A Natural Evolution Strategy for Multi-Objective Optimization

Tobias Glasmachers, Tom Schaul, and Jürgen Schmidhuber
{tobias,tom,juergen}@idsia.ch

IDSIA, University of Lugano, Switzerland

Abstract. The recently introduced family of *natural* evolution strategies (NES), a novel stochastic descent method employing the natural gradient, is providing a more principled alternative to the well-known covariance matrix adaptation evolution strategy (CMA-ES). Until now, NES could only be used for single-objective optimization. This paper extends the approach to the multi-objective case, by first deriving a $(1+1)$ hillclimber version of NES which is then used as the core component of a multi-objective optimization algorithm. We empirically evaluate the approach on a battery of benchmark functions and find it to be competitive with the state-of-the-art.

1 Introduction

The last decade has seen a shift in research focus from single-objective to multi-objective optimization (MOO) [1, 2, 5, 7, 12, 13]. While many problems can very naturally be viewed as multi-objective (e.g., minimizing cost while simultaneously maximizing utility), they have traditionally been traded off into a single objective to be optimized. Numerous arguments have been put forward in favor of handling the multiple objectives explicitly, especially in the context of evolutionary computation. For one, the diversity of solutions found is larger than for single-objective optimization with fixed trade-offs, which in turn can improve over the single-objective optimization performance at its own game, as it may allow the search to circumnavigate local optima [7]. Furthermore, in many practical applications it is more advantageous to choose among the non-dominated solutions within the Pareto-front, rather than deciding upon a trade-off a priori and then maximizing it. Among the broad range of MOO algorithms that have been proposed (see e.g. [1] for an overview, omitted here for space reasons), approaches based on evolution strategies [5] are of particular interest for the present paper. They show how algorithms like the covariance matrix adaptation evolution strategy (CMA-ES [4, 6]), that shine on non-seperable optimization problems, can be utilized for MOO.

The recently introduced family of *natural evolution strategies* (NES [3,8–11]), consists in an optimization method that follows a sampled natural gradient of the expected fitness, and as such, provides a more principled alternative to CMA-ES. In this paper we combine the well-founded framework of NES with the proven approach of tackling MOO using evolution strategies. This both significantly broadens the applicability of NES, and establishes a novel, elegant MOO algorithm. Our contribution is two-fold: First, we turn the most recent NES algorithm into a hillclimber. Second, we use this hillclimber as a module of an evolutionary algorithm for multi-objective optimization, following an established scheme. We benchmark both algorithms against their CMA-ES counterparts and obtain competitive results.

2 Natural Evolution Strategies

Natural evolution strategies (NES) [3,8–11] are a class of evolutionary algorithms for real-valued optimization. They maintain a Gaussian search distribution with fully adaptive covariance matrix. The principal idea is to adapt the search distribution to the problem at hand by following the natural gradient of expected fitness. Although relying exclusively on function value evaluations, the resulting optimization behavior closely resembles second order optimization techniques. This avoids drawbacks of regular gradients which are prone to slow or even premature convergence. Just like CMA-ES [4, 6], NES algorithms are invariant under monotone transformations of the fitness function and linear transformations of the search space (given that the initial search distribution is transformed accordingly).

In this paper we build upon the most recent NES variant, *exponential NES* (xNES), first presented in [3]. We start with stating its working principles, which are needed later on to cleanly derive its hillclimber variant.

In each generation the algorithm samples a population of $n \in \mathbb{N}$ individuals $x_i \sim \mathcal{N}(\mu, C)$, $i \in \{1, \dots, n\}$, i.i.d. from its search distribution, which is represented by the center $\mu \in \mathbb{R}^d$ and a factor $A \in \mathbb{R}^{d \times d}$ of the covariance matrix $C = AA^T$. These points are obtained by sampling $z_i \sim \mathcal{N}(0, I)$ and setting $x_i = \mu + A \cdot z_i$. In this paper, I always denotes the d-dimensional unit matrix. Let $p(x \,|\, \mu, A)$ denote the density of the search distribution $\mathcal{N}(\mu, AA^T)$. Then,

