Sanyang ML, Kabán A. Large-scale estimation of distribution algorithms with adaptive heavy tailed random projection ensembles. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(6): 1241–1257 Nov. 2019. DOI 10.1007/s11390-019-1973-1

Large-Scale Estimation of Distribution Algorithms with Adaptive Heavy Tailed Random Projection Ensembles

Momodou L. Sanyang[1,2] and Ata Kabán[1], *Member*, *IEEE*

[1] *School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K.*
[2] *School of Information Technology and Communication, University of The Gambia, Serekunda, The Gambia*

E-mail: MLSanyang@utg.edu.gm, A.Kaban@cs.bham.ac.uk

Received September 6, 2018; revised September 13, 2019.

Abstract We present new variants of Estimation of Distribution Algorithms (EDA) for large-scale continuous optimisation that extend and enhance a recently proposed random projection (RP) ensemble based approach. The main novelty here is to depart from the theory of RPs that require (sub-)Gaussian random matrices for norm-preservation, and instead for the purposes of high-dimensional search we propose to employ random matrices with independent and identically distributed entries drawn from a t-distribution. We analytically show that the implicitly resulting high-dimensional covariance of the search distribution is enlarged as a result. Moreover, the extent of this enlargement is controlled by a single parameter, the degree of freedom. For this reason, in the context of optimisation, such heavy tailed random matrices turn out to be preferable over the previously employed (sub-)Gaussians. Based on this observation, we then propose novel covariance adaptation schemes that are able to adapt the degree of freedom parameter during the search, and give rise to a flexible approach to balance exploration versus exploitation. We perform a thorough experimental study on high-dimensional benchmark functions, and provide statistical analyses that demonstrate the state-of-the-art performance of our approach when compared with existing alternatives in problems with 1 000 search variables.

Keywords covariance adaptation, estimation of distribution algorithm, random projection ensemble, t-distribution

1 Introduction

Optimisation over high-dimensional search spaces is a key task in many modern applications, including scientific, engineering and management problems. Estimation of Distribution Algorithms (EDAs) are black-box optimisation methods based on probabilistic modelling, which are becoming increasingly important in the evolutionary computation (EC) community[1]. EDAs avoid the usage of arbitrary operators of traditional EC approaches; instead they estimate a probabilistic model from the fittest individuals, and advance the search for the global optimum by sampling the estimated model. In this way, EDAs will not only simplify the design of an optimisation method, but also allow the algorithms to learn the structure of the search space and guide the search in further promising directions[2].

However, the performance of most existing EDAs deteriorates significantly in high-dimensional search spaces. This is because EDA requires model building from empirical data, which is susceptible to the curse of dimensionality[3]. Unless an enormous budget of fitness evaluations is available, the sample size is insufficient for reliable model estimation. Consequently the structure of the problem is often severely mis-estimated from limited samples (see [4, 5] for examples). In turn, a badly estimated model typically results in premature convergence[5].

To mitigate the curse of dimensionality, simplified models such as univariate models, or limited dependency models are often preferred instead of a full multivariate model in practice. The simplest of such approach is described in [6], called UMDAc. The

Regular Paper

The work of Momodou L. Sanyang was partly funded by a Ph.D. scholarship from the Islamic Development Bank. The work of Ata Kabán is funded by the Engineering and Physical Sciences Research Council of UK under Fellowship Grant EP/P004245/1.

model building in this univariate approach is simply done under the independence assumption[6]: $P(\boldsymbol{x}) = \prod_{i=1}^{d} P(x_i)$, where d is the dimensionality of the search space. More refined approaches such as the sep-CMA-ES[7] and the univariate AMaLGaM[8] use only diagonal elements of a rotated full covariance matrix. However, the univariate models assume that all the variables are independent in some coordinate basis; hence they may be expected to perform well on problems that have no dependency between variables (i.e., separable) in that basis.

Non-separable problems are those in which the above assumption does not hold, that is, the design variables have multiple inter-dependencies. The approach known as MIMIC, proposed by De Bonet *et al.*[9] sets out to capture bivariate interactions between decision variables by sampling from the pairwise joint distribution between variables. Despite that MIMIC is able to outperform univariate models, the majority of optimisation problems will have larger groups of interacting design variables. Several proposals have been put forth to explicitly capture multivariate dependencies by building graphical dependency networks, for example Bayesian Networks[10]. But, from statistical, computational and memory points of view, learning probabilistic graphical models from the sample of fit individuals is highly expensive and requires a large sample[11]. Thus, the scaling-up of these model building processes to high-dimensional problems remains challenging.

1.1 Related Work in Large-Scale Black-Box Optimisation

There have been many efforts to alleviate the problems of high-dimensional search, resulting in several algorithms being proposed recently, including EDA-type methods. Here we limit ourselves to a few that are most relevant to the present work. Approaches can be grouped into cooperative co-evolution based methods, evolutionary strategies, latent variable models, hybrid methods, and compression based divide and conquer.

One of the first cooperative co-evolution (CC) methods is CC with Variable Interaction Learning (CCVIL) proposed by Weicker and Weicker in [12]. It is a deterministic method to uncover dependencies between decision variables. A more recent model-based formulation of a similar idea is EDA with Model Complexity Control (EDA-MCC)[13], which also employs a deterministic algorithm; it groups variables into two disjoint subsets, those that are weakly correlated with other variables, and those that are strongly correlated. The strength of correlations is estimated relative to a threshold.

Multilevel cooperative co-evolution (MLCC) proposed in [14] groups the decision variables of a problem randomly to tackle problems in high-dimensional optimisation. The groups are then optimised jointly, but separately from other groups.

MA-SW-Chain[15] is a hybrid algorithm that assigns local search intensity to each individual by chaining dissimilar local search applications. It is an extension of the MA-CMA-Chain algorithm for high-dimensional regime, and was the winner of the CEC2010-2012 large-scale global optimisation competition on 1 000 dimensional benchmark problems.

Another related method employs latent variable models to sample new individuals with low-dimensional latent vectors. Shin *et al.*[16] proposed a latent variable model such as the Helmholtz machine and probabilistic principal component analysis to estimate the probability distribution of given data. The model considers latent variables which have a much lower dimension than input variables for optimising a high-dimensional function. In doing so, the method will mitigate the curse of dimensionality. A different way to exploit hidden low-dimensional structure was proposed in [17], suggesting that a notion of the intrinsic dimension of a function can replace the ambient dimension of the inputs to the function, provided that the function has a special structure. However, in the absence of this structure this method is unsuitable.

A recent approach is the random matrix theory based EDA in [4], which devises a compression-based divide and conquer strategy. The idea is to project the high-dimensional selected individuals to many independent low-dimensional random subspaces, carry out the costly sampling and estimation operations concurrently in these subspaces, and combine the new samples to form the next generation. Such ensemble can exhibit a strong smoothing effect that facilitates the estimation from small samples.

1.2 Contributions

In this paper, we further enhance the random projection (RP) ensemble approach of [4], by employing a parameterised family of heavy-tailed random matrices to replace the more conventional sub-Gaussian ones. Although this results in a minor change in the implementation, as we shall see, it allows for a better exploration of high-dimensional search spaces. An early version with promising empirical results of this idea ap-

peared in [18]①.

The increased exploration abilities of heavy tailed distributions are well known in evolutionary search, starting from pioneering work by Yao *et al.*[19] in the univariate setting, and more recent work in the multivariate EDA framework (see, e.g., [20–22] and references therein). However, in the regime of large search spaces of dimensionality well beyond 100 variables, the use of such heavy tailed distributions in EDA is not straightforward – as demonstrated in [22], a heavy tailed search distribution becomes increasingly counterproductive as it loses sight of the direction of the search. Here we avoid these problems as we employ heavy tailed distributions in a combination of RPs rather than directly in the role of a search distribution.

Our approach is based on a detailed analysis of the effect of the distribution of entries of random matrices used in the algorithm in terms of the resulting aggregated covariance of the multivariate search distribution (which is not heavy tailed but Gaussian, as we shall see later). We should also note that this type of analysis is entirely different, and complementary with analyses of the dynamics or the running time, as the latter is currently feasible only in univariate models on some very specific problems[23]. The purpose of our analysis in this work is 1) to shed light on some limitations of borrowing existing tools from the area of RPs as done in previous work, and 2) to construct novel algorithm variants that bypass these limitations.

In particular, our analysis reveals that, while the use of RPs is useful for dealing with high-dimensional optimisation problems, the existing RP theory is not well aligned with the needs of optimisation. More precisely, the use of sub-Gaussian RP matrices has been originally borrowed from a literature that aims at approximately preserving Euclidean geometry. This goal is different from ours and, as we shall see, adopting the same approach falls short of exploration ability for search. In turn, the use of heavy tailed random matrices is very unusual in the context of the RP literature, as they do not have good distance preservation properties, but they will turn out to have advantages for high-dimensional search.

The specific contributions of this paper are as follows.

• We analytically show that the excess kurtosis of the entries of the RP matrices in the ensemble has a precise role in the aggregated covariance of the ensemble. The latter is the covariance of the high-dimensional search distribution; therefore this analysis reveals the following: a distribution with negative excess kurtosis will shrink the aggregated covariance in comparison with that obtained when using the Gaussian distribution, and a distribution with positive excess kurtosis will enlarge it.

• We then propose to employ random matrices with i.i.d. t-distributed entries — this subsumes the Gaussian ensemble of [4] as a special case, when the degree of freedom is infinite. However, the degree of freedom parameter offers a means to enlarge the aggregated covariance in favour of better exploration in high dimensions. In particular, we show that the extent of this enlargement is a decreasing function of the degree of freedom parameter.

• Based on this, we then develop simple methods to adapt the degree of freedom parameter during the search, which essentially implement novel covariance adaptation schemes specifically for high-dimensional search.

• We present a thorough experimental study on 1 000-dimensional multi-modal test functions from the CEC2010-2012 large-scale global optimisation competition benchmark, and demonstrate the superior or state-of-the-art performance in statistical comparisons with a number of existing alternatives.

2 Heavy-Tailed Random Matrices for Continuous EDA Optimisation

Random projections (RP) have already been used with success in large-scale continuous EDA[4]. However the theory of RP in the literature is aimed at the preservation of the Euclidean distances and norms. The goal in optimisation is different: it is more important to have a good exploration-exploitation trade-off.

With this goal in mind, in this section we shall present an RP ensemble based large-scale multivariate Gaussian EDA that employs random matrices with heavy tailed entries drawn i.i.d. from the family of t-distributions. We shall present this algorithm first, to fix ideas, so that we see how these random matrices enter into the search algorithm. We will then analytically justify our choice for the family of t-distributions afterwards in the subsequent subsection.

