Scaling Up Covariance Matrix Adaptation Evolution Strategy Using Cooperative Coevolution

Jinpeng Liu and Ke Tang*

USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI),
School of Computer Science and Technology,
University of Science and Technology of China (USTC),
Hefei 230027, China
jpl@mail.ustc.edu.cn, ketang@ustc.edu.cn

Abstract. Covariance matrix adaptation evolution strategy (CMA-ES) has demonstrated competitive performance especially on multimodal non-separable problems. However, CMA-ES is not capable of dealing with problems having several hundreds dimensions. Motivated by that cooperative coevolution (CC) has scaled up many kinds of evolutionary algorithms (EAs) to high dimensional optimization problems effectively, we propose an algorithm called CC-CMA-ES which apply CC to CMA-ES in order to scale up CMA-ES to large scale problems. CC-CMA-ES adopts a new sampling scheme which does not divide population into small subpopulations and conducts mutation and crossover operations in subpopulation to generate offspring, but extracts a subspace Gaussian distribution from the global Gaussian distribution for subspace sampling. Also in CC-CMA-ES, two new decomposition strategies are proposed in order to balance exploration and exploitation. Lastly, an adaptive scheme is adopted to self-adapt appropriate decomposition strategy during evolution process. Experimental studies on a series of benchmark functions with different characteristic have been conducted and verified the excellent performance of our newly proposed algorithm and the effectiveness of the new decomposition strategies.

1 Introduction

Evolutionary algorithms (EAs) have been widely used in the field of numerical optimization [1]. However, they also suffer from "the curse of dimensionality" [2]. Cooperative coevolution (CC) [3] is an evolutionary framework based on divide-and-conquer strategy which has been used to scaling up some of EAs successfully [4–7]. DECC-G [4], proposed a decomposition scheme called random grouping which divides D-dimensional solution vector into m s-dimensional subvectors ($D = m \times s$) and conducts this many times each at the beginning

* Corresponding author.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 350–357, 2013.

of an optimization cycle to improves the probability of distributing interacting variables into a common subspace. MLCC [5] extended DECC-G by incorporating a self-adaptation strategy to select appropriate dimension for the subspace. CCPSO2 [6], imitating MLCC, adopted Cauchy and Gaussian based PSO as subspace optimizer. CCVIL [7] proposed a method to learn interacting variables other than random decomposition.

In this paper, we consider scaling up a representative kind of EAs, namely covariance matrix adaptation evolution strategy (CMA-ES) [8]. CMA-ES samples offspring via a multivariate Gaussian distribution and use offspring to update this distribution. It has demonstrated competitive performance compared with other EAs particular on multimodal non-separable problems [9]. However, CMA-ES also suffers from the curse of dimensionality. A variant of CMA-ES, namely sep-CMA-ES [10], has been proposed which can achieve linear time and space complexity. Unfortunately, experimental results have revealed the performance of sep-CMA-ES drops significantly with increasing dimensionality [11]. Motivated by all of those, we proposed an algorithm called CC-CMA-ES which apply CC to CMA-ES in order to scale up CMA-ES to large scale optimization problems. CC-CMA-ES adopts a new sampling scheme which does not divide population into small subpopulations and conducts mutation and crossover operations like that in DECC-G etc., but extracts a subspace Gaussian distribution from the global Gaussian distribution for subspace sampling. Also, two new decomposition strategies are proposed and an adaptive scheme is used to self-adapt appropriate decomposition strategy among two new decomposition strategies together with random based decomposition strategy during evolution process.

The rest of this paper is organized as follows. Section 2 describes CMA-ES briefly. Section 3 presents our new algorithm - CC-CMA-ES in detail, including new decomposition strategies and their adaptation scheme, sampling and update of distribution. Section 4 describes experimental setup and experimental results analysis. Finally, Section 5 concludes this paper.

2 Covariance Matrix Adaptation Evolution Strategy

As a respective of EAs, different from other EAs having mutation and crossover operations, CMA-ES firstly estimates a distribution from the samples and then takes sample to generate offspring using this distribution and repeats it until a stop criteria is satisfied.

$$x_k^{(g+1)} \sim N(x_w^g, (\sigma^g)^2 \cdot C^g), \qquad k = 1, 2, \cdots, \lambda \tag{1}$$

Equation 1 shows the sampling process at generation g. μ^g, C^g and σ^g are the Gaussian mean, covariance matrix and global step size respectively at generation g. CMA-ES has rotation invariant feature that is produced by C. This feature is the key of success for CMA-ES. When updating covariance matrix C, CMA-ES not only adopt current sample with better fitness but use history distribution information. Namely, CMA-ES combines *rank-μ* update and *rank-one* update [8]. The disadvantage of CMA-ES is its high time complexity, $O(D^3)$, where

D is the dimension of the solution vector, which avoids its application on high dimensional problems.

