A Simple Yet Efficient Evolution Strategy for Large-Scale Black-Box Optimization

Zhenhua Li and Qingfu Zhang, *Fellow, IEEE*

***Abstract*—We propose an evolution strategy algorithm using a sparse plus low rank model for large-scale optimization in this paper. We first develop a rank one evolution strategy using a single principal search direction. It is of linear complexity. Then we extend it to multiple search directions, and develop a rank-*m* evolution strategy. We illustrate that the principal search direction accumulates the natural gradients with respect to the distribution mean, and acts as a momentum term. Further, we analyze the optimal low rank approximation to the covariance matrix, and experimentally show that the principal search direction can effectively learn the long valley of the function with predominant search direction. Then we investigate the effects of Hessian on the algorithm performance. We conduct experiments on a class of test problems and the CEC'2010 LSGO benchmarks. The experimental results validate the effectiveness of our proposed algorithms.**

***Index Terms*—Evolution path, evolution strategy, low rank model, principal search direction.**

I. Introduction

THE COVARIANCE matrix adaptation evolution strategy (CMA-ES) is a very popular and successful evolutionary algorithm for continuous optimization [1], [2]. During the search, it maintains and evolves a Gaussian distribution for generating new solutions. It approximates the contours of the objective function by that of its Gaussian distribution, which increases the probability of generating successful solutions. Recent years have witnessed significant increasing needs for solving large-scale optimization problems with over thousands or more variables. As CMA-ES considers all the variable covariances, it has to store $O(n^2)$ (where n is the number of variables) elements for its Gaussian distribution at each generation. Moreover, to generate new solutions, it has to do eigen-decomposition on the $n \times n$ covariance matrix and the resultant computational complexity is $O(n^3)$. Although an outdated covariance matrix can be used in practical implementation [1], the complexity is still $O(n^2)$ on average. The high space and time complexity makes the CMA-ES unsuitable for large-scale optimization.

A very natural way to remedy this shortcoming is to restrict the $n \times n$ covariance matrix to some specific forms to reduce the number of model parameters. Among various matrix forms which can be used, a simplest one is

$$\mathbf{C} = \alpha \mathbf{I} + \beta \mathbf{v}\mathbf{v}^T \tag{1}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix, and $\mathbf{v} \in \mathbb{R}^n$ is a principal search direction, and $\alpha, \beta > 0$ are control parameters. In this paper, we call it the rank one model. Many difficult optimization problems have a long valley in their fitness landscapes [3]. For example, in the case of minimizing $f(x_1, x_2) = x_1^2 + 100x_2^2$, the changes in x_2 produce much larger variations in the objective value than the same changes in x_1. It may become even more difficult to optimize nonconvex functions with long valleys, such as the Rosenbrock function with a parabolic-shaped valley [4]. If the principal search direction in the rank one model (1) can effectively capture the valley, it may solve the problem with low complexity.

A main research issue in using the rank one model (1) is how to effectively adapt the principal search direction and mutation strength. To the best of our knowledge, only main vector adaptation evolution strategy (MVA-ES) [5] and rank one natural evolution strategy (R1-NES) [6] exploit this form of covariance matrix. MVA-ES updates the main vector as the covariance matrix in CMA-ES, and adapts the mutation strength using cumulative step-size adaptation (CSA) [7]. CSA exploits the standard Gaussian distribution, and does not use the search information along the main vector. This would result in performance loss. R1-NES adapts the distribution parameters using natural gradient [8], and may lose population diversity quickly on some objective functions with ill-conditioned Hessian matrices.

In this paper, we study how to use the rank one model (1) and a generalized sparse plus low rank model, to design simple and yet efficient algorithms for handling large-scale optimization problems. Our main contributions are as follows.

1) Using the rank one model (1), we develop an algorithm, called rank one evolution strategy (R1-ES). It is of linear time complexity $O(n)$ and low space complexity. We further generalize it and design an algorithm with multiple principal search directions, called rank-*m* evolution strategy (Rm-ES).
2) We illustrate that the principal search direction in R1-ES accumulates the natural gradients with respect to the distribution mean, and acts as a momentum term.

Manuscript received July 18, 2016; revised December 29, 2016, May 14, 2017, and August 5, 2017; accepted October 10, 2017. Date of publication October 23, 2017; date of current version September 28, 2018. This work was supported by the ANR/RGC Joint Research Scheme under Grant A-CityU101/16. *(Corresponding author: Zhenhua Li.)*

The authors are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: zhenhua.li@my.cityu.edu.hk; qingfu.zhang@cityu.edu.hk).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2017.2765682

3) We study the optimal low rank approximation to the general covariance matrix. Furthermore, we experimentally demonstrate that the principal search direction in R1-ES can efficiently capture the long valley of the problem. We also investigate the effects of Hessian on the performance of R1-ES.
4) We experimentally investigate the algorithms behaviors and performances, compared with related algorithms. The experimental results show that Rm-ES with $m = 2$ performs competitively to limited memory CMA (LM-CMA) [9]. We also evaluate Rm-ES with a restart strategy on the CEC'2010 LSGO benchmarks, compared with state-of-the-art algorithms for large-scale optimization. The results validate the effectiveness of the proposed algorithm.

This paper is organized as follows. Section II presents the background and related work. In Section III, we propose R1-ES and extend it to Rm-ES with multiple principal search directions. Section IV analyzes the proposed algorithms. Section V presents the experimental results. Section VI concludes this paper.

II. Background and Related Work

We consider the following continuous black-box optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{2}$$

where no gradient information available, and the only information accessible is the objective value $f(\mathbf{x})$.

A. Full Matrix Adaptation Methods

CMA-ES is one of the most successful evolution strategies that uses the full covariance matrix of the Gaussian distribution [1], [2]. It adapts the covariance matrix to capture the information of dependencies and scales [1]. At each generation, to sample new solutions from distribution $\mathcal{N}(\mathbf{m}_t, \sigma_t^2\mathbf{C}_t)$, eigen-decomposition is conducted $\mathbf{C}_t = \mathbf{B}\mathbf{D}^2\mathbf{B}^T$, where $\mathbf{B}$ is an orthogonal matrix, and $\mathbf{D}$ is a diagonal matrix. Then a new solution is generate as

$$\mathbf{x} = \mathbf{m}_t + \sigma_t\mathbf{B}\mathbf{D}\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

An alternative is to conduct Cholesky decomposition [10]. Then the covariance matrix is updated by both rank-1 and rank-μ terms [2]. The rank-1 update involves the accumulation of movements of the distribution mean, and the rank-μ term is the weighted maximum likelihood estimation of covariance using the current selected solutions. By increasing the variances along successful search directions, it is more likely to generate promising solutions.

The computational complexity of eigen-decomposition is $O(n^3)$. In practical implementations, the covariance matrix is updated at every generation, while the eigen-decomposition is conducted every $O(n)$ generations. During this procedure, the outdated orthogonal matrix $\mathbf{B}$ and diagonal matrix $\mathbf{D}$ are used to generate new solutions, and current search directions are not used immediately [11].

Instead of explicitly adapting and decomposing the covariance matrix, Cholesky CMA-ES [12] maintains and adapts a Cholesky factor. It considers a symmetric rank one update to the covariance matrix, and obtains a rank one Cholesky update. To use the evolution path which involves history search information, it has to additionally maintain the inverse Cholesky factor [13]. In [14], the Cholesky factor is restricted to be upper triangular and does not need to maintain the corresponding inverse Cholesky factor. In [15], a new evolution path is constructed to replace the inverse vector of the evolution path, simplifying the algorithm by removing the inverse Cholesky factor. The matrix adaptation method [16] removes the evolution path, and obtains an approximated update methods for the mutation matrix $\mathbf{A}$.

A class of natural evolution strategies (NESs) can be obtained by taking the natural gradients of the expected fitness function on the distribution parameters [17], [18]. The natural gradient represents the steepest direction on the manifold of Gaussian distributions equipped with Fisher information matrix as metric [8]. The distribution parameters are updated along the natural gradients estimated using the currently sampled solutions.

Using a full matrix enables the algorithm to learn the rotated coordinate system and scalings. Generally, maintaining and evolving a full matrix requires a time and space complexity of $O(n^2)$, as the sampling procedure involves matrix-vector multiplications. Thus, it becomes more difficult to maintain and adapt the full matrix as the number of variables increases.