2.1 tRP-Ens-EDA Algorithm: A Basic Variant

Our algorithm is a modification of the random projection ensemble based EDA (RP-Ens-EDA) of [4], and

①Received a Runner-Up Student Paper Award of the CEC2015 Conference.

we refer to our new variant as tRP-Ens-EDA. The pseudo-code of tRP-Ens-EDA is given in Algorithm 1.

Algorithm 1. Algorithm with Entries of $\boldsymbol{R}_i$ from t-Distribution (tRP-Ens-EDA)

Input: $k, M, N, MaxFE$
Output: best individual $\boldsymbol{x}^*$; best fitness f^*
1 $\mathcal{P} \leftarrow$ Initialize a population uniformly at random in the search space
2 $\nu \leftarrow$ Initialize a degree of freedom
3 **while** $MaxFE > 0$ **do**
4 $\quad MaxFE \leftarrow MaxFE - N$
5 $\quad \boldsymbol{f} \leftarrow$ Evaluate the fitnesses of $\mathcal{P}$
6 $\quad \mathcal{P}^{\text{fit}} \leftarrow$ Select the T individuals of $\mathcal{P}$ with best fitness using truncation selection
7 $\quad \boldsymbol{x}^* \leftarrow$ the individual with the optimal fitness so far
8 $\quad f^* \leftarrow$ the optimal fitness value so far
9 $\quad \nu \leftarrow$ Decide a new ν with an adaptive method
10 $\quad \boldsymbol{\mu} \leftarrow$ Estimate the mean of $\mathcal{P}^{\text{fit}}$ using maximum likelihood estimation (MLE)
11 $\quad \mathcal{P}^{\text{fit}} \leftarrow \mathcal{P}^{\text{fit}} - \boldsymbol{\mu}$
12 $\quad \{\boldsymbol{R}_i\}_{i=1:M} \leftarrow$ Generate M independent random matrices with entries drawn i.i.d. from a t-distribution with mean 0 and variance $\frac{1}{d}$
13 $\quad$ **foreach** $i \in \{1, 2, \ldots, M\}$ **do**
14 $\quad\quad \mathcal{Y}^{\boldsymbol{R}_i} \leftarrow \boldsymbol{R}_i \mathcal{P}^{\text{fit}}$ projects $\mathcal{P}^{\text{fit}}$ to a random k-dimensional subspace
15 $\quad\quad \boldsymbol{\Sigma}^{\boldsymbol{R}_i} \leftarrow$ Estimate the $k \times k$ covariance matrix of $\mathcal{Y}^{\boldsymbol{R}_i}$ using MLE
16 $\quad\quad$ Sample N new points $\boldsymbol{y}_1^{\boldsymbol{R}_i}, ..., \boldsymbol{y}_N^{\boldsymbol{R}_i}$ i.i.d. from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{R}_i})$
17 $\quad \mathcal{P}^{\text{new}} \leftarrow \sqrt{\frac{dM}{k}}[\frac{1}{M}\boldsymbol{\Sigma}_{i=1}^{M} \boldsymbol{R}_i^{\mathrm{T}} \boldsymbol{y}_1^{\boldsymbol{R}_i}, ..., \frac{1}{M}\boldsymbol{\Sigma}_{i=1}^{M} \boldsymbol{R}_i^{\mathrm{T}} \boldsymbol{y}_N^{\boldsymbol{R}_i}] + \boldsymbol{\mu}$
18 $\quad \mathcal{P} \leftarrow \mathcal{P}^{\text{new}}$
19 $\quad$ Replace an individual in $\mathcal{P}$ with $\boldsymbol{x}^*$ (elitism)

The tRP-Ens-EDA proceeds by initially generating a population of individuals randomly everywhere in the search space, and selects the T fittest ones based on their fitness values. This is the set $\mathcal{P}^{\text{fit}}$ in Algorithm 1. M independent random projection (RP) matrices $\boldsymbol{R}_i$ are then generated, and each defines a random subspace of dimension $k \ll d$, where d is the dimension of the search space. These are created in order to project the fittest individuals onto low-dimensional subspaces defined by them.

The number of such subspaces, M, and their dimension, k, are parameters of the method along with the population size N. For all of these parameters we will use the default values determined in [4], i.e., $M = \lceil \frac{3d}{k} \rceil$, $k = 3$, $N = 300$, and $|\mathcal{P}^{\text{fit}}| = N/4$, as the context and meaning of these parameters is the same as in [4]. Another input parameter is the maximum fitness evaluations allowed, MaxFE, which will be varied depending on the goal of the experiments.

Once the set $\mathcal{P}^{\text{fit}}$ is determined, its sample mean is estimated in step 10 and used to center the points in step 11. The entries of the RP matrices $\boldsymbol{R}_i$ are drawn i.i.d. from a t-distribution with mean 0 and variance $\frac{1}{d}$. This is the only difference from the original RP-Ens-EDA[4]. It is implemented by sampling from $t(0, 1, \nu)$ first, and then multiplying the samples by $\sqrt{\frac{\nu-2}{\nu d}}$ so that their variance becomes $1/d$. This choice for the variance is to ensure that we recover the original scale in step 17 without having to modify the scaling factor of the original RP-Ens-EDA. When d is large, $\boldsymbol{R}_i$ has nearly orthonormal rows, as a result of the concentration of norms in high dimensions, since the coordinates in each row are independent – this is a very generic property of high-dimensional probability spaces that both [4] and our method exploit. Therefore, premultiplying with $\boldsymbol{R}_i$ is almost like orthogonally projecting the points from the d-dimensional space to a k-dimensional subspace, which shortens the lengths of vectors by a factor of $\sqrt{\frac{k}{d}}$, and the standard deviation gets reduced by a factor of $\sqrt{M}$ after averaging. Therefore, the scaling factor needed to recover the original scale is $\sqrt{\frac{dM}{k}}$.

Step 14 projects the good samples, $\mathcal{Y}^{\boldsymbol{R}_i}$ down to the subspaces of dimension k, and then estimates the $k \times k$ covariance matrices in each subspace and samples N new points in each subspace using k-dimensional multivariate Gaussian distributions. Step 17 averages the obtained points from these subspaces to produce the new population $\mathcal{P}$. In addition, we will use elitism in practice, thereby the best fitness individual always survives to the next generation.

2.2 Why t-Distributed Random Projections

Let us restrict our attention to the procedure by which the new generation is created, as this is what distinguishes our algorithm from the vanilla multivariate EDA. We want to analyze the aggregated covariance that the ensemble of RP creates implicitly, which is in fact the covariance of the high-dimensional search distribution that the new generation follows.

Our analysis will allow quite general RP matrices, and only require their entries to be drawn i.i.d. from a 0-mean symmetric distribution having finite first four moments. This includes the Gaussian, all sub-Gaussians, and the family of t-distributions with the degree of freedom at least 5. We take advantage of this generality to develop our arguments and explain our choice for the t-distributions, which, for readers familiar with the random projection literature, may seem very unusual.

We start by computing the covariance of the new population in step 17 of Algorithm 1, conditional on fixing the random projection matrices $\boldsymbol{R}_i, i = 1 : M$. We will then condition on the sample of fit individuals and look at the effect of $\boldsymbol{R}_i, i = 1 : M$ by computing the expectation of this ensemble covariance with respect to $\boldsymbol{R}_i, i = 1 : M$. If M is large enough, this expectation approximates well the finite M case. Moreover, although not pursued here, tools from [24] may be used to determine the sufficient ensemble size.

Proposition 1. *Conditionally on $\boldsymbol{R}_i$, $i = 1 : M$, the new generation produced at step* 17 *of Algorithm* 1 *is i.i.d. Gaussian with mean $\boldsymbol{\mu}$ and $d \times d$ covariance of the following form*:

$$\boldsymbol{\Sigma}_{rp} = \frac{d}{kM} \sum_{i=1}^{M} \boldsymbol{R}_i^{\mathrm{T}} \boldsymbol{R}_i \boldsymbol{\Sigma} \boldsymbol{R}_i^{\mathrm{T}} \boldsymbol{R}_i, \tag{1}$$

where $\boldsymbol{\Sigma}$ is the maximum likelihood sample covariance of the original selected individuals in $\mathcal{P}^{\mathrm{fit}}$.

The proof is given in Appendix A1.

Next, we want to see the effect of the random $\boldsymbol{R}_i$'s on $\boldsymbol{\Sigma}_{rp}$. To this end, we condition on $\boldsymbol{\Sigma}$, and look at the expectation $E_{\boldsymbol{R}}[\boldsymbol{\Sigma}_{rp}]$.

We require the following definition.

Definition 1. *The excess kurtosis of a random variable x is defined as*:

$$K = \frac{E[x^4]}{E[x^2]^2} - 3.$$

We have the following result.

Theorem 1. *Let $\boldsymbol{R}$ be a $k \times d$ random matrix, $k < d$, with entries drawn i.i.d. from a symmetric distribution with* 0*-mean, finite first four moments, and excess kurtosis K. Let $\boldsymbol{\Sigma}$ be a $d \times d$ fixed positive semi-definite matrix with eigenvalues $\lambda_1, ..., \lambda_d$.*

$$E_{\boldsymbol{R}}[\boldsymbol{\Sigma}_{rp}] = \frac{1}{d}[(k+1)\boldsymbol{\Sigma} + Tr(\boldsymbol{\Sigma})\boldsymbol{I}_d + K \sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i], \tag{2}$$

where $\boldsymbol{A}_i$ is $d \times d$ diagonal matrices with their j-th diagonal elements being $\sum_{a=1}^{d} U_{ai}^2 U_{aj}^2$ and U_{ai} is the a-th entry of the i-th eigenvector of $\boldsymbol{\Sigma}$.

Proof. From (1), the expectation of $\boldsymbol{\Sigma}_{rp}$ is $\frac{d}{k} E_{\boldsymbol{R}}[\boldsymbol{R}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\Sigma} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{R}]$, which we compute using a result from [25], originally derived in a very different context (namely to analyze compressive least square regression).

Under the condition on $\boldsymbol{R}$ stated in Theorem 2, Lemma 2 from [25] proved that

$$E[\boldsymbol{R}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{\Sigma} \boldsymbol{R}^{\mathrm{T}} \boldsymbol{R}] = k \times E[R_{i,j}^2]^2 [(k+1)\boldsymbol{\Sigma} + Tr(\boldsymbol{\Sigma}) I_d + K \sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i], \tag{3}$$

where $\boldsymbol{A}_i$ is the diagonal matrices defined as in Theorem 1, and R_{ij} is a generic entry of matrix $\boldsymbol{R}$.