3 Proposed Algorithm

At the beginning of each optimization cycle of CC-CMA-ES, a decomposition strategy will be selected from a decomposition pool according their history performance and then the search space will be decomposed into several non-overlapping subspaces. Next, each subspace will be optimized sequently for a fixed number of fitness evaluations by a subspace CMA-ES whose parameters are extracted from a global CMA-ES and updated using the samples. At the end of each cycle, the performance record for the current selected decomposition strategy will be updated using its performance in this cycle. The same cycle will be conducted many times until a stop criteria is satisfied. The framework of CC-CMA-ES is showed in Algorithm 1 and each part of CC-CMA-ES will be described in detail as follows.

Algorithm 1. CC-CMA-ES

Require: space dimension dimension D, subspace dimension s, offspring size λ and fitness evaluation maximum $maxfes$

1: Initialize $D \times 1$ mean vector x_w, $D \times D$ covariance matrix C, global step size σ and other control parameters for global CMA-ES;
2: Initialize best solution *best* and the performance record;
3: Choose a decomposition strategy and decompose solution vector into m disjoint parts (so, each subspace has dimension $s = D/m$) (Sect. 3.2 in detail);
4: Set $sub = 1$ to start an optimization cycle;
5: Extract $s \times 1$ subspace mean vector x_{sub} and $s \times s$ covariance matrix C_{sub} from x_w and C and generate λ offspring using $N(x_{sub}, \sigma^2 \cdot C_{sub})$ (Sect. 3.3 in detail);
6: Evaluate offspring and update *best* if exists better offspring;
7: Update x_{sub}, C_{sub} and take them back to x_w, C (Sect. 3.4 in detail);
8: If the number of fitness evaluations reaches $maxfes$, stop. Else, if $sub < m$, $sub{+}{+}$ and go to step 5. Others, update performance record and go to step 3.

3.1 New Decomposition Strategies

In this paper, we propose two new decomposition strategies called Min-Variance decomposition strategy (MiVD) and Max-Variance decomposition strategy (MaVD) respectively. The emphasis of those new decomposition strategies is on the balance of exploration and exploitation.

MiVD and MaVD decompose space based on the diagonal of the covariance matrix. They firstly rank variables in the solution vector according to their variances in the diagonal of the covariance matrix and then separate the solution vector into several sub-vectors. MiVD decomposes variables with similar variances into a subspace to minimize the diversity among their variances. Opposite,

```
MaVD(D, m, C)
diagC ← diag(C);
s ← D/m;
subInfo ← ∅;
[sortedDiagC, sortedIndex]
  ← sort(diagC);
for i ← 1 to m
    S_i ← sortedIndex(i : s : D);
    subInfo ← subInfo ∪ {S_i};
end
return subInfo;
```

```
MiVD(D, m, C)
diagC ← diag(C);
s ← D/m;
subInfo ← ∅;
[sortedDiagC, sortedIndex]
  ← sort(diagC);
for i ← 1 to m
    S_i ← sortedIndex((i − 1) · s + 1 : i · s);
    subInfo ← subInfo ∪ {S_i};
end
return subInfo;
```

Fig. 1. Pseudocode of MaVD (left) and MiVD (right)

MaVD guarantees the diversity of the diagonal values of the variables in the same subspace as large as possible. Figure 1 shows their pseudocode. The sum of the volume of the tolerance region (hyperellipsoid) for each subspace with the same confidence by MiVD is larger than that of MaVD, which leads to MiVD is appropriate for exploration while MaVD for exploitation. From this perspective, random decomposition (RD) is a compromise between MiVD and MaVD.

3.2 Adaptive Decomposition Strategy Scheme

In CC-CMA-ES, three decomposition strategies – MaVD, MiVD and RD, will construct a pool and be used in an adaptive manner in order to accommodate various demands for decomposition strategy during environmental changeable evolution process like that in [5].

