How to Assess Step-Size Adaptation Mechanisms in Randomised Search

Nikolaus Hansen, Asma Atamna, Anne Auger

How to Assess Step-Size Adaptation Mechanisms in Randomised Search

Nikolaus Hansen, Asma Atamna, and Anne Auger

Inria**
LRI (UMR 8623), University of Paris-Sud (UPSud), France

Abstract. Step-size adaptation for randomised search algorithms like evolution strategies is a crucial feature for their performance. The adaptation must, depending on the situation, sustain a large diversity or entertain fast convergence to the desired optimum. The assessment of step-size adaptation mechanisms is therefore non-trivial and often done in too restricted scenarios, possibly only on the sphere function. This paper introduces a (minimal) methodology combined with a practical procedure to conduct a more thorough assessment of the overall population diversity of a randomised search algorithm in different scenarios. We illustrate the methodology on evolution strategies with σ-self-adaptation, cumulative step-size adaptation and two-point adaptation. For the latter, we introduce a variant that abstains from *additional* samples by constructing two particular individuals within the *given* population to decide on the step-size change. We find that results on the sphere function alone can be rather misleading to assess mechanisms to control overall population diversity. The most striking flaws we observe for self-adaptation: on the linear function, the step-size increments are rather small, and on a moderately conditioned ellipsoid function, the adapted step-size is 20 times smaller than optimal.

1 Introduction

In this paper we consider a fitness or objective function, $f : \mathbb{R}^n \to \mathbb{R}$, to be minimised in a black-box optimisation scenario, and an evolutionary algorithm, or randomised search method, generating λ offspring according to

$$\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \times \boldsymbol{z}_k^{(t)} \ , k = 1, \ldots, \lambda \ , \tag{1}$$

where $\boldsymbol{x}^{(t)} \in \mathbb{R}^n$ denotes the incumbent solution at iteration t and $\boldsymbol{z}_k^{(t)} \in \mathbb{R}^n$ are i.i.d. random vectors. The "overall variance" of the offspring population in (1) is determined by the diversity parameter $\sigma^{(t)}$. More generally, we rely on two assumptions: (i) we have a valid measurement for the "global diversity" of the offspring population, denoted as $\sigma^{(t)}$, and (ii) the shape of the offspring population (determined by the distribution of $\boldsymbol{z}_k^{(t)}$ in (1)) does not change remarkably during the investigated time range of t.

Controlling the overall diversity in the population plays a crucial role in randomised search and has been typically approached by step-size adaptation. Two conflicting objectives are in place. On the one hand, diversity should be as large as possible to prevent

** Research centre Saclay–Île-de-France, TAO team, lastname@lri.fr

premature convergence or convergence to the very next local optimum. On the other hand, fast convergence to a global (or a good local) optimum is desired which is usually accompanied and facilitated by a fast decrease of diversity.

While adaptation of the *shape* of the sample distribution appears to be a solved problem in moderate dimension [6,10,11] (e.g. by CMA), the effective adaptation of the *overall* population diversity seems yet to pose open questions, in particular with recombination or without entire control over the realised distribution. For example, cumulative step-size adaptation is prone to fail when repair or rejection sampling is used.

In this context, we propose a basic *assessment procedure* to evaluate the capability of step-size control, or the entire search algorithm for that matter, to keep the overall diversity, or step-size $\sigma^{(t)}$, within reasonable limits. This procedure might be used during an algorithm designing process, however we like to remind the general scientific principle that a procedure used to systematically *tune parameters* of an algorithm is forfeited to *assess* the resulting algorithm.

In the next section we introduce the assessment methodology. Section 3 introduces the algorithms used in the case study in Section 4. We also introduce a simplified two-point adaptation and tune its damping parameter on the sphere function in Section 3. Section 5 provides a short discussion and summary.