Now, by design, we have $E[R_{ij}^2] = \frac{1}{d}$. Replacing (3) into (1) completes the proof. □

Observe that from the definition of the matrices $\boldsymbol{A}_i$, and since $\boldsymbol{\Sigma}$ is positive semi-definite, λ_i is non-negative – consequently the sum in the last term in the brackets in (2), that is, $\sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i$, is always a diagonal matrix with non-negative diagonal entries.

Now, let us look at the multiplier of this term, K. In previous work[4] the entries of $\boldsymbol{R}$ were Gaussian or sub-Gaussian. In the Gaussian case, $K = 0$. Hence in that case the last term cancels out: $K \sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i = 0$. Nevertheless, what remains is a regularised version of the sample covariance estimate $\boldsymbol{\Sigma}$. This is always non-singular, thereby the search does not get stuck in a subspace of the search space.

However, if the entries of $\boldsymbol{R}$ are drawn from a sub-Gaussian distribution, we find that certain choices may be a bad idea — it depends on the excess kurtosis of the distribution.

Indeed, take the Rademacher distribution below, and note this is sub-Gaussian:

$$R_{ij} \sim_{i.i.d.} \begin{cases} 1, & \text{with probability } 1/2, \\ -1, & \text{with probability } 1/2. \end{cases}$$

Random matrices with i.i.d. entries from this distribution have been originally proposed by [26] as an implementation-friendly alternative to the Gaussian RP, and it was subsequently considered for RP-EDA-Ensemble optimisation in [4].

It is easy to check that for this distribution we have $E[R_{ij}^4] = 1$, and $E[R_{ij}^2] = 1$. Therefore, the excess kurtosis is $K = 1 - 3 = -2$. Consequently, $K \sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i = -2 \sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i$. This is a diagonal matrix with all non-positive diagonal entries. Therefore this term diminishes the regularisation effect of the term $Tr(\boldsymbol{\Sigma})\boldsymbol{I}_d$ in (2). It also shrinks the ensemble covariance matrix in comparison with that of a Gaussian RP ensemble.

We can go even further, and construct a simple example where the term $K \sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i$ cancels out the effect of $Tr(\boldsymbol{\Sigma})\boldsymbol{I}_d$ completely in some directions of the

search space, thereby we are left with maximum likelihood estimates in those directions that may be not sufficiently accurate due to the small number of high-fitness individuals relative to the high-dimensionality of the search space. We give such an example in Appendix A2.

Although such examples may seem contrived, the shrinking effect of distributions with negative kurtosis is undesirable for search, and instead we would like to use the insights of this analysis to come up with a better alternative.

If $K > 0$, then the term, $K\sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i$, is a diagonal positive semi-definite matrix; hence it always adds a non-negative quantity to the diagonals of the ensemble covariance. Hence, our idea is to use a family of distributions with positive kurtosis, and moreover, adapt K during the search. Based on the discussion above, this would enlarge the covariance $E_{\boldsymbol{R}}[\boldsymbol{\Sigma}_{rp}]$ adaptively to the necessary extent. Covariance adaptation has been a successful technique in the EDA literature[27], and our idea makes it feasible to implement it in new ways, specifically for large-scale EDA search through a more suitable random projection ensemble.

To achieve this, we have chosen the family of t-distributions with degree of freedom at least 5. Indeed, for this family of distributions the following holds.

Proposition 2. *The excess kurtosis of $t(0, 1, \nu)$ distribution with the degree of freedom ν is*:

$$K = \frac{6}{\nu - 4}, \quad \textit{provided } \nu > 4.$$

The proof is given in Appendix A3.

Corollary 1 (to Proposition 2). *Changing the variance leaves the excess kurtosis unchanged.*

The proof is immediate and given in Appendix A3 for completeness.

We replace this into Theorem 1, and obtain the following.

Theorem 2. *If $\boldsymbol{R}$ is sampled i.i.d. from a t-distribution with degree of freedom $\nu \geqslant 5$, mean 0 and variance $1/d$, then the ensemble covariance for the high-dimensional search distribution is*:

$$\begin{aligned} & E_{\boldsymbol{R}}[\boldsymbol{\Sigma}_{rp}] \\ = \ & \frac{1}{d}[(k+1)\boldsymbol{\Sigma} + Tr(\boldsymbol{\Sigma})\boldsymbol{I}_d + \frac{6}{\nu - 4}\sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i], \quad (4) \end{aligned}$$

where the matrices $\boldsymbol{A}_i$ are the same as defined in Theorem 1.

The Gaussian RP corresponds to $\nu \to \infty$, in which case the last term in the bracket in (4) vanishes: $\lim_{\nu\to\infty} \frac{6}{\nu-4}\sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i = 0$. In turn, by any finite choice of the degree of freedom parameter for the t-distribution we will be adding a positive semi-definite matrix to this covariance — this makes it larger and gives it more chance to explore the search space.

3 Adaptive Degree of Freedom Parameter

In this section we devise methods to adapt the parameter ν during the search.

Parameter setting methods are dichotomised into tuning and controlling[28]. Tuning means finding a good value by trial and error before running the algorithm and then fixing this value throughout the evolutionary process. While it may seem convenient, the trial-and-error phase consumes part of the budget of function evaluations. On the other hand, parameter control starts with an initial value which is then updated during the search, based on partial outcomes of the algorithm[28]. In other words, controlling tries to adapt the control parameters automatically in order to adjust the algorithm to the problem during the search[29]. Hence controlling methods need us to devise an algorithm that acts as an adaptation strategy. This is what we pursue in the remainder of this section.

In the sequel we present three different adaptation schemes. In each of these, the parameter we try to control is the degree of freedom of the t-distributed entries of our random projection matrices $(\boldsymbol{R}_i, i = 1 : M)$.

To simplify the computations we shall approximate $\boldsymbol{\Sigma}_{rp}$ by an upper bound on it in positive semi-definite ordering, which is independent of the eigenvectors of $\boldsymbol{\Sigma}$. Observing that $\sum_{i=1}^{d} \lambda_i \boldsymbol{A}_i \preceq Tr(\boldsymbol{\Sigma}) \cdot I_d$, we have:

$$E_{\boldsymbol{R}}[\boldsymbol{\Sigma}_{rp}] \preccurlyeq \frac{1}{d}[(k+1)\boldsymbol{\Sigma} + Tr(\boldsymbol{\Sigma})\boldsymbol{I}_d(1 + \frac{6}{\nu - 4})\boldsymbol{I}_d].$$

3.1 Adaptive Variance Scaling (AVS) Based Method

An adaptive variance scaling (AVS) method was originally proposed for EDA by Grahl *et al.*[27] In each generation, it decides how to update a parameter for the next generation based on the "difference in the best fitness" between the latest two consecutive generations. The original approach[27] is to scale the sample covariance matrix $\boldsymbol{\Sigma}$ multiplicatively, by a value $c \in [1, 10]$, where c is adapted (i.e., either increased or decreased) by a coefficient $\eta \in (0, 1)$ as follows. If the best fitness improved then c is increased as c/η in order to explore; otherwise c is decreased as $c\eta$ for exploitation. The authors of [27] put c to 10 if it becomes greater than 10 or

smaller than 1, in order to stimulate exploration while preventing a blow-up of the covariance.

Note in our context of RP ensemble based algorithm, we do not need to artificially introduce the c parameter as the original AVS does, as we can use the degree of freedom parameter ν to emulate the updating heuristic of the original AVS. The pseudo-code of this adaptation scheme is given in Algorithm 2.

Algorithm 2. Adaptive Variance Based Method (AVS)

Input: index of current generation t; degree of freedom ν^t; best fitness b^t, b^{t-1}; coefficient η

Output: degree of freedom ν^{t+1}

```
c ← 1 + 6/(ν^t − 4)
if b^t is no better than b^{t−1} then
    c ← cη //c gets smaller, ν gets larger
else
    c ← c/η //c gets larger, ν gets smaller
ν^{t+1} ← 6/(c−1) + 4
if ν^{t+1} < 5 or ν^{t+1} > 124 then
    ν^{t+1} ← 5 //to stimulate exploration
```

From (4) we have seen that smaller values for ν allow for exploration and larger ν-values encourage exploitation in tRP-Ens-EDA. We also observe that ν has a non-linear effect on the extent of enlarging the covariance of the new population. We set:

$$c = 1 + 6/(\nu - 4), \tag{5}$$

which ensures that $\nu \geqslant 5$ as required so that we can directly apply the same adaptation rules as the original AVS approach of [27] as follows. Suppose the tRP-Ens-EDA is at its t-th generation, and we denote by b^t and b^{t-1} the best fitness values of the latest two generations respectively. Recall, tRP-Ens-EDA uses the elitism, thereby we will never observe a worse-than previous value of the best fitness. Therefore, our AVS will decrease c (hence increase ν) for exploitation if b^t is no better than b^{t-1} before the elitism is applied, and increase c (hence decrease ν) for exploration if b^t is strictly better than b^{t-1}. This adaptation strategy is exactly the same as that originally proposed in [27], and the only difference is that it is now built into our RP-ensemble based EDA for large-scale searches.

Finally, from (5) we have:

$$\nu = 6/(c - 1) + 4, \tag{6}$$

and as in the original method of [27], we will also constrain the range of $\nu \in [5, 124]$, which corresponds to $c \in [1.05, 7]$. If ν goes beyond this range, then it is put to 5 to trigger exploration, which is the same strategy as in [27].

3.2 1/5-Success Rule Based Method

A classic alternative adaptation scheme is the well-known 1/5-success rule (or 1/5 rule). It is one of the oldest adaptation heuristics, and it is widely used in evolutionary strategies (ES)[30]. The 1/5 rule does not look at the improvement in the best fitness, but instead it looks at the overall fitness improvement of the population. It measures this as the following success probability:

$$P_s = \frac{\text{number of individuals with improved fitness}}{\text{population size}}.$$

If $P_s > 1/5$ then it is said that an improvement is detected. In our case, if such improvement is detected then we decrease ν for exploration; otherwise we increase ν in order to exploit — in the same manner as we did in AVS in Subsection 3.1.

However, in ESs it is easy to compute P_s, because the individuals can be compared with their parents after a mutation; but in EDA there is no explicit connection between a particular individual and its descendant in the next generation, as the new individuals are sampled from a probability distribution that the whole previous generation contributed to. To apply the 1/5 rule in EDAs we use the following heuristic. At the t-th generation, given the fitness values of the population with N individuals $\boldsymbol{f}^t$, and the fitness values of the individuals of the previous generation $\boldsymbol{f}^{t-1}$, we sort both $\boldsymbol{f}^t$ and $\boldsymbol{f}^{t-1}$, and compare between the corresponding elements f_i^t and f_i^{t-1}, where $i \in \{1, \ldots, N\}$. This way we count the number of fitness values f_i^t that are better than the corresponding f_i^{t-1}, and use this number in the expression of P_s.