A performance record maintains the history performance for each decomposition strategy for a fixed number of optimization cycles (set to 5 in experiment), showed in Table 1. Rows indicate different decomposition strategies and columns indicate different cycles. Each item in the record demonstrates the fitness improvement rate of the best solution in this optimization cycle and is set to 1 initially. The probability of applying certain decomposition strategy is calculated by (3). At the beginning of each optimization cycle, we use stochastic universal selection method to select a decomposition strategy and record the fitness of the best solution found so far as $bestval_{old}$. After an optimization cycle, the fitness $bestval_{new}$ for new *best* will be obtained. The improvement rate for the decomposition strategy used in this cycle will be calculated according to (2) and the oldest history improvement of this decomposition strategy will be deleted to leave room for the new.

$$\delta_{ij} = \left| \frac{bestval_{new} - bestval_{old}}{bestval_{old}} \right| \tag{2}$$

$$probability_i = \frac{e^{\theta_i}}{\sum_{p=1}^{3} e^{\theta_p}} \qquad \text{where} \qquad \theta_i = \sum_{j=1}^{n} \delta_{ij} \ . \tag{3}$$

Table 1. Performance record

	$cycle_1$	$cycle_2$	$cycle_j$	$cycle_n$
Max-Variance decomposition	δ_{11}	δ_{12}	δ_{1j}	δ_{1n}
Random decomposition	δ_{21}	δ_{22}	δ_{2j}	δ_{2n}
Min-Variance decomposition	δ_{31}	δ_{32}	δ_{3j}	δ_{3n}

3.3 Subspace Sampling and Fitness Evaluation

After decomposition, subspaces have been obtained denoted as $S_i = \{x_{i1}, x_{i2}, \ldots, x_{is}\}$, $i = 1, 2, \ldots, m$, $D = m \times s$ where D is the dimension of the problem, m is the number of subspaces and s is the dimension of subspace. When optimizing subspace S_i, use Gaussian distribution $N(x_{S_i}, \sigma^2 \cdot C_{S_i})$ to generate λ offsprings. σ is the global step size. x_{S_i} and C_{S_i} are extracted from the global mean vector x_w and the covariance matrix C via the related dimension indexes in this subspace. For example, when $D = 4$, $s = 2$ and being optimized subspace S_i consists of dimension 1 and 3, namely $S_i = \{x_1, x_3\}$, the x_{S_i} and C_{S_i} will be extracted from the $D \times 1$ global mean vector x_w and the $D \times D$ covariance matrix C as follows.

$$\begin{bmatrix} 1.8 \\ 0.9 \\ 3.2 \\ 5.3 \end{bmatrix} \text{ and } \begin{bmatrix} 1.2 & 0.2 & 0.4 & 0.9 \\ 0.2 & 1.7 & 0.7 & 0.85 \\ 0.4 & 0.7 & 2.3 & 0.56 \\ 0.9 & 0.85 & 0.56 & 2.8 \end{bmatrix} \xrightarrow{extract} \begin{bmatrix} 1.8 \\ 3.2 \end{bmatrix} \text{ and } \begin{bmatrix} 1.2 & 0.4 \\ 0.4 & 2.3 \end{bmatrix}$$

We use greedy strategy [12] to evaluate the fitness for subspace samples. Namely, we take place of the values of the variables related to current subspace in the best solution *best* found so far using sampled offspring and use the fitness of this complete solution as the fitness of offspring. After optimization of each subspace, the best solution *best* will be updated if better solution is found.

3.4 Updating CMA-ES Parameters

x_{S_i}, C_{S_i} and σ will be updated using offspring [8]. Then we will take place the portion of the global mean x_w and covariance matrix C related to current subspace while other portions are fixed by those subspace parameters to guarantee the newest correlation information for subsequent usage, that is, this procedure is the inverse process of extracting x_{S_i} and C_{S_i} from x_w and C. As the example in 3.3, after optimizing subspace $S_i = 1, 3$, updating the subspace mean vector x_{S_i} and the subspace covariance matrix C_{S_i} and take those back to the global mean vector x_w and the global covariance matrix C is showed as follows.

$$\begin{bmatrix} 1.8 \\ 3.2 \end{bmatrix} \begin{bmatrix} 1.2 & 0.4 \\ 0.4 & 2.3 \end{bmatrix} \xrightarrow{update} \begin{bmatrix} 2.8 \\ 4.2 \end{bmatrix} \begin{bmatrix} 2.2 & 0.88 \\ 0.88 & 1.9 \end{bmatrix} \xrightarrow{reverse} \begin{bmatrix} 2.8 \\ 0.9 \\ 4.2 \\ 5.3 \end{bmatrix} \begin{bmatrix} 2.2 & 0.2 & 0.88 & 0.9 \\ 0.2 & 1.7 & 0.7 & 0.85 \\ 0.88 & 0.7 & 4.2 & 0.56 \\ 0.9 & 0.85 & 0.56 & 2.8 \end{bmatrix}$$

Table 2. The results are the mean best values averaged over 25 independent runs and standard deviation. Wilcoxon signed rank test with significance 0.05 for CC-CMA-ES and each other algorithm have been conducted and statistically significant results are labeled by superscript *.

