High Dimensions and Heavy Tails for Natural Evolution Strategies

Tom Schaul
IDSIA, University of Lugano
Manno, Switzerland
tom@idsia.ch

Tobias Glasmachers
IDSIA, University of Lugano
Manno, Switzerland
tobias@idsia.ch

Jürgen Schmidhuber
IDSIA, University of Lugano
Manno, Switzerland
juergen@idsia.ch

ABSTRACT

The family of natural evolution strategies (NES) offers a principled approach to real-valued evolutionary optimization by following the natural gradient of the expected fitness on the parameters of its *search distribution*. While general in its formulation, existing research has focused only on multivariate Gaussian search distributions. We address this shortcoming by exhibiting problem classes for which other search distributions are more appropriate, and then derive the corresponding NES-variants.

First, we show how simplifying NES to separable distributions reduces its complexity from $\mathcal{O}(d^3)$ to $\mathcal{O}(d)$, and apply it to problems of previously unattainable dimensionality, recovering lowest-energy structures on the Lennard-Jones atom clusters and state-of-the-art results on neuro-evolution benchmarks. Second, we develop a new, equivalent formulation based on invariances, which allows us to generalize NES to heavy-tailed distributions, even if their variance is undefined. We then investigate how this variant aids in overcoming deceptive local optima.

Categories and Subject Descriptors

[**Evolution Strategies and Evolutionary Programming**]

General Terms

Algorithms

Keywords

evolution strategies, natural gradient, black-box optimization, global optimization

1. INTRODUCTION

Research on Evolution Strategies (ES) for real-valued black box search and optimization has traditionally focused on Gaussian search distributions of various kinds, ranging from the simplicity of rotation symmetric radial Gaussians to the flexibility of fully adaptive covariance matrices [1]. The family of Gaussian search distributions is the canonical choice for its convenient properties among distributions with existing first and second moments. Natural Evolution Strategies (NES [14, 3]) are a recent class of evolution strategies that adapt the search distribution by ascending the natural gradient of expected fitness. NES is a principled approach that offers the flexibility to adapt arbitrary classes of search distributions, without being restricted to continuous search spaces, as long as they have continuous parameters.

Previous research on NES has focused exclusively on multivariate Gaussian search distributions. In this respect it has been in line with state-of-the-art algorithms such as CMA-ES [8, 5]. In this article we demonstrate the flexibility of NES by extending it to various types of search distributions. The goal of these extensions is to overcome two different limitations of existing algorithms.

One type of shortcoming of the NES algorithm is its limited applicability to high-dimensional problems. This limitation has two facets. The computational complexity of a typical NES variant with fully adaptive Gaussian search distribution is $\mathcal{O}(d^3)$ operations per generation, where d in the dimensionality of the search space. A second effect is that the adaptation of $\mathcal{O}(d^2)$ parameters in the covariance matrix limits the sample efficiency of the adaptation scheme. Thus, adapting the full covariance matrix is not necessarily a good strategy for solving high-dimensional optimization problems. The remedy we suggest is to limit the search distribution to diagonal covariance matrices (as was done for CMA-ES in [9]). The resulting algorithm, separable NES (SNES), allows for linear time updates and makes the algorithm applicable to extremely high-dimensional problems – under the implicit assumption that the problem variables are mostly separable. Furthermore, this variant permits the straightforward extension of NES to separable but non-Gaussian search distributions.

To overcome a different type of limitation, we extend NES to heavy-tailed search distributions (e.g., the Cauchy distribution). In contrast to distributions with bounded variance, such distributions search the space with an emphasis on big jumps. An ES relying on search distributions with bounded variance is known to follow the so-called ‘global trend’ of the fitness landscape. This behavior can be understood as moving the high probability region of the search distribution to a related (typically shrunken) region with good average fitness[1], where the averaging takes place on the scale of the

GECCO '11, Dublin, Ireland

[1] With rank-based selection this amounts to focusing on a quantile, like fitness values better than the median fitness

current standard deviation. In contrast, for heavy-tailed distributions such a scale does not exist, due to a substantially increased chance of sampling huge variations. Thus, heavily tailed distributions are promising candidates for solving highly multi-modal problems that can not be solved solely by following a global trend, while maintaining the capability of ESs to descend into local optima.

The paper is organized as follows. We start with an introduction to the general NES scheme, including two recent variants, the exponential version (xNES, [3]) and the hill-climber version [2]. We then introduce SNES with its computationally efficient updates, and generalize NES to heavy-tailed search distributions. We evaluate the new algorithms on a broad collection of benchmark problems, illustrating both the usefulness of heavy-tailed distributions for global optimization, and the power of the SNES algorithm on the Lennard-Jones potential [13], which is high-dimensional and highly multi-modal at the same time.

2. NATURAL EVOLUTION STRATEGIES

Natural evolution strategies (NES) [14, 12, 11, 2, 3] are a class of evolutionary algorithms for real-valued optimization that maintain a search distribution, and adapt the distribution parameters by following the natural gradient of expected fitness. This avoids drawbacks of the plain gradient which are prone to slow or even premature convergence. Although relying exclusively on function evaluations, the resulting optimization behavior closely resembles second order optimization techniques. Just like CMA-ES [8, 5], NES algorithms are invariant under monotone transformations of the fitness function and linear transformations of the search space.