The pseudo-code of this adapted 1/5 rule for tRP-Ens-EDA is given in Algorithm 3. The algorithm first counts the number of improved individuals between the sorted fitness vectors of the two latest generations, and then calculates the success probability P_s. If $P_s \leqslant 1/5$, then the adaptive method decreases the value of c (defined as before in (5)) and thus increases ν (defined in (6)) for the purpose of exploitation. Otherwise c is increased, and thus ν is decreased, for exploration. This method shares with AVS the schemes of updating c and thresholding ν, namely it increases or decreases c with a coefficient $\eta \in (0, 1)$, keeps $\nu \in [5, 124]$ and resets ν

to 5 if it goes beyond the range in order to stimulate exploration.

Algorithm 3. 1/5-Success Rule Based Adaptive Method

Input: population size N; index of current generation t; degree of freedom ν^t, sorted fitnesses of the latest 2 populations $\boldsymbol{f}^t$ & $\boldsymbol{f}^{t-1}$; coefficient η
Output: degree of freedom ν^{t+1}

$c \leftarrow 1 + \frac{6}{\nu^t - 4}$
$N_p \leftarrow 0$ //counter of improved individuals
for $i = 1 : N$ **do**
  if f_i^t is better than f_i^{t-1} **then**
    $N_p \leftarrow N_p + 1$
$P_s \leftarrow N_p / N$ //success probability
if $P_s \leqslant 1/5$ **then**
  $c \leftarrow c\eta$ //c gets smaller, ν gets larger
else
  $c \leftarrow c/\eta$ //c gets larger, ν gets smaller
$\nu^{t+1} \leftarrow \frac{6}{c-1} + 4$
if $\nu^{t+1} < 5$ or $\nu^{t+1} > 124$ **then**
  $\nu^{t+1} \leftarrow 5$ //to stimulate exploration

3.3 Adaptive Degree of Freedom (ADF)

Our third adaptation method will be a scheme where some backtracking is possible. The parameter updating decisions are made after trying out a list of alternative choices and observing their effects on the best fitness within each generation. We drew inspiration from [29] initially, and devise our own adaptation rules to fit the problem, bearing in mind that our parameter of interest must obey $\nu \geqslant 5$. The pseudo-code of our proposed adaptive method is given in Algorithm 4.

The adaptive method given in Algorithm 4 maintains a list of L different alternative values of ν in ascending order, and in each generation it creates a separate trial (mock) new-generation using tRP-Ens-EDA with each of the L different ν values on the current list. The list L is initialised by taking the first ν values to be $6, 7, \ldots, 6 + L - 1$ (all close to the smallest allowable value). The list is then updated based on the current best value, to be tried in the next generation. If the current list of ν-values produces no difference to the fitness, then the list of values is spread out more (by decreasing the smallest and increasing all other ν-values). Otherwise the algorithm updates the list of values to be tried in the next generation by bringing them closer to the current best ν-value.

This adaptation scheme is direct and generic; however it has a tuning parameter that presents a trade-off: the length of the list of the ν-values to try. Obviously, the longer the list, the higher the chance to find a good value within the generation, but the price is a quicker drainage of the budget of fitness evaluations. We experimented with list lengths of $L = 2$ and $L = 5$ as representatives of this algorithm in our early work[18], and had found $L = 2$ to be a good value. Therefore we employ this in the experiments reported here, and will refer to our adaptation scheme as ADF(2df).

Algorithm 4. Adaptive Degree of Freedom (ADF)

Input: ϑ (initial list of ν-values in ascending order)
Output: ν

$L \leftarrow |\vartheta|$
for $i = 1 : L$ **do**
  Run steps 11–17 and 5 of Algorithm 1 with ν_i
  $f_i \leftarrow$ best fitness from step 5 of Algorithm 1
if $\min(\boldsymbol{f}) == \max(\boldsymbol{f})$ **then**
  $\vartheta_1 \leftarrow \max(5, round(\vartheta_1/2))$
  for $j = 2 : L$ **do**
    $\vartheta_j \leftarrow \vartheta_j \times 2$
else
  $\ell \leftarrow \arg\min(\boldsymbol{f})$; $\nu \leftarrow \vartheta_\ell$
  if $\ell == 1$ **then**
    $\vartheta_1 \leftarrow \max(5, round(\vartheta_1/2))$
    for $j = 2 : L$ **do**
      $\vartheta_j \leftarrow \max(5, round((\vartheta_{j-1} + \vartheta_j)/2))$
  else
    for $j = 1 : \ell - 1$ **do**
      $\vartheta_j \leftarrow \max(5, round((\vartheta_j + \vartheta_{j+1})/2))$
    $\vartheta_\ell \leftarrow \vartheta_\ell + \max(5, round(\vartheta_\ell/2))$
    for $j = \ell + 1 : L$ **do**
      $\vartheta_j \leftarrow \max(5, round((\vartheta_{j-1} + \vartheta_j)/2))$

It follows from the construction of this method that ADF uses L times as many fitness evaluations per generation as a fixed choice of ν would do, while the two adaptation heuristics presented in the previous subsections keep the same per-generation budget of fitness evaluations, since they make adaptation decisions sequentially, without the possibility of backtracking a parameter value after observing its effect on the fitness. Therefore we expect ADF to be more advantageous in terms of reaching a better fitness value when a sufficiently large budget is available, whereas the sequential heuristics may be preferable when the budget is relatively small. This will be assessed in more detail in Section 4.

4 Experiments

Initial experiments have indicated that the best choice of degree of freedom parameter is problem-dependent[18]. Therefore we turn our attention to the adaptation approaches. We conduct a thorough set of experiments to assess our tRP-Ens-EDA with adaptation schemes on a battery of large-scale multi-modal benchmark problems, from the 1 000-

dimensional CEC2010–2012 competition test suite, as described in [31]. All problems are minimisations. Table 1 lists the test functions used in our experiments.

Table 1. 1 000-Dimensional Multi-Modal Test Functions from the CEC2010–2012 Collection

Problem	Name
F1	Shifted Rastrigin's function
F2	Shifted Ackley's function
F3	Single-group Shifted & m-rotated Rastrigin's function
F4	Single-group Shifted & m-rotated Ackley's function
F5	$\frac{D}{2m}$-group Shifted & m-rotated Rastrigin's function
F6	$\frac{D}{2m}$-group Shifted & m-rotated Ackley's function
F7	$\frac{D}{2m}$-group Shifted & m-dimensional Rosenbrock's function
F8	$\frac{D}{2m}$-group Shifted & m-rotated Rastrigin's function
F9	$\frac{D}{m}$-group Shifted & m-rotated Ackley's function
F10	$\frac{D}{m}$-group Shifted & m-dimensional Rosenbrock's function
F11	Fully nonseparable Rosenbrock

In each experiment we initialize the population randomly in the search space. All the results in comparative experiments are obtained through fresh runs of the methods, except MA-SW-Chains whose results were obtained from the literature.

4.1 Results and Analysis

4.1.1 Comparative Assessment of Our Adaptation Schemes

In the first set of experiments we are interested in finding out whether our adaptive schemes can outperform the existing RP-Ens-EDA, and to learn about their respective advantages and disadvantages.

Table 2 provides a statistical analysis of comparisons on the 1 000-dimensional benchmarks. For each function, we compare our three different adaptive methods and the existing RP-Ens-EDA of [4] ($\nu = \infty$). We report the average and the standard deviation (Std) of the best fitness achieved with the evaluation budget of 6×10^5 fitness evaluations, which is considered as medium budget in the benchmark problem description.

We perform a two-tailed t-test for each pair of methods involved in this comparison, and the symbols in the rightmost column of the table record statistically significant differences detected at the 0.05 level as follows. A symbol "+" indicates that the method in the corresponding row performed significantly better than the method in the corresponding column. Likewise, "–" means that the method in the corresponding row performed significantly worse than the method in the corresponding column. Recall that all the test problems are minimisation problems, thereby lower fitness means better performance. The symbol ∅ appears where statistical comparison is not applicable. Positions left blank in the last column mean that no statistically significant difference was detected between the pair of methods tested.

Table 2. Statistical comparison of Our Adaptation Schemes and the Existing RP-Ens-EDA[4] ($\nu = \infty$) on 1 000-Dimensional Test Functions, with a Budget of 6×10^5 Function Evaluations

	Method	Mean	Std	t Test vs.		
				AVS	1/5	ADF(2df)
F1	AVS	750.318 0	57.995 8	∅		-
	1/5	758.636 0	48.709 0		∅	-
	ADF(2df)	579.000 0	24.000 0	+	+	∅
	$\nu = \infty$	784.210 0	76.017 0			-
F2	AVS	2.518E−13	4.390E−15	∅		
	1/5	2.527E−13	4.390E−15		∅	
	ADF(2df)	2.480E−13	4.760E−15			∅
	$\nu = \infty$	2.537E−13	4.810 4E−15			
F3	AVS	1.308E+07	3.486E+06	∅	+	+
	1/5	2.149E+07	9.803E+06	-	∅	+
	ADF(2df)	3.040E+08	9.490E+06	-	-	∅
	$\nu = \infty$	1.239 7E+07	3.186 4E+06		+	+
F4	AVS	674.339 0	126.295 0	∅	+	-
	1/5	1.417E+04	2 023.350 0	-	∅	-
	ADF(2df)	548.000 0	75.000 0	+	+	∅
	$\nu = \infty$	117.780 0	17.151 0	+	+	+
F5	AVS	765.787 0	52.446 2	∅		-
	1/5	785.701 0	55.950 6		∅	-
	ADF(2df)	601.000 0	26.600 0	+	+	∅
	$\nu = \infty$	832.180 0	62.543 0	-	-	-
F6	AVS	34.346 6	8.330 9	∅		
	1/5	35.513 0	10.622 9		∅	
	ADF(2df)	35.000 0	9.490 0			∅
	$\nu = \infty$	41.664 0	8.700 3	-	-	-
F7	AVS	1.330 6E+06	7.169E+04	∅		+
	1/5	1.330 5E+06	6.865E+04		∅	+
	ADF(2df)	1.400E+06	5.670E+04	-	-	∅
	$\nu = \infty$	1.444 2E+06	5.394 5E+04	-	-	-
F8	AVS	765.966 0	46.523 1	∅		-
	1/5	742.170 0	45.863 1		∅	-
	ADF(2df)	599.000 0	24.600 0	+	+	∅
	$\nu = \infty$	802.110 0	49.957 0	-	-	-
F9	AVS	78.552 1	14.088 6	∅		+
	1/5	78.364 0	13.620 3		∅	+
	ADF(2df)	91.800 0	15.700 0	-	-	∅
	$\nu = \infty$	90.481 0	17.178 0	-	-	
F10	AVS	4.363E+04	8 810.530 0	∅		
	1/5	4.011E+04	1.106E+04		∅	
	ADF(2df)	4.610E+04	9.72E+03			∅
	$\nu = \infty$	5.410 5E+04	1.087 7E+04	-	-	-
F11	AVS	1 022.220 0	63.462 2	∅		
	1/5	987.983 0	1.076 8		∅	+
	ADF(2df)	1.10E+03	67.600 0		-	∅
	$\nu = \infty$	1 614.700 0	249.440 0	-	-	-

From the results listed in Table 2 we observe the followings.