2 Step-Size Evaluation Methodology

General demands on the behaviour of evolutionary algorithms were suggested previously, e.g. in [4,11]. Here, we propose a methodology to specifically investigate and assess the overall population diversity, or step-size, towards meeting reasonable demands via the following scenarios:

Random fitness (and flat fitness). On the random fitness, all f-values, $f(\boldsymbol{x})$, are i.i.d., independently of $\boldsymbol{x}$ as a continuous random variable. For algorithms invariant under order-preserving transformations of f, i.e., algorithms based on *f-rankings* only (as those investigated in this paper), testing a single continuous f-distribution is sufficient. Generally, we desire stationarity or unbiasedness of parameters under random fitness [11] and here we expect to see an unbiased random walk in log-scale. For the flat fitness, where f is constant, we expect the same behaviour. In contrast, [4] argues for an exponentially increasing step-size on the flat fitness which, however, involves the risk of divergence when the selection pressure is weak [7].

The linear function, where $f : \boldsymbol{x} \mapsto x_1$ is the prototypical instantiation (see paragraph *Invariance* below). A linear function tests whether and how quickly the diversity can increase. With step-size to zero, any smooth function appears to be an instantiation of the linear function (unless at a local optimum or saddle point) and the diversity should increase in this case. We demand a fast exponential increase, that is, a linear increase on the log-scale [4]. The rate should be at least comparable to the rate of decrease on the sphere function or at least a factor of 1.1 within n evaluations or at least a factor of 2 in n iterations.

The sphere function, $f : \boldsymbol{x} \mapsto \sum_{i=1}^{n} x_i^2 = \|\boldsymbol{x}\|^2$, is the most simple quadratic function, demanding a rapid decrease of the step-size. Arguably, no other function requires a faster step-size decrement. Step-size control should not reduce the fastest possible (optimal) convergence rate on the sphere function by more than a factor of about three.

To achieve linear (i.e. fast) convergence on the sphere function we need to have, at least approximately, $\sigma^{(t)} \propto f(\boldsymbol{x}^{(t)})^{1/2}$, implying that σ and $f^{1/2}$ converge at the same rate. More specifically, on the sphere function with isotropic sample distribution, there is a constant $\sigma^*_{\mathrm{opt}}(n)$ such that the step-size

$$\sigma^{(t)} = \frac{\sigma^*_{\mathrm{opt}}(n)}{n} \times f(\boldsymbol{x}^{(t)})^{1/2} \tag{2}$$

achieves optimal convergence speed and $\sigma^*_{\mathrm{opt}}(n) = \Theta(n^0) = \Theta(1)$. When running a real algorithm, the proportionality can only be satisfied in a stochastic sense, i.e. the random variable $\sigma^{(t)}/f(\boldsymbol{x}^{(t)})^{1/2}$ is stable (for example when $\boldsymbol{x}^{(t)}/\sigma^{(t)}$ is an irreducible, recurrent and ergodic Markov Chain [3]).

A similar reasoning on $\sigma^{(t)}$ holds true on the ellipsoid function, where the direct link between $\sigma^{(t)}$ and $f(\boldsymbol{x}^{(t)})^{1/2}$ is less obvious, however presumed in the following to obtain the *optimal* convergence rates to compare with.

The ellipsoid function, $f : \boldsymbol{x} \mapsto \sum_{i=1}^{n} \alpha^{(i-1)/(n-1)} x_i^2$, is arguably the most basic function where, for $\alpha \neq 1$, an isotropic distribution of the new offspring is not optimal. The parameter α represents the condition number of the Hessian matrix of f.

With isotropic sample distribution in (1) and $\alpha > 10$, the realised convergence rates are roughly proportional to $10/\alpha$ [12]. Recalling that $f^{1/2}(\boldsymbol{x}^{(t)})$ and the optimal value for $\sigma^{(t)}$, are linked to each other (Eq. (2)), we observe that with larger α, when approaching the optimum, the optimal step-size changes more slowly (because the realised convergence rate is small). The task to *estimate the optimal step-size* becomes more relevant than the task to *follow the change* of the optimal step-size. In this paper, experiments are done for $\alpha = 1, 10, 100$.

The stationary sphere is an artificial model, resembling the sphere function in that an isotropic sample distribution is optimal, but with *stationary optimal step-size*. While the sphere function tests the ability to decrease the step-size quickly, the stationary sphere function tests the ability to adapt the step-size close to the optimal step-size in the same sphere-like topography without approaching the optimum. With global intermediate or weighted recombination, as used below, the stationary sphere is simulated by setting the norm of the resulting recombination vector (super-parent) to one and re-normalisation of all other individuals or solutions in the algorithm's state by the same factor (see, e.g., lines 5–6 in Algorithm 3). When the population is never reduced to a single point, an appropriate normalisation factor needs to be identified (omitted due to space restrictions). The stationary sphere model is arguably the easiest model for step-size adaptation and we expect to observe close to the optimal step-sizes.