In each generation the algorithm samples a population of $\lambda \in \mathbb{N}$ individuals $\mathbf{z}_k \sim \pi(\mathbf{z}|\theta)$, $k \in \{1, \dots, \lambda\}$, i.i.d. from its search distribution, which is parameterized by θ, with the goal to maximize a fitness function $f : \mathbb{R}^d \to \mathbb{R}$. The expected fitness under the search distribution is

$$J(\theta) = \mathbb{E}_\theta[f(x)] = \int f(x)\ \pi(x \,|\, \theta)\ dx \ \ .$$

The gradient w.r.t. the parameters can be rewritten as

$$\begin{aligned}\nabla_\theta J(\theta) =& \nabla_\theta \int f(\mathbf{z})\ \pi(\mathbf{z} \,|\, \theta)\ dx \\ =& \mathbb{E}_\theta \left[f(\mathbf{z})\ \nabla_\theta \log \pi(\mathbf{z} \,|\, \theta) \right] \ \ ,\end{aligned}$$

(see [14] for the full derivation) from which we obtain the Monte Carlo estimate

$$\nabla_\theta J(\theta) \approx \frac{1}{\lambda} \sum_{k=1}^{\lambda} f(\mathbf{z}_k)\ \nabla_\theta \log \pi(\mathbf{z}_k \,|\, \theta)$$

of the search gradient. The key step then consists in replacing this gradient, pointing into the direction of (locally) steepest ascent w.r.t. the given parameterization, by the natural gradient

$$\widetilde{\nabla}_\theta J = \mathbf{F}^{-1} \nabla_\theta J(\theta) \ \ .$$

where $\mathbf{F} = \mathbb{E}\left[\nabla_\theta \log \pi\left(\mathbf{z}|\theta\right) \nabla_\theta \log \pi\left(\mathbf{z}|\theta\right)^\top \right]$ is the Fisher information matrix; leading to a straightforward scheme of natural gradient ascent for iteratively updating the search distribution

$$\theta \leftarrow \theta + \eta \widetilde{\nabla}_\theta J = \theta + \eta \mathbf{F}^{-1} \nabla_\theta J(\theta) \ \ ,$$

with learning rate parameter η. Note that this general formulation is applicable to arbitrary (parameterizable) search distributions. The sequence of 1) sampling an offspring population, 2) computing the corresponding Monte Carlo estimate of the fitness gradient, 3) transforming it into the natural gradient, and 4) updating the search distribution, constitutes one generation of NES.

For the case of the multi-variate Gaussian search distribution, a specific formulation was introduced in [3], called "exponential NES" (xNES), which removes the requirement of the costly inversion of $\mathbf{F}$, using a two-fold change of coordinate system. The exponential map $M \mapsto \exp(M) = \sum_{n=0}^{\infty} \frac{1}{n!} M^n$ for symmetric matrices is used to encode the covariance matrix, resulting in a multiplicative form of the covariance matrix update (see [3] for details). At the same time all updates are computed in a local "natural" coordinate system, with respect to which the current search distribution has zero mean and unit covariance, which results in the trivial Fisher matrix $\mathbf{F} = \mathbb{I}$.

In the resulting algorithm, the distribution parameters $\theta = \langle \boldsymbol{\mu}, \boldsymbol{\Sigma} \rangle$, where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector and $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ is the covariance matrix, are split canonically into three invariant components. This amounts to a (non-redundant) representation similar to CMA-ES, that is, we split off a global step size variable from the covariance matrix in the form $\boldsymbol{\Sigma} = \sigma^2 \cdot \mathbf{B}^\top \mathbf{B}$, with $\sigma \in \mathbb{R}^+$ and $\det(\mathbf{B}) = 1$. We write $\mathbf{A} = \sigma \cdot \mathbf{B}$ for a factor of the covariance matrix, fulfilling $\boldsymbol{\Sigma} = \mathbf{A}^\top \mathbf{A}$. We obtain the corresponding gradient components

$$\begin{aligned}\nabla_{\boldsymbol{\mu}} J &= \sum_{k=1}^{\lambda} f(\mathbf{z}_k) \cdot \mathbf{s}_k \\ \nabla_{\mathbf{M}} J &= \sum_{k=1}^{\lambda} f(\mathbf{z}_k) \cdot (\mathbf{s}_k \mathbf{s}_k^\top - \mathbb{I}) \\ \nabla_\sigma J &= \operatorname{tr}(\nabla_{\mathbf{M}} J)/d \\ \nabla_{\mathbf{B}} J &= \nabla_{\mathbf{M}} J - \nabla_\sigma J \cdot \mathbb{I} \ \ ,\end{aligned}$$

where samples are drawn from the standard multinormal distribution $\mathbf{s}_k \sim \mathcal{N}(0, \mathbb{I})$, before being mapped back into the original coordinate system $\mathbf{z}_k = \boldsymbol{\mu} + \sigma \mathbf{B}^\top \mathbf{s}_k$. The updates then become:

$$\begin{aligned}\boldsymbol{\mu} \leftarrow& \boldsymbol{\mu} + \eta_{\boldsymbol{\mu}} \cdot \sigma \mathbf{B} \cdot \nabla_{\boldsymbol{\mu}} J \\ \sigma \leftarrow& \sigma \cdot \exp(\eta_\sigma / 2 \cdot \nabla_\sigma J) \\ \mathbf{B} \leftarrow& \mathbf{B} \cdot \exp(\eta_{\mathbf{B}} / 2 \cdot \nabla_{\mathbf{B}} J) \ \ ,\end{aligned}$$

where $\eta_{\boldsymbol{\mu}}$, η_σ, and $\eta_{\mathbf{B}}$ denote learning rates for the different components (refer to [3] for further details).

If raw fitness values are used, the algorithm will be prone to getting stuck on plateaus and to systematically jumping over steep optima. Thus, *fitness shaping* [14] is used to normalize the fitness values by replacing them by rank-based utility values $u_k \in \mathbb{R}$, $k \in \{1, \dots, \lambda\}$. For this purpose we order the individuals by fitness, with $\mathbf{z}_{1:\lambda}$ denoting the best and $\mathbf{z}_{\lambda:\lambda}$ denoting the worst offspring. We then use the "fitness-shaped" gradient

$$\nabla_\theta J = \sum_{k=1}^{\lambda} u_k \cdot \nabla_{(\theta)} \log \pi(\mathbf{z}_{k:\lambda} \,|\, \theta)$$

under the current distribution.

to update the parameters of the search distribution. Typically, the utility values are either non-negative numbers that add to one, or a shifted variant with zero mean.

In addition to robustness, these utility values provide us with an elegant formalism to describe a (1+1) hill-climber within the same framework, by using different utility values, depending on success (see [2] for details, and also algorithm 2 which instantiates the principle).