B. Large-Scale Variants

Typically, algorithms using a full matrix cannot scale well as the number of variables increases. Hence, it is necessary to simplify the mutation matrix to handle large-scale optimization problems. By simplifying the structure of the covariance matrix and reducing the number of distribution parameters, we can generate new solutions efficiently.

A simple model is to restrict the covariance matrix to the rank one model (1). It aims to capture the most important search direction and parameter dependencies with low complexity. MVA-ES [5] first tried to exploit this rank one model. The main vector is adapted by the evolution path, and mutation strength is adapted using CSA [1]. CSA does not make use of the information along the main vector, which may cause a performance loss. A similar idea is used in R1-NES [6], where the distribution parameters are updated by estimated natural gradients. It does not make use of the accumulation of search directions, and may suffer a performance loss on some ill-conditioned objective functions.

Some variants use more search directions. L-CMA-ES [19] maintains the m largest eigenvalues and corresponding eigenvectors of the covariance matrix, while keeping all other eigenvalues identical. It uses a low rank singular value decomposition method to update the largest eigenvalues and eigenvectors [20], and its time complexity is $O(m^2n)$. LM-CMA [9], [21] uses $m < n$ search directions to reconstruct the Cholesky factor of the covariance to sample new solutions.

The reconstruction follows the idea of L-BFGS, and its computational complexity is $O(mn)$.

Another commonly used model is to restrict the covariance matrix to be diagonal to exploit the problem separability. The separable CMA-ES uses a diagonal covariance to detect the variable scalings along axes. It achieves good performances on separable objective functions [22]. This idea is also used in NES [23]. The VD-CMA, derived from the information geometric optimization (IGO) framework [24], exploits the scalings along axes as well as a search direction [25]. It is extended to use multiple directions [26] and online adaptation of the direction number [26]. As these methods focus on the variable scalings along axes and abandon the parameter dependencies, their performances would significantly degenerate on nonseparable objective functions [27].

The cooperative co-evolution [28] method is commonly used in large-scale optimization to exploit a block-diagonal dependency structure. It detects the parameter dependencies before the optimization procedure, and decomposes a solution into a number of subcomponents accordingly [29]. Then each subcomponent is optimized individually using optimizers, such as CMA-ES [30], estimation distribution algorithms [31], and differential evolution [32]. Various decomposition methods have been proposed [29]. As the dependencies among subcomponents are neglected, the algorithm performance would degenerate significantly if the decomposition procedure does not capture the parameter dependency structure appropriately. In the cases of Gaussian distributions, the time complexity of these algorithms generally depends quadratically on the number of variables of each subcomponent, and linearly on the number of subcomponents [30].

Instead of using a fixed parameter dependency structure, the random projection estimation distribution algorithm [33] makes use of an ensemble of random projections on low dimension subspaces to exploit the parameter dependencies randomly. It performs well on some nonseparable objective functions, and scales quadratically on the dimension.

III. Proposed Algorithms

We propose to simplify the covariance matrix as

$$\mathbf{C} = \alpha\mathbf{S} + \mathbf{L} \tag{4}$$

where $\mathbf{S}$ is a sparse matrix, $\mathbf{L}$ is a symmetric low rank matrix, and $\alpha > 0$. When $\mathbf{S} = \mathbf{I}$, and $\mathbf{L}$ is of rank one, it becomes rank one model. Further, if set $\mathbf{L}$ to be rank m, we have

$$\mathbf{C} = \alpha\mathbf{I} + \sum_{i=1}^{m} \beta_i \mathbf{v}_i \mathbf{v}_i^T \tag{5}$$

where $\mathbf{v}_i$ $(i = 1, \ldots, m)$ are m principal search directions. In the following, we first develop R1-ES using rank one model, and then generalize it to rank-m model.

A. R1-ES

We first propose to set the covariance matrix at generation t as

$$\mathbf{C}_t = (1 - c_{\text{cov}})\mathbf{I} + c_{\text{cov}}\mathbf{p}_t\mathbf{p}_t^T \tag{6}$$

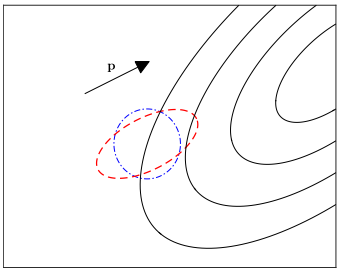

Fig. 1. Black solid circles are the contours of the objective function, the blue dashed circle is the contour of probability density function (PDF) with the identity covariance matrix, and the red dashed ellipse is the contour of the PDF with covariance matrix (6).

where $\mathbf{p}_t$ is the principal search direction, and $c_{\text{cov}} \in (0, 1)$ is the changing rate of covariance matrix. This model aims at capturing the most important search direction and parameter dependencies.

R1-ES maintains and evolves the following parameters at generation t.

1) $\mathbf{m}_t \in \mathbb{R}^n$: The distribution mean.
2) $\mathbf{p}_t \in \mathbb{R}^n$: The principal search direction.
3) $\sigma_t > 0$: The mutation strength.
4) $s_t \in \mathbb{R}$: The cumulative rank rate.
5) $\mathbf{x}_{best}$: The best solution found so far.

The distribution parameters $\mathbf{m}_t$, $\mathbf{p}_t$, and σ_t define the following Gaussian distribution model:

$$\mathcal{N}\left(\mathbf{m}_t, \sigma_t^2\mathbf{C}_t\right) \tag{7}$$

where $\mathbf{C}_t$ is given by (6). It should be noted that the covariance matrix is not explicitly stored in the optimization process. In the following, we give the details of the major steps of R1-ES.

1) Sampling New Solutions: R1-ES samples λ new solutions at each generation. In this paper, a new solution is generated as follows:

$$\mathbf{x} = \mathbf{m}_t + \sigma_t\left(\sqrt{1 - c_{\text{cov}}}\mathbf{z} + \sqrt{c_{\text{cov}}}r\mathbf{p}_t\right) \tag{8}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a n-dimensional standard Gaussian random vector, and $r \sim \mathcal{N}(\mathbf{0}, 1)$ is a random variable independent of $\mathbf{z}$. This sampling technique has been analyzed in [1] and used in [5].

This sampling procedure only involves scalar-vector multiplications, and its complexity is $O(n)$. If $\mathbf{p}_t$ learns the long valley well, one can expect, as shown in Fig. 1, that it is more likely to generate solutions with low objective values.

2) Selection: The λ newly sampled solutions are evaluated by the objective function. Then the objective values are sorted as

$$f(\mathbf{x}_{1:\lambda}) \leqslant f(\mathbf{x}_{2:\lambda}) \leqslant \ldots \leqslant f(\mathbf{x}_{\lambda:\lambda}) \tag{9}$$

where $\mathbf{x}_{i:\lambda}$ is the ith best solution. The best μ solutions are selected to update the algorithm parameters. If $\mathbf{x}_{i:\lambda}$ is better than $\mathbf{x}_{\text{best}}$, it replaces $\mathbf{x}_{\text{best}}$. We denote $F_{t+1} = [f(\mathbf{x}_{1:\lambda}), \ldots, f(\mathbf{x}_{\mu:\lambda})]$ the fitness of selected solutions. It is used to adapt the mutation strength.

3) Update the Distribution Mean: The best μ selected solutions are used to compute the distribution mean for the next generation as follows:

$$\mathbf{m}_{t+1} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}. \tag{10}$$

The weights are designed such that $w_1 \geqslant \cdots \geqslant w_\mu > 0$ for emphasizing better solutions, and $\sum_{i=1}^{\mu} w_i = 1$.

4) Update the Principal Search Direction: The principal search direction is updated as

$$\mathbf{p}_{t+1} = (1-c)\mathbf{p}_t + \sqrt{c(2-c)}\sqrt{\mu_{\text{eff}}}\frac{\mathbf{m}_{t+1} - \mathbf{m}_t}{\sigma_t} \tag{11}$$

with changing rate $c \in (0, 1)$ and $\mu_{\text{eff}} = 1/\sum_{i=1}^{\mu} w_i^2$. The factor $\sqrt{\mu_{\text{eff}}}$ normalizes $(\sqrt{\mu_{\text{eff}}}/\sigma_t)(\mathbf{m}_{t+1} - \mathbf{m}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_t)$, if $x_{i:\lambda}$ in (10) are randomly selected.

Clearly, $\mathbf{p}_{t+1}$ is a weighted combination of the previous principal search direction $\mathbf{p}_t$ and the current movement direction of the distribution mean $(\mathbf{m}_{t+1} - \mathbf{m}_t)/\sigma_t$. Thus, $\mathbf{p}_{t+1}$ is a weighted combination of all the previous mean movement directions. It is called the evolution path in [1].

