MOEA/D Using Covariance Matrix Adaptation Evolution Strategy for Complex Multi-Objective Optimization Problems

Ting-Chen Wang, Rung-Tzuo Liaw, and Chuan-Kang Ting
Department of Computer Science and Information Engineering and
Advanced Institute of Manufacturing with High-tech Innovations
National Chung Cheng University
Chia-Yi 621, Taiwan
Email: {wtc102p, lrt101p, ckting}@cs.ccu.edu.tw

***Abstract*—Multi-objective optimization is a blooming research area since many real-world problems comprise multiple objectives. Multi-objective evolutionary algorithms (MOEAs) have been widely used to solve the multi-objective optimization problems. In particular, the decomposition based MOEA (MOEA/D) has achieved considerable successes in tackling multi-objective optimization problems. The covariance matrix adaptation evolution strategy (CMAES) is known for its effectiveness in solving complex numerical optimization problems. This study integrates CMAES into MOEA/D as the MOEA/D-CMAES for the merits of MOEA/D framework in multi-objective optimization and CMAES in complex numerical optimization. In MOEA/D-CMAES, each subproblem is handled with one CMAES. To avoid the drastic increase in the number of offspring generated and their fitness evaluations, MOEA/D-CMAES generates only one offspring in each subproblem. The multivariate normal distribution in each CMAES is updated by the collaboration of the offspring generated in the present subproblem and those of other subproblems. Experimental results show that MOEA/D-CMAES outperforms MOEA/D using differential evolution in terms of hypervolume and convergence speed, which validate the effectiveness and efficiency of MOEA/D-CMAES in multi-objective optimization.**

***Index Terms*—Multi-objective optimization, MOEA/D, CMAES, evolution strategy, complex objective functions.**

I. Introduction

Multi-objective optimization is an important and blooming research area. Unlike the traditional single-objective optimization problems, multi-objective optimization problems (MOPs) need to optimize multiple conflicting objectives, in which the improvement on a certain objective may cause deterioration on other objectives. A classic example is the trade-off between cost and performance while buying a car. The mathematical form of MOPs can be formulated by

$$\min_{\boldsymbol{x}\in\Omega} F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T,$$

where Ω denotes the design space, and $f_1, \ldots, f_k$ are the k objective functions. The MOPs have not only one optimal solution but several non-dominated solutions. A solution $\boldsymbol{x}$ is said to *dominate* a solution $\boldsymbol{y}$ if $\boldsymbol{x}$ is better than $\boldsymbol{y}$ in one objective and is not worse than $\boldsymbol{y}$ in all other objectives. The non-dominated solutions represent the solutions that are not dominated by any solutions. The set of non-dominated solutions forms the Pareto set and Pareto front in design space and objective space, respectively.

Multi-objective evolutionary algorithms (MOEAs) have been widely used to solve the MOPs. According to the criterion used in survival selection, MOEAs can be classified into three categories, i.e., decomposition based MOEA [1, 2], Pareto dominance based MOEA [3], and indicator based MOEA [4]. Each category of MOEAs has different selection pressures and search behaviors. The MOEA based on decomposition (MOEA/D) [1] achieves many successes in tackling the MOPs. MOEA/D divides an MOP into several scalar subproblems and each subproblem serves as a single-objective optimization problem. Therefore, MOEA/D solves each subproblem individually by canonical EAs such as genetic algorithm, particle swarm optimization, and differential evolution. A state-of-the-art decomposition based MOEA is the MOEA/D-DE [2], which uses differential evolution (DE) to deal with subproblems.

Covariance matrix adaptation evolution strategy (CMAES) [5, 6] is a powerful evolutionary algorithm for complex numerical optimization. Different from conventional evolution strategy (ES), CMAES adopts a multivariate normal distribution to model the population. The offspring is generated by sampling from the multivariate normal distribution, instead of the crossover operator that is widely used in evolutionary algorithms. CMAES has gained great successes and is viewed as a state-of-the-art evolutionary algorithm for complex numerical optimization problems.