3. SEPARABLE NES

Multi-variate normal distributions arguably constitute the most important class of search distributions in modern evolution strategies. However, adapting the full covariance matrix of the search distribution can be disadvantageous, particulary in high-dimensional search spaces, for two reasons.

For many problems it can be safely assumed that the computational costs are governed by the number of fitness evaluations. This is particularly true if such evaluations rely on expensive simulations. However, for applications where fitness evaluations scale gracefully with the search space dimension, the $\mathcal{O}(d^3)$ exponential NES update (due to the matrix exponential[2]) can dominate the computation. One such application is the evolutionary training of recurrent neural networks (i.e., neuroevolution), where the number of weights in the network can grow quadratically with the number of neurons n, resulting in a complexity of $\mathcal{O}(n^6)$ for a single NES update.

A second reason not to adapt the full covariance matrix in high dimensional search spaces is sample efficiency. The covariance matrix has $d(d{+}1)/2 \in \mathcal{O}(d^2)$ degrees of freedom, which can be a huge number in large dimensions. Obtaining a stable estimate of this matrix based on samples may thus require many (costly) fitness evaluations, in turn requiring very small learning rates. As a result, the algorithm may simply not have enough time to adapt its search distribution to the problem with a given budget of fitness evaluations. In this case, it may be advantageous to restrict the class of search distributions in order to adapt at all, even if this results in a less steep learning curve in the (then practically irrelevant) limit of infinitely many fitness evaluations.

The only two distinguished parameter subsets of a multi-variate distribution that do not impose the choice of a particular coordinate system onto our search space are the 'size' of the distribution, corresponding to the $(2d)$-th root of the determinant of the covariance matrix, and its orthogonal complement, the covariance matrix normalized to constant determinant (corresponding to all remaining shape properties of the distribution, but size) [2]. The first of these candidates results in a standard evolution strategy without covariance adaptation at all, which may indeed be a viable option in some applications, but is often too inflexible. The set of normalized covariance matrices, on the other hand, is not interesting because it is clear that the size of the distribution needs to be adjusted in order to ensure convergence to an optimum.

Thus, it has been proposed to give up some invariance properties of the search algorithm, and to adapt the class of search distribution with diagonal covariance matrices *in some predetermined coordinate system* [9]. Such a choice is justified in many applications where a certain degree of independence can be assumed among the fitness function parameters. It has even been shown in [9] that this approach can work surprisingly well even for highly non-separable fitness functions.

Restricting the class of search distributions to Gaussians with diagonal covariance matrix corresponds to restricting a general class of multi-variate search distributions to separable distributions

$$p(\mathbf{z} \,|\, \theta) = \prod_{i=1}^{d} \tilde{p}(\mathbf{z}_i \,|\, \theta_i) \ ,$$

where $\tilde{p}$ is a family of densities on the reals, and $\theta = (\theta_1, \dots, \theta_d)$ collects the parameters of all of these distributions. In most cases these parameters amount to $\theta_i = (\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$, where $\boldsymbol{\mu}_i \in \mathbb{R}$ is a position and $\boldsymbol{\sigma}_i \in \mathbb{R}^+$ is a scale parameter (i.e., mean and standard deviation, if they exist), such that $\mathbf{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \cdot \mathbf{s}_i \sim \tilde{p}(\cdot \,|\, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ for $\mathbf{s}_i \sim \tilde{p}(\cdot \,|\, 0, 1)$.

Algorithm 1: Separable NES (SNES)

```
input: $f$, $\boldsymbol{\mu}_{init}$, $\boldsymbol{\sigma}_{init}$
repeat
    for $k = 1 \dots \lambda$ do
        draw sample $\mathbf{s}_k \sim \mathcal{N}(0, \mathbb{I})$
        $\mathbf{z}_k \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma}\mathbf{s}_k$
        evaluate the fitness $f(\mathbf{z}_k)$
    end
    sort $\{(\mathbf{s}_k, \mathbf{z}_k)\}$ with respect to $f(\mathbf{z}_k)$ and compute
    utilities $u_k$
    compute gradients   $\nabla_{\boldsymbol{\mu}} J \leftarrow \sum_{k=1}^{\lambda} u_k \cdot \mathbf{s}_k$
                        $\nabla_{\boldsymbol{\sigma}} J \leftarrow \sum_{k=1}^{\lambda} u_k \cdot (\mathbf{s}_k^2 - 1)$
    update parameters   $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \eta_{\boldsymbol{\mu}} \cdot \boldsymbol{\sigma} \cdot \nabla_{\boldsymbol{\mu}} J$
                        $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} \cdot \exp(\eta_{\boldsymbol{\sigma}}/2 \cdot \nabla_{\boldsymbol{\sigma}} J)$
until stopping criterion is met;
```

Obviously this allows us to sample new offspring in $\mathcal{O}(d)$ time. Since the adaptation of each component's parameters is independent, the strategy update step also takes only $\mathcal{O}(d)$ time. At the same time the number of parameters in the covariance matrix shrank from $d(d+1)/2 \in \mathcal{O}(d^2)$ to $d \in \mathcal{O}(d)$, which allows us to increase the learning rate $\eta_{\boldsymbol{\sigma}}$ by a factor of $d/3 \in \mathcal{O}(d)$. This choice has proven robust in practice [9].

Thus, the sacrifice of invariance, amounting to the selection of a distinguished coordinate system, allows for a linear time algorithm (per individual), that still maintains a reasonable amount of flexibility in the search distribution, and allows for a considerably faster adaptation of its parameters. The resulting NES variant, called *separable* NES (SNES), is illustrated in algorithm 1 for Gaussian search distributions. Note that each of the steps requires only $\mathcal{O}(d)$ operations. In the next section we extend this algorithm to other search distributions, without affecting its computational complexity.

4. HEAVY-TAILED NES

Natural gradient ascent and 'vanilla' gradient ascent in natural coordinates provide two equivalent views on the working principle of NES. In this section we introduce yet another

[2] Even if we sacrifice the exponential parameterization (which leads to other problems, see [3]) to perform more efficient updates, a cost of $\mathcal{O}(d^2)$ is unavoidable, already for sampling offspring.

interpretation, with the goal to extend NES to heavy-tailed distributions, in particular distributions with infinite variance, like the Cauchy distribution. The problem posed by these distributions within the NES framework is that they do not induce a Riemannian structure on the parameter space of the distribution via their Fisher information, which renders the information geometric interpretation of natural coordinates and natural gradient ascent invalid.

