in the resulting success rate of S-PSO for $\alpha = 1000$. In 40-D already for $\alpha = 100$ the success rate drops to zero on the rotated

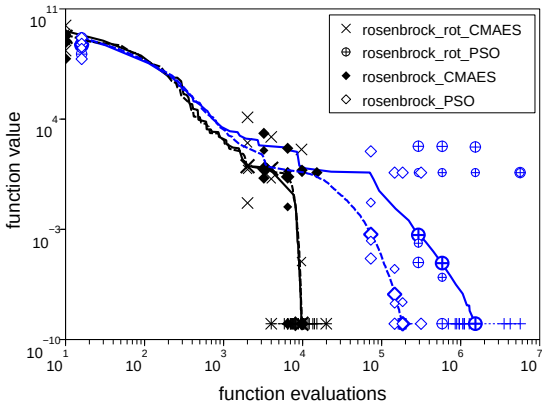

Figure 4: Rosenbrock function in 10-D with the (original) function condition parameter $\alpha = 100$: median function value versus function evaluations, rotated (solid lines) and non-rotated (dashed lines) case.

function. For $\alpha \geq 10^5$ S-PSO fails to reach the target function value all together. In contrast, for CMA-ES there is no significant influence of $\alpha \geq 100$, of the dimension, or of the rotation on the success rate. Unsuccessful trials of CMA-ES are those that end up in the local minimum of the Rosenbrock function, with a function value of just under 3.987 for the used dimensions.

The empirical distribution of the number of function evaluations to reach the target function value is depicted in **Fig. 5** for $\alpha = 1; 10; 100$ in 10-D. Even for $\alpha = 1$ the rotation significantly compromises the performance of S-PSO and the effect becomes slightly more pronounced with increasing α. In all cases the necessary number of evaluations increases with increasing α as the distribution graphs move to the right. For $\alpha = 100$ we can conjecture that some runs of S-PSO might reach the target value slightly after the maximum number of function evaluation is exceeded.

Figure 6 shows all $\widehat{\mathrm{SP1}}$ measures on the Rosenbrock function, where one line in Fig. 5 collapses to a point in Fig. 6. The rotation leads roughly to an increase of $\widehat{\mathrm{SP1}}$ by a factor of five to ten independent of α or the search space dimensionality. The sensitivity to the condition parameter α is quite pronounced: the number of function evaluations increases roughly linearly with α. From these graphs we can conjecture that the drop of the success probability for large α is most likely induced by the termination criterion of 10^7 function evaluations and does not indicate a principle failure of S-PSO. Nevertheless S-PSO is distinctly outperformed by CMA-ES on the Rosenbrock function in particular in the rotated case and for large values of α. For CMA-ES the number of function evaluations increases more moderately with about $\alpha^{0.27}$, $\alpha^{0.3}$, and $\alpha^{0.33}$ for $n = 10$, 20, and 40.

4.3 Diff-Powers Function

Experiments on the Diff-Powers function are shown in **Fig. 7** for $\alpha = 0, 2, 10$. For $\alpha = 0$ the isotropic and quadratic sphere function is recovered. For $\alpha = 2$ the

Table 2: Rosenbrock function: percentage of successful trials (number in parentheses), out of 21, for S-PSO above and CMA-ES below. A dash (–) means that no trial reached the value 10^{-9} within 10^7 evaluations.

dimension	α	1	10	100	300	1000	10000	10^5	10^6	10^8
10		100% (21)	95% (20)	81% (17)	43% (9)	81% (17)	81% (17)	–	–	–
	rotated	100% (21)	90% (19)	71% (15)	86% (18)	–	–	–	–	–
20		100% (21)	90% (19)	76% (16)	62% (13)	86% (18)	–	–	–	–
	rotated	100% (21)	86% (18)	62% (13)	48% (10)	–	–	–	–	–
40		100% (21)	86% (18)	67% (14)	62% (13)	24% (5)	–	–	–	–
	rotated	100% (21)	86% (18)	–	–	–	–	–	–	–
10		100% (21)	90% (19)	71% (15)	86% (18)	86% (18)	76% (16)	71% (15)	81% (17)	86% (18)
	rotated	100% (21)	100% (21)	100% (21)	90% (19)	76% (16)	81% (17)	81% (17)	100% (21)	86% (18)
20		100% (21)	71% (15)	81% (17)	90% (19)	90% (19)	76% (16)	81% (17)	86% (18)	76% (16)
	rotated	100% (21)	76% (16)	76% (16)	86% (18)	86% (18)	86% (18)	86% (18)	67% (14)	76% (16)
40		100% (21)	95% (20)	71% (15)	81% (17)	81% (17)	62% (13)	95% (20)	81% (17)	67% (14)
	rotated	100% (21)	100% (21)	86% (18)	81% (17)	86% (18)	90% (19)	71% (15)	71% (15)	76% (16)