Convergence rate and optimal step-size. On the last three functions, we compute from a single run with t iterations the consistent estimator

$$\widehat{c} = -\frac{1}{T}\sum_{s=t-T}^{t-1}\frac{1}{2}\ln\left(\frac{f(\boldsymbol{x}^{(s+1)})}{f(\boldsymbol{x}^{(s)})}\right) \tag{3}$$

for the convergence rate [2, Eq. (24)], where $\boldsymbol{x}^{(s)} \in \mathbb{R}^n$ is the solution proposed at time step s, and the burn-in time $t-T$ diminishes the possible bias due to initialisation. In this paper we use $T = \lceil t/2 \rceil$, i.e. half of the overall time steps for aggregated measurements. If necessary (e.g., when we terminate due to numerical precision, but want more data), we average $\widehat{c}$ over several runs.

We obtain the values for the *optimal* step-size and convergence rate empirically by measuring the convergence rate with $\sigma^{(t)}$ set according to (2) and sweeping through different values for σ^*_{opt}. Generally, we demand the "real" algorithm to perform within a factor of three of this optimal convergence rate, and we prefer larger step-sizes to smaller ones, given the same performance is observed.

Invariance is an important concept in the assessment of algorithms. For example, all linear functions are identical for the below assessed algorithms, because the algorithms are invariant under affine transformations of f and under rotations of the search space. In the case where algorithms do not exhibit certain invariances (e.g. rotation invariance), it is advisable to test different instantiations (e.g. different rotations) of the above scenarios. Scale invariance on the other hand is a prerequisite to measure (3) independently of initial step-size or the distance to the optimum.

We now apply our methodology to three step-size adaptation methods. Due to the space limits, we do not always display single runs, but we consider investigating the evolution of f and σ (both displayed in the log scale) in single runs in all scenarios part of the assessment procedure [15].

3 Considered Step-Size Adaptation Methods

In the following, we consider the $(\mu/\mu, \lambda)$-ES with weighted recombination [11]. The offspring are generated as in (1) where the i.i.d. $\boldsymbol{z}_k^{(t)}$ follow the standard multivariate normal distributions, i.e., $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$. They are sorted according to their fitness such that

$$f\left(\boldsymbol{x}_{1:\lambda}^{(t)}\right) \leq f\left(\boldsymbol{x}_{2:\lambda}^{(t)}\right) \leq \cdots \leq f\left(\boldsymbol{x}_{\lambda:\lambda}^{(t)}\right) \quad , \tag{4}$$

thereby defining the index $k:\lambda$ used in the following. The μ best individuals are then recombined according to

$$\boldsymbol{x}^{(t+1)} = \sum_{k=1}^{\mu} w_k \boldsymbol{x}_{k:\lambda}^{(t)} \quad , \tag{5}$$

where w_k's are chosen to be optimal on the infinite-dimensional sphere function [1]. We set $\mu = \lfloor \lambda/2 \rfloor$ and therefore have only positive weights while $\lambda = 4 + \lfloor 3 \ln n \rfloor$.

We consider here three ways to adapt the step-size in (1). Self-Adaptation (SA) [14] and Cumulative Step-size Adaptation (CSA) [11] are given in Algorithm 1 and 2. The used default parameter settings for the latter are taken from [9] as $c_\sigma = \frac{\mu_w+2}{n+\mu_w+5}$,

Algorithm 1 The $(\mu/\mu, \lambda)$-σSA-ES

0 **given** $n \in \mathbb{N}_+$, λ, μ, w_k, $\tau = 1/\sqrt{2n}$
1 **initialize** $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\sigma^{(0)} \in \mathbb{R}_+$
2 **while** not happy
3 **if** *stationary_sphere* :
4 $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t)}/\|\boldsymbol{x}^{(t)}\|$
5 **for** $k \in \{1, \ldots, \lambda\}$
6 $\xi_k^{(t)} = \tau \mathcal{N}_{t,k}(0, 1)$
7 $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$
8 $\sigma_k^{(t)} = \sigma^{(t)} \times \exp(\xi_k^{(t)})$
9 $\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma_k^{(t)} \times \boldsymbol{z}_k^{(t)}$
10 $\sigma^{(t+1)} = \sum_{k=1}^{\mu} w_k \sigma_{k:\lambda}^{(t)}$
11 $\boldsymbol{x}^{(t+1)} = \sum_{k=1}^{\mu} w_k \boldsymbol{x}_{k:\lambda}^{(t)}$
12 $t = t + 1$