• In nine (out of 11) of the test problems, no statistically significant difference is detected between the AVS and the 1/5 adaptation heuristics.

• In the remaining two problems, AVS is statistically superior to 1/5. On further investigation we observe that the 1/5 approach has periods of stagnation where AVS is able to progress. This is most likely because the definition of 1/5 in the context of EDA is rather ad hoc.

• AVS wins with statistical significance against the existing RP-Ens-EDA in seven (out of 11) test problems, and only loses in one (F4). The remaining three cases are ties.

• ADF performs on-par with AVS at this budget size: four wins and three losses against AVS, and it outperforms the existing RP-Ens-EDA in seven of the functions; it is outperformed by RP-Ens-EDA in two functions (F4 and F3), and two ties.

We then run the two most successful adaptation methods, ADF(2df) and AVS, versus the existing RP-Ens-EDA for a larger budget size of 3×10^6. Fig.1 shows the detailed trajectories, in order to visualize the behaviour of these methods and gain insight into the reasons for the observed differences. In Fig.1 the best fitness is plotted against the elapsed number of function evaluations, averaged over 25 independent restarts. For clarity of visual inspection, these are zoomed into the interesting ranges of the axes: the fitness values of the first few generations are not shown, and the plotting halts when no more interesting change occurs.

We see from the trajectory plots on Fig.1 that, given a sufficient budget, the direct adaptation method ADF(2df) often reaches a better fitness value with a slight delay. This makes good sense intuitively too, based on the construction of the algorithm that spends part of its resources on exploring. Interestingly, it often goes through a period of stagnation before it wins over. For instance, on F3, with the medium budget size reported in Table 2 it was outperformed by all the other methods, including the previously existing RP-Ens-EDA, while from the trajectory plot we see that it does win eventually, given a larger budget, whereas the Gaussian variant makes no further progress. Indeed, we see from the trajectories in Fig.1 that the Gaussian RP-Ens-EDA is more prone to premature convergence than our adaptive tRP-Ens-EDA variants. More differences between ADF and AVS will be seen in multi-comparison tests in the presence of other competing state-of-the-art algorithms in Subsection 4.1.2.

4.1.2 *Comparison with State-of-the-Art Methods on 1 000-Dimensional Problems*

We compare our best performing adaptive degree of freedom methods tRP-Ens-EDA(ADF) and tRP-Ens-EDA(AVS) with the existing state-of-the-art on the 1 000-dimensional CEC2010–2012 competition multimodal benchmark problems.

Tables 3 and 4 and Tables 5 and 6 summarize the results from experiments with two different budged sizes respectively, from 25 repeated independent runs each. The average and the standard deviation of the best fitness values achieved are reported. By the Friedman's test and a subsequent multi-comparison, the means and the standard deviations of the best performing algorithms are marked in bold. The two budget sizes tested are 6×10^5 and 3×10^6, that is, a medium size budget and a large budget, according to the definitions in the benchmark suite.

The results are summarized in Tables 3–6. We see from these tables and Fig.1 that our tRP-Ens-EDA(ADF) is most competitive in the high budget setting.

• In the medium budget setting (0.6×10^6 function evaluations), tRP-Ens-EDA(ADF) wins with statistical significance over all competitors tested in two of the test functions in the medium budget setting. AVS wins over all competitors tested in one function. In fact, tRP-Ens-EDA(AVS) does better than tRP-Ens-EDA(ADF) in terms of achieving lower fitness on a larger number of functions, but statistically significant out-performance is not observed in most of these cases in the multi-comparison test.

• In the large budget setting (3×10^6 function evaluations), tRP-Ens-EDA(ADF) wins over all the other methods tested, including the winner of the CEC2010–2012 competitions[15], in four test functions (out of 11 tested).

We can therefore conclude that a direct adaptation scheme like ADF turns most beneficial when we allow slightly larger budget sizes. In medium or limited budget settings we can recommend the AVS method.

Naturally, one cannot expect any method to perform best on all problems in the light of well known "no free lunch" theorems. However, the observed empirical results together with their statistical significance analysis support the conclusion that the methods presented in this paper are competitive with state-of-the-art large-scale optimisation heuristics, and therefore

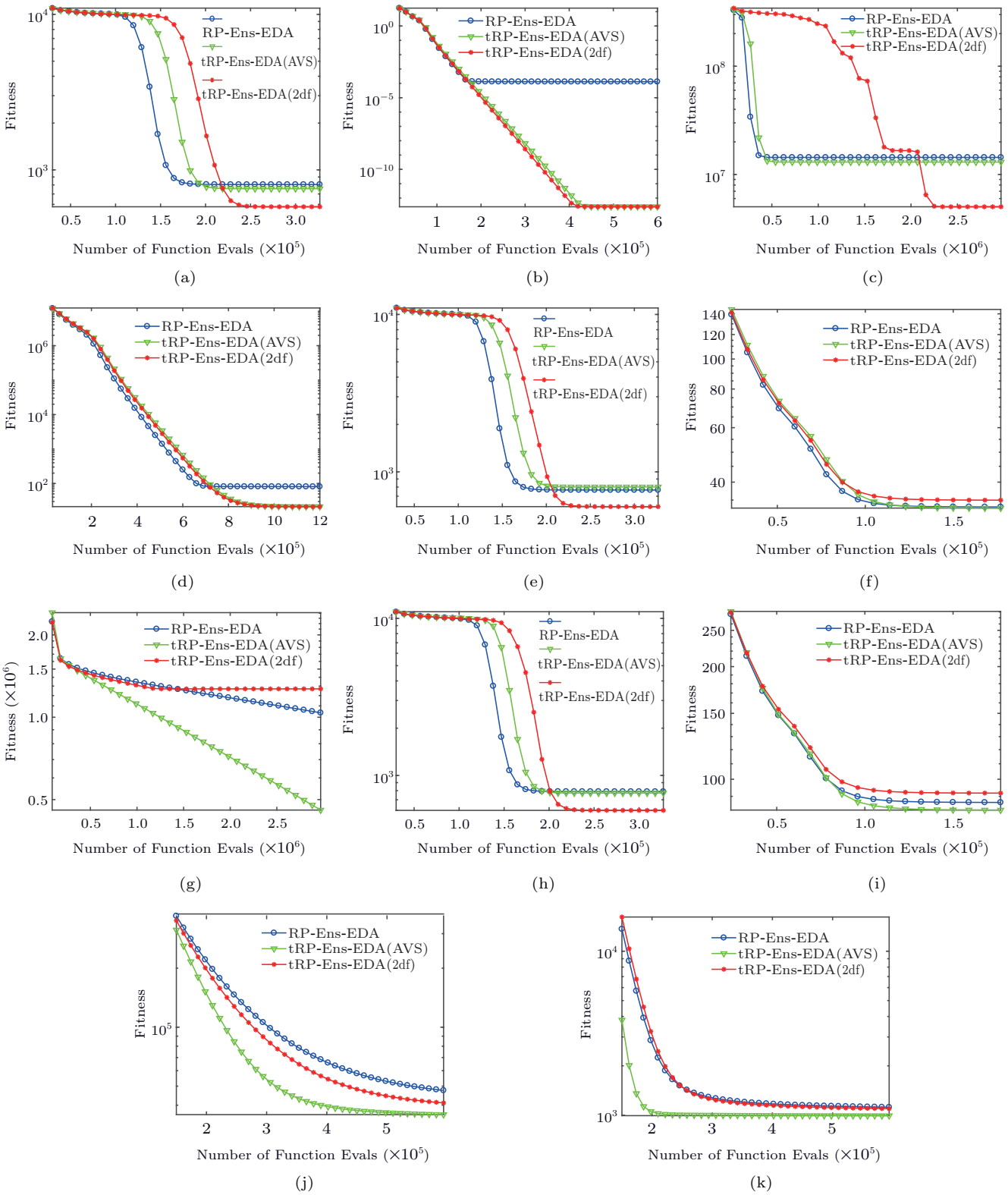

Fig. 1. Fitness trajectories of the RP-Ens-EDA, tRP-Ens-EDA(AVS) and tRP-Ens-EDA(2df) for a maximum of 3×10^6 function evaluations. For a statistical analysis of significance of the differences observed halfway through the budget, i.e., after 6×10^5 function evaluations (see Table 3 and Table 4). (a) Function F1. (b) Function F2. (c) Function F3. (d) Function F4. (e) Function F5. (f) Function F6. (g) Function F7. (h) Function F8. (i) Function F9. (j) Function F10. (k) Function F11.