$$J(\mu, A) = \mathbb{E}[f(x) \,|\, \mu, A] = \int f(x) \; p(x \,|\, \theta) \; dx$$

is the expected fitness under the current search distribution. The so-called 'log-likelihood trick' enables us to write

$$\begin{aligned} \nabla_{(\mu,A)} J(\mu, A) &= \int \Big[f(x) \; \nabla_{(\mu,A)} \log(p(x \,|\, \mu, A)) \Big] \; p(x \,|\, \mu, A) \; dx \\ &\approx \frac{1}{n} \sum_{i=1}^{n} f(x_i) \; \nabla_{(\mu,A)} \log(p(x \,|\, \mu, A)) \ . \end{aligned}$$

Using raw fitness values endangers the algorithm to get stuck on plateaus and to systematically overjump steep optima. Thus, *fitness shaping* [11] is used to normalize the fitness values by shaping them into rank-based utility values $u_i \in \mathbb{R}$, $i \in \{1, \dots, n\}$. For this purpose we order the individuals by fitness,

Algorithm 1: The xNES Algorithm

Input: $d \in \mathbb{N}$, $f : \mathbb{R}^d \to \mathbb{R}$, $\mu \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$
$\sigma \leftarrow \sqrt[d]{|\det(A)|};\ B \leftarrow A/\sigma$
while *stopping condition not met* **do**
  for $i \in \{1, \dots, n\}$ **do** $z_i \leftarrow \mathcal{N}(0, I);\ x_i \leftarrow \mu + \sigma B \cdot z_i$
  sort $\{(z_i, x_i)\}$ with respect to $f(x_i)$
  $G_\mu \leftarrow \sum_{i=1}^n u_i \cdot z_i$
  $G_A \leftarrow \sum_{i=1}^n u_i \cdot (z_i z_i^T - I);\ G_\sigma \leftarrow \mathrm{tr}(G_A)/d;\ G_B \leftarrow G_A - G_\sigma \cdot I$
  $\mu \leftarrow \mu + \eta_\mu \cdot \sigma B \cdot G_\mu;\ \sigma \leftarrow \sigma \cdot \exp(\eta_\sigma \cdot G_\sigma);\ B \leftarrow B \cdot \exp(\eta_B \cdot G_B)$
end

with $x_{1:n}$ denoting the best and $x_{n:n}$ denoting the worst offspring. We then use the "fitness-shaped" gradient $G = \sum_{i=1}^n u_i \cdot \nabla_{(\mu,A)} \log(p(x_{i:n} \mid \mu, A))$ 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.

The xNES algorithm introduces a number of novel techniques for its updates. In each step, the coordinate system is transformed such that the search distribution has zero mean and unit variance. This results in the Fisher information matrix being the unit matrix and the natural gradient coinciding with the 'standard' gradient. 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).

The parameters (μ, A) of the distribution can be split canonically into 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 $A = \sigma \cdot B$, with $\det(B) = 1$. We obtain the corresponding gradient components

$$G_\mu = \sum_{i=1}^n u_i \cdot z_i \qquad G_A = \sum_{i=1}^n u_i \cdot (z_i z_i^T - I)$$

with sub-components $G_\sigma = \mathrm{tr}(G_A)/d$ and $G_B = G_A - G_\sigma \cdot I$ (refer to [3] for the full derivation).

Let η_μ, η_σ, and η_B denote learning rates for the different parameter components. Putting everything together, the resulting xNES update rules for the search distribution read

$$\mu \leftarrow \mu + \eta_\mu \cdot \sigma B \cdot G_\mu \qquad \sigma \leftarrow \sigma \cdot \exp(\eta_\sigma \cdot G_\sigma) \qquad B \leftarrow B \cdot \exp(\eta_B \cdot G_B)\ .$$

The full xNES algorithm is summarized in Algorithm 1.

As indicated earlier, xNES is closely related to CMA-ES. However, conceptually xNES constitutes a much more principled approach to covariance matrix adaptation. This is because the updates of all parts of the search distribution, center, global step size, and full covariance matrix, result from the same principle of natural gradient descent.