This paper proposes the MOEA/D-CMAES by integrating CMAES into MOEA/D, aiming for their advantages in numerical optimization and multi-objective optimization, respectively. To accommodate CMAES to the multiple subproblems of MOEA/D, we modify the reproduction scheme and survival selection of CMAES and utilize the decomposition structure of MOEA/D. To evaluate the performance of MOEA/D-CMAES, this study presents six complex multi-objective optimization problems (CMOPs), each consisting of two complex single-

978-1-5090-0623-6/16/$31.00

objective problems. A series of experiments is conducted to examine the effectiveness of MOEA/D-CMAES

The remainder of this paper is organized as follows. Section II reviews the related work on MOEA/D and CMAES. Section III describes the proposed MOEA/D-CMAES in detail. Section IV presents and discusses the experimental results of MOEA/D-CMAES. Finally, Section V concludes this study.

II. Related Work

A. MOEA/D

Based on the concept of decomposition, MOEA/D [1] divides an MOP into several single-objective problems by a set of weight vectors. Each weight vector determines a scalar subproblem. The population of MOEA/D renders a set of solutions, each of which is associated with a subproblem. Hence, the population size N in MOEA/D is usually equal to the number of weight vectors. In generating the weight vector set, a typical way is to distribute weight vectors uniformly on the objective space. The population size N is decided by

$$N = \begin{pmatrix} H+k-1 \\ k-1 \end{pmatrix}, \tag{1}$$

where H denotes the fidelity for splitting the objective space. The fitness with respect to each subproblem is defined by the decomposition approach. A common decomposition approach is the Tchebycheff (TCH) approach. For a given solution x and weight vector w, the TCH approach calculates the fitness by

$$g^{TCH}(x|w, z^{\min}) = \max_{1 \le i \le k} \left\{ w_i \left| f_i(x) - z_i^{\min} \right| \right\},$$

where $z^{\min}$ is a vector consisting of best-so-far solutions for each objective, i.e., $z^{\min} = \left(z_1^{\min}, z_2^{\min}, \ldots, z_k^{\min}\right)$ and $z_i^{\min}$ stands for the best-so-far solution for the i-th objective. Nonetheless, several studies point out the issue that the convergence of the TCH approach is non-uniform [7, 8]. The subproblem along weight vector w defined by the TCH approach converges on the vector $\left(\frac{1}{w_1}, \frac{1}{w_2}, \ldots, \frac{1}{w_k}\right)$ rather than $(w_1, w_2, \ldots w_k)$. Li et al. [9] proposed a revised TCH approach to address this issue by

$$g^{TCH}(x|w, z^{\min}) = \max_{1 \le i \le k} \left\{ \frac{\left| f_i(x) - z_i^{\min} \right|}{w_i} \right\}.$$

This study adopts the revised TCH approach due to its generally better performance than the original TCH approach. Through the decomposition approach, solutions for a specific subproblem can be compared according to the above fitness function.

Algorithm 1 shows the framework of MOEA/D. At beginning, MOEA/D initializes the weight vector set $\mathbf{W}$ and population $\mathbf{X}$. In each generation, every subproblem generates an offspring and performs survival selection. There exist two types of update operations for generating offspring and performing survival selection: global update and local update. Global update selects parents from the whole population $\mathbf{X}$ for reproduction; by contrast, local update considers only the parents within neighborhood $\mathbf{X}^w$. The neighborhood $\mathbf{X}^w$ represents the set of solutions associated with the neighboring subproblems of w. The survival selection in MOEA/D is also operated globally or locally, depending upon the type of update. Global and local survival selection considers the population $\mathbf{X}$ and neighborhood $\mathbf{X}^w$, respectively. The process of MOEA/D continues until the termination criterion is met.

Algorithm 1 Framework of MOEA/D

N: Population size
$\mathbf{W}$: Weight vectors set
$\mathbf{X}$: Population of MOEA/D
$\mathbf{X}^w$: Solutions from subproblems neighboring w
Υ: Parent set

Initialize $\mathbf{W}$ ▷ compute N by (1)
Initialize $\mathbf{X}$ ▷ $\mathbf{X} = \{x_1, \ldots, x_N\}$
for $t \leftarrow 1$ **to** T **do**
  for each $w \in \mathbf{W}$ **do**
    if Global update **then**
      $\Upsilon \leftarrow \mathrm{Selection}\,(\mathbf{X})$
      $\hat{x} \leftarrow \mathrm{Variation}\,(x_w, \Upsilon)$
      Survival$(\hat{x}, \mathbf{X})$ ▷ update population
    else
      $\Upsilon \leftarrow \mathrm{Selection}\,(\mathbf{X}^w)$
      $\hat{x} \leftarrow \mathrm{Variation}\,(x_w, \Upsilon)$
      Survival$(\hat{x}, \mathbf{X}^w)$ ▷ update neighbors
    end if
  end for
end for