Table 3. Mean of Best Fitness Values from 25 Independent Runs in Comparison with State-of-the-Art Under Equal Budget of 0.6×10^6 Function Evaluations

	MA-SW-Chains	CCVIL	MLCC	sep-CMA-ES	EDA-MCC	tRP-Ens-EDA(AVS)	RP-Ens-EDA	tRP-Ens-EDA(ADF)
F1	2.67E+03	**1.77E+01**	6.22E+03	8.15E+03	1.24E+03	7.57E+02	8.05E+02	5.79E+02
F2	3.84E+00	1.77E+01	1.45E+01	2.15E+01	3.86E+00	2.50E−13	1.30E−04	2.48E−13
F3	2.17E+08	9.79E+08	4.19E+08	1.41E+08	1.84E+07	1.30E+07	1.44E+07	3.04E+08
F4	8.14E+04	2.12E+07	1.41E+07	1.04E+07	**1.78E+01**	6.69E+02	2.51E+02	5.48E+02
F5	3.22E+03	1.31E+04	1.10E+04	8.36E+03	1.38E+03	8.01E+02	7.66E+02	**6.01E+02**
F6	3.83E+01	2.32E+02	2.16E+02	2.25E+02	3.42E+01	2.30E+01	3.33E+01	3.50E+01
F7	**4.34E+03**	1.84E+11	1.21E+09	5.02E+05	1.89E+08	1.33E+06	1.43E+06	1.40E+06
F8	3.19E+03	1.86E+04	1.84E+04	8.13E+03	1.49E+03	7.72E+02	7.89E+02	**5.99E+02**
F9	1.02E+02	4.26E+02	4.14E+02	4.30E+02	7.92E+01	8.27E+01	8.67E+01	9.18E+01
F10	**5.53E+03**	4.76E+11	2.67E+10	2.19E+04	1.65E+09	3.58E+04	4.74E+04	4.61E+04
F11	1.21E+03	4.48E+11	5.26E+10	1.16E+03	2.16E+09	**9.94E+02**	1.12E+03	1.10E+03

Table 4. Standard Deviation of Best Fitness Values from 25 Independent Runs in Comparison with State-of-the-Art under Equal Budget of 0.6×10^6 Function Evaluations

	MA-SW-Chains	CCVIL	MLCC	sep-CMA-ES	EDA-MCC	tRP-Ens-EDA(AVS)	RP-Ens-EDA	tRP-Ens-EDA(ADF)
F1	1.63E+02	**1.40E+01**	1.75E+03	3.63E+02	4.77E+01	6.76E+01	7.17E+01	2.40E+01
F2	2.13E−01	1.97E+00	1.90E+00	2.07E−02	3.66E−01	6.39E−015	2.24E−06	4.76E−15
F3	8.56E+07	7.60E+07	1.76E+08	3.00E+07	3.39E+06	4.09E+06	3.29E+06	9.49E+06
F4	2.84E+05	3.76E+05	8.29E+06	2.33E+06	**1.90E−01**	1.30E+02	2.68E+01	7.50E+01
F5	1.85E+02	3.29E+02	1.97E+03	3.85E+02	6.28E+01	9.17E+01	5.91E+01	**2.66E+01**
F6	7.23E+00	8.08E−01	1.79E+01	5.05E+00	3.12E+00	9.48E+00	7.31E+00	9.49E+00
F7	**3.21E+03**	1.23E+11	2.15E+09	8.66E+04	7.86E+07	5.45E+04	4.72E+04	5.67E+04
F8	1.46E+02	3.76E+02	1.02E+04	3.49E+02	7.53E+01	3.79E+01	4.85E+01	**2.46E+01**
F9	1.42E+01	9.41E−01	1.22E+01	2.88E+00	8.38E+00	1.29E+01	1.43E+01	1.57E+01
F10	**3.94E+03**	2.86E+11	5.31E+10	1.12E+04	3.97E+08	5.43E+03	1.12E+04	9.72E+03
F11	1.42E+02	2.64E+11	7.80E+10	1.34E+02	4.97E+08	**1.55E+01**	1.04E+02	6.76E+01

Table 5. Mean of Best Fitness Values from 25 Independent Runs in Comparison with State-of-the-Art under Equal Budget of 3×10^6 Function Evaluations

	MA-SW-Chains	CCVIL	MLCC	sep-CMA-ES	EDA-MCC	tRP-Ens-EDA(AVS)	RP-Ens-EDA	tRP-Ens-EDA(ADF)
F1	**2.10E−14**	4.00E−07	5.57E−01	5.68E+03	1.24E+03	7.56E+02	8.05E+02	5.94E+02
F2	7.28E−13	4.00E−07	9.88E−13	2.15E+01	3.86E+00	2.52E−13	1.30E−04	**2.46E−13**
F3	1.68E+08	5.51E+08	3.84E+08	1.19E+08	1.84E+07	1.30E+07	1.44E+07	**4.99E+06**
F4	8.14E+04	4.14E+05	1.62E+07	6.39E+06	**1.78E+01**	2.13E+01	8.05E+01	3.20E+01
F5	2.07E+03	1.43E+03	3.43E+03	6.28E+03	1.38E+03	8.01E+02	7.66E+02	**5.96E+02**
F6	3.80E+01	**7.44E+00**	1.98E+02	2.12E+02	3.42E+01	3.30E+01	3.33E+01	3.10E+01
F7	1.25E+03	2.98E+11	2.08E+03	**2.94E+02**	1.89E+08	4.51E+05	1.03E+06	1.00E+06
F8	2.74E+03	2.78E+03	7.11E+03	6.76E+03	1.49E+03	7.72E+02	7.89E+02	**6.05E+02**
F9	9.98E+01	**1.31E+01**	3.76E+02	4.21E+02	7.92E+01	8.27E+01	8.67E+01	9.18E+01
F10	1.30E+03	6.42E+11	7.09E+03	**9.16E+02**	1.65E+09	2.87E+04	4.02E+04	4.05E+04
F11	1.07E+03	1.75E+11	2.05E+03	9.04E+02	2.16E+09	9.89E+02	1.04E+03	1.06E+03

make a worthwhile addition to the practitioner's toolbox for use in difficult high-dimensional problems that no specialised algorithm is known to be able to solve. Moreover, our algorithm construction comes with a good intuition of its working strategy, which consists in a controlled increase of exploration in a similar spirit as the adaptive variance scaling idea previously pursued in [27], but now supporting much larger-scale problems through an ensemble of RPs.

4.1.3 Scalability of ADF(2df)

Since our ADF(2df) is the most expensive in terms of function evaluations per generation, our final set of experiments is to assess its scalability. We measure the number of function evaluations needed (search cost) to reach a specified distance from the global optimum

Table 6. Standard Deviation of Best Fitness Values from 25 Independent Runs in Comparison with State-of-the-Art under Equal Budget of 3×10^6 Function Evaluations

	MA-SW-Chains	CCVIL	MLCC	sep-CMA-ES	EDA-MCC	tRP-Ens-EDA(AVS)	RP-Ens-EDA	tRP-Ens-EDA(ADF)
F1	**1.99E−14**	0.00E+00	2.21E+00	4.89E+02	4.77E+01	6.76E+01	7.17E+01	1.76E+01
F2	3.40E−13	6.32E−07	3.70E−12	1.08E−01	3.66E−01	6.30E−15	2.33E−06	**4.58E−15**
F3	1.04E+08	1.55E+08	6.93e+07	2.92E+07	3.39E+06	4.09E+06	3.29E+06	**1.32E+06**
F4	2.84E+05	6.54E+05	4.97E+06	3.85E+06	**1.90E−01**	2.82E−02	2.13E+00	5.32E−01
F5	1.44E+02	6.34E+01	8.72E+02	2.51E+02	6.28E+01	9.17E+01	5.91E+01	**1.71E+01**
F6	7.35E+00	**2.41E+00**	6.98E−01	6.16E+00	3.12E+00	9.48E+00	7.31E+00	1.09E+01
F7	5.72E+02	8.57E+11	7.27E+02	**9.20E+01**	7.86E+07	6.82E+04	5.74E+04	5.74E+04
F8	1.22E+02	8.78E+01	1.34E+03	2.76E+02	7.52E+01	3.79E+01	4.84E+01	**2.74E+01**
F9	1.40E+01	**2.92E+00**	4.71E+01	1.59E+01	8.39E+00	1.29E+01	1.43E+01	1.57E+01
F10	4.36E+02	2.19E+12	4.77E+03	**1.94E+02**	3.96E+08	5.13E+03	9.86E+03	9.82E+03
F11	7.29E+01	8.77E+11	1.80E+02	3.91E+01	4.97E+08	1.06E+01	5.08E+01	5.45E+01

as the problem dimension varies. We fix the value to reach (VTR) to 10^{-3}, and vary the dimensionality of the problem $d \in [50, 1\,000]$. We repeat the same experiment with three other choices of VTR: 10^{-2}, 10^2 and 10^3 to avoid biasing the conclusions towards a particular choice of the VTR. Throughout, the budget was fixed to 3×10^6 thereby the algorithm stops either when the budget is exhausted or upon reaching the VTR. All other parameters are kept at their default values.

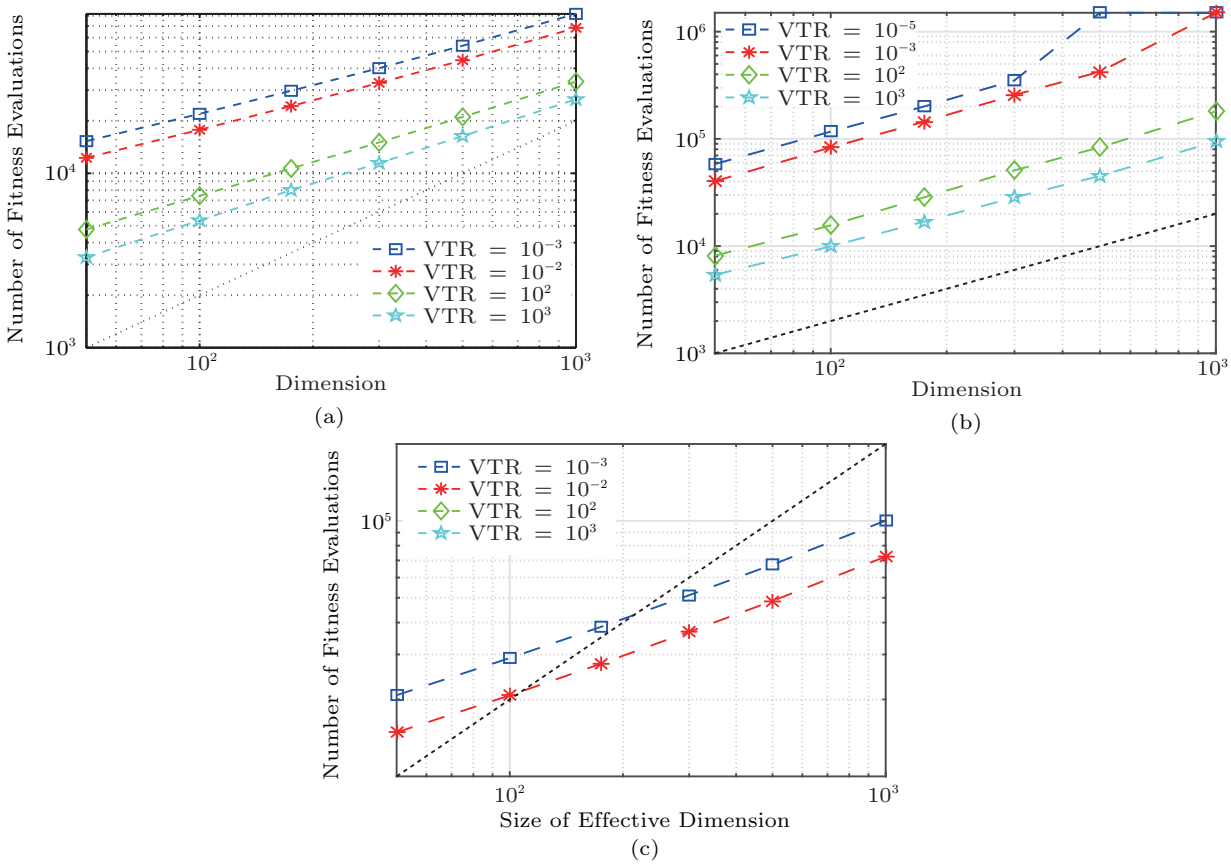

Fig.2. Number of function evaluations taken by successful runs of our tRP-Ens-EDA with ADF to reach a pre-specified "value to reach (VTR)" as the problem dimensionality is varied in $d \in [50, 1\,000]$. The markers represent averages computed from 25 independent repetitions. The black dotted line corresponds to linear scaling (slope = 1). (a) Shifted sphere function. (b) Shifted elliptic function. (c) Shifted Ackley function.