Algorithm 2: $(1+1)$-xNES

Input: $d \in \mathbb{N}$, $f : \mathbb{R}^d \to \mathbb{R}$, $\mu \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$

```
σ ← 1
while stopping condition not met do
    z ← N(0, I)
    x ← μ + σA · z
    if f(x) is better than f(μ) then
        G_μ ← z
        G_A ← zz^T − I
        μ ← μ + 1 · A · G_μ
        A ← A · exp(η_A · G_A)
        σ ← σ · exp(η_σ^+)
    else σ ← σ / exp(η_σ^−)
end
```

Algorithm 3: $(1+1)$-xNES with natural gradient descent

Input: $d \in \mathbb{N}$, $f : \mathbb{R}^d \to \mathbb{R}$, $\mu \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$

```
while stopping condition not met do
    z ← N(0, I);  x ← μ + A · z
    if f(x) is better than f(μ) then
        succ ← +;  z_{1:2} ← z;  z_{2:2} ← 0
    else
        succ ← −;  z_{1:2} ← 0;  z_{2:2} ← z
    G_μ ← Σ_{i=1}^{2} u_i^{(μ)} · z_{i:2}
    G_A ← Σ_{i=1}^{2} u_i^{(A,succ)} (z_{i:2} z_{i:2}^T − I)
    μ ← μ + 1 · A · G_μ
    A ← A · exp(η_A · G_A)
end
```

3 An Elitist Variant for the NES Family

In this section we introduce $(1+1)$-xNES, a hillclimber variant of xNES. Our goal is to use this algorithm as a building block for a multi-objective optimization scheme, in analogy to the development of the $(1+1)$-CMA-ES. The main motivation for this work is that $(1+1)$-xNES is conceptually simpler and more unified than $(1+1)$-CMA-ES. We apply a number of techniques to xNES that were used to derive $(1+1)$-CMA-ES from its population-based variant. In a second step we show that the resulting algorithm can be derived from the NES principle of following the natural fitness gradient.

The resulting algorithm implements the following principles to adapt its search strategy (as usual, an offspring is considered successful if its fitness is better than the fitness of the parent):

1. A successful offspring becomes the center of the search distribution.
2. Sampling a successful offspring results in a covariance matrix update.
3. Global step size adaptation is used to sustain a success rate of about 1/5.

The elitist $(1+1)$-xNES algorithm, stated in Algorithm 2, incorporates the above principles into the xNES algorithm in a straightforward way. It is designed such that its state is completely determined by its current search distribution. We use a global step size σ and a factor A to represent the covariance matrix $C = \sigma^2 \cdot A^T A$. We stick to this redundant representation for the sake of clarity, as it allows us to separate the mechanisms of xNES-style covariance matrix adaptation and success rule-based step size adaptation. Also note that the learning rate for the center has been fixed to one in order to satisfy the elitism rule. We set the other learning rates to $\eta_A = 1/4 \cdot d^{1.5}$, $\eta_\sigma^- = 1/5 \cdot d^{1.5}$, and $\eta_\sigma^+ = d^{1.5}$. The form of the dependency of the learning rates on the problem dimension is inspired by the learning rates of xNES, divided by its population size.

In the special case of $(1+1)$-xNES the matrix exponential in the covariance matrix update can be computed analytically (that is, without resorting to iterative matrix decomposition techniques) and in quadratic time. This is a consequence of the special form $M = v \cdot zz^T + w \cdot I$ of the argument: We exploit its eigen-decomposition, which consists of a one-dimensional eigenspace along z with eigenvalue $v \cdot \|z\|^2 + w$, while the space orthogonal to z forms an eigenspace with eigenvalue w. With the definitions $S = (1/\|z\|^2) \cdot zz^T$ and $R = I - S$ we obtain $\exp(M) = \exp(v) \cdot R + \exp(v \cdot \|z\|^2 + w) \cdot S$, which can be computed in $\mathcal{O}(d^2)$ operations. The decisive advantage of this computation is numerical stability, even if the time per generation remains cubic (matrix multiplication).