B. CMAES

Unlike other evolutionary algorithms, CMAES exploits a multivariate normal distribution to model the population [10]:

$$\mathcal{N}\left(\boldsymbol{m}, \boldsymbol{C}\right) \sim \boldsymbol{m} + \mathcal{N}\left(0, \boldsymbol{C}\right).$$

The multivariate normal distribution consists of the mean vector $\boldsymbol{m}$ and the covariance matrix $\boldsymbol{C}$. The mean vector is the weighted center of population $\mathbf{X}_c = \{x_1, \ldots, x_\mu\}$ according to the rank of fitness. The covariance matrix $\boldsymbol{C}$, constructed by eigendecomposition, is associated with the rotation and scaling of design space. More details about the multivariate normal distribution refer to [6]. In CMAES, offspring $\hat{\mathbf{X}}_c = \{\hat{x}_1, \ldots, \hat{x}_\lambda\}$ are sampled from the multivariate normal distribution rather than variation operators.

Algorithm 2 shows the pseudocode of CMAES. First, CMAES initializes a multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$. The offspring $\hat{\mathbf{X}}_c$ are sampled from $\mathcal{N}\left(\boldsymbol{m}, \boldsymbol{C}\right)$ in each generation. The best μ individuals of $\hat{\mathbf{X}}_c$ are used to update the multivariate normal distribution $\mathcal{N}\left(\boldsymbol{m}, \boldsymbol{C}\right)$. This evolutionary process continues until the predetermined stopping criterion is satisfied.

Algorithm 2 CMAES.

Initialize $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$
$t \leftarrow 0$
while $t < T$ **do**
  Generate $\hat{\mathbf{X}}_c \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ ▷ $\hat{\mathbf{X}}_c = \{\hat{x}_1, \ldots, \hat{x}_\lambda\}$
  Evaluate $\hat{\mathbf{X}}_c$
  $\mathbf{X}_c \leftarrow \text{Survival}(\hat{\mathbf{X}}_c)$ ▷ $\mathbf{X}_c = \{x_1, \ldots, x_\mu\}$
  Update $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ using $\mathbf{X}_c$
  $t \leftarrow t + \lambda$ ▷ #evaluations
end while

III. MOEA/D Using CMAES

In view of the advantages of CMAES in complex numerical optimization, this study proposes the MOEA/D-CMAES by integrating CMAES into MOEA/D. In MOEA/D-CMAES, each subproblem is handled by an individual CMAES. Thus, the number of CMAES instances is equal to the population size N. To avoid the drastic increase in the number of offspring generated and their fitness evaluations, MOEA/D-CMAES generates only one offspring in each subproblem. However, one offspring is insufficient to update the multivariate normal distribution. To address this issue, MOEA/D-CMAES includes the offspring generated from other subproblems in the update of multivariate normal distribution. Following the notation of original CMAES(μ, λ), the MOEA/D-CMAES(μ,$1+(N-1)$) represents that each CMAES uses $\lambda = N$ offspring, where one offspring is generated from the present subproblem and $N-1$ offspring are borrowed from other subproblems. According to the fitness values with respect to the present weight vector, μ out of N offspring are selected to model the multivariate normal distribution for a certain subproblem. In MOEA/D-CMAES, the fitness used in each CMAES is defined by the decomposition approach described in Section II. Restated, the multivariate normal distribution $\mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$ for the subproblem with weight vector w is updated by the best μ offspring according to the decomposition approach $g\left(\cdot|w, z^{\min}\right)$.

Algorithm 3 presents the proposed MOEA/D-CMAES. At initialization, the weight vector set $\mathbf{W}$ is first defined. The multivariate normal distribution $\mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$ for each subproblem w is then initialized, from which the first population $\mathbf{X}$ is constructed by sampling. In each generation, offspring $\hat{\mathbf{X}}_c$ for a subproblem w are generated by sampling from the multivariate normal distribution $\mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$. Following MOEA/D, the survival selection includes global and local selection: The $\text{Survival}(\hat{x}, \mathbf{X})$ acts upon the whole population $\mathbf{X}$, whereas the $\text{Survival}(\hat{x}, \mathbf{X}^w)$ considers only the solutions in the neighborhood $\mathbf{X}^w$. Afterward, each multivariate normal distribution $\mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$ associated with weight vector w is updated with the μ surviving offspring $\mathbf{X}_c$ according to the decomposition approach. The process of MOEA/D-CMAES continues until the predefined number of generations is reached.