Algorithm 2 The $(\mu/\mu, \lambda)$-CSA-ES

0 **given** $n \in \mathbb{N}_+$, λ, μ, w_k, c_σ, d
1 **initialize** $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\sigma^{(0)} \in \mathbb{R}_+$, $\boldsymbol{p}_\sigma^{(0)} = \mathbf{0}$
2 **while** not happy
3 **if** *stationary_sphere* :
4 $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t)}/\|\boldsymbol{x}^{(t)}\|$
5 **for** $k \in \{1, \ldots, \lambda\}$
6 $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$
7 $\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \times \boldsymbol{z}_k^{(t)}$
8 $\boldsymbol{p}_\sigma^{(t+1)} = (1 - c_\sigma)\boldsymbol{p}_\sigma^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)/\sum_k^\mu w_k^2} \sum_{k=1}^{\mu} w_k \boldsymbol{z}_{k:\lambda}^{(t)}$
9 $\sigma^{(t+1)} = \sigma^{(t)} \times \exp^{\frac{c_\sigma}{d}} \left(\frac{\|\boldsymbol{p}_\sigma^{(t+1)}\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1 \right)$
10 $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \sum_{k=1}^{\mu} w_k \boldsymbol{z}_{k:\lambda}^{(t)}$
11 $t = t + 1$

$d = 1 + 2 \max\left(0, \sqrt{\frac{\mu_w - 1}{n+1}} - 1\right) + c_\sigma$. The third method considered for step-size adaptation is presented in the following.

Two-Point Step-Size Adaptation (TPA). We consider a tidied version of Two-Point Step-Size Adaptation (TPA) based on [8,13]. Conceptually, TPA implements a very coarse line search along the direction of the latest mean shift from $\boldsymbol{x}^{(t-1)}$ to $\boldsymbol{x}^{(t)}$. In our version, we sample the first two offspring *of the next iteration* along this line. These two offspring are generated as a mirrored pair, symmetric about the current mean vector $\boldsymbol{x}^{(t)}$,

$$\boldsymbol{x}_{1/2}^{(t)} = \boldsymbol{x}^{(t)} \pm \sigma^{(t)} \times \|\mathcal{N}_t(\mathbf{0}, \mathbf{I})\| \frac{\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}}{\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\|} \quad , \tag{6}$$

instead of (1). Their ranking according to the fitness is used to adapt the step-size: if $\boldsymbol{x}_1^{(t)}$ is better than $\boldsymbol{x}_2^{(t)}$ the step-size is increased, because there are better points in the direction of the mean shift vector, beyond of where the mean has been moved. Otherwise, the step-size is decreased. By using individuals that are likely to be sampled by the current distribution, information on the "signal strength" is available, because we can compare their fitness to the fitness of the remaining population. Accordingly, we take the difference between the f-ranks of $\boldsymbol{x}_1^{(t)}$ and $\boldsymbol{x}_2^{(t)}$ in the population, $\frac{\text{rank}(\boldsymbol{x}_2^{(t)}) - \text{rank}(\boldsymbol{x}_1^{(t)})}{\lambda - 1} \in [-1, 1]$. This normalised rank difference is averaged in $s^{(t)}$ and used to finally update the step-size $\sigma^{(t+1)} = \sigma^{(t)} \times \exp\left(s^{(t)}/d_\sigma\right)$, where the damping, d_σ, moderates the step-size changes. The details are shown in Algorithm 3.

The constant for which $\sigma^{(t)}$ in (2) achieves optimal convergence rate depends on the sampling. For TPA-like sampling, we denote it $\sigma^*_{\text{opt TPA}}$.