From a conceptual point of view Algorithm 2 is still unsatisfactory, because the step size adaptation mechanism is not derived from the NES principle of updating the search distribution by following the natural gradient of expected fitness. Fortunately, the NES update scheme (and fitness shaping in particular) is flexible enough to cover the elitist case, including the success-based update rule, as we will see in the following.

In the $(1+1)$-selection scheme we need to assign a rank-based utility value not only to the offspring, but also to the parent, giving us an additional degree of freedom. Using different utility values for different parts of the $(1+1)$-xNES update (analogous to using different learning rates) allows us to derive the full update rule from the principle of natural gradient descent. Note that multiplicative factors in the learning rate and the utility values are exchangable, because they have the exact same effect.

The simplest case is the update of the center μ, which is completely determined by the elitism rule. This leaves us with a single choice, amounting to $(u_1^{(\mu)}, u_2^{(\mu)}) = (1, 0)$ for the utility values. Note that in this notation the utility value u_1 automatically refers to the better individual, such that it may correspond to either parent or offspring. This reflects the intuitive notion that in the $(1+1)$-selection scheme only the better individual is of use (has positive utility), while the worse individual is discarded (has zero utility).

The covariance matrix update is a bit more involved, as here the utility values are success dependent. In the simpler case where the offspring is not successful (the **else** part in Algorithm 2), it does not have an impact on the update, and the corresponding utility value is $u_2^{(A,-)} = 0$. Interestingly, we can use the utility of the parent to encode the shrinking of the global step size. The calculation

$$-\eta_\sigma^- \cdot I = \eta_A \cdot G_A = \eta_A \cdot \Big(u_1^{(A,-)} \cdot (00^T - I) + u_2^{(A,-)} \cdot (zz^T - I) \Big)$$

shows that the choice $u_1^{(A,-)} = \eta_\sigma^- / \eta_A$ does the job (with $0 \in \mathbb{R}^d$ denoting the zero vector). In case of the offspring being successful the analog calculation

$$\eta_\sigma^+ \cdot I + \eta_A \cdot (zz^T - I) = \eta_A \cdot G_A = \eta_A \cdot \Big(u_1^{(A,+)} \cdot (zz^T - I) + u_2^{(A,+)} \cdot (-I) \Big)$$

results in $u_1^{(A,+)} = 1$ and $u_2^{(A,-)} = -\eta_\sigma^+ / \eta_A$. These rules amount to a natural adaptation of the notion of utility to the $(1+1)$ elitist selection scheme. This means that Algorithm 2 is fully compatible with the principle of strategy adaptation by following the (utility shaped) gradient of expected fitness. It can be turned into the equivalent Algorithm 3.

4 Experimental Evaluation of $(1+1)$-xNES

We compared the $(1+1)$-xNES hillclimber to both xNES and $(1+1)$-CMA-ES a number of standard benchmark functions. We initialized the algorithms by drawing the initial center of the search distribution from a Gaussian with zero mean and unit variance, and setting the covariance matrix to I. Each algorithm was run until it reached the target fitness of 10^{-10} (-10^3 for the unbounded functions ParabR and SharpR), in which case the trial is counted as a success. A trial is said to fail if it reaches the maximum number of 10^7 iterations or shrinks the search distribution below numerical limits, which amounts to premature convergence. The results are shown in Figure 1.

The plots show $(1+1)$-xNES practically reaching the performance of $(1+1)$-CMA-ES on some benchmarks, while it falling behind on others, particularly in high dimensions. We attribute this to a better tuning of the parameters of (1+1)-CMA-ES, and the lack of evolution paths in xNES. Not surprisingly, the new elitist algorithm improves on the original xNES on nearly all unimodal problems studied here. Interestingly, both elitist algorithms have severe problems with the sharp ridge benchmark, on which they prematurely converge due to their too greedy strategy adaptation.

5 Multi-Objective NES

We now posess all the ingredients to construct a natural evolution strategy for multi-objective optimization, named MO-NES. Our construction follows the successful scheme developed in [5].