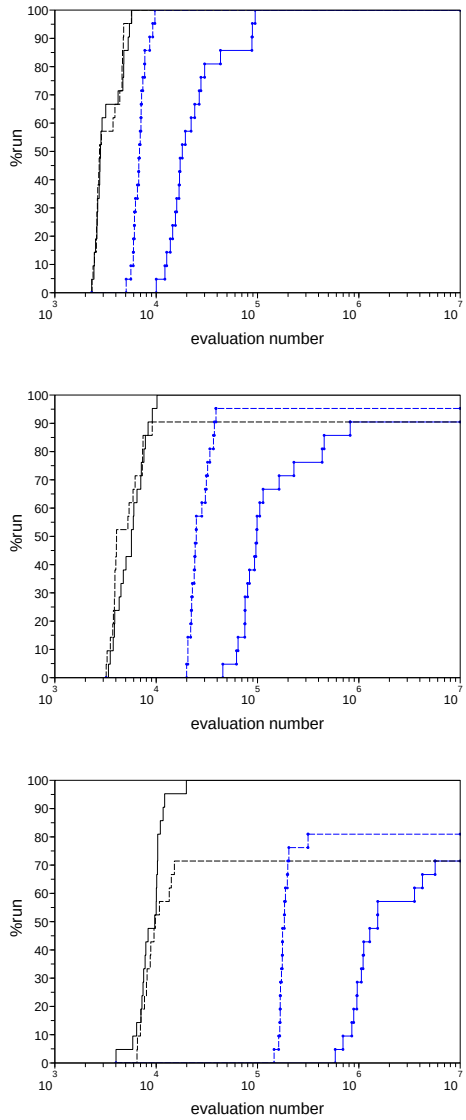

Figure 5: Rosenbrock function in 10-D: empirical cumulative distribution of the number of function evaluations to reach the target function value with a function conditioning parameter α of 1, 10, and 100 from top to bottom. S-PSO (with small bullets) corresponds to the two lines to the right in each case. Solid lines: rotated, dashed lines: non-rotated function.

used summands are between x^2 and x^4. The sensitivity to the rotation is again well visible. For the non-rotated function the performance is even comparable to the sphere function. Taking into account the logarithmic scale of the x-axis we find the convergence rate slowing down significantly when approaching the optimum in the rotated case. In the rotated case S-PSO needs 500 times more

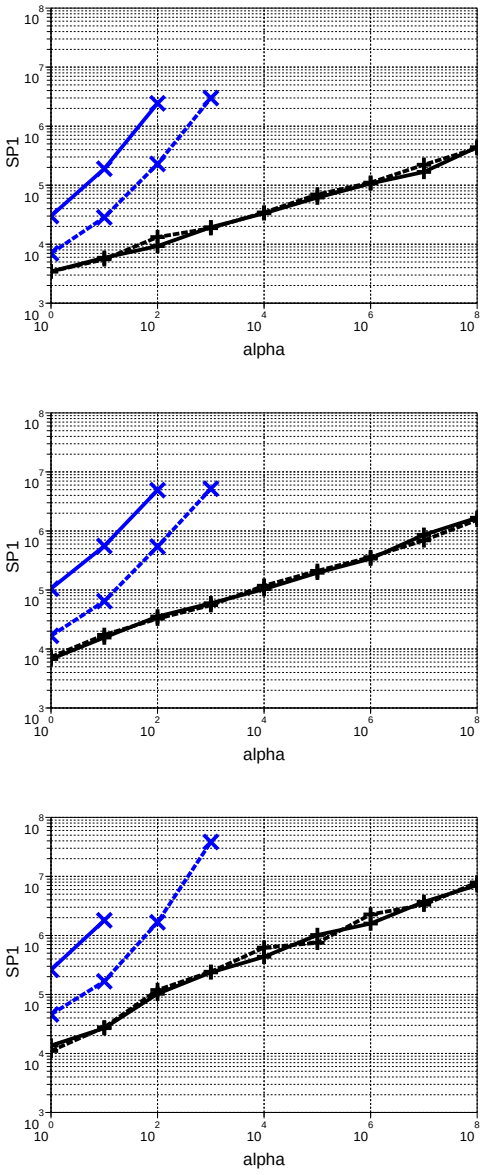

Figure 6: Rosenbrock function: $\widehat{\mathrm{SP1}}$ versus conditioning parameter α in 10, 20, and 40-D, from top to bottom. S-PSO corresponds to the two upper lines in each case. Solid lines: rotated, dashed lines: non-rotated.

function evaluations than in the non-rotated case to reach the target function value. As to be expected, for $\alpha = 10$ the effect becomes even more pronounced.