Algorithm 3 The $(\mu/\mu, \lambda)$-ES with TPA

given $n \in \mathbb{N}_+$, λ, μ, $c_\sigma = 0.3$, $d_\sigma = \sqrt{n}$, w_k
init $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$, $\sigma^{(0)} \in \mathbb{R}_+$, $t = 0$, $s^{(0)} = 0$
while not happy
  if *stationary_sphere* :
    if $t > 0$:
      $\boldsymbol{x}^{(t-1)} = \boldsymbol{x}^{(t-1)} / \|\boldsymbol{x}^{(t)}\|$
    $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t)} / \|\boldsymbol{x}^{(t)}\|$
  for $k \in \{1, \ldots, \lambda\}$
    $\boldsymbol{z}_k^{(t)} = \mathcal{N}_{t,k}(\mathbf{0}, \mathbf{I})$
    if $t > 0$ and $k = 1$:
      $\boldsymbol{z}_1^{(t)} = \|\mathcal{N}_t(\mathbf{0}, \mathbf{I})\| \times \dfrac{(\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)})}{\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^{(t-1)}\|}$
    if $t > 0$ and $k = 2$:
      $\boldsymbol{z}_2^{(t)} = -\boldsymbol{z}_1^{(t)}$
    $\boldsymbol{x}_k^{(t)} = \boldsymbol{x}^{(t)} + \sigma^{(t)} \times \boldsymbol{z}_k^{(t)}$
  $\boldsymbol{x}^{(t+1)} = \sum_{k=1}^{\mu} w_k \boldsymbol{x}_{k:\lambda}^{(t)}$
  if $t > 0$:
    $s^{(t)} = (1 - c_\sigma) s^{(t-1)} + c_\sigma \dfrac{\text{rank}(\boldsymbol{x}_2^{(t)}) - \text{rank}(\boldsymbol{x}_1^{(t)})}{\lambda - 1}$
    $\sigma^{(t+1)} = \sigma^{(t)} \times \exp\left(\dfrac{s^{(t)}}{d_\sigma}\right)$
  $t = t + 1$

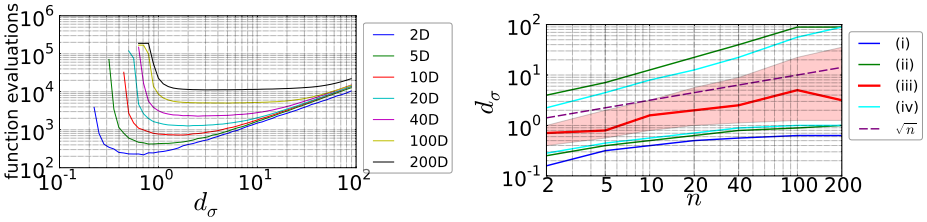

Fig. 1. Left: number of function evaluations versus damping d_σ for TPA, averaged over 101 runs with target f-value 10^{-8}. Right: solid lines depict, from bottom to top, (i) the smallest damping where all runs reached the target value; (ii) the smallest and largest "reasonable" damping with a performance not worse than three times the best (lowest) value in the respective graph on the left; (iii) the damping with best performance, d_σ^*; (iv) the smallest and largest damping with performance no more than two times worse than the best value in the respective graph on the left, all plotted against dimension. The dashed line depicts $\sqrt{n}$. The filled area corresponds to damping values with at most 20% performance loss compared to the optimal damping.

The Damping Factor. Here we identify a default value for the damping d_σ. To this aim, we follow a standard procedure: d_σ is tuned on the sphere function. For each value of d_σ, the algorithm is run 101 times with target f-value 10^{-8} (the f-value that stops the algorithm when reached), and if all runs reached the target, the average number of f-evaluations is recorded, see Figure 1, left. We observe a steep incline to the left (small values of d_σ), where missing points indicate the failure of at least one run to reach the target. To the right, the number of f-evaluations increases linearly with the damping and no failures are observed. We extract four damping values per dimension as shown and described in Figure 1, right. We then choose the damping to be (a) more than three times larger than the smallest "reasonable" value and (b) larger than the optimal value such that (c) reducing d_σ by a factor of two leads to a better performance than increasing it by a factor of two without (d) loosing more than a factor of two in performance compared to the best damping (see also [5]). The default choice becomes $d_\sigma = \sqrt{n}$. Note that we