Function	Dim		DECC-G	MLCC	CCPSO2	CC-CMA-ES
f_1	1000	mean	$6.28e-06^*$	**1.24e-23***	$2.94e-05^*$	$5.77e-09$
		std	$9.62e-06$	$4.94e-23$	$4.39e-05$	$1.00e-09$
f_2	1000	mean	$1.32e+03$	**5.67e+00***	$6.63e+00^*$	$1.33e+03$
		std	$5.23e+01$	$8.92e+00$	$3.14e+01$	$1.11e+02$
f_3	1000	mean	$1.11e+00^*$	$1.41e-10$	$8.41e-06^*$	**1.51e-13**
		std	$2.70e-01$	$7.02e-10$	$7.54e-06$	$6.73e-15$
f_4	1000	mean	$2.25e+11^*$	$1.03e+11^*$	$7.21e+11^*$	**2.20e+09**
		std	$1.43e+11$	$4.41e+10$	$9.66e+11$	$1.31e+09$
f_5	1000	mean	$7.28e+14$	$7.28e+14$	$7.28e+14^*$	$7.28e+14$
		std	$1.65e+07$	$1.68e+06$	$3.17e+05$	$4.65e+06$
f_6	1000	mean	**1.84e+05***	$9.07e+05$	$8.60e+05$	$5.83e+05$
		std	$2.89e+04$	$1.84e+05$	$3.03e+05$	$4.79e+05$
f_7	1000	mean	$9.92e+08^*$	$8.64e+08^*$	$3.21e+09^*$	**7.45e+06**
		std	$5.62e+08$	$8.10e+08$	$2.18e+09$	$1.21e+07$
f_8	1000	mean	$6.82e+15^*$	$5.25e+15^*$	$1.47e+16^*$	**3.88e+14**
		std	$3.17e+15$	$2.83e+15$	$1.47e+16$	$2.87e+14$
f_9	1000	mean	$5.78e+08^*$	$8.51e+08^*$	$7.66e+08^*$	**3.71e+08**
		std	$1.52e+08$	$1.73e+08$	$3.82e+08$	$1.83e+08$
f_{10}	1000	mean	$2.91e+07^*$	$5.08e+07^*$	$4.02e+07^*$	**7.55e+05**
		std	$1.14e+07$	$2.51e+07$	$3.82e+07$	$5.02e+05$
f_{11}	1000	mean	$1.21e+11^*$	$1.08e+11^*$	$4.49e+11^*$	**1.59e+08**
		std	$7.35e+10$	$8.53e+10$	$3.04e+11$	$1.47e+08$
f_{12}	1000	mean	$9.58e+03^*$	$3.33e+03^*$	$1.46e+03$	**1.27e+03**
		std	$6.30e+03$	$2.98e+03$	$5.17e+02$	$4.26e+02$
f_{13}	905	mean	$8.88e+09^*$	$7.20e+09^*$	$2.73e+10^*$	**6.70e+08**
		std	$3.21e+09$	$2.96e+09$	$7.43e+09$	$1.14e+09$
f_{14}	905	mean	$1.34e+11^*$	$1.09e+11^*$	$5.50e+11^*$	**7.10e+07**
		std	$5.18e+10$	$6.98e+10$	$2.55e+11$	$1.25e+08$
f_{15}	1000	mean	$1.20e+07^*$	**7.44e+06***	$5.30e+08^*$	$3.03e+07$
		std	$9.91e+05$	$1.33e+06$	$9.26e+08$	$6.08e+06$

4 Experimental Studies

4.1 Experimental Setup

We have chosen the benchmark functions provided by *CEC2013 Special Session on large scale global optimization* [13] for evaluating CC-CMA-ES and comparisons with other algorithms. The algorithms for comparisons consist of DECC-G, MLCC and CCPSO2 all of which are state-of-art CC based algorithms and have demonstrated remarkable performance on large scale global optimization. The control parameters used in those algorithms could be found in [4–6]. For CC-CMA-ES, the size of offspring is set to 50 and the number of subspaces is set to 20. We set the maximum fitness evaluations for each iteration of optimizing a certain subspace to $10 \times s$ where s is the dimension of this subspace. For fair comparison, the stop criterion for all algorithms is the same number of fitness evaluations $3e6$ and each algorithm will run for 25 times independently.