Fig.2 displays the average number of function evaluations as computed from the successful runs out of 25 independent repetitions for each problem, for each dimension tested. That is, runs terminated by exhausting the budget do not contribute to these plots.

From Fig.2 we observe a linear shape on the scalability measurements (dashed lines on the log-log plots). The black dotted line corresponds to the slope of 1 (that is, linear scaling) for reference. This means that our tRP-Ens-EDA with ADF scheme scales at most polynomially (with a degree indicated by the slope of lines) in the dimension of the problem, which matches the best scaling known for sep-CMA-ES[7] and the original RP-Ens-EDA[4].

5 Conclusions

We devised new adaptive approaches for high-dimensional continuous black-box optimisation which extend and generalize the random projection ensemble based EDA by means of using unconventional random matrices. We analytically showed that this results in enlarging the covariance of the high-dimensional search distribution, which then increases exploration. Hence our adaptive schemes in fact implement novel means of covariance adaptation. We demonstrated superior performance against the original version RP-Ens-EDA on both large and medium budget instances, and with a state-of-the-art performance in comparison with a number of existing methods on 1 000-dimensional problems. Future work remains to investigate other aggregation schemes, and to extend the method to multi-objective problems. A very interesting question for future work would be to analyze the dynamics and the running time of such high-dimensional EDA optimisation algorithms.

Acknowledgments The authors would like to thank Qi Xu (M.Sc. student at Birmingham) for implementing the AVS method and carrying out part of the experiments for Table 5 and Table 6.

References

[1] Larrañaga P, Lozano J A. Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation (2002 edition). Kluwer Academic Publishers, 2002.

[2] Yuan B, Gallagher M. On the importance of diversity maintenance in estimation of distribution algorithms. In *Proc. the 2005 Genetic and Evolutionary Computation Conference*, June 2005, pp.719-729.

[3] Friedman J H. An overview of predictive learning and function approximation. In *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Cherkassky V, Friedman J H, Wechsler H (eds.), Springer-Verlag Berlin Heidelberg, 1994, pp.1-61.

[4] Kabán A, Bootkrajang J, Durrant R J. Toward large-scale continuous EDA: A random matrix theory perspective. *Evolutionary Computation*, 2016, 24(2): 255-291.

[5] Dong W, Yao X. Covariance matrix repairing in Gaussian based EDAs. In *Proc. the 2007 IEEE Congress on Evolutionary Computation*, September 2007, pp.415-422.

[6] Paul T K, Iba H. Linear and combinatorial optimizations by estimation of distribution algorithms. In *Proc. the 9th MPS Symposium on Evolutionary Computation*, Jan. 2002, pp.99-106.

[7] Ros R, Hansen N. A simple modification in CMA-ES achieving linear time and space complexity. In *Proc. the 10th International Conference on Parallel Problem Solving from Nature*, September 2008, pp.296-305.

[8] Bosman P. On empirical memory design, faster selection of Bayesian factorizations and parameter-free Gaussian EDAs. In *Proc. the 2009 Genetic and Evolutionary Computation Conference*, July 2009, pp.389-396.

[9] de Bonet J S, Isbell C L, Viola P. MIMIC: Finding optima by estimating probability densities. In *Proc. the 1997 International Conference on Neural Information Processing Systems*, December 1996, pp.424-430.

[10] Bosman P A N, Thierens D. An algorithmic framework for density estimation based evolutionary algorithms. Technical Report, Utrecht University, 1999. https://homepages.cwi.nl/∼bosman/publications/1999_analgorithmicframework.pdf, June 2019.

[11] Armañanzas R, Inza I, Santana R *et al.* A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 2008, 1: Article No. 6.

[12] Weicker K, Weicker N. On the improvement of coevolutionary optimizers by learning variable interdependencies. In *Proc. the 1999 Congress on Evolutionary Computation*, July 1999, pp.1627-1632.

[13] Dong W, Chen T, Tiňo P, Yao X. Scaling up estimation of distribution algorithm for continuous optimisation. *IEEE Transaction of Evolutionary Computation*, 2013, 17(6): 797-822.

[14] Yang Z, Tang K, Yao X. Multilevel cooperative convolution for large scale optimization. In *Proc. IEEE World Congress on Computational Intelligence*, June 2008, pp.1663-1670.

[15] Molina D, Lozano M, Sánchez A M, Herrera F. Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. *Soft Computing*, 2011, 15(11): 2201-2220.

[16] Shin S Y, Cho D Y, Zhang B T. Function optimization with latent variable models. In *Proc. the 3rd International Symposium on Adaptive Systems*, March 2001, pp.145-152.

[17] Sanyang M L, Kabán A. REMEDA: Random embedding EDA for optimising functions with intrinsic dimension. In *Proc. the 14th International Conference on Parallel Problem Solving from Nature*, September 2016, pp.859-868.

[18] Sanyang M L, Kabán A. Heavy tails with parameter adaptation in random projection based continuous EDA. In *Proc. the 2015 IEEE Congress on Evolutionary Computation*, May 2015, pp.2074-2081.

[19] Yao X, Liu Y, Lin G. Evolutionary programming made faster. *IEEE Transaction on Evolutionary Computation*, 1999, 3(2): 82-102.

[20] Gao B, Wood I. TAM-EDA: Multivariate t distribution, archive and mutation based estimation of distribution algorithm. *ANZIAM Journal*, 2014, 54: 720-746.

[21] Gao B. Estimation of distribution algorithms for single- and multi-objective optimization [Ph.D. Thesis]. School of Mathematics and Physics, The University of Queensland, 2014.

[22] Sanyang M L, Durrant R J, Kabán A. How effective is Cauchy-EDA in high dimensions? In *Proc. the 2016 IEEE Congress on Evolutionary Computation*, July 2016, pp.3409-3416.

[23] Dang D C, Lehre P K, Nguyen P T H. Level-based analysis of the univariate marginal distribution algorithm. *Algorithmica*, 2018, 81(2): 668-702.

[24] Kabán A. On compressive ensemble induced regularisation: How close is the finite ensemble precision matrix to the infinite ensemble? In *Proc. the 2017 International Conference on Algorithmic Learning Theory*, October 2017, pp.617-628.

[25] Kabán A. New bounds for compressive linear least squares regression. In *Proc. the 17th International Conference on Artificial Intelligence and Statistics*, April 2014, pp.448-456.

[26] Achlioptas D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 2003, 66(4): 671-687.

[27] Grahl J, Bosman P A, Rothlauf F. The correlation-triggered adaptive variance scaling IDEA. In *Proc. the 2006 Genetic and Evolutionary Computation Conference*, July 2006, pp.397-404.

[28] Eiben A E, Hinterding R, Michalewicz Z. Parameter control in evolutionary algorithms. *IEEE Transaction on Evolutionary Computation*, 1999, 3(2): 124-141.

[29] Wong Y Y, Lee K H, Leung K S, Ho C W. A novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Computing*, 2003, 7(8): 506-515.

[30] Beyer H G, Schwefel H P. Evolution strategies — A comprehensive introduction. *Natural Computing*, 1999, 1(1): 3-52.

[31] Tang K, Li X D, Suganthan P N, Yang Z, Weise T. Benchmark functions for the CEC'2010 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, 2012. http://www.tflsgo.org/assets/cec2018/cec2-013-lsgo-benchmark-tech-report.pdf, June 2019.

[32] Abramowitz M, Stegun I A, Morse P M (eds.). Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables. National Bureau of Standards, 1964.

Momodou L. Sanyang received his Ph.D. degree in computer science from the University of Birmingham, Birmingham, in 2017. Prior to that, he received his B.Sc. degree in mathematics and physics from the University of The Gambia (UTG), Serekunda, The Gambia, in 2006, and his M.Sc. degree in information systems and applications from the "National" Tsing Hua University, Hsinchu, in 2010. He is currently a senior lecturer in computer science at the University of The Gambia (UTG), and a honorary research fellow at the University of Birmingham, UK. His research interests include machine learning and data mining, dimensionality reduction, random projections, evolutionary computation and black box optimisation in high-dimensional settings.

Ata Kabán is a professor in computer science at the University of Birmingham, Birmingham. Her main research interests are in statistical machine learning, data mining, and black-box optimisation in high-dimensional settings. She holds a Ph.D. degree in computer science (2002) from the University of Paisley, Scotland, and a Ph.D. degree in musicology (2000) from the Music Academy "Gh. Dima" of Cluj-Napoca, Romania. She is a member of the IEEE CIS Technical Committee on Data Mining and Big Data Analytics, and a vice-chair of the IEEE CIS Task Force on High Dimensional Data Mining. She has been awarded a 5-year EPSRC Fellowship (2017–2022), and a part-time Turing Fellowship (2019–2021).

Appendix

A.1

Proof of Proposition 1. Recall from step 14 of Algorithm 1 that the set of projected points in the i-th subspace is:

$$\mathcal{Y}^{\boldsymbol{R}_i} = \{\boldsymbol{R}_i(\boldsymbol{x}_1 - \boldsymbol{\mu}), \boldsymbol{R}_i(\boldsymbol{x}_2 - \boldsymbol{\mu}), ..., \boldsymbol{R}_i(\boldsymbol{x}_N - \boldsymbol{\mu})\}.$$

Conditionally on $\boldsymbol{R}_i$, the sample covariance matrix of $\mathcal{Y}^{\boldsymbol{R}_i}$ is:

$$\boldsymbol{\Sigma}^{\boldsymbol{R}_i} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{R}_i(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{R}_i(\boldsymbol{x}_n - \boldsymbol{\mu}))^{\mathrm{T}} = \boldsymbol{R}_i\boldsymbol{\Sigma}\boldsymbol{R}_i^{\mathrm{T}}.$$

Thereby the samples in step 16 are $\boldsymbol{y}_1^{\boldsymbol{R}_i}, \ldots, \boldsymbol{y}_N^{\boldsymbol{R}_i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^{\boldsymbol{R}_i})$.