The MO-NES algorithm maintains a population of (1+1)-xNES hillclimbers, with the goal of maximally approximating the Pareto front. Each generation, the $N \in \mathbb{N}$ hillclimbers generate one offspring each. As in MOO there is no unique notion of success due to multiple contradicting objectives, the selection scheme of the individual hillclimbers becomes meaningless. Instead, we adopt the indicator-based selection scheme used in [5] which consists of two stages. Parents and offspring are merged into a single population and ranked according to (1) the dominance relation, and (2) an indicator (see, e.g, [12]) that permits aggregating the relative value of each individual within its front into a single number (in contrast to the m-dimensional fitness vector).

For this purpose, the population is split into fronts $F_1, \dots, F_k$ using non-dominated sorting. Within each front no individual weakly dominates another, while F_i weakly dominates F_j for $i < j$. Thus, in this notation the set F_1 consists of the non-dominated solutions. A secondary sorting criterion is needed to rank individuals within each front. In this study we use the S-measure or hypervolume contribution [13], which is the Laplace measure of the volume of

Algorithm 4: The MO-NES Algorithm

Input: $f : \mathbb{R}^d \to \mathbb{R}^m$, $(\mu_i, \sigma_i, A_i) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^{d \times d}$ for $i \in \{1, \dots, N\}$
while *stopping condition not met* **do**
  for $i \in \{1, \dots, N\}$ **do**
    $z_i \sim \mathcal{N}(0, I);\ \mu'_i \leftarrow \mu_i + \sigma_i A_i z_i;\ \sigma'_i \leftarrow \sigma_i;\ A'_i \leftarrow A_i$
  use non-dominated sorting to compute fronts $F_1, \dots, F_k$
  compute the S-measure of each individual within its front
  compute ranks $R_1, \dots, R_N, R'_1, \dots, R'_N \in \{1, \dots, 2N\}$
  for $i \in \{1, \dots, N\}$ **do**
    if $R'_i < R_i$ **then**
      $\sigma_i \leftarrow \sigma_i \cdot \exp(\eta_\sigma^+);\ \sigma'_i \leftarrow \sigma'_i \cdot \exp(\eta_\sigma^+);\ A'_i \leftarrow A'_i \cdot \exp(\eta_A \cdot [z_i z_i^T - I])$
    else
      $\sigma_i \leftarrow \sigma_i / \exp(\eta_\sigma^-);\ \sigma'_i \leftarrow \sigma'_i / \exp(\eta_\sigma^-)$
  copy best ranked N individuals into (μ_i, σ_i, A_i) for $i \in \{1, \dots, N\}$
end

the set dominated by some point in objective space, but not by any other point in the front. The hypervolume depends on a reference point, which is chosen adaptively such that it is dominated by the whole population, and such that the best individuals w.r.t. a single objective are always preferred (which amounts to elitism w.r.t. each single objective). Then selection amounts to keeping the best N out of $2N$ individuals according to this ranking.

Care has to be taken when adapting the $(1+1)$-xNES hillclimber to this selection scheme, because the notion of success differs from the condition for survival. We say that an offspring that is ranked higher than its parent is successful, resulting in a covariance matrix update. In contrast, depending on the success of the mutation, the step size is updated *for parent and offspring*.

The resulting MO-NES algorithm is summarized in Algorithm 4. It is derived straightforwardly by removing the $(1+1)$-CMA-ES module from MO-CMA-ES and replicing it with $(1+1)$-xNES. An individual is represented by the triplet $(\mu, \sigma, A) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^{d \times d}$, which is the state of the corresponding hillclimber. We denote parents by (μ_i, σ_i, A_i) and offspring by $(\mu'_i, \sigma'_i, A'_i)$ for $i \in \{1, \dots, N\}$.

6 Experimental Evaluation of MO-NES

We assess the performance of MO-NES compared to MO-CMA-ES on a collection of test problems found in [5], namely the standard benchmarks FON, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, coming with rectangular feasible regions, and the unbounded problems ELLI_1, ELLI_2, CIGTAB_1, and CIGTAB_2. These benchmarks cover a number of typical challenges such as concave and disconnected pareto fronts, as well as highly correlated variables.