Figure 8 depicts $\widehat{\mathrm{SP1}}$ and reveals that only for $\alpha \leq 2$ S-PSO reaches the target function value on the rotated function. Where CMA-ES slows down by a factor of up to three for α approaching ten, in the non-rotated case S-PSO becomes with increasing α even slightly faster. Can we explain this behavior? The

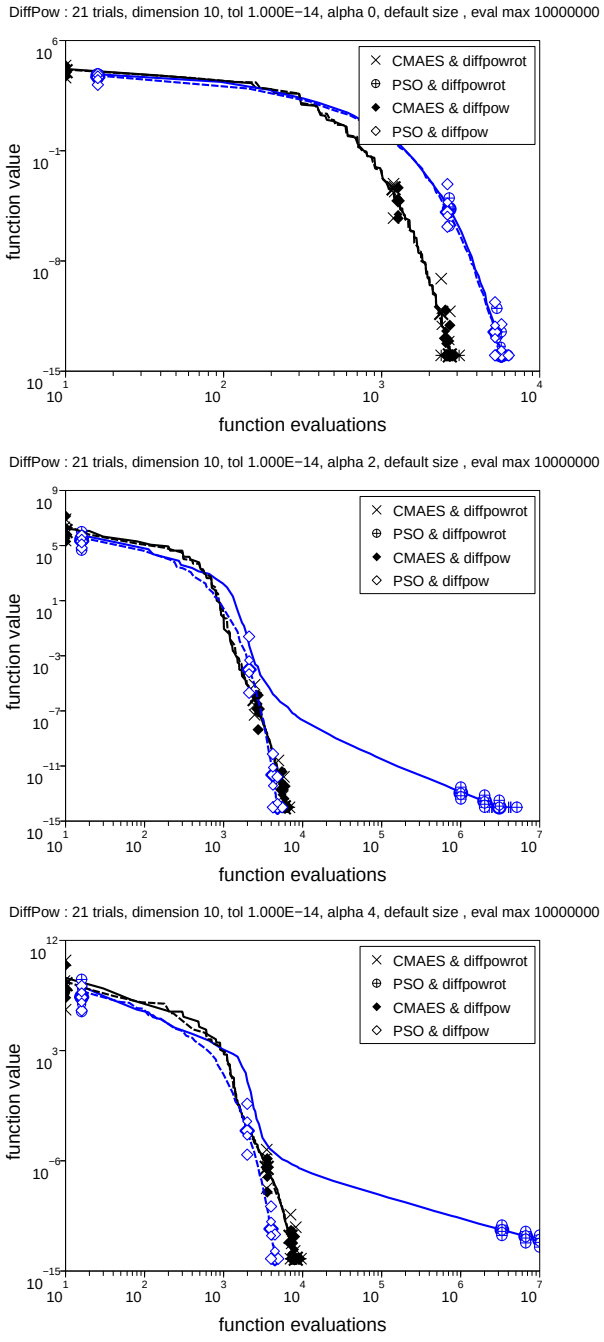

Figure 7: Diff-Powers function in 10-D: time evolution of the median function value, rotated and non-rotated case, with parameter $\alpha = 0, 2, 4$, from top to bottom. Small symbols indicate the 25%- and 75%-ile, large symbols indicate smallest and largest value from 21 trials. S-PSO: $\oplus, \diamond$; CMA-ES: $\times, \blacklozenge$

Diff-Powers function becomes generally more difficult to solve with increasing α. The sensitivity difference between the parameters becomes more pronounced

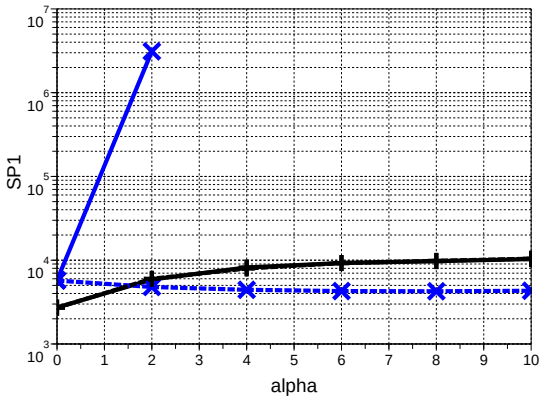

Figure 8: Diff-Powers function in 10-D: $\widehat{\mathrm{SP1}}$ versus parameter α. S-PSO: $\times$, CMA-ES: $+$; solid lines: rotated, dashed lines: non-rotated function. For CMA-ES both graphs are virtually identical. Missing points indicate that 10^7 function evaluations were exceeded in all trials. The axis parallel function becomes with increasing α even more easy to solve for S-PSO.