To find the distribution of the individuals in $\mathcal{P}^{\text{new}}$ at step 17, we look at the n-th individual:

$$\boldsymbol{x}_n := \sqrt{\frac{dM}{k}}[\frac{1}{M}\sum_{i=1}^{M}\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{y}_n^{\boldsymbol{R}_i}] + \boldsymbol{\mu}. \tag{A1}$$

Conditionally on $\boldsymbol{R}_i, i = 1 : M$, this is a linear combination of independent Gaussian random variables, which is again a Gaussian②. Hence, $\boldsymbol{x}_n$ is Gaussian distributed, with mean $\boldsymbol{\mu}$ (since $\boldsymbol{y}_n^{\boldsymbol{R}_i}$ has zero mean), and we compute its covariance below.

In (7), denote $\boldsymbol{A}_i := \sqrt{\frac{dM}{k}}\frac{1}{M}\boldsymbol{R}_i^{\mathrm{T}}$, then we have that $\boldsymbol{y}_n^{\boldsymbol{R}_i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_i\boldsymbol{\Sigma}\boldsymbol{R}_i^{\mathrm{T}})$. Therefore,

$$\boldsymbol{A}_i\boldsymbol{y}_n^{\boldsymbol{R}_i} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{A}_i\boldsymbol{R}_i\boldsymbol{\Sigma}\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{A}_i^{\mathrm{T}}), \tag{A2}$$

for all $n = 1, \ldots N$. Replacing $\boldsymbol{A}_i$ in (A2), we have the i-th summand in (A1):

$$\boldsymbol{A}\boldsymbol{y}_n^{\boldsymbol{R}_i} \sim \mathcal{N}(\boldsymbol{0}, \sqrt{\frac{dM}{k}}\frac{1}{M}\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{\Sigma}\boldsymbol{R}_i^{\mathrm{T}}\sqrt{\frac{dM}{k}}\frac{1}{M}\boldsymbol{R}_i),$$

which simplifies to

$$\mathcal{N}(\boldsymbol{0}, \frac{d}{kM}\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{\Sigma}\boldsymbol{R}_i^{\mathrm{T}}\boldsymbol{R}_i).$$

Finally, the summation operation yields the ensemble covariance in the d-dimensional search space as stated in (1). □

A.2

Here we give an example to demonstrate that random matrices with negative excess kurtosis can cancel the regularisation effect of the ensemble in some directions of the search space.

Let $R_{ij} \sim \{-1, +1\}$ with a probability of 1/2 each. Hence we have $E[R_{ij}^4] = 1$, $E[R_{ij}^2] = 1$, and therefore the excess kurtosis is $K = 1 - 3 = -2$.

For the sake of this example, suppose that $rank(\boldsymbol{\Sigma}) = 2$, thereby $\lambda_1 \geqslant \lambda_2 > 0, \lambda_3 = \ldots \lambda_d = 0$, and suppose the first two eigenvectors of $\boldsymbol{\Sigma}$ are

$$\begin{aligned}\boldsymbol{u}_1 &= \left(1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots, 0\right),\\ \boldsymbol{u}_2 &= \left(-1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots, 0\right),\end{aligned}$$

and the remaining eigenvectors have their first two coordinates 0. This is well defined, since one can easily verify that $\boldsymbol{u}_1 \perp \boldsymbol{u}_2$, $\|\boldsymbol{u}_1\| = \|\boldsymbol{u}_2\| = 1$. Denote by $\boldsymbol{U}$ the matrix having $\boldsymbol{u}_i$ in its rows, with entries U_{ai}, where $a, i = 1, \ldots, d$.

Now, we have:

$$\begin{aligned}&\boldsymbol{A}_1\\ &= \begin{pmatrix} \sum_{a=1}^{d} U_{a1}^4 & 0 & 0 & \ldots & 0\\ 0 & \sum_{a=1}^{d} U_{a1}^2U_{a2}^2 & 0 & \ldots & 0\\ 0 & 0 & \sum_{a=1}^{d} U_{a1}^2U_{a3}^2 & \ldots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \ldots & \sum_{a=1}^{d} U_{a1}^2U_{ad}^2 \end{pmatrix}\\ &= \begin{pmatrix} 1/2 & 0 & 0 & \ldots & 0\\ 0 & 1/2 & 0 & \ldots & 0\\ 0 & 0 & 0 & \ldots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix} = \boldsymbol{A}_2.\end{aligned}$$

Then, for this example we get:

$$\begin{aligned}K\sum_{i=1}^{d}\lambda_i\boldsymbol{A}_i &= -2(\lambda_1\boldsymbol{A}_1 + \lambda_2\boldsymbol{A}_2)\\ &= -Tr(\boldsymbol{\Sigma})\begin{pmatrix} 1 & 0 & 0 & \ldots & 0\\ 0 & 1 & 0 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix}.\end{aligned}$$

This cancels the effect of the term $Tr(\boldsymbol{\Sigma})\boldsymbol{I}_d$ in the first two coordinate directions, leaving $E_{\boldsymbol{R}}[\boldsymbol{\Sigma}_{rp}]$ to be a multiple of maximum likelihood covariance estimates in those directions.

A.3

Proof of Proposition 2. Let $k \in \{2, 4\}$ and $x \sim t(0, 1, \nu)$ a t-distributed random variable, then by definition,

$$E[x^k] = \int_{-\infty}^{\infty} x^k f(x)\mathrm{d}x,$$

where $f(x)$ is the pdf of the t-distribution,

$$f(x) = c\left(1 + \frac{x^2}{\nu}\right)^{\frac{-1}{2}(\nu+1)}, \tag{A3}$$

where

$$c = \frac{1}{\sqrt{\nu}B(\frac{\nu}{2}, \frac{1}{2})} = \frac{\Gamma(\nu/2 + 1/2)}{\sqrt{\nu}\ \Gamma(\nu/2)\Gamma(1/2)}, \tag{A4}$$

and $B(\cdot, \cdot)$ is the beta function[32]. Observing that $f(x) = f(-x)$, we can re-write the integral as:

$$E[x^k] = 2\int_{0}^{\infty} x^k f(x)\mathrm{d}x. \tag{A5}$$

②Assume $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}_x, \boldsymbol{\Sigma}_x)$ and $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{m}_y, \boldsymbol{\Sigma}_y)$, then $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} + \boldsymbol{c} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{m}_x + \boldsymbol{B}\boldsymbol{m}_y + \boldsymbol{c}, \boldsymbol{A}\boldsymbol{\Sigma}_x\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{B}\boldsymbol{\Sigma}_y\boldsymbol{B}^{\mathrm{T}})$.

Writing out the expression of f as defined in (A3), we have that (A5) equals

$$E[x^k] = 2c \int_0^\infty x^k \left(1 + \frac{x^2}{\nu}\right)^{\frac{-1}{2}(\nu+1)} \mathrm{d}x.$$

Now, we make the change of variable: $t = \frac{x^2}{\nu}$. Hence $x = (\nu t)^{\frac{1}{2}}$, and we get:

$$\begin{aligned} E[x^4] &= c\nu^{\frac{k+1}{2}} \int_0^\infty t^{\frac{k+1}{2}-1}(1+t)^{-\frac{1}{2}-\frac{\nu}{2}} \mathrm{d}t \\ &= c\nu^{\frac{k+1}{2}} \int_0^\infty t^{\frac{k+1}{2}-1}(1+t)^{-(\frac{k+1}{2})-(\frac{\nu-k}{2})} \mathrm{d}t, \end{aligned}$$

where we added and subtracted $k/2$ in the exponent of the last term so that the integral represents a beta function[32]:

$$B(m+1, n+1) = \int_0^\infty u^m (1+u)^{-(m+n)-2} \mathrm{d}u,$$

with $m := \frac{k}{2} - \frac{1}{2}$ and $n := \frac{\nu}{2} - 3$. Therefore we can write:

$$\begin{aligned} E[x^k] &= c\nu^{\frac{k+1}{2}} B\left(\frac{k+1}{2}, \frac{\nu-k}{2}\right) \\ &= c\nu^{\frac{k+1}{2}} \frac{\Gamma(\frac{k+1}{2})\Gamma(\frac{\nu-k}{2})}{\Gamma(\frac{1+\nu}{2})}. \end{aligned}$$

We need to evaluate this for $k = 4$ and $k = 2$. Replacing the expression of c from (A4), we get for $k = 4$ the following:

$$E[x^4] = \nu^2 \frac{\Gamma(\frac{5}{2})\Gamma(\frac{\nu-4}{2})}{\Gamma(\frac{\nu}{2})\Gamma(\frac{1}{2})},$$

which is finite provided that $\nu - 4 > 0$ (since $\Gamma(0) = \infty$).

To simplify, we use the identities $\Gamma(\frac{5}{2}) = \frac{3}{4}\sqrt{\pi}$ and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$[32]. Therefore,

$$E[x^4] = \frac{\frac{3\nu^2}{4}\ \Gamma(\frac{\nu-4}{2})}{\Gamma(\frac{\nu}{2})}.$$

Furthermore, by a property of the Gamma function, $\Gamma(x) = (x-1)\Gamma(x-1)$[32], we have after substitution and simplification:

$$E[x^4] = \frac{3\nu^2}{(\nu-2)(\nu-4)}.$$

Analogously, for $k = 2$, we arrive at

$$E[x^2] = \frac{\nu}{\nu-2}.$$

Therefore, the excess kurtosis is:

$$K = \frac{E[x^4]}{(E[x^2])^2} - 3 = \frac{6}{\nu-4}. \qquad \square$$

Proof of Corollary 1. Let $c > 0$ be a constant. Then $c \cdot x$ has variance $c^2 var(x)$. The excess kurtosis of $c \cdot x$ is

$$\frac{E[(c\cdot x)^4]}{(E[(c\cdot x)^2])^2} - 3.$$

Taking the constant out, we have

$$\frac{c^4 E\left[x^4\right]}{c^4 E\left[x^2\right]^2} - 3 = \frac{E\left[x^4\right]}{E\left[x^2\right]^2} - 3,$$

thus the excess kurtosis did not change. $\square$