Algorithm 3 MOEA/D-CMAES

N: Population size
$\mathbf{W}$: Weight vectors set
$\mathbf{X}$: Population of MOEA/D
$\mathbf{X}^w$: Solutions from subproblems neighboring w
$\mathbf{X}_c$: Population of CMAES

Initialize $\mathbf{W}$ ▷ compute N by (1)
$\mathbf{X} \leftarrow \phi$
for each $w \in \mathbf{W}$ **do**
  Initialize $\mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$
  Generate $\hat{x} \sim \mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$
  $\mathbf{X} \leftarrow \mathbf{X} \cup \{\hat{x}\}$
end for
for $t \leftarrow 1$ **to** T **do**
  $\hat{\mathbf{X}}_c \leftarrow \phi$
  for each $w \in \mathbf{W}$ **do**
    Generate $\hat{x} \sim \mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$
    $\hat{\mathbf{X}}_c \leftarrow \hat{\mathbf{X}}_c \cup \{\hat{x}\}$
    if Global update **then**
      $\text{Survival}(\hat{x}, \mathbf{X})$ ▷ update population
    else
      $\text{Survival}(\hat{x}, \mathbf{X}^w)$ ▷ update neighbors
    end if
  end for
  for each $w \in \mathbf{W}$ **do**
    Evaluate $\hat{\mathbf{X}}_c$ by decomposition approach
    $\mathbf{X}_c \leftarrow \text{Survival}(\hat{\mathbf{X}}_c)$
    Update $\mathcal{N}(\boldsymbol{m}^w, \boldsymbol{C}^w)$ using $\mathbf{X}_c$
  end for
end for

IV. Experimental Results

This study conducts several experiments to examine the effectiveness and efficiency of the proposed MOEA/D-CMAES, in comparison to the state-of-the-art MOEA/D-DE. Considering that the objective functions used in the conventional MOP benchmarks are relatively simple, this study presents six test problems CMOPs (cf. Table I) consisting of complex single-objective numerical optimization problems selected from the CEC2014 test suite [11]. All the objective functions in the six CMOPs are shifted and rotated multi-modal functions; thus, they are asymmetric to the main axes. The experiments are carried out on the functions of low ($n = 10$) and medium ($n = 30$) dimensions.

In this study, we use hypervolume as the performance metric due to the lack of true Pareto front. Hypervolume computes the area covered by the obtained front to the reference point. Since the optimal solution of each objective is known, the reference point for hypervolume is set to $\{f_1(f_2^*), f_2(f_1^*)\}$, where f_1^* and f_2^* are the optimal solutions of the first objective f_1 and second objective f_2, respectively.

Table II lists the parameter setting for MOEA/D-DE and MOEA/D-CMAES. The population size (N), fidelity (H), neighborhood size (S), selection probability (δ), and the

Table I: Objective functions of complex multi-objective numerical optimization problems (CMOPs)

Problem	Objective 1	Objective 2
CMOP1	Shifted and Rotated Ackley's Function (f_5)	Shifted and Rotated Weierstrass Function (f_6)
CMOP2	Shifted and Rotated Katsuura Function (f_{12})	Shifted and Rotated HappyCat Function (f_{13})
CMOP3	Shifted and Rotated Ackley's Function (f_5)	Shifted and Rotated Katsuura Function (f_{12})
CMOP4	Shifted and Rotated Ackley's Function (f_5)	Shifted and Rotated HappyCat Function (f_{13})
CMOP5	Shifted and Rotated Weierstrass Function (f_6)	Shifted and Rotated Katsuura Function (f_{12})
CMOP6	Shifted and Rotated Weierstrass Function (f_6)	Shifted and Rotated HappyCat Function (f_{13})

Table II: Parameter setting

Algorithm	Parameter	Value
MOEA/D	Population size of MOEA/D (N)	100
	Fidelity (H)	99
	Neighborhood size (S)	20
	Selection probability (δ)	0.9
	#Replacements (n_r)	2
DE	Crossover rate (CR)	1.0
	Factor of deference vector (F)	0.5
	Distribution index (η)	20
	Mutation rate (p_m)	$1/n$
CMAES	Population size (μ)	10
	Offspring size (λ)	1+99

Table III: Hypervolume obtained