4.2 Experimental Results

From Table 2, CC-CMA-ES performs significantly better on eleven out of fifteen functions compared with DECC-G, which demonstrates that CC-CMA-ES is more effective in capturing the correlation between variables through the use of covariance matrix. CC-CMA-ES was outperformed significantly by DECC-G on function 6 and 15 where function 6 are partially separable function and function 15 are fully non-separable function. The reason may be the decomposition of space in CC-CMA-ES results into the correlation among variables could not be updated timely and accurately. CC-CMA-ES outperforms MLCC significantly on nine functions and was outperformed significantly on function 1, 2 and 15. It is noteworthy that function 1 and 2 are fully-separable functions. And on the other fully-separable function 3, CC-CMA-ES does not outperform MLCC significantly. Combing the results of DECC-G, we could find the effectiveness of MLCC derived from the self-adaptative scheme of subspace dimension. The comparison result on function 15 is similar to that of DECC-G, which reveals the disadvantage of CC-CMA-ES on fully non-separable problems. CC-CMA-ES outperforms CCPSO2 significantly on all functions except function 2 and 6. The phenomenon of CC-CMA-ES was outperformed on function 2 shows similar result to that of MLCC, which demonstrated the relative weakness of CC-CMA-ES on fully separable functions compared with its strength on partially additive separable functions.

It is noteworthy CC-CMA-ES outperforms best on all three overlapping functions 12, 13 and 14. Overlapping functions were included in *CEC2013* benchmark functions for the first time, and no related results on those functions have been reported. The effectiveness of CC-CMA-ES on overlapping functions shows the advantage of combing CMA-ES and CC framework where CMA-ES can capture the interdependence among variables and CC can alleviate the hardness of CMA-ES on high dimension problems.

5 Conclusion

In this paper, we propose a new algorithm CC-CMA-ES which scales up CMA-ES to high dimensional problems using CC. In this algorithm, two new decomposition strategies are proposed which are based on the diagonal of the covariance matrix of CMA-ES to keep the balance between exploration and exploitation in evolution process. To coordinate different decomposition strategy, an adaptive decomposition strategy scheme is adopted which could select appropriate decomposition strategy adaptively. Comprehensive experiment studies on large scale problems demonstrated the effectiveness of our proposed algorithm and verified the validity of the new decomposition strategies and adaptive decomposition strategy scheme. Although the effectiveness of the two new decomposition strategies has been verified on the whole, it still needs studies on fully non-separable problems. Moreover, those two new decomposition strategies could be used in other EAs in order to study their performance in various circumstances.

Acknowledgment. This work was supported in part by the 973 Program of China under Grant 2011CB707006, the National Natural Science Foundation of China under Grant 61175065, the Program for New Century Excellent Talents in University under Grant NCET-12-0512, and the Science and Technological Fund of Anhui Province for Outstanding Youth under Grant 1108085J16.

References

1. Sarker, R., Mohammadian, M., Yao, X.: Evolutionary optimization, vol. 48. Kluwer Academic Pub. (2002)
2. Van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 225–239 (2004)
3. Potter, M.A., De Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evolutionary Computation 8(1), 1–29 (2000)
4. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Sciences 178(15), 2985–2999 (2008)
5. Yang, Z., Tang, K., Yao, X.: Multilevel cooperative coevolution for large scale optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 1663–1670. IEEE (2008)
6. Li, X., Yao, Y.: Cooperatively coevolving particle swarms for large scale optimization. IEEE Transactions on Evolutionary Computation 16(2), 1–15 (2011)
7. Chen, W., Weise, T., Yang, Z., Tang, K.: Large-scale global optimization using cooperative coevolution with variable interaction learning. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 300–309. Springer, Heidelberg (2010)
8. Hansen, N.: The cma evolution strategy: A tutorial. Vu le 29 (2005)
9. Hansen, N., Kern, S.: Evaluating the CMA evolution strategy on multimodal test functions. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 282–291. Springer, Heidelberg (2004)
10. Ros, R., Hansen, N.: A simple modification in CMA-ES achieving linear time and space complexity. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 296–305. Springer, Heidelberg (2008)
11. Omidvar, M.N., Li, X.: A comparative study of cma-es on large scale global optimisation. In: Li, J. (ed.) AI 2010. LNCS, vol. 6464, pp. 303–312. Springer, Heidelberg (2010)
12. Wiegand, R.P., Liles, W.C., De Jong, K.A.: An empirical analysis of collaboration methods in cooperative coevolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), vol. 2611, pp. 1235–1245 (2001)
13. Li, X., Tang, K., Omidvar, M.N., Yang, Z., Qin, K., China, H.: Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. Gene 7, 33 (2013)