The setting of the coefficients in (11) follows the so-called stationarity condition for evolution paths [1]. If $\mathbf{p}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_t)$ and $x_{i:\lambda}$ in (10) are randomly selected, then $\mathbf{p}_{t+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_t)$ as well.

5) Rank-Based Adaptation for Mutation Strength: As CSA exploits standard Gaussian distribution, it is unsuitable for R1-ES. We propose a rank-based success rule (RSR) to adapt the mutation strength. It works as follows.

1) Set $F = F_t \cup F_{t+1}$ and sort all the elements in F in ascending order. The ranks of the ith best solutions from F_t and F_{t+1} in F are denoted by $R_t(i)$ and $R_{t+1}(i)$, respectively.
2) Compute the rank difference

$$q = \frac{1}{\mu}\sum_{i=1}^{\mu} w_i (R_t(i) - R_{t+1}(i)) \tag{12}$$

where the weights w_i are the same as used in (10), and the factor $1/\mu$ is used to normalize $q \in [-1, 1]$.
3) Compute the cumulative rank rate

$$s_{t+1} = (1-c_s)s_t + c_s\left(q - q^*\right) \tag{13}$$

where $q^* \in (0, 0.5)$ is a constant target ratio, $c_s > 0$ is a constant changing rate, and $s_0 = 0$.
4) Adapt the mutation strength as

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{s_{t+1}}{d_\sigma}\right) \tag{14}$$

where $d_\sigma \geqslant 1$ is a damping factor.

The rank difference q measures how much the current parents are better or worse than the previous ones in terms of the rank. If $q > 0$, the current selected solutions generally rank before their counterparts of the previous generation, and hence F_{t+1} is generally better than F_t. The cumulative rank rate s is used to average $(q - q^*)$ over generations to alleviate the effects of randomness.

Algorithm 1 R1-ES

1: **Initialize:** $\mathbf{m}_0, \sigma_0, \mathbf{p}_0 = \mathbf{0}, \mathbf{x}_{\text{best}} = \mathbf{m}_0, F_0, s_0 = 0, t = 0$
2: **repeat**
3: **for** $i = 1$ to λ **do**
4: $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad r_i \sim \mathcal{N}(\mathbf{0}, 1)$
5: $\mathbf{x}_i = \mathbf{m}_t + \sigma_t(\sqrt{1-c_{cov}}\mathbf{z}_i + \sqrt{c_{cov}}r_i\mathbf{p}_t)$
6: **if** $f(\mathbf{x}_i) < f(\mathbf{x}_{\text{best}})$ **then**
7: $\mathbf{x}_{\text{best}} = \mathbf{x}_i$
8: **end if**
9: **end for**
10: sort $\mathbf{x}_i$ as $f(\mathbf{x}_{1:\lambda}) \leqslant f(\mathbf{x}_{2:\lambda}) \leqslant \cdots \leqslant f(\mathbf{x}_{\lambda:\lambda})$
11: $\mathbf{m}_{t+1} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$
12: $\mathbf{p}_{t+1} = (1-c)\mathbf{p}_t + \sqrt{c(2-c)\mu_{\text{eff}}}\frac{\mathbf{m}_{t+1} - \mathbf{m}_t}{\sigma_t}$
13: $R_t, R_{t+1} \leftarrow$ ranks of F_t, F_{t+1} in $F_t \cup F_{t+1}$
14: $q = \frac{1}{\mu}\sum_{i=1}^{\mu} w_i\big(R_t(i) - R_{t+1}(i)\big)$
15: $s_{t+1} = (1-c_s)s_t + c_s(q - q^*)$
16: $\sigma_{t+1} = \sigma_t \exp\left(\frac{s_{t+1}}{d_\sigma}\right)$
17: $t = t + 1$
18: **until** stopping criterion is met
19: **return**

The parameters for RSR are discussed in the following.

1) q^**:* The target ratio is a major factor for determining how the mutation strength is changed. In principle, the optimal setting for q^* depends on the problem. When $q - q^* > 0$, the cumulative rank rate is more likely to be positive, and the mutation strength tends to increase in the following generations. Otherwise, the mutation strength tends to decrease. In this paper, we use $q^* = 0.3$.
2) c_s*:* The changing rate c_s determines the number of generations involved in the accumulation of s as argued in [1]. If $c_s = 1$, no accumulation takes place. This would cause an immediate change in the mutation strength. For small c_s, it accumulates and averages $q - q^*$ over a long time, and the mutation strength may change slowly. In this paper, we set $c_s = 0.3$.
3) d_σ*:* The damping parameter $d_\sigma \geq 1$ determines the change magnitude of $\ln \sigma_t$. If $d_\sigma = 1$, it could induce a rapid adaptation in the mutation strength. On the contrary, large d_σ could slow down the mutation strength adaptation, which would be beneficial for multimodel functions. In this paper, we set $d_\sigma = 1$.

RSR is inspired by the median success rule [34] and population success rule [21], and can be considered as a population version of the traditional success rules [35]. There are two major differences between RSR and these two rules: RSR considers the rank differences between the selected solutions in two consecutive generations, and uses weights to favor the top ranked solutions. A comparison on mutation strength adaptation methods is presented in Section V.

B. R1-ES Framework

R1-ES is summarized in Algorithm 1. The distribution parameters are initialized in line 1, $F_0 = (f(\mathbf{m}_0), \ldots, f(\mathbf{m}_0))$ containing μ values. Then, a population of λ solutions are sampled in lines 4 and 5, and $\mathbf{x}_{\text{best}}$ is updated if a better solution is found in lines 6–8. The solutions are sorted according

TABLE I
PARAMETERS FOR R1-ES

$\lambda = 4 + \lfloor 3 \ln n \rfloor, \quad \mu = \lfloor \frac{\lambda}{2} \rfloor, w_i = \frac{\ln(\mu+1) - \ln i}{\mu \ln(\mu+1) - \sum_{j=1}^{\mu} \ln j}, i = 1, \ldots, \mu$
$\mu_{\text{eff}} = \frac{1}{\sum_{i=1}^{\mu} w_i^2}, \quad c_{cov} = \frac{1}{3\sqrt{n}+5}, \quad c_c = \frac{2}{n+7}$
$q^* = 0.3, \quad c_s = 0.3, \quad d_\sigma = 1$

to the objective values in line 10, then the best μ solutions are used to update the evolution path and mutation strength in lines 11–15.

1) Parameter Settings: All algorithm parameters are presented in Table I.

The changing rates c_{cov} and c are discussed as follows.

1) c_{cov}*:* The changing rate of the covariance matrix c_{cov} is designed according to the convergence analysis of the IGO flow equation in [36]. It has proven that, on quadratic objective functions, the covariance matrix converges to the inverse Hessian at the rate of $\sqrt{2/n}$ with the optimal weights [37]. Hence, it is reasonable to set the changing rate to the order of $1/\sqrt{n}$. In this paper, we use $c_{\text{cov}} = 1/(3\sqrt{n}+5)$.
2) c*:* The changing rate of the evolution path c determines the weight of the most recent search direction, and the backward time horizon of the cumulation c^{-1} [1]. It is designed to be inverse proportional to the number of parameters to be adapted such that $c^{-1} \propto n$ [1]. In this paper, we use $c = 2/(n+7)$.

2) Time and Space Complexity: The time complexity refers to the time consumption of one objective function evaluation (FE). R1-ES involves only scalar-vector multiplications to generate new solutions, hence its time complexity is $O(n)$.

At each generation, the major data to be recorded is a set of λ solutions. Thus, its space complexity is $O(\lambda n)$.

C. Rm-ES With Multiple Evolution Paths

In the following, we extend R1-ES to a general framework to use a number of m evolution paths to generate new solutions at each generation, and propose a Rm-ES.

1) Basic Idea: Let $\mathbf{C}_{t-1}$ be the covariance matrix at generation $(t-1)$ and $\mathbf{p}_t$ be the evolution path at generation t, the rank one update of $\mathbf{C}_t$ in CMA-ES [1] gives

$$\mathbf{C}_t = (1 - c_{\text{cov}})\mathbf{C}_{t-1} + c_{\text{cov}}\mathbf{p}_t\mathbf{p}_t^T \tag{15}$$

where $\mathbf{C}_0 = \mathbf{I}$. Thus, the covariance matrix can be constructed by using current and all previous evolution paths.