The most important types of search distributions have strong invariance properties, this includes multi-variate and separable heavy-tailed distributions. In this still very general case the NES principle can be derived solely based on invariance properties, without ever referring to information geometric concepts.

The direction of the gradient $\nabla_\theta J(\theta)$ depends on the inner product $\langle \cdot, \cdot \rangle$ in use, corresponding to the choice of a coordinate system or an (orthonormal) set of basis vectors. Thus, expressing a gradient ascent algorithm in arbitrary coordinates results in (to some extent) arbitrary and often sub-optimal updates. NES resolves this dilemma by relying on the natural gradient, which corresponds to the distinguished coordinate system (of the tangent space of the family of search distributions) corresponding to the Fisher information metric.

The natural coordinates of a multi-variate Gaussian search distribution turn out to be those local coordinates w.r.t. which the current search distribution has zero mean and unit covariance. This coincides with the coordinate system in which the invariance properties of multi-variate Gaussians are most apparent. This connection turns out to be quite general. We exploit this property systematically and apply it to distributions with infinite (undefined) Fisher information.

4.1 Groups of Invariances

The invariances of a search distribution can be expressed by a group $\mathcal{G}$ of (affine) linear transformations. Typically, $\mathcal{G}$ is a sub-group of the group of orthogonal transformations (i.e., rotations) w.r.t. a local coordinate system. For example, let $q : \mathbb{R}^d \to \mathbb{R}_0^+$ be the density of a rotation symmetric distribution (e.g., a Gaussian), then

$$p(\mathbf{z} \,|\, \boldsymbol{\mu}, \mathbf{A}) = \frac{1}{\det(\mathbf{A})} \cdot q\left(\mathbf{A}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right)$$

with $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\mathbf{A} \in \mathbb{R}^{d \times d}$, $\det(\mathbf{A}) \neq 0$, is the family of corresponding multi-variate distributions. Let $\mathcal{G}_{(\boldsymbol{\mu},\mathbf{A})}$ be the group of invariances of $p(\mathbf{z} \,|\, \boldsymbol{\mu}, \mathbf{A})$, that is, $\mathcal{G}_{(\boldsymbol{\mu},\mathbf{A})} = \left\{ g \,\middle|\, p(g(\mathbf{z}) \,|\, \boldsymbol{\mu}, \mathbf{A}) = p(\mathbf{z} \,|\, \boldsymbol{\mu}, \mathbf{A}) \,\forall \mathbf{z} \in \mathbb{R}^d \right\}$. We have $\mathcal{G}_{(0,\mathbb{I})} = \mathbb{O}_{\langle\cdot,\cdot\rangle}(\mathbb{R}^d) = \left\{ g \,\middle|\, \langle g(\mathbf{z}), g(\mathbf{z}')\rangle = \langle \mathbf{z}, \mathbf{z}' \rangle \,\forall \mathbf{z}, \mathbf{z}' \in \mathbb{R}^d \right\}$, where the right hand side is the group of orthogonal transformations w.r.t. an inner product, defined as the (affine) linear transformations that leave the inner product (and thus the properties induced by the orthonormal coordinate system) invariant. Here the inner product is the one w.r.t. which the density q is rotation invariant. For general $(\boldsymbol{\mu}, \mathbf{A})$ we have $\mathcal{G}_{(\boldsymbol{\mu},\mathbf{A})} = h \circ \mathcal{G}_{(0,\mathbb{I})} \circ h^{-1}$, where $h(\mathbf{z}) = \mathbf{A}\mathbf{z} + \boldsymbol{\mu}$ is the affine linear transformation corresponding to the current search distribution. In general, the group of invariances is only a subgroup of an orthogonal group, e.g., for a separable distribution q, $\mathcal{G}$ is the finite group generated by coordinate permutations and flips.

We argue that it is most natural to rely on a gradient or coordinate system which is *compatible* with the invariance properties of the search distribution in use. In other words, we should ensure the compatibility condition

$$\mathcal{G}_{(\boldsymbol{\mu},\mathbf{A})} \subset \mathbb{O}_{\langle\cdot,\cdot\rangle}(\mathbb{R}^d)$$

for the inner product $\langle \cdot, \cdot \rangle$ with respect to which we compute the gradient ∇J. This condition has a straight-forward connection to the natural coordinate system deduced in [3]: It is fulfilled by performing all updates in local coordinates, in which the current search distribution is expressed by the density $p(\cdot \,|\, 0, \mathbb{I}) = q(\cdot)$. In these coordinates, the distribution is already rotation symmetric by construction (or similar for separable distributions), where the rotation symmetry is defined in terms of the 'standard' inner product of the local coordinates. Local coordinates save us from the cumbersome explicit construction of an inner product that is left invariant by the group $\mathcal{G}_{(\boldsymbol{\mu},\mathbf{A})}$.

Note, however, that $q(\mathbf{z})$ and $q(\sigma \cdot \mathbf{z})$ have the same invariance properties. Thus, the invariance properties make only the gradient components $\nabla_{\boldsymbol{\mu}} J$ and $\nabla_{\mathbf{B}} J$ unique, but not the scale component $\nabla_\sigma J$. Luckily this does not affect the (1+1) hill-climber variant of NES, which relies on a success-based step size adaptation rule. Also note that this derivation of the NES updates works only for families of search distributions with strong invariance properties, while natural gradient ascent extends to much more general distributions, such as mixtures of Gaussians.