Experimental Setup We largely follow the experimental procedure of [5]. The population size was set to $N = 100$, with individuals initialized uniformly at random in the feasible region. For the unbounded benchmarks the population was sampled from $[-10, 10]^d$. We used a search space dimension of $d = 10$ for all problems except FON, where it is fixed to $d = 3$. Resembling [5], we set the component-wise standard deviation of the initial search distribution to 0.6 times the edge length of the hyper-rectangle from which the initial population is sampled. Constraints were handled by evaluating the closest feasible point and adding 10^{-6} times the squared norm of the distance to the feasible region to each fitness component.

Each algorithm was granted 50.000 fitness evaluations, and assigned a score according to the hypervolume dominated by the final population. To achieve comparability with other studies, we fix the reference point for the hypervolume computation to $(1, 1)^T$ for FON and the ZDT-benchmarks, with the exception of $(1, 20)^T$ for ZDT4 (which is not solved satisfactory by any of the two algorithms), and to $(10, 10)^T$ for the unconstrained ELLI and CIGTAB problems.[1] We performed 25 independent trials for each experiment.

Results and Discussion The results are summarized in Table 1. There is no clear trend indicating that one algorithm would be generally preferable to the other. In most cases the differences between the algorithms are negligible, in the sense that they are below the range of the inter-trial deviations, and only in the fifth digit of the hypervolume. We found four significant differences (Wilcoxon rank sum test, $p = 0.01$): The MO-CMA-ES performs better on benchmarks with quadratic objectives, such as FON and CIGTAB, while MO-NES is superior on the ZDT6 problem. We conclude that the Pareto front approximations obtained by the two algorithms are generally of comparable quality. Taking the conceptual parallels of the two algorithms into account, this result does not come as a surprise. It shows that our novel MO-NES algorithm achieves state-of-the-art performance.

7 Conclusion

We presented two novel algorithms. The $(1 + 1)$-xNES hillclimber constituts a minimal elitist variant of the xNES algorithm which can be derived completely from the principle of natural gradient descent, despite its success-based step size adaptation rule. Like other NES algorithms, $(1+1)$-xNES is more principled than its canonical counterpart $(1 + 1)$-CMA-ES. Combining our new hillclimber with the multi-objective optimization scheme established for MO-CMA-ES results in the MO-NES algorithm. We empirically find both algorithms to exhibit state-of-the-art performance.

The impact of these contributions be seen from two perspectives: On the one hand, they make NES capable of multi-objective optimization, on the other

[1] Note that adaptively computed reference points are used in both algorithms to compute the S-measure for selection, and that we use these fixed reference points only for the evalution of the final fronts. This procedure is chosen to foster comparability with future studies. In particular, it does not expose any additional information to the search algorithms.