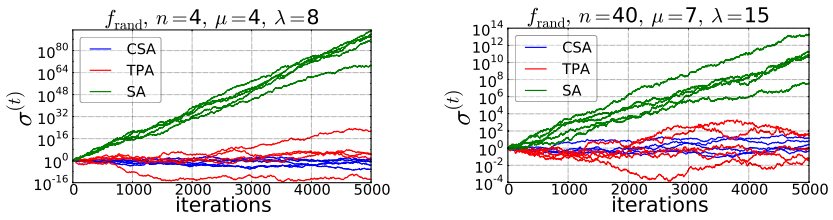

Fig. 2. Evolution of $\sigma^{(t)}$ on the random fitness for 5 runs of SA (green), CSA (blue), and TPA (red) in 4-D (left) and 40-D (right).

identified the damping only for the given default population size. The same procedure needs to be repeated to identify a damping parameter for different population sizes.

4 A Case Study

Experiments are conducted in dimensions between 2 and 100. The algorithms are run with the default parameter settings (Section 3) and initial $\boldsymbol{x}^{(0)} = (1, 0, \ldots, 0)^{\mathsf{T}}$. On random, linear, and ellipsoid function we have $\sigma^{(0)} = 1$, on the sphere and stationary sphere we have $\sigma^{(0)} = \sigma^*_{\text{opt}}/n$ (respectively $\sigma^*_{\text{opt TPA}}/n$) for SA and CSA (respectively TPA). Interquartile ranges are depicted as notched bars with the median at the notch.

Random Fitness. Figure 2 displays the evolution of $\sigma^{(t)}$ for 5000 iterations in 4- and 40-D, five runs for each algorithm. As expected by design, CSA and TPA show an unbiased random walk of $\log \sigma$, where TPA reveals a larger variance. In contrast, due to the combination of geometric mutation and arithmetic recombination of the step-sizes, the random walk of SA is biased [7] and $\log \sigma$ increases linearly with a rate of a little above (below) $10^{0.07} \approx 1.17$ in n iterations for $n = 40$ ($n = 4$, respectively).

Linear Function. On the linear function, the algorithms are run 100 times for 400 iterations. Figure 3 shows geometric average and quartiles of the step-size change realised after n evaluations, $(\sigma^{(t+1)}/\sigma^{(t)})^{n/\lambda}$, compared to results obtained on the random and the sphere function.

For CSA and TPA, the step-size increases by at least a factor of 1.14 within n evaluations. This factor increases slowly with increasing dimension (but never exceeds a factor of two) and the increment on the linear function is at least about three times faster than the decrement on the sphere function.

Self-Adaptation realises only an increment of a factor between 1.03 and 1.05 within n function evaluations, where also decrements appears frequently. The step-size grows faster than on the random function but up to four times slower than it shrinks on the sphere function. This latter observation, together with the observed slow changes rates, fails to meet our original demand.

Sphere. On the sphere function, the target f-value is 10^{-100}. Figure 4 shows nine single runs (left) with $\sigma^{(0)} = 10^{-5}$, the step-size as geometric average (middle), and the convergence rate $\widehat{c} \times n/\lambda$ (right, see (3)), both averaged over 100 runs.

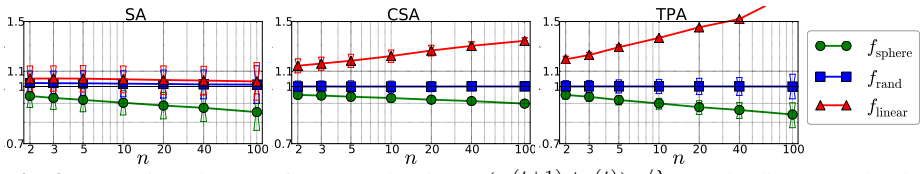

Fig. 3. Step-size change after n evaluations, $(\sigma^{(t+1)}/\sigma^{(t)})^{n/\lambda}$, on the linear (red), the sphere (green), and the random function (blue).