Algorithm 2: (1+1)-NES with multi-variate Cauchy search distribution

```
input: $f$, $\boldsymbol{\mu}_{init}$, $\boldsymbol{\Sigma}_{init} = \mathbf{A}^\top \mathbf{A}$
$f_{\text{best}} \leftarrow -\infty$
repeat
    draw sample  $\mathbf{s} \sim \mathcal{N}(0, \mathbb{I})$
                 $r \sim \pi_{Cauchy}(0, 1)$
                 $\mathbf{z} \leftarrow r\mathbf{A}^\top \mathbf{s} + \boldsymbol{\mu}$
    evaluate the fitness $f(\mathbf{z})$
    $(\mathbf{s}_1, \mathbf{s}_2) \leftarrow (0, \mathbf{s})$
    $(\mathbf{z}_1, \mathbf{z}_2) \leftarrow (\boldsymbol{\mu}, \mathbf{z})$
    calculate log-derivatives
    $\nabla_{\mathbf{M}} \log \pi(\mathbf{z}_k|\theta) \leftarrow \frac{1}{2}\left(\frac{d+1}{r^2+1}\mathbf{s}_k\mathbf{s}_k^\top - \mathbb{I}\right)$
    if $f(\mathbf{z}) > f_{best}$ then
        update mean $\boldsymbol{\mu} \leftarrow \mathbf{z}$
        $f_{\text{best}} \leftarrow f(\mathbf{z})$
        $\mathbf{u} \leftarrow (-4, 1)$
    else
        $\mathbf{u} \leftarrow (\frac{4}{5}, 0)$
    end
    $\nabla_{\mathbf{M}} J \leftarrow \frac{1}{2}\sum_{k=1}^{2} \nabla_{\mathbf{M}} \log \pi(\mathbf{z}_k|\theta) \cdot u_k$
    $\mathbf{A} \leftarrow \mathbf{A} \cdot \exp\left(\frac{1}{2}\eta_{\mathbf{A}} \nabla_{\mathbf{M}} J\right)$
until stopping criterion is met;
```

4.2 Cauchy Distributions

Given these results, NES, formulated in local coordinates, can be used with heavy-tailed search distributions without modification. This applies in particular to the (1+1) hill-climber, which is the most attractive choice for heavy-tailed search distributions, because when the search distribution

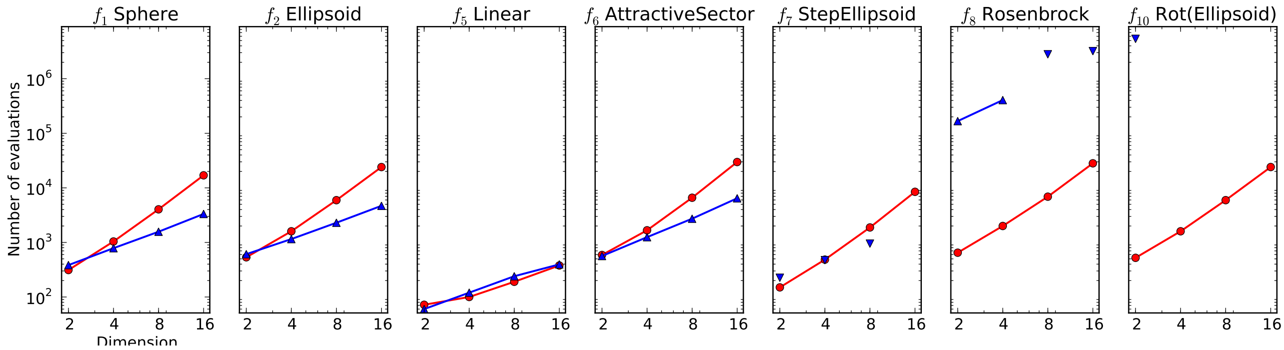

Figure 1: Comparison of the performance of xNES (red circles) and SNES (blue triangles) on a subset of the unimodal BBOB benchmark functions. The log-log plots show the median number of evaluations required to reach the target fitness 10^{-8}, for problem dimensions ranging from 2 to 16 (over 20 runs). The inverted triangles indicate cases where SNES converged to the optimum in less than 90% of the runs.

converges to a local optimum and a better optimum is located by a mutation, then averaging this step over the offspring population will usually result in a sub-optimal step that stays within the same basin of attraction. In contrast, a hill-climber can jump straight into the better basin of attraction, and can thus make better use the specific advantages of heavy-tailed search distributions.

Of course, the computation of the vanilla gradient changes depending on the distribution in use. Once this gradient is computed in the local coordinate system respecting the invariances of the current search distribution, it can be used for updating the search parameters $\boldsymbol{\mu}$ and $\mathbf{B}$ without further corrections like multiplying with the (in general undefined) inverse Fisher matrix. For the multi-variate Cauchy distribution we have

$$q(\mathbf{s}) = \frac{\Gamma((d+1)/2)}{\pi^{(d+1)/2}} \cdot \left(\|\mathbf{s}\|^2 + 1\right)^{-(d+1)/2} \quad ,$$

which results in the gradient components

$$\begin{aligned}\nabla_{\boldsymbol{\delta}} J =& \frac{d+1}{\|\mathbf{s}\|^2+1} \cdot \mathbf{s} \\ \nabla_{\mathbf{M}} J =& \frac{d+1}{2\cdot(\|\mathbf{s}\|^2+1)} \cdot \mathbf{s}\mathbf{s}^\top - \frac{1}{2}\cdot\mathbb{I} \ .\end{aligned}$$

The full NES hill-climber with multi-variate Cauchy mutations is provided in algorithm 2.

5. EXPERIMENTS

We now proceed to empirically validate the new algorithms. In the first part of this section we evaluate the performance of SNES on a whole collection of benchmarks. Then we address the question of global optimization and to what degree a heavy-tailed search distribution (namely multivariate Cauchy) can alleviate the problem of getting stuck in local optima.

If not mentioned otherwise, in the setup of the algorithm we use the following parameters: population size $\lambda = 4 + \lfloor 3\log(d)\rfloor$, learning rates $\eta_{\boldsymbol{\mu}} = 1$, $\eta_\sigma = \eta_{\mathbf{B}} = \eta_{\mathbf{A}} = \frac{(9+3\log(d))}{5d\sqrt{d}}$ (all in line with [3]) and the increased $\eta_{\boldsymbol{\sigma}} = \frac{(3+\log(d))}{5\sqrt{d}}$ for SNES (see section 3). The five algorithm variants that are evaluated below are xNES (as in [3]), (1+1)-NES-Gauss (as in [2]), (1+1)-NES-Cauchy (as in algorithm 2), SNES (as in algorithm 1) and its hill-climber variant (1+1)-SNES (pseudocode not shown). A Python implementation of all these is available within the open-source machine learning library PyBrain [10].