and the topography becomes less spherical. On the contrary, in order to reach the same target function value, when α increases, the parameters with large an exponent need to be located with a lesser precision. For the same reason the search space volume for which the function value is smaller than the target function value increases with increasing α. For this reason S-PSO becomes slightly faster on the non-rotated function and for the same reason also the target function value was chosen smaller than 10^{-9}.

The condition numbers realized by the covariance matrix of CMA on the Diff-Powers function are comparatively large as shown in **Figure 9**. For $\alpha = 2$ a simulated condition number of above 10^5 can be observed when the target function value is reached. Nevertheless, the performance loss is moderate. The observed condition number seems to overestimate the "real difficulty". Still, similar as for the Ellipsoid function, for a simulated condition number of larger than 10^5 S-PSO is not able to reach the target function value within 10^7 function evaluations also on the rotated Diff-Powers function.

4.4 Rastrigin Function

The Rastrigin function is a highly multi-modal test function and not easy to solve. Our tests on the Rastrigin function serve to check whether a trade off between local and global search performance can be observed.

On multi-modal functions the swarm size becomes a decisive factor. Therefore different swarm size between 10 and 1000 were investigated. **Figure 11** shows the time evolution of the function value from all 21 runs on the rotated (middle) and the non-rotated (left) Rastrigin functions for swarm sizes of 30, 100, 300, and 1000. Even on the Rastrigin function the results are entirely different on the rotated versus the non-rotated function. The separable Rastrigin function can be solved reliably, if the swarm size is chosen large enough, say,

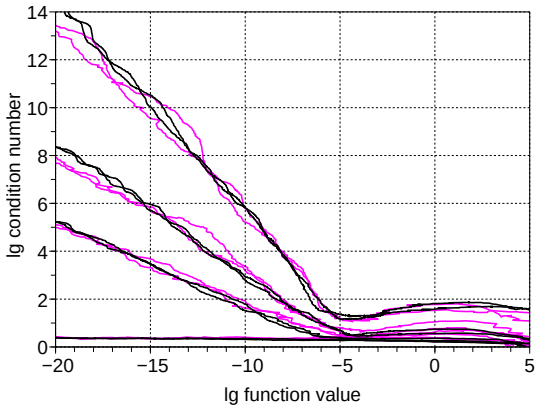

Figure 9: Diff-Powers function: log-log plot (base 10) of the condition number of the covariance matrix in CMA-ES versus the best function value per generation where $\alpha = 0, 1, 2, 10$ (bottom to top), in 10- and 30-D, two runs in each case. For large α the larger dimension yields slightly larger final condition numbers. The target function value, 10^{-14} corresponds to a condition number of somewhat above 10^5 for $\alpha = 2$, and somewhat below 10^{10} for $\alpha = 10$.

Table 3: Rastrigin function in 10-D: percentage of successful trials (number in parentheses), out of 21, for different swarm/population sizes

S, λ	10&16	30	100	300	1000
PSO	–	5% (1)	71% (15)	100% (21)	100% (21)
rotated	–	–	5% (1)	–	–
CMA-ES	–	–	24% (5)	76% (16)	100% (21)
rotated	–	5% (1)	10% (2)	90% (19)	100% (21)

not smaller than $100 = 10n$. In contrast, the rotated Rastrigin function was solved by only one trial with swarm size 100. Conducting 160 more trials for this set-up found another single success and we conclude that the success rate is roughly 1%. This is in agreement with all the presented data, but beyond the sensitivity of our experimental set-up. Figuring small success rates was not the goal of our investigation. Table 3 tabulates the success rates of spotting the global optimum from all experiments (excluding the 160 additional trials). The effect of rotation becomes visible for swarm size 100, where the non-rotated Rastrigin function is solved in 71%. Swarm size in S-PSO and population size in CMA-ES seem to be comparable, however S-PSO needs somewhat smaller swarm sizes to be successful on the separable Rastrigin function compared to CMA-ES.