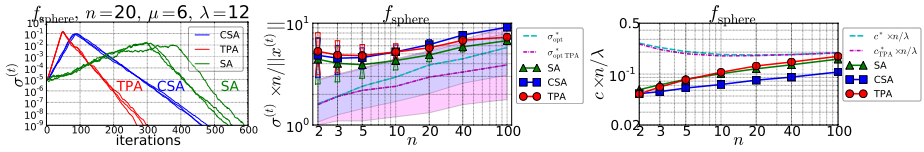

Fig. 4. Single runs (left), step-size (middle) and convergence rate (right) on the sphere function, for SA (green), CSA (blue), and TPA (red) and the respective optimal values. Filled areas correspond to step-size values with at most 20% performance loss compared to σ_{opt} (or $\sigma_{\text{opt TPA}}$, respectively).

All algorithms realise a too large step-size. In small dimensions, this leads to a loss in performance by about a factor of five, thereby failing our original demand. Fortunately, with increasing dimension the effect diminishes. For $n = 100$, TPA and SA reveal close to optimal convergence rates, whereas CSA is about two times slower.

Supposedly, we observe larger-than-optimal step-sizes, because the optimal step-size changes during the run and is therefore a *moving target*. Indeed, decreasing the damping parameters d or d_σ in CSA or TPA by a factor of two or increasing τ in SA improves the convergence speeds thereby meeting just about the original demand. However for SA, this impairs the performance on the ellipsoid function with $\alpha = 10$.

Ellipsoid. Complementing the observations on the sphere function, which coincides with the ellipsoid function with $\alpha = 1$, the algorithms are investigated on the ellipsoid function with $\alpha \in \{10, 100\}$. These are very moderate condition numbers, where an isotropic distribution can still realise comparatively high convergence rates. We conduct 100 runs with target f-value[1] of 10^{-50}. Figure 5 shows the step-size as geometric average and the convergence rate $\widehat{c}$ from (3).

With increasing condition number the realised step-sizes become across the board smaller (compared to the optimal step-size). For $\alpha = 10$, the step-size is still slightly too large with CSA and TPA, while SA shows already too small step-sizes. With $\alpha = 100$, SA realises a 20 times smaller than optimal step-size. Then, for $n \geq 10$, SA performs four to six times slower than optimal, while the other two methods reveal close-to-optimal convergence rates.

Stationary Sphere. On the stationary sphere model, the algorithms are run for $t = 5000$ iterations. The convergence rate $\widehat{c}$ from (3) is estimated from 100 runs.

Figure 6 shows step-sizes (as geometric average) and convergence rates. The CSA achieves close to optimal step-sizes and convergence rates in all dimensions. The TPA

[1] In general, we can use such a small target f-value only because the optimum is located at zero and because the distribution shape does not change over the iterations (see Section 1).

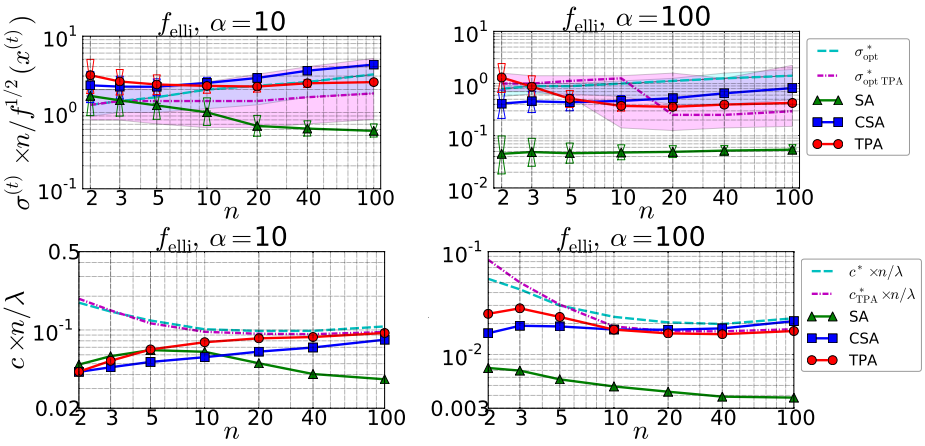

Fig. 5. Results on the ellipsoid with condition number 10 (left) and 100 (right). Top: normalised step-size. Shaded areas depict the step-size range with at most 20% loss in convergence rate. Bottom: convergence rate according to (3).