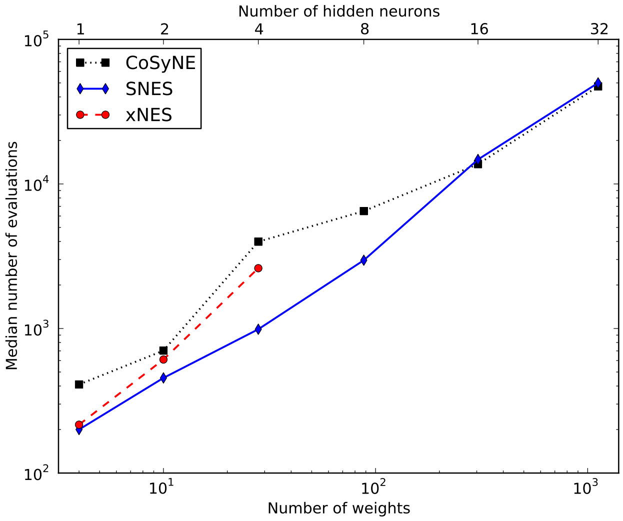

Figure 2: Median number of evaluations required to solve the non-Markovian pole-balancing task (over 100 runs), with increasing number of neurons (and corresponding number of weights). We limited the runtime to one hour per run, which explains why no results are available for xNES on higher dimensions (cubic time complexity). The fact that SNES quickly outperforms xNES, also in number of function evaluations, indicates that the benchmark is (sufficiently close to) separable, and it is unnecessary to use the full covariance matrix. For reference we also plot the results of the previously best performing algorithm CoSyNE [4] (population size 40).

5.1 Separable NES

First, we evaluate SNES on a subset of the unimodal benchmark problems from the BBOB framework [7, 6]. These

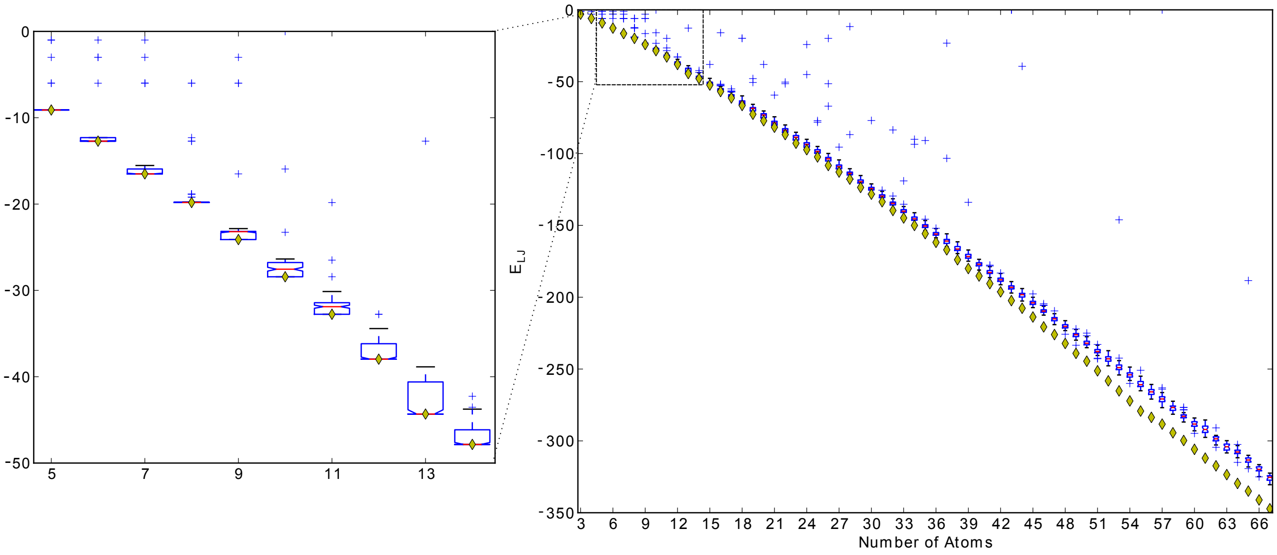

Figure 3: Performance of (1+1)-SNES on the Lennard-Jones benchmark for atom clusters ranging from 3 to 67 atoms (corresponding to problem dimensions d of 9 to 201). The yellow diamonds indicate the best known configurations (taken from [13]), and the box-plots show upper and lower quartile performance (the red line being the median) of SNES, over 100 runs. The inset is a zoom on the behavior in small dimensions, where SNES succeeds in locating the true optimum in a large fraction of the runs.

benchmarks test the capability of SNES to descend quickly into local optima, a key property of most evolution strategies. The results in figure 1 show how SNES dominates when the function is separable (f_1 through f_6), and converges much slower than xNES in non-separable benchmarks, as expected. In particular, on the rotated ellipsoid function f_{10}, which is designed to make separable methods fail, SNES requires 4 orders of magnitude more evaluations. In dimensions $d > 2$ it fails completely because the resolution of double precision numbers is insufficient for this task.

In the second experiment, we show how SNES is well-suited for neuroevolution problems because they tend to be high-dimensional, multi-modal, but with highly redundant global optima (there is not a unique set of weights that defines the optimal behavior). In particular, we run it on non-Markovian double pole balancing, a task which involves balancing two differently sized poles hinged on a cart that moves on a finite track. The single control consists of the force F applied to the cart and observations include the cart's position and the poles' angles, but no velocity information, which makes this task partially observable. The controller is represented by a simple recurrent neural network, with three inputs, (position x and the two poles' angles β_1 and β_2), and a variable number n of tanh units in the output layer, which are fully connected (recurrently), resulting in a total of $n(n+3)$ weights to be optimized. The activation of the first of these recurrent neurons directly determines the force to be applied. We use the implementation found in [10]. An evaluation is considered a success if the poles do not fall over for $100,000$ time steps.