In Rm-ES, we maintain a number of $m \ll n$ evolution paths $\mathbf{P} = [\hat{\mathbf{p}}_1, \ldots, \hat{\mathbf{p}}_m]$ and their generations $\hat{\mathbf{t}} = [\hat{t}_1, \ldots, \hat{t}_m]$ at each generation, where $\hat{t}_i < \hat{t}_j$ if $i < j$. Inspired by the rank-1 update (15), it is reasonable to define the covariance matrix as

$$\mathbf{C} = (1 - c_{\text{cov}})^m \mathbf{I} + c_{\text{cov}} \sum_{i=1}^{m} (1 - c_{\text{cov}})^{m-i} \hat{\mathbf{p}}_i \hat{\mathbf{p}}_i^T. \tag{16}$$

In this way, we can efficiently adapt the evolution paths directly to generate new solutions, instead of storing the covariance matrix explicitly.

Algorithm 2 Update

1: **Input:** $\mathbf{P}, \hat{\mathbf{t}}, T, m, \mathbf{p}_t, t$
2: $T_{\min} = \min_{1 \leqslant i \leqslant m-1} (\hat{t}_{i+1} - \hat{t}_i)$
3: **if** $T_{\min} > T$ or $t < m$ **then**
4: $\quad \hat{\mathbf{p}}_i \leftarrow \hat{\mathbf{p}}_{i+1}, \ \hat{t}_i \leftarrow \hat{t}_{i+1}, \ i = 1, \ldots, m-1$
5: **else**
6: $\quad i' \leftarrow \operatorname{argmin}_{1 \leqslant i \leqslant m-1} (\hat{t}_{i+1} - \hat{t}_i)$
7: $\quad \hat{\mathbf{p}}_i \leftarrow \hat{\mathbf{p}}_{i+1}, \ \hat{t}_i \leftarrow \hat{t}_{i+1}, \ i = i', \ldots, m-1$
8: **end if**
9: $\hat{\mathbf{p}}_m \leftarrow \mathbf{p}_t, \ \hat{t}_m \leftarrow t$
10: **Output:** $\mathbf{P}, \hat{\mathbf{t}}$

2) Sampling: Using the low rank model $\mathcal{N}(\mathbf{m}_t, \sigma_t^2 \mathbf{C}_t)$ with covariance matrix $\mathbf{C}_t$ given by (16), we can sample a new solution $\mathbf{x}$ similar to (8) as

$$\mathbf{x} = \mathbf{m}_t + \sigma_t \left(a^m \mathbf{z} + b \sum_{i=1}^{m} a^{m-i} r_i \hat{\mathbf{p}}_i \right) \tag{17}$$

where $a = \sqrt{1 - c_{\text{cov}}}$, $b = \sqrt{c_{\text{cov}}}$, and $r_i \sim \mathcal{N}(0, 1)$. It can be easily verified that $\mathbf{x} \sim \mathcal{N}(\mathbf{m_t}, \sigma_t^2 \mathbf{C}_t)$, with $\mathbf{C}_t$ given by (16). The complexity of the sampling procedure is $O(mn)$ for each solution.

3) Update of Directions: [9], [21] noticed that the evolution paths from the last several generations are similar and suggested that separate evolution paths with about the same generation distance should be used. We use the technique proposed in [9] to maintain the direction set. It works as follows.

Each evolution path in $\mathbf{P} = [\hat{\mathbf{p}}_1, \ldots, \hat{\mathbf{p}}_m]$ is initialized to zero. At the first m generations, Rm-ES records all the generated evolution paths $\mathbf{p}_t$. Thereafter, P is updated by the current evolution path at each generation. If generation gap between a recorded evolution path $\hat{\mathbf{p}}_{i+1}$ and $\hat{\mathbf{p}}_i$ is less than fixed generation gap T, then $\hat{\mathbf{p}}_{i+1}$ is removed from the set. Otherwise, the oldest recorded evolution path $\hat{\mathbf{p}}_1$ is removed. Then the current evolution path is added to the set, and the recorded generations are updated accordingly. The detailed procedure is presented in Algorithm 2.

4) Rm-ES Framework: The procedure of Rm-ES is given in Algorithm 3.

5) Parameter Settings: The population size, weights, changing rates c, c_{cov}, and parameters for mutation strength adaptation are given in Table I. We discuss the number of evolution paths m and the generation gap T in the following.

1) m*:* To obtain an appropriate approximation to the covariance matrix, one needs multiple evolution paths. However, using a large number of evolution paths would increase the computational complexity in the sampling procedure (17). We investigate the effects of m on the algorithm performance in Section V.
2) T*:* The generation gap T is used to keep the recorded evolution paths uncorrelated. As the backward time horizon of the cumulation in evolution path is c^{-1}, it is reasonable to set $T \geqslant c^{-1} \approx n/2$. In this paper, we set $T = n$.

Algorithm 3 Rm-ES

Initialize: $\mathbf{m}_0, \sigma_0, \mathbf{p}_0 = \mathbf{0}, \mathbf{x}_{\text{best}} = \mathbf{m}_0, F_0, s_0, P, t = 0$
repeat
  for $i = 1$ to λ **do**
    sample $\mathbf{x}_i$ using (17)
    if $f(\mathbf{x}_i) < f(\mathbf{x}_{\text{best}})$ **then**
      $\mathbf{x}_{\text{best}} = \mathbf{x}_i$
    end if
  end for
  sort $\mathbf{x}_i$ as $f(\mathbf{x}_{1:\lambda}) \leqslant f(\mathbf{x}_{2:\lambda}) \leqslant \cdots \leqslant f(\mathbf{x}_{\lambda:\lambda})$
  $\mathbf{m}_{t+1} = \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$
  $\mathbf{p}_{t+1} = (1-c)\mathbf{p}_t + \sqrt{c(2-c)\mu_{\text{eff}}}\frac{\mathbf{m}_{t+1}-\mathbf{m}_t}{\sigma_t}$
  Update()
  update σ_t
  $t = t + 1$
until stopping criterion is met
return

IV. Algorithm Analysis

A. Insights Into Evolution Path

We use evolution path as the principal search direction in the low rank model. We illustrate in the following that it is reasonable as the evolution path accumulates natural gradients, and acts as a momentum term.

1) Natural Gradient: The NES [17] and IGO framework [24] provide a principled way to design stochastic optimization algorithms. NES minimizes the expected fitness with respect to the distribution parameter. IGO transforms the original minimization problem to a maximization problem

$$\max_{\theta} \quad J(\theta) = \int W^f_{\theta_t}(\mathbf{x}) P_\theta(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{18}$$

where $W^f_{\theta_t}(\mathbf{x})$ determines the selection scheme, θ_t is the current distribution parameter, and θ is the distribution parameter to be optimized.

To optimize $J(\theta)$, IGO takes the natural gradient with respect to the distribution parameter θ as

$$\hat{\nabla}_\theta J(\theta) = \mathbf{F}^{-1}(\theta_t) \int W^f_{\theta_t}(\mathbf{x}) \frac{\partial \ln P_\theta(x)}{\partial \theta} P_{\theta_t}(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{19}$$

where $\mathbf{F}$ is the Fisher information matrix. In practice, the value $W^f_{\theta_t}$ can be approximated by samples drawn from P_{θ_t}. Given a set of N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$ drawn from P_{θ_t}, one can estimate the natural gradient as

$$\hat{\nabla}_\theta J(\theta) = \mathbf{F}^{-1}(\theta_t) \sum_{i=1}^{N} \hat{w}_i \frac{\partial \ln P_\theta(\mathbf{x}_{i:N})}{\partial \theta}\bigg|_{\theta=\theta_t} \tag{20}$$

where $\mathbf{x}_{i:N}$ denotes the ith sample according to objective value, and the weights $\hat{w}_i$ are set to

$$\hat{w}_i = \frac{1}{N} w\left(\frac{i-1/2}{N}\right) \tag{21}$$

where $w(q) = 1_{q \leq 1/2}$ (see [24] for more details).