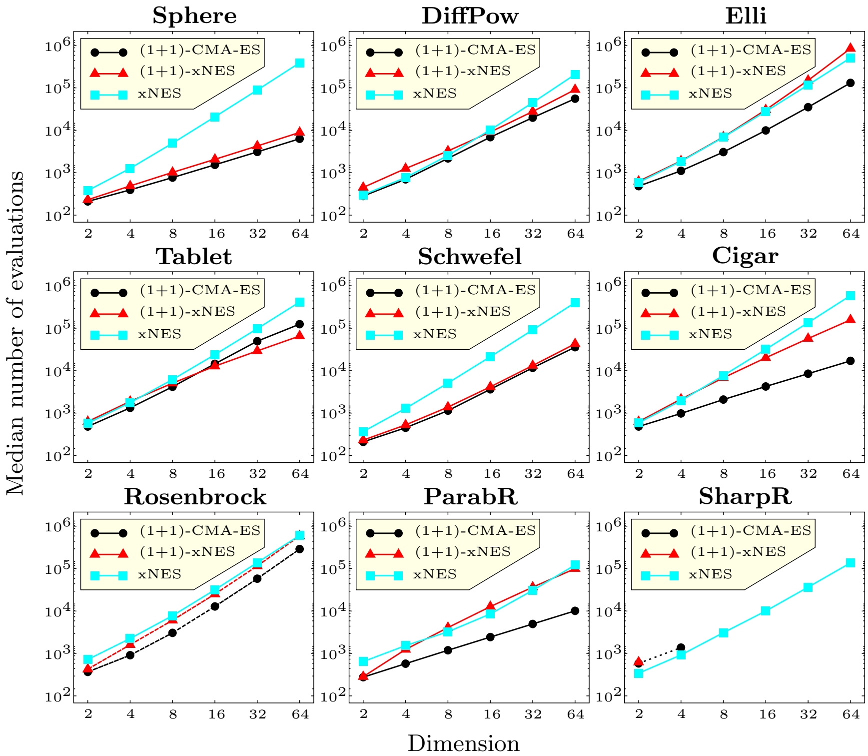

Fig. 1: Log-log plot of fitness evaluations required to reach target fitness (see text) over search space dimension for 9 different benchmark functions. The level of opacity of dashed connections indicates the fraction of successful runs. Setups for which no single run converged are not shown at all.

	MO-NES			MO-CMA-ES		
benchmark function	25% quantile	50% quantile	75% quantile	25% quantile	50% quantile	75% quantile
FON	0.337443	0.337453	0.337479	**0.337496**	**0.337511**	**0.337539**
ZDT1	0.661945	0.661962	0.661972	0.661934	0.661958	0.661972
ZDT2	0.328698	0.328703	0.328713	0.328697	0.328707	0.328720
ZDT3	1.042180	1.042180	1.042190	1.042180	1.042180	1.042190
ZDT4	0.661836	0.661860	0.661885	0.661834	0.661857	0.661879
ZDT4	10.93040	12.80430	15.95730	9.53145	11.77960	12.46610
ZDT6	**0.3225640**	**0.3225750**	**0.3225810**	0.0863215	0.3225550	0.3225770
$ELLI_1$	95.5311	95.5527	95.5593	95.5466	95.5553	95.5604
$ELLI_2$	99.9872	99.9905	99.9922	99.9897	99.9907	99.9931
$CIGTAB_1$	97.2286	97.2302	97.2310	**97.2303**	**97.2312**	**97.2318**
$CIGTAB_2$	99.9981	99.9985	99.9987	**99.9984**	**99.9988**	**99.9990**

Table 1: Hypervolume covered by the populations of MO-NES and MO-CMA-ES after 50,000 fitness evaluations. Statistically superior results are marked bold.

hand, they enrich the field of evolutionary MOO by the NES principle of descent along the natural fitness gradient.

Acknowledgments

This work was funded through the 7th framework programme of the European Union, under grant number 231576 ("STIFF" project), and SNF grant number 200021-113364/1.

References

1. C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer, New York, 2007.
2. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182 – 197, 2002.
3. T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2010.
4. N. Hansen and A. Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
5. C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, 2007.
6. 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.
7. J. Knowles, R. Watson, and D. Corne. *Evolutionary Multi-Criterion Optimization*, volume 1993 of *Lecture Notes in Computer Science*. Springer, 2001.
8. T. Rückstieß, F. Sehnke, T. Schaul, D. Wierstra, Sun Yi, and J. Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn Journal of Behavioral Robotics*, 1(1):14–24, 2010.
9. Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient Natural Evolution Strategies. In *Genetic and Evolutionary Computation Conference (GECCO)*, 2009.
10. Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic Search using the Natural Gradient. In *International Conference on Machine Learning (ICML)*, 2009.
11. 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.
12. E. Zitzler and S. Künzli. Indicator-Based Selection in Multiobjective Search. In *Parallel Problem Solving from Nature (PPSN VIII)*, pages 832–842. Springer, 2007.
13. E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms – A Comparative Case Study. In *Parallel Problem Solving from Nature (PPSN V)*, pages 292–301. Springer, 1998.