In practical scenarios we cannot know the best network size a priori, and thus the conservative is choice of overestimating the required number of neurons. An algorithm that graciously scales with problem dimension is therefore highly desirable, and we find that SNES (unlike xNES) is exhibiting precisely that behavior.We experimented with recurrent layers of sizes $n = 1$ to $n = 32$ (corresponding to between 4 and 1120 weights). Figure 2 shows the results, which are in fact state-of-the-art for SNES, outperforming the previously best algorithm CoSyNE [4] by a factor 2 on small and medium dimensions.

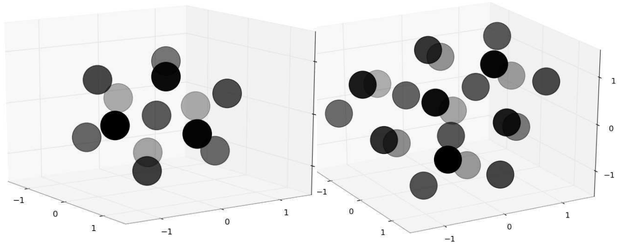

Figure 4: Illustration of the best configuration found for 13 atoms (symmetric, left), and 22 atoms (asymmetric, right).

As our third benchmark for illustrating the performance of (1+1)-SNES we chose the widely studied problem of minimizing the Lennard-Jones atom cluster potentials, which is known for being extremely multi-modal [13]. It consists of finding that configuration of N atoms which minimizes the potential energy function

$$E_{LJ} \propto \sum_{i,j \leq N} \left[\left(\frac{1}{r_{ij}} \right)^{12} - \left(\frac{1}{r_{ij}} \right)^{6} \right] ,$$

where r_{ij} is the distance between atoms i and j (see also figure 4 for an illustration). For the setup here, we initialized $\boldsymbol{\mu}$ near 0 the step-sizes at $\boldsymbol{\sigma}_i = 0.01$ to avoid jumping into a local optimum in the fist generation. The results are plotted in figure 3, showing how SNES scales convincingly

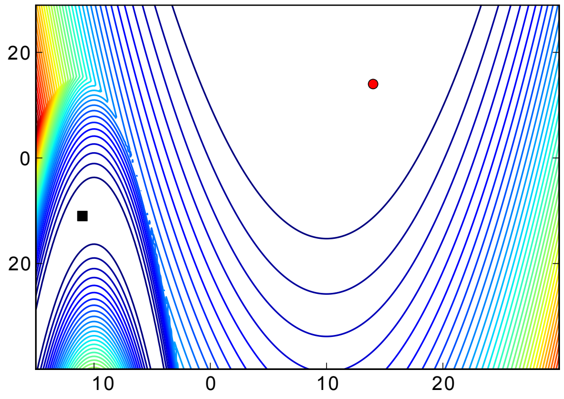

Figure 5: Contour plot of the 2-dimensional Double-Rosenbrock function, illustrating its deceptive double-funnel structure. The global structure leads the search to the local optimum ((14,14), red circle), whereas the true optimum ((-11,-11), black square) is located in the smaller valley.

to hundreds of parameters (each run up to 500d function evaluations).

5.2 Heavy Tails and Global Optimization

We illustrate the power of using a heavy-tailed search distribution on the synthetic benchmark function

$$f_{2Rosen}(\mathbf{z}) = \min\left\{ f_8(-\mathbf{z}-10), 5 + f_8\left(\frac{\mathbf{z}-10}{4}\right)\right\} \quad ,$$

where f_8 is the well-known Rosenbrock function [7]. Our variant has a deceptive double-funnel structure, with a large valley containing a local optimum and a smaller but deeper valley containing the global optimum. The global structure will tend to guide the search towards the local optimum (see also figure 5 for an illustration). For this experiment, the search distribution is initialized at mid-distance between the two optima, and the initial step-size σ is varied. Figure 6 shows the proportion of runs that converge to the global optimum, instead of the (easier to locate) local one, comparing for a multivariate Cauchy and Gaussian (1+1)-NES.

The last experiment uses the following 'random-basin' benchmark function:

$$f_{rb}(\mathbf{z}) = 1 - \frac{9}{10} r\left(\left\lfloor \frac{\mathbf{z}_1}{10}\right\rfloor, \ldots, \left\lfloor \frac{\mathbf{z}_d}{10}\right\rfloor\right) - \frac{1}{10} r(\lfloor \mathbf{z}_1 \rfloor, \ldots, \lfloor \mathbf{z}_d \rfloor) \cdot \prod_{i=1}^{d} \sin^2(\pi \mathbf{z}_i)^{\frac{1}{20d}}$$

to investigate the degree to which a heavy-tail distribution can be useful when the objective function is highly multi-modal, but there is no global structure to exploit. Here $r : \mathbb{Z}^d \to [0,1]$ is a pseudo-random number generator, which approximates an i.i.d. uniformly random distribution for each tuple of integers, while still being deterministic, i.e., each tuple evaluates to the same value each time. In practice, we implement it as a Mersenne twister seeded with the hash-value of the integers. Further, to avoid axis-alignment, we rotate the function by multiplying with an orthonormal random $d \times d$ matrix (as in [14]).

One interesting property of this function is that each unit-sized hypercube is an "attractor" of a local optimum. Thus, while sampling points from one hypercube, an ES will contract its search distribution, making it harder to escape from that local optimum. Furthermore, the values of the local optima are uniformly distributed in $[0,1]$, and do not provide a systematic global trend (in contrast to the Rastrigin function). If the optimization results in a value of, say, 0.11, then we know that only 11% of the local optima are better than this.