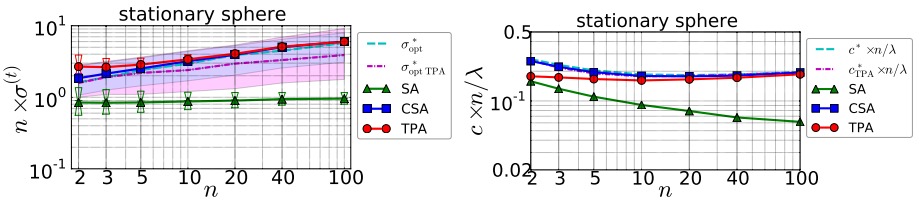

Fig. 6. Step-size (left) and convergence rate (right) of SA (green), CSA (blue), and TPA (red) on the stationary sphere together with the respective optimal values. Shaded areas reflect step-sizes with no more than 20% loss in the achieved convergence rate.

reveals very similar step-sizes in larger dimensions, however for TPA they are somewhat too large, because the optimal step-size is somewhat smaller. Yet, only in smaller dimensions a (moderate) performance loss is observed.

In contrast, SA adapts always a too small step-size. The gap to the optimal step-size is a factor of two in 2-D and increases to a factor of 6 in 100-D. The loss in convergence rate is (slightly) above a factor of three only in 100-D. These observations are (qualitatively) similar to those on the ellipsoid function with condition number 100.

Compared to the sphere function, the observed step-sizes are *in all cases* considerably smaller, again supporting the hypothesis that too large step-sizes are observed on the sphere function mainly because the optimal step-size is a moving target.[2]

5 Discussion and Summary

We have introduced a methodology to assess the overall population diversity, for example determined via step-size adaptation, by describing the desired outcomes on basic

[2] Experiments with varying damping- or τ-values give additional strong support. Increasing damping impairs the performance on the sphere function (cp. Fig. 1) by reducing the change rate of the step-size, while it (slightly) improves the performance on the stationary sphere.

scenarios. We conducted a case study assessing evolution strategies with weighted recombination and three different step-size adaptation mechanisms.

Despite the small number of investigated algorithms, we find in each test scenario, arguably with exception of the random function, limitations of at least one method: a (too) slow step-size increase on the linear function; a (too) slow step-size decrease on the sphere function in small dimensions; adaptation of a far too small step-size on the ellipsoid and stationary sphere. The results suggest that both, design and assessment of step-size adaptation methods is more intricate than one would have hoped for.

Acknowledgments. This work was supported by the grant ANR-2012-MONU-0009 (NumBBO) of the French National Research Agency.

References

1. D. V. Arnold. Optimal weighted recombination. In *Foundations of Genetic Algorithms*, pages 215–237. Springer, 2005.
2. A. Auger and N. Hansen. Reconsidering the progress rate theory for evolution strategies in finite dimensions. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 445–452. ACM, 2006.
3. A. Auger and N. Hansen. On Proving Linear Convergence of Comparison-based Step-size Adaptive Randomized Search on Scaling-Invariant Functions via Stability of Markov Chains, 2013. ArXiv eprint.
4. H.-G. Beyer and K. Deb. On self-adaptive features in real-parameter evolutionary algorithms. *Evolutionary Computation, IEEE Transactions*, 5(3):250–270, 2001.
5. D. Brockhoff, A. Auger, N. Hansen, D. V. Arnold, and T. Hohm. Mirrored sampling and sequential selection for evolution strategies. In *Parallel Problem Solving from Nature, PPSN XI*, pages 11–21. Springer, 2010.
6. T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber. Exponential natural evolution strategies. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 393–400. ACM, 2010.
7. N. Hansen. An analysis of mutative σ-self-adaptation on linear fitness functions. *Evolutionary Computation*, 14(3):255–275, 2006.
8. N. Hansen. CMA-ES with two-point step-size adaptation. *CoRR*, abs/0805.0231, 2008.
9. N. Hansen. The CMA evolution strategy: A tutorial. 2011.
10. N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 282–291. Springer, 2004.
11. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
12. N. Hansen, R. Ros, N. Mauny, M. Schoenauer, and A. Auger. Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11(8):5755–5769, 2011.
13. R. Salomon. Evolutionary algorithms and gradient search: Similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.
14. H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley Interscience, New York, 1995.
15. http://hal.inria.fr/hal-00997294.