Figure 12 shows the empirical cumulative distribution functions of the number of function evaluations to reach the target function value on the left part, and of the best function value achieved at maximum number of evaluations 10^7 on the right from all experiments with swarm/population sizes between 30

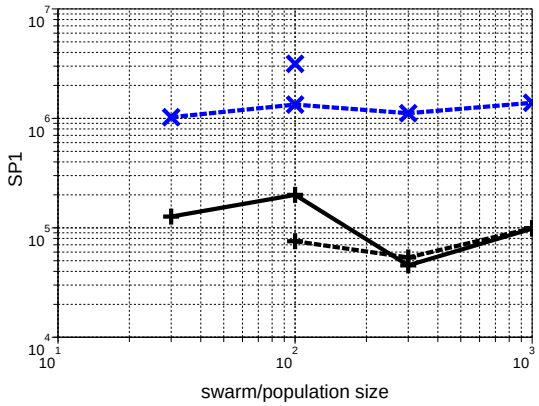

Figure 10: Rastrigin function in 10-D: $\widehat{\text{SP1}}$ versus swarm size/population size. S-PSO: $\times$ (above lines), CMA-ES: $+$ (below lines); solid lines: rotated, dashed lines: non-rotated function. For missing points no trial reached the target function value.

and 1000. The value at the graph transition between left and right reflects the success rate. For the smallest swarm/population size all results are fairly similar. Again, for swarm sizes $S \geq 100$ the dependency of S-PSO on the rotation becomes clearly visible. For larger swarm/population sizes a difference to CMA-ES becomes clearly visible. Even on the separable Rastrigin function S-PSO is approximately ten times slower than CMA-ES to reach the target function value.

These result are summarized in terms of $\widehat{\text{SP1}}$ in **Fig. 10**. The variance of $\widehat{\text{SP1}}$ for a small swarm or population is large as the outcome depends on a small number of successful runs. Yet the graphs are rather flat in that the $\widehat{\text{SP1}}$ measure is comparatively insensitive to the swarm or population size, as long as the function was solved at all. Invariably S-PSO is roughly ten times slower than CMA-ES while the statistical significance of this difference is only asserted for $S = \lambda \geq 300$.

The strong dependency of S-PSO on the rotation of the Rastrigin function comes as a surprise to us. On the non-rotated Rastrigin function it is generally not sufficient to align the swarm along one coordinate axis. Large steps must be taken along different coordinate directions, if the global optimum shall be located based on the functions separability. In order to make such steps in coordinate direction the three vectors $\boldsymbol{x}^t$, $\boldsymbol{x}^{t-1}$, and $\frac{\boldsymbol{p}+\boldsymbol{g}}{2}$ must be similar in all but a few components, say two or three. This seems to be the case systematically. Additional experiments in 20-D with swarm size 2000 also gave a 100% success rate and make the moderate dimension as explanation factor implausible.

5 Summary and Conclusion

We summarize the findings of this article, where we investigated Standard PSO 2006 (PSO) and conducted numerical experiments on four parameterized test

functions with condition numbers of up to 10^{10}, and search space dimensionality between 10 and 40. No parameters were varied besides the swarm size on the Rastrigin function, as well as the population size for the accompanying experiments with a contemporary Evolution Strategy (ES), the CMA-ES. Our results are mainly expressed in terms of function evaluations needed to reach a target function value and therefore yield a quantitative assessment.

Invariance We believe invariance is an important aspect of continuous domain search algorithms. Invariances induce equivalence classes of objective functions and consequently guaranty the generalization of performance results within each class, thereby considerably strengthening its relevance. PSO is invariant under order-preserving transformations of the objective function value. PSO is invariant under a scaling of the search space and a scaling of single variables. PSO is not invariant under rotations of the search space in its standard formulation [25].

Ill-Conditioned, Separable Functions PSO performs very well on ill-conditioned, separable functions, where the design variables of the objective function are independent. The result corresponds to an invariance under the scaling of variables (diagonal invariance). On ill-conditioned, separable functions PSO outperforms the rotationally invariant ES by a factor of about three. The diagonal invariance of PSO supports the generalizability of this empirical finding.

Coordinate System Rotation The performance of PSO on even moderately ill-conditioned functions declines remarkably with coordinate system rotations, where the design variables of the objective function become dependent. Also on the per-se non-separable Rosenbrock function, a coordinate system rotation leads to a decline in performance by a factor of about ten, nearly independently of search space dimension and conditioning parameter.