In the space of multivariate Gaussian distributions with parameter $\theta = (\mathbf{m}, \mathbf{C})$, the natural gradient with respect to the distribution mean $\mathbf{m}$ is estimated by

$$\hat{\nabla}_{\mathbf{m}} J(\theta) = \sum_{i=1}^{N} \hat{w}_i (\mathbf{x}_i - \mathbf{m}_t). \tag{22}$$

2) Evolution Path Acts As Momentum: As shown in the update of evolution path (11), the update direction for the evolution path is

$$\mathbf{m}_{t+1} - \mathbf{m}_t = \sum_{i=1}^{\mu} w_i (\mathbf{x}_{i:\lambda} - \mathbf{m}_t). \tag{23}$$

It shows remarkable similarities with (22), indicating that $(\mathbf{m}_{t+1} - \mathbf{m}_t)$ is an estimation to the natural gradient $\hat{\nabla}_{\mathbf{m}} J(\theta)$. Hence, the evolution path accumulates the natural gradients with respect to the distribution mean, which is known as the momentum term [38]. Consequently, the evolution path acts as a momentum term under the stationarity condition. This can be seen more clearly by the update equations of the evolution path and the momentum (for maximization problems)

$$\begin{aligned} \text{momentum:} \quad & \mathbf{d}_{t+1} = \beta \mathbf{d}_t + \alpha \nabla f(\mathbf{x}) \\ \text{evolution path:} \quad & \mathbf{p}_{t+1} = \beta_p \mathbf{p}_t + \alpha_p \hat{\nabla}_m J(\theta) \end{aligned}$$

where $\alpha, \beta > 0$, $\beta_p = 1 - c$, and $\alpha_p = \sqrt{c(2-c)\mu_{\text{eff}}}/\sigma_t$.

The momentum method is commonly used for optimization in machine learning [38], [39]. The evolution path has the advantages of the momentum. The opposite search directions in successive generations are mutually canceled out as they are averaged over generations, and these components will remain small. Hence, the evolution path accumulates contributions in directions of persistent descent over many generations. The consistent search directions generations represents a promising search direction. Therefore, searching along the evolution path is likely to generate solutions with good quality.

B. Low Rank Approximation to General Covariance

Consider the following quadratic objective function:

$$\min_{x \in \mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} \tag{24}$$

where $\mathbf{H}$ is its $n \times n$ positive-definite Hessian matrix. The theoretical convergence analysis on IGO flow in [36] indicates that the covariance matrix converges to $\mathbf{H}^{-1}$ (up to a scalar factor) on quadratic objective functions. When using the low rank model, it cannot converge to a general inverse Hessian. The following issues naturally arise.

1) What is the optimal low rank model for approximating a given positive-definite matrix $\mathbf{C}$?
2) Can the evolution path in R1-ES approximate the optimal low rank model?
3) How does the Hessian matrix $\mathbf{H}$ affect the algorithm performance?

1) Optimal Low Rank Approximation: We present the main result on the optimal low rank approximation to a given positive-definite matrix in the following. A detailed proof can be found in the supplementary.

Proposition 1: For a given positive definite matrix $\mathbf{C}$, consider the low rank approximation

$$D_m = \min_{a>0, \mathbf{V} \in \mathbb{R}^{n \times m}} \|\mathbf{C} - \left(a\mathbf{I} + \mathbf{V}\mathbf{V}^T\right)\|_F. \tag{25}$$

An optimal solution to (25) is given by

$$\mathbf{V}^* = \mathbf{B}_m \left(\Lambda_m - a^* \mathbf{I}\right)^{1/2}, \; a^* = \frac{1}{n-m} \sum_{i=m+1}^{n} r_i \tag{26}$$

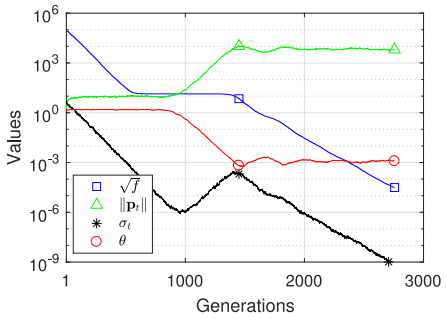

Fig. 2. Algorithm behavior of R1-ES on Cigar function with $n = 200$.

where $\mathbf{B}_m$ is a $n \times m$ matrix of orthogonal column eigenvectors, and $\Lambda_m = \text{diag}(r_1, \ldots, r_m)$ is $m \times m$ diagonal matrix of largest m eigenvalues of $\mathbf{C}$. The optimal value of D_m is

$$D_m^* = \sqrt{\sum_{i=m+1}^{n} (r_i - a^*)^2}. \tag{27}$$

Note that the optimal solution is not unique, as $\mathbf{VRR}^T\mathbf{V}^T = \mathbf{VV}^T$ holds for any $m \times m$ orthogonal matrix $\mathbf{R}$. A result equivalent to (26) (but from the point of view of maximum likelihood estimation) has been used in [45].

We can remark that the optimal rank m approximation of (25) is given by the subspace expanded by the leading m eigenvectors of $\mathbf{C}$, and D_m^* represents the lower bound of (25) determined by the eigenvalues of $\mathbf{C}$.

2) Experiments on the Evolution Path: When $\mathbf{V}$ is of rank one, the optimal approximation in (26) is given by the first principal component of the inverse Hessian. We investigate whether the evolution path in R1-ES can effectively approximate the predominant search direction of the Cigar function $f(\mathbf{x}) = x_1^2 + 10^6 \cdot \sum_{i=2}^{n} x_i^2$ with $n = 200$. The first principal component of inverse Hessian is given by $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$. The similarity between $\mathbf{e}_1$ and the evolution path $\mathbf{p}_t$ is measured by

$$\theta = \arccos\left(\frac{|\mathbf{p}_t^T \mathbf{e}_1|}{\|\mathbf{p}_t\|}\right) \tag{28}$$

where $\theta \approx 0$ indicates that $\mathbf{p}_t$ is close to $\mathbf{e}_1$.

Fig. 2 plots the evolution of the objective value, the mutation strength, the norm of evolution path, and θ of a typical run of R1-ES. It is clear that the evolution path has approximated $\mathbf{e}_1$ very well at generation 1500, where θ has decreased from 1 to 10^{-3} and the length of the evolution path has increased three orders of magnitude from 10 to 10^4. Since $\mathbf{p}_t$ has been very close to $\mathbf{e}_1$, the objective value decreases rapidly after generation 1500. Consequently, the evolution path in R1-ES can effectively approximate the first principal component of inverse Hessian on Cigar function.

V. Experimental Studies

In this section, we experimentally investigate the behaviors and performances of the proposed algorithms. The test problems are given in Table II. In the experiments, the target f-value is set to 10^{-8}, the distribution mean $\mathbf{m}_0$ is randomly initialized in the range $[-10, 10]^n$, and $\sigma_0 = 20/3$, unless specifically stated otherwise. Each algorithm runs 31 times independently, and we conduct Wilcoxon rank sum test at the 5% significance level to assess the experimental results. Further, we conduct experiments on the CEC'2010 LSGO benchmarks, compared with the state-of-the-art large-scale variants.

TABLE II
Test Problems

$f_{\text{Ell}}(\mathbf{x})$	$= \sum_{i=1}^{n} 10^{6 \cdot \frac{i-1}{n-1}} \cdot x_i^2$
$f_{\text{Cig}}(\mathbf{x})$	$= x_1^2 + 10^6 \cdot \sum_{i=2}^{n} x_i^2$
$f_{\text{Ctb}}(\mathbf{x})$	$= x_1^2 + 10^4 \cdot \sum_{i=2}^{n-1} x_i^2 + 10^6 x_n^2$
$f_{\text{Tab}}(\mathbf{x})$	$= 10^6 \cdot x_1^2 + \sum_{i=2}^{n} x_i^2$
$f_{\text{Sch}}(\mathbf{x})$	$= \sum_{i=1}^{n} (\sum_{j=1}^{i} x_i)^2$
$f_{\text{Ros}}(\mathbf{x})$	$= \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$

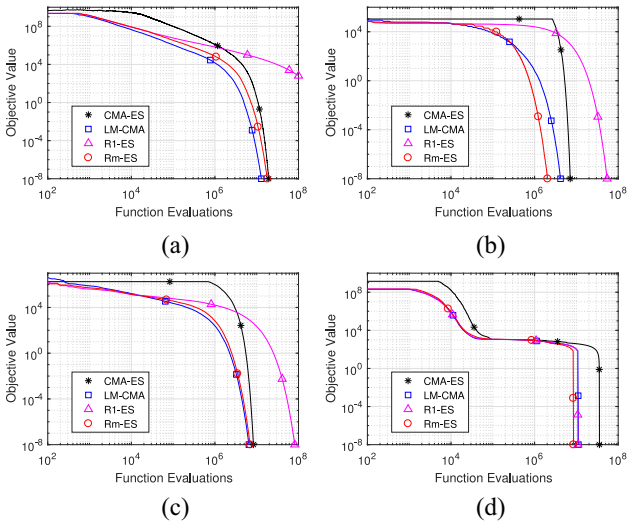

Fig. 3. Median run out of 31 runs of each algorithm on each test problem with $n = 1000$. (a) f_{Ell}. (b) f_{Tab}. (c) f_{Sch}. (d) f_{Ros}.