Figure 8 shows the results: not surprisingly, employing the Cauchy distribution for search permits longer jumps, and thus enables the algorithm to find better local optima on average. The Cauchy version outperforms the Gaussian

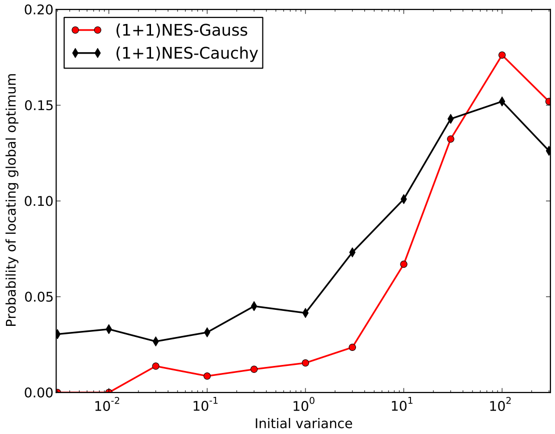

Figure 6: Empirical success probabilities (of locating the global optimum), evaluated over 200 runs on the f_{2Rosen} benchmark, while varying the size of the initial search distribution. The results clearly show the robustness of using a heavy-tailed distribution.

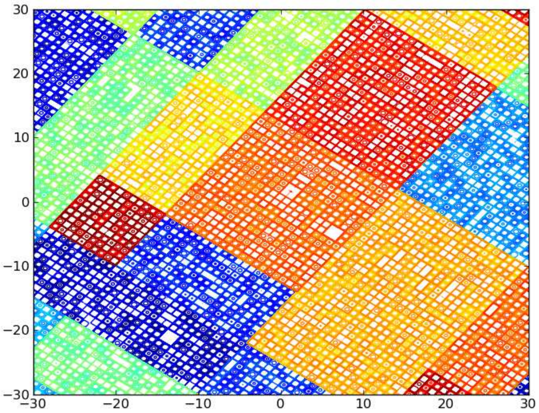

Figure 7: Contour plot of (one instantiation of) the deceptive global optimization benchmark function f_{rb}, in two dimensions. It is constructed to contain local optima in unit-cube-sized spaces, whose best value is uniformly random. In addition, it is superimposed on 10^d-sized regional plateaus, also with uniformly random value.

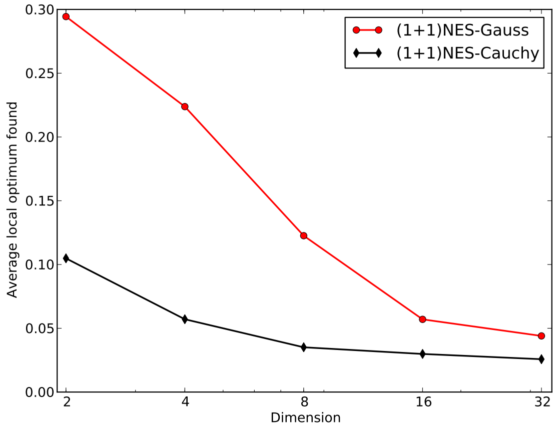

Figure 8: Value of the local optimum discovered on f_{rb} (averaged over 250 runs) as a function of problem dimension. Since the locally optimal values are uniformly distributed in $[0, 1]$, the results can equivalently be interpreted as the top percentile in which the found local optimum is located. E.g., on the 4-dimensional benchmark, NES with the Cauchy distribution tends to find one of the 6% best local optima, whereas employing the Gaussian distribution only leads to one of the best 22%.

version by a factor 2-3, depending on the problem dimension. Note that the improvement (for both distributions) is due to each unit-cube having exponentially more neighbor-cubes as dimension increases, and therefore an increasing chance that a relatively small jump will reach a better local optimum. At the same time, the adaptation of the step size is slowed down by the dimension-dependency of the learning rate, which leaves the algorithm more time to explore before it eventually converges into one of the local optima.

6. CONCLUSION

We addressed two shortcomings of NES by deriving variants that employ different search distributions. First we introduced separable NES (SNES), reducing the complexity of NES from $\mathcal{O}(d^3)$ to $\mathcal{O}(d)$, and making it applicable to problems of previously intractable dimensionality. We showed its performance improvement on separable benchmark functions, demonstrated how it finds lowest-energy structures on high-dimensional Lennard-Jones atom clusters, and achieved state-of-the-art results on the non-Markovian double-pole benchmark. Second, we developed a new, equivalent formulation of NES based on invariances instead of the natural gradient. This allowed us to generalize NES to heavy-tailed distributions, even if their variance is undefined. The resulting NES variant maintains a multi-variate Cauchy search distribution, and we validated that it does indeed find better local optima (and more often than its Gaussian counterpart) on two deceptively multi-modal benchmarks.

Acknowledgments.

We thank Sun Yi for constructive discussions and Faustino Gomez for helpful suggestions and for providing us with his CoSyNE data. This work was funded through the 7th framework program of the EU under grant number 231576 (STIFF project), and SNF grant number 200021-113364/1.

7. REFERENCES

[1] H.-G. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[2] T. Glasmachers, T. Schaul, and J. Schmidhuber. A Natural Evolution Strategy for Multi-Objective Optimization. In *Parallel Problem Solving from Nature (PPSN)*, 2010.

[3] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Genetic and Evolutionary Computation Conference (GECCO)*, Portland, OR, 2010.

[4] F. Gomez, J. Schmidhuber, and R. Miikkulainen. Accelerated Neural Evolution through Cooperatively Coevolved Synapses. *Journal of Machine Learning Research*, 2008.

[5] N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[6] N. Hansen and A. Auger. Real-parameter black-box optimization benchmarking 2010: Experimental setup, 2010.

[7] N. Hansen and S. Finck. Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions, 2010.

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

[9] R. Ros and N. Hansen. A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity. In R. et al., editor, *Parallel Problem Solving from Nature, PPSN X*, pages 296–305. Springer, 2008.

[10] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

[11] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient Natural Evolution Strategies. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2009.

[12] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic Search using the Natural Gradient. In *International Conference on Machine Learning (ICML)*, 2009.

[13] D. Wales and J. Doye. Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms. *The Journal of Physical Chemistry A*, 101(28):8, 1998.

[14] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural Evolution Strategies. In *Proceedings of the Congress on Evolutionary Computation (CEC08), Hongkong*. IEEE Press, 2008.