Non-Separable Functions On non-separable functions the performance of PSO slows down about proportional with the condition number of the problem, for condition numbers larger than 100. This holds true on quadratic and non-quadratic problems and is independent of the problem dimension. Our data do not indicate that PSO converges prematurely and fails to solve ill-conditioned, non-separable functions. However, in comparison, a contemporary ES achieves slightly better performance on well conditioned problems and scales roughly between $\alpha^{0.1}$ and $\alpha^{0.33}$ with condition number α. Consequently, PSO is outperformed roughly by a factor of $\alpha^{0.7}$—for still a moderate condition number, say of slightly above 10^4, a factor of a thousand.

Multi-Modal Functions On the multi-modal 10-D Rastrigin function PSO is able to locate the optimum with a large swarm size reliably in the separable case, where it is approximately ten times slower than the ES with a large population size. In the rotated case the success probability for PSO drops to roughly 1% while the ES performs invariant under rotations. No trade-off between performance on unimodal versus multi-modal functions can therefore be reported.

A Common Concept A close parallel of the velocity update in PSO and the update of an evolution path in ESs is elaborated. While the updates are virtually identical in their concept and formulation, two aspects are different. The velocity is utilized in an intuitive directional way, while the evolution path is utilized point symmetrically, thereby not effecting the mean displacement. The time constant for the evolution path increases proportionally with the search space dimension while the time constant for the velocity is independent of the search space dimension about 3.6.

No Free Lunch On the non-separable functions of our small test function set, PSO is consistently outperformed by a contemporary evolution strategy. Our benchmark functions were specifically selected to test the behavior on non-separable, ill-conditioned problems, where dependencies between the design parameters play a decisive role. One might argue that this implies PSO must be better on other benchmark functions or on real world problems as the no free lunch (NFL) theorem seems to implicate [18, 26]. This would render our results fairly meaningless and would indeed be true, if the necessary conditions for NFL would hold, namely, if the set of all considered functions were closed under permutation [20]. Fortunately, we do not have much of a reason to believe that this condition holds on any interesting set of test and/or real world problems (including all interesting or all ever considered problems) [13, 12]. Additionally, in continuous domain, NFL seems not to be available at all [3]. Summing up, we do not see that NFL theorems can have any grave impact on the relevance of our empirical findings.

Implications We highly appreciate the effort to provide a standard version of PSO, Standard PSO 2006. We believe that a standard algorithm with standard parameter settings that works well over a wide range of objective functions is an essential feature for the applicability of a search algorithm and we regard Standard PSO 2006 as a step forward along this line. We believe our results are largely independent of specific parameter settings and the implications of our work are not limited to Standard PSO 2006. In order to solve ill-conditioned, non-separable problems the swarm shape must be elongated, but not aligned to a coordinate axis. This is impeded by coordinate wise independent sampling used in most PSO variants. Recognizing the importance of invariance properties, Standard PSO 2007 provides an option for a rotation of the random step in order to make it less sensitive to rotations. This will make the performance of Standard PSO 2007, first of all, more predictable. Additionally, a rotation procedure most likely impedes the sampling of an elongated distribution shape—preventing the swarm shape from getting remarkably elongated all together. We conjecture that in our experimental set-up, results on separable functions will become similar to those on the rotated ones, but not vice versa.

Final Word We believe that ill-conditioning is a prevalent property of difficult real-world problems and that there is no trivial solution to searching non-separable, ill-conditioned problems efficiently. Achieving rotational invariance alone is not sufficient as isotropic algorithms necessarily perform poorly on ill-conditioned problems. First, rapid and effective adaptivity must be provided such that an elongated, narrow swarm shape can be realized. Second, reliability

must be assured in that the swarm does not collapse into low-dimensional subspaces. Finally, rotational invariance confirms this behavior for any coordinate system rotation and constitutes its generalizability. The invariance-diversity dilemma needs yet to be solved for PSO.

References

[1] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In Proceedings of the IEEE Congress on Evolutionary Computation, 2005.

[2] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In Proceedings of the IEEE Congress on Evolutionary Computation, 2005.

[3] A. Auger and O. Teytaud. Continuous lunches are free! In Proceedings of the 9th annual conference on Genetic and evolutionary computation, pages 916–922. ACM Press New York, NY, USA, 2007.

[4] M. Clerc and J. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. Evolutionary Computation, IEEE Transactions on, 6(1):58–73, 2002.

[5] B. Efron and R. Tibshirani. An Introduction to the Bootstrap. Chapman & Hall/CRC, 1993.

[6] V. Feoktistov. Differential Evolution: In Search of Solutions. Optimization and Its Applications. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.

[7] Sylvain Gelly, Sylvie Ruette, and Olivier Teytaud. Comparison-based Algorithms are Robust and Randomized Algorithms are Anytime. Evolutionary Computation Journal, 15(4):411–434, 2007.