A. Performance Comparison

We compare Rm-ES with LM-CMA and CMA-ES on the test functions with $n = 1000$. In the experiments, we use $m = 2$ for Rm-ES. Fig. 3 plots the median run[1] out of 31 independent runs of each algorithm.

The experimental results show that Rm-ES outperforms LM-CMA on f_{Tab} and f_{Ros}, and performs similarly to LM-CMA on f_{Sch}. Rm-ES slightly outperforms CMA-ES on f_{Ell}, while both methods cost slight more FEs than LM-CMA. It is very impressive, considering that we only use two evolution paths in Rm-ES, and that the fitness landscape of f_{Ell} is far different from the low rank model. On f_{Tab}, Rm-ES costs only a half of FEs of LM-CMA and less than a third of CMA-ES. It is interesting as the landscape of f_{Tab} is opposite to the low rank model. Rm-ES outperforms R1-ES by a factor about ten on f_{Tab}, f_{Ell}, and f_{Sch}, whose fitness landscapes are different to low rank model. This indicates the effectiveness of using $m = 2$ evolution paths.

Based on these observations, we claim that Rm-ES with $m = 2$ achieves competitive performance to LM-CMA, while

[1]By median run, we mean the run with median number of FEs to reach the required accuracy. We consider the failed runs on f_{Ros} cost more FEs than any success one.

Rm-ES is much simpler. It does not need to update the inverse Cholesky factor, does not use the subset selection, and simplifies the sampling procedure in LM-CMA [9].

B. Experiments on the Number of Evolution Paths

The number of evolution paths in Rm-ES is a key parameter. We investigate its effects on algorithm performance, compared with LM-CMA. LM-CMA stores a set of $m = 4 + \lfloor 3 \ln n \rfloor$ evolution paths and corresponding inverse vectors, which is the same number to the population size. To sample new solutions, a subset selection procedure is used to select the latest m^* directions to reconstruct the Cholesky factor, where $m^* = \min(\lfloor m_\sigma |\mathcal{N}(0, 1)| \rfloor, m)$ with $m_\sigma = 4$ [9, Algorithm 6]. We denote LM-CMA using all the stored m directions by LM-CMA-m. The experiments are conducted on $n = 1000$, with m ranging from 2 to $\lfloor \sqrt{n} \rfloor$. For f_{Ros}, the success performance $\hat{SP}_1 = T_s/p_s$ is used [40], where T_s is the mean FEs of success runs, and p_s is the ratio of success runs.

1) Experimental Results: Fig. 4 presents the average FEs to reach accuracy 10^{-8} with different m settings. We have the following observations.

1) The algorithm performance improvement of Rm-ES is marginal as the number of evolution paths increases from 2 to $m = 24$. Its performance even becomes worse on f_{Ros} when using more evolution paths. LM-CMA-m shows a similar trends with the increasing of directions on f_{Ros}.
2) The subset selection procedure of LM-CMA significantly improves the performance of LM-CMA-m on all test functions. Rm-ES outperforms LM-CMA-m on most of the test functions, while performs competitively to LM-CMA with the subset selection procedure.

2) Discussion: It is generally expected to obtain a better algorithm performance if more evolution paths are used. However, the experimental results show that the algorithm performance improvement is marginal as the number of evolution paths increases. The reason may be that the recorded evolution paths are correlated. In Rm-ES, we record evolution paths in separated generations to keep them independent. Yet, the evolution paths would be correlated even after n generations. As the Rm-ES ($m = 2$) outperforms or performs competitively on quadratic objective functions to CMA-ES and the covariance matrix of CMA-ES can be constructed by all the previous evolution paths, it is required the properties between the evolution paths to further improvement the algorithm performances.

Further, the algorithm performance becomes worse on f_{Ros} when more evolution paths are used. The reason is that the old search directions have negative effects on the algorithm performance. In CMA-ES with rank-1 update (15), the variances along historical directions are damped by factor $(1 - c_{\text{cov}})$ at every generation. While in the rank-m model, the variances of old evolution paths are only damped less than $m \ll n$ times. Although we use a much larger changing rate, the outdated evolution paths may affect algorithm search for quite a long time. This may significantly degenerate algorithm performance when it requires rapid adaptation as the f_{Ros}.

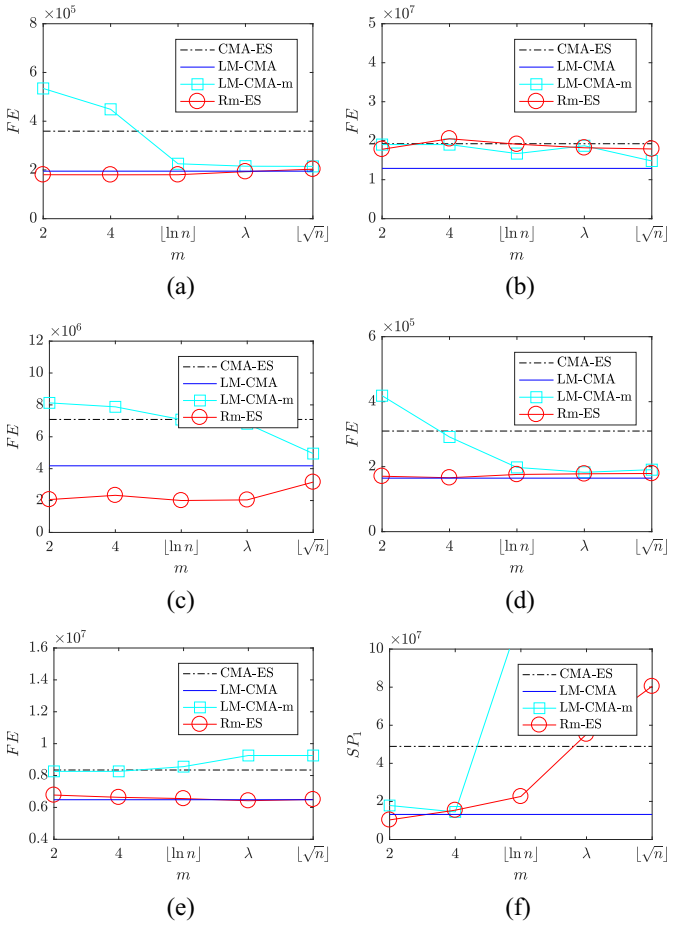

Fig. 4. Average number of FEs on the number of evolution paths m with $n = 1000$. For f_{Ros}, the success performance $\hat{SP}_1 = T_s/p_s$ is used. (a) f_{Cig}. (b) f_{Ell}. (c) f_{Tab}. (d) f_{Ctb}. (e) f_{Sch}. (f) f_{Ros}.

C. Experiments on LSGO 2010

We present the experimental results on the CEC'2010 LSGO benchmarks [41]. The benchmarks include a set of specifically designed group-rotated objective functions with $n = 1000$. The maximum FE is set to 3×10^6. The proposed Rm-ES ($m = 2$) is compared with CMA-ES, CC-GDG-CMAES [30] (denoted by GDG-CMA), and the winners of CEC'2010 and CEC'2012 competitions, MA-SW-Chains [42] (denoted by MA-SW) and MOS [43]. The experimental results of GDG-CMA and CMA-ES are from [30].

As the mutation strength decays quickly to zero if it gets stuck at some local minima, Rm-ES is randomly restarted if the maximum FE is not reached. The restart strategy is commonly used in stochastic search (e.g., [40]). We restart the algorithm if $f(\mathbf{x}_{\text{best}}) > 10^{-10}$ and: 1) the mutation strength $\sigma_t < 10^{-10}$ or 2) the objective improvement over n generations is less than 10^{-10}.

Every time the algorithm is restarted, we increase the population size and damping parameter as $\lambda^{(i)} = 2\lambda^{(i-1)}$ and $d_\sigma^{(i)} = 2d_\sigma^{(i-1)}$, where the superscript i denotes the number of restarts. This is reasonable because increasing the population size [44] and slowing down the mutation strength adaptation are beneficial for multimodal problems.