[8] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In Xin Yao et al., editors, Parallel Problem Solving from Nature - PPSN VIII, LNCS 3242, pages 282–291. Springer, 2004.

[9] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation, 11(1):1–18, 2003.

[10] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolutionstrategies: the covariance matrix adaptation. In Proceedings of the IEEE Congress on Evolutionary Computation, pages 312–317, 1996.

[11] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation, 9(2):159–195, 2001.

[12] C. Igel and M. Toussaint. On classes of functions for which No Free Lunch results hold. Information processing letters, 86(6):317–321, 2003.

[13] C. Igel and M. Toussaint. A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions. Journal of Mathematical Modelling and Algorithms, 3(4):313–322, 2004.

[14] J. Kennedy and R. Eberhart. Particle swarm optimization. In Neural Networks, 1995. Proceedings., IEEE International Conference on, volume 4, pages 1942–1948, 1995.

[15] S. Kern, S.D. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms–a comparative review. Natural Computing, 3(1):77–112, 2004.

[16] S. Kok, DN Wilke, and AA Groenwold. Recent Developments of the Particle Swarm Optimization Algorithm. In Proc of International Conference on Computational Intelligence, volume 2005, 2005.

[17] V. Miranda. Evolutionary algorithms with particle swarm movements. In Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on, 2005.

[18] NJ Radcliffe and PD Surry. Fundamental Limitations on Search Algorithms: Evolutionary Computing in Perspective. Lecture Notes in Computer Science, Springer Verlag, New York, NY, 1000:275–291, 1995.

[19] R. Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. BioSystems, 39(3):263–278, 1996.

[20] C. Schumacher, MD Vose, and LD Whitley. The no free lunch and problem description length. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), pages 565–570. Morgan Kaufmann, 2001.

[21] Yun-Wei Shang and Yu-Huang Qiu. A note on the extended rosenbrock function. Evol. Comput., 14(1):119–126, 2006.

[22] Y. Shi and R. Eberhart. Modified particle swarm optimizer. In The 1998 IEEE International Conference on Evolutionary Computation, ICEC'98, pages 69–73, 1998.

[23] Y. Shi, RC Eberhart, E.D.S.I.T. Center, and IN Carmel. Empirical study of particle swarm optimization. In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, volume 3, 1999.

[24] D.N. Wilke, S. Kok, and A.A. Groenwold. Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on diversity. Int. J. Numer. Meth. Engng, 70:962–984, 2007.

[25] D.N. Wilke, S. Kok, and A.A. Groenwold. Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance. Int. J. Numer. Meth. Engng, 70:985–1008, 2007.

[26] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. Evolutionary Computation, IEEE Transactions on, 1(1):67–82, 1997.