TABLE III
MED, MEAN, AND STANDARD DEVIATION OF THE FITNESS VALUES OBTAINED FROM 25 INDEPENDENT RUNS ON CEC'2010 LSGO BENCHMARKS. THE RANK IS OBTAINED BY COMPARING THE MEDIAN VALUE OF THE FIVE ALGORITHMS. THE MED OF THE BEST PERFORMING ALGORITHMS ARE MARKED IN BOLD, UNDER THE PAIRWISE WILCOXON SIGNED-RANK TEST WITH SIGNIFICANCE LEVEL $p = 0.05$

Fun		CMA-ES	GDG-CMA	Rm-ES	MA-SW	MOS
f1	med	8.49E+06	**0.00E+00**	4.70E+06	1.50E-14	**0.00E+00**
	mean	8.60E+06	0.00E+00	4.81E+06	2.10E-14	0.00E+00
	std	8.01E+05	0.00E+00	3.38E+05	1.99E-14	0.00E+00
f2	med	5.20E+03	1.61E+03	4.85E+02	7.90E+02	**1.95E+02**
	mean	5.21E+03	1.60E+03	4.89E+02	8.10E+02	1.97E+02
	std	2.20E+02	5.29E+01	2.37E+01	5.88E+01	1.59E+01
f3	med	2.17E+01	2.17E+01	6.05E-04	**6.11E-13**	1.29E+00
	mean	2.17E+01	2.17E+01	5.09E-04	7.28E-13	1.12E+00
	std	1.06E-02	1.06E-02	2.50E-04	3.40E-13	1.00E+00
f4	med	5.22E+13	**1.17E+10**	3.72E+11	3.54E+11	1.88E+10
	mean	5.90E+13	1.60E+10	4.03E+11	3.53E+11	1.91E+10
	std	1.98E+13	1.25E+10	3.13E+10	3.12E+10	8.08E+09
f5	med	**6.44E+07**	1.02E+08	9.35E+07	2.31E+08	6.86E+08
	mean	6.37E+07	1.02E+08	7.57E+07	1.68E+08	6.81E+08
	std	1.31E+07	1.32E+07	1.75E+07	1.04E+08	1.42E+08
f6	med	2.17E+01	**0.00E+00**	2.16E+01	1.60E+00	1.98E+07
	mean	2.58E+06	6.03E+06	2.16E+01	8.14E+04	2.00E+07
	std	7.13E+06	9.88E+06	1.41E-02	2.84E+05	5.67E+04
f7	med	1.41E+09	**0.00E+00**	2.85E+07	9.04E+01	**0.00E+00**
	mean	1.35E+09	0.00E+00	1.89E+07	1.03E+02	0.00E+00
	std	3.29E+08	0.00E+00	5.25E+06	8.70E+01	0.00E+00
f8	med	6.47E+08	2.20E+07	7.68E+06	3.43E+06	**2.74E-01**
	mean	1.08E+09	2.90E+07	8.21E+06	1.41E+07	1.12E+06
	std	1.35E+09	2.59E+07	4.81E+05	3.68E+07	1.79E+06
f9	med	9.55E+06	**1.57E+03**	6.26E+06	1.40E+07	8.83E+06
	mean	9.59E+06	1.74E+03	6.74E+06	1.41E+07	8.78E+06
	std	1.07E+06	6.95E+02	4.54E+05	1.15E+06	1.01E+06
f10	med	5.17E+03	1.80E+03	**4.91E+02**	2.07E+03	7.83E+03
	mean	5.20E+03	1.81E+03	4.92E+02	2.07E+03	7.86E+03
	std	2.83E+02	8.89E+01	1.53E+01	1.44E+02	2.43E+02
f11	med	2.38E+02	6.44E+01	1.36E+02	**3.75E+01**	1.99E+02
	mean	2.38E+02	6.53E+01	1.36E+02	3.80E+01	1.99E+02
	std	1.97E-01	2.82E+01	2.21E+01	7.35E+00	4.52E-01
f12	med	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.50E-06	**0.00E+00**
	mean	0.00E+00	0.00E+00	0.00E+00	3.62E-06	0.00E+00
	std	0.00E+00	0.00E+00	0.00E+00	5.92E-07	0.00E+00
f13	med	7.85E+04	**1.59E+02**	3.28E+05	1.07E+03	1.18E+03
	mean	9.15E+04	2.72E+02	5.15E+05	1.25E+03	1.36E+03
	std	6.77E+04	1.77E+02	1.65E+05	5.72E+02	9.37E+02
f14	med	1.07E+07	**0.00E+00**	6.53E+06	3.09E+07	1.85E+07
	mean	1.06E+07	0.00E+00	6.24E+06	3.11E+07	1.82E+07
	std	1.11E+06	0.00E+00	9.43E+05	1.93E+06	1.18E+06
f15	med	5.16E+03	2.01E+03	**4.90E+02**	2.72E+03	1.54E+04
	mean	5.12E+03	2.00E+03	4.95E+02	2.74E+03	1.54E+04
	std	1.98E+02	6.74E+01	3.12E+01	1.22E+02	5.36E+02
f16	med	4.33E+02	**8.46E+01**	1.38E+02	9.44E+01	3.97E+02
	mean	4.33E+02	9.67E+01	1.42E+02	9.98E+01	3.97E+02
	std	2.83E-01	3.78E+01	3.82E+01	1.40E+01	2.10E-01
f17	med	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.26E+00	4.83E-05
	mean	0.00E+00	0.00E+00	0.00E+00	1.24E+00	4.66E-05
	std	0.00E+00	0.00E+00	0.00E+00	1.25E-01	6.24E-06
f18	med	2.51E+03	**5.79E+01**	3.25E+02	1.19E+03	3.55E+03
	mean	3.36E+03	8.63E+01	8.79E+02	1.30E+03	3.91E+03
	std	1.81E+03	8.76E+01	9.58E+02	4.36E+02	2.18E+03
f19	med	2.81E+06	2.81E+06	1.60E+05	2.85E+05	**3.40E+04**
	mean	2.87E+06	2.87E+06	1.36E+05	2.85E+05	3.41E+04
	std	6.61E+05	6.61E+05	3.69E+04	1.78E+04	2.63E+03
f20	med	8.29E+02	8.29E+02	**6.63E+02**	1.06E+03	7.26E+02
	mean	8.54E+02	8.54E+02	7.51E+02	1.07E+03	8.31E+02
	std	6.71E+01	6.71E+01	9.85E+01	7.29E+01	3.76E+02
No. of Best		3	11	5	2	5
Avg. of Rank		3.9	1.9	2.4	3.2	3.0

Table III presents the median, mean and standard deviation of the fitness values obtained from 25 independent runs on the CEC'2010 LSGO benchmarks. We conduct Wilcoxon rank sum test for equality of medians of the experimental results of any two algorithms. It clearly shows that, on five out of 20 benchmarks, Rm-ES achieves the best results. Rm-ES achieves an average rank (in terms of median value of the experimental results) of 2.4 out of the five compared algorithms, outperforming CMA-ES, MA-SW, and MOS. Rm-ES outperforms CMA-ES, MA-SW, and MOS on 16, 11, and 13 benchmarks out of 20, respectively. The restart strategy plays an important role in the remarkable performances of Rm-ES on multimodal functions. Rm-ES is inferior to GDG-CMA, MA-SW, and MOS on some of the separable objective functions as it does not use any separability property of the objective functions. Using a nonidentity sparse matrix in the low rank model (4) may significantly improve the algorithm performance on separable objective functions.

More experimental results can be found in supplementary material.

VI. CONCLUSION

In this paper, we have proposed a general framework of using sparse and low rank decomposition of the covariance to achieve low computational complexity. We have presented R1-ES and Rm-ES using single and multiple search directions, respectively. We have illustrated that the evolution path accumulates natural gradients of the expected fitness with respect to the distribution mean, and acts as a momentum term under the stationarity condition. Hence, it is reasonable to use evolution path as the principal search direction. We have illustrated the optimal approximation of low rank model to a general covariance is given by the subspace expanded by the top eigenvectors. We have conducted experiments to show that the evolution path effectively approximates the eigenvector corresponding to the largest eigenvalue on the Cigar function, and investigated the effects of Hessian on the algorithm performances.