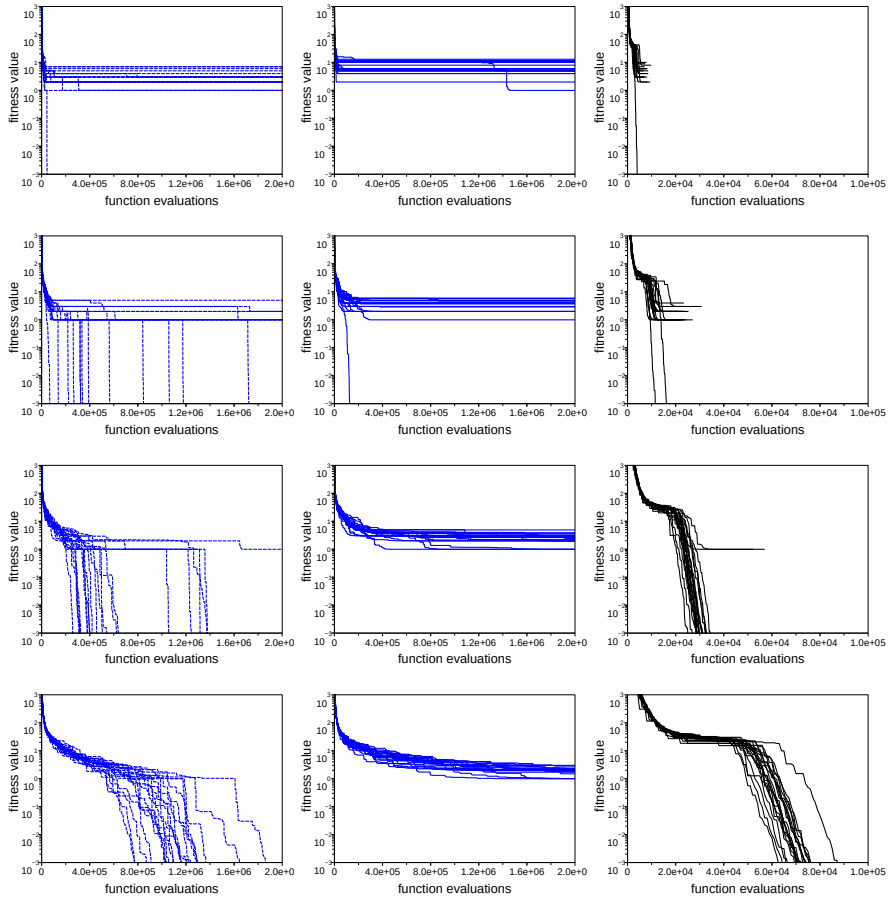

Figure 11: Rastrigin function in 10-D: time evolution of the objective function value of all 21 runs. Left column: S-PSO on the non-rotated Rastrigin function until up to 2×10^6 function evaluations. Middle column: S-PSO on the rotated Rastrigin function until up to 2×10^6 function evaluations. Right column: CMA-ES on the rotated Rastrigin function until up to 10^5 function evaluations. Swarm/population size of 30, 100, 300, 1000 from top to bottom. The CMA-ES was terminated before the maximum number of function evaluations was reached also in the unsuccessful trials.

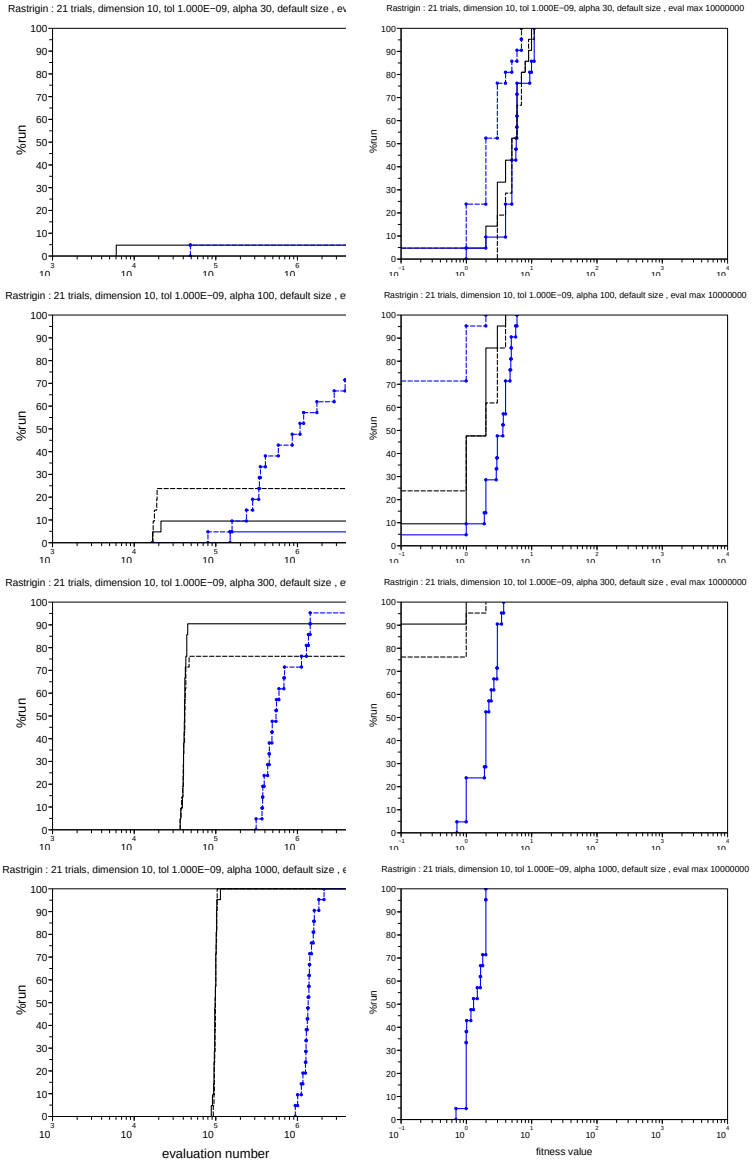

Figure 12: Rastrigin function in 10-D: empirical cumulative distribution (ECDF) of the number of function evaluations to reach the target function value 10^{-9} (left column) and ECDF of the best function value after the maximum number of function evaluations is exceeded (right column). Swarm/population size of 30, 100, 300, 1000 from top to bottom. S-PSO with small bullets, CMA-ES without; solid lines: rotated, dashed lines: non-rotated function.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399