We have conducted experiments to investigate the performances of RSR, R1-ES, and Rm-ES, respectively. It shows that RSR has some advantages than CSA on sphere and ellipsoid functions with low condition numbers. Further, R1-ES achieves remarkable performance on the well-known Rosenbrock function, and ill-conditioned objective functions with predominant search direction. We have investigated the effects of the number of evolution paths, compared with LM-CMA. Rm-ES with $m = 2$ generally performs competitively to LM-CMA on most test functions, yet much simpler. Further, we have investigated the performance of Rm-ES ($m = 2$) on the CEC'2010 LSGO benchmarks with a restart strategy. The experimental results validate the effectiveness of our proposed algorithm.

We will further investigate two important research issues to improve the algorithm performance, i.e., how to maintain the orthogonality and conjugacy properties between evolution paths, and how to use other sparse matrices.

REFERENCES

[1] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.
[2] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, 2003.
[3] 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," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5755–5769, 2011.

[4] Y.-W. Shang and Y.-H. Qiu, "A note on the extended Rosenbrock function," *Evol. Comput.*, vol. 14, no. 1, pp. 119–126, 2006.
[5] J. Poland and A. Zell, "Main vector adaptation: A CMA variant with linear time and space complexity," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, San Francisco, CA, USA, 2001, pp. 1050–1055.
[6] Y. Sun, T. Schaul, F. Gomez, and J. Schmidhuber, "A linear time natural evolution strategy for non-separable functions," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Amsterdam, The Netherlands, 2013, pp. 61–62.
[7] D. V. Arnold and H.-G. Beyer, "Performance analysis of evolutionary optimization with cumulative step length adaptation," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 617–622, Apr. 2004.
[8] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
[9] I. Loshchilov, "LM-CMA: An alternative to L-BFGS for large-scale black box optimization," *Evol. Comput.*, vol. 25, no. 1, pp. 143–171, 2017.
[10] H.-G. Beyer and B. Sendhoff, "Covariance matrix adaptation revisited—The CMSA evolution strategy," in *Proc. Parallel Problems Solving Nat. (PPSN)*, Dortmund, Germany, 2008, pp. 123–132.
[11] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a New Evolutionary Computation*. Berlin, Germany: Springer, 2006, pp. 75–102.
[12] C. Igel, T. Suttorp, and N. Hansen, "A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Seattle, WA, USA, 2006, pp. 453–460.
[13] T. Suttorp, N. Hansen, and C. Igel, "Efficient covariance matrix update for variable metric evolution strategies," *Mach. Learn.*, vol. 75, no. 2, pp. 167–197, 2009.
[14] O. Krause and C. Igel, "A more efficient rank-one covariance matrix update for evolution strategies," in *Proc. ACM Conf. Found. Genet. Algorithms (FOGA)*, Aberystwyth, U.K., 2015, pp. 129–136.
[15] Z. Li and Q. Zhang, "An efficient rank-1 update for Cholesky CMA-ES using auxiliary evolution path," in *Proc. Congr. Evol. Comput. (CEC)*, 2017, pp. 1777–1784.
[16] H.-G. Beyer and B. Sendhoff, "Simplify your covariance matrix adaptation evolution strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 746–759, Oct. 2017.
[17] D. Wierstra *et al.*, "Natural evolution strategies," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 949–980, 2014.
[18] T. Glasmachers, T. Schaul, S. Yi, D. Wierstra, and J. Schmidhuber, "Exponential natural evolution strategies," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Portland, OR, USA, 2010, pp. 393–400.
[19] J. N. Knight and M. Lunacek, "Reducing the space-time complexity of the CMA-ES," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, London, U.K., 2007, pp. 658–665.
[20] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra Appl.*, vol. 415, no. 1, pp. 20–30, 2006.
[21] I. Loshchilov, "A computationally efficient limited memory CMA-ES for large scale optimization," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Vancouver, BC, Canada, 2014, pp. 397–404.
[22] R. Ros and N. Hansen, "A simple modification in CMA-ES achieving linear time and space complexity," in *Proc. Parallel Problems Solving Nat. (PPSN)*, Dortmund, Germany, 2008, pp. 296–305.
[23] T. Schaul, T. Glasmachers, and J. Schmidhuber, "High dimensions and heavy tails for natural evolution strategies," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Dublin, Ireland, 2011, pp. 845–852.
[24] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen, "Information-geometric optimization algorithms: A unifying picture via invariance principles," *J. Mach. Learn. Res.*, vol. 18, no. 18, pp. 1–65, 2017.
[25] Y. Akimoto, A. Auger, and N. Hansen, "Comparison-based natural gradient optimization in high dimension," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Vancouver, BC, Canada, 2014, pp. 373–380.
[26] Y. Akimoto and N. Hansen, "Projection-based restricted covariance matrix adaptation for high dimension," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Denver, CO, USA, 2016, pp. 197–204.
[27] M. N. Omidvar and X. Li, "A comparative study of CMA-ES on large scale global optimisation," in *Proc. Aust. Conf. Artif. Intell.*, Adelaide, SA, Australia, 2010, pp. 303–312.
[28] M. A. Potter and K. A. D. Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Parallel Problems Solving Nat. (PPSN)*, Jerusalem, Israel, 1994, pp. 249–257.
[29] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization," *Inf. Sci.*, vol. 295, pp. 407–428, Feb. 2015.
[30] Y. Mei, M. N. Omidvar, X. Li, and X. Yao, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Trans. Math. Softw.*, vol. 42, no. 2, pp. 1–24, 2016.
[31] W. Dong, T. Chen, P. Tino, and X. Yao, "Scaling up estimation of distribution algorithms for continuous optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 797–822, Dec. 2013.
[32] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Inf. Sci.*, vol. 178, no. 15, pp. 2985–2999, 2008.
[33] A. Kabán, J. Bootkrajang, and R. J. Durrant, "Toward large-scale continuous EDA: A random matrix theory perspective," *Evol. Comput.*, vol. 24, no. 2, pp. 255–291, 2016.
[34] O. A. Elhara, A. Auger, and N. Hansen, "A median success rule for non-elitist evolution strategies: Study of feasibility," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Amsterdam, The Netherlands, 2013, pp. 415–422.
[35] H.-P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*. New York, NY, USA: Wiley, 1993.
[36] H.-G. Beyer, "Convergence analysis of evolutionary algorithms that are based on the paradigm of information geometry," *Evol. Comput.*, vol. 22, no. 4, pp. 679–709, 2014.
[37] D. V. Arnold, "Weighted multirecombination evolution strategies," *Theor. Comput. Sci.*, vol. 361, no. 1, pp. 18–37, 2006.
[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
[39] A. Bhaya and E. Kaszkurewicz, "Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method," *Neural Netw.*, vol. 17, no. 1, pp. 65–71, 2004.
[40] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 3. Edinburgh, U.K., 2005, pp. 1769–1776.
[41] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization," Nat. Inspired Comput. Appl. Lab., Univ. Sci. Technol. China, Hefei, China, Tech. Rep., 2009. [Online]. Available: https://titan.csit.rmit.edu.au/∼e46507/publications/lsgo-cec10.pdf
[42] D. Molina, M. Lozano, and F. Herrera, "MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization," in *Proc. Congr. Evol. Comput. (CEC)*, Barcelona, Spain, 2010, pp. 1–8.
[43] A. LaTorre, S. Muelas, and J. M. Peña, "Multiple offspring sampling in large scale global optimization," in *Proc. Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, 2012, pp. 1–8.
[44] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions," in *Proc. Parallel Problems Solving Nat. (PPSN)*, Birmingham, U.K., 2004, pp. 282–291.
[45] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

Zhenhua Li received the B.S. degree in mathematics and applied mathematics from Zhengzhou University, Zhengzhou, China, in 2009, and the M.Sc. degree in applied mathematics from the Guangdong University of Technology, Guangzhou, China, in 2013. He is currently pursuing the Ph.D. degree in the Department of Computer Science, City University of Hong Kong, Hong Kong.

His current research interests include evolutionary computation and optimization.

Qingfu Zhang (M'01–SM'06–F'17) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984, and the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi'an, China, in 1991 and 1994, respectively.

He is a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, where he is leading the Metaheuristic Optimization Research Group. He was selected as a Changjiang Chair Professor and a Thousand Talents Program Professor in 2011 and 2015, respectively, in China. His current research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications.

Dr. Zhang was a recipient of the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper. He is a Highly Cited Researcher in Computer Science (Thomson Reuters, 2016, 2017). MOEA/D, a multiobjective optimization algorithmic framework, developed in his group, is one of the most widely used and researched multiobjective framework. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. He is also an Editorial Board Member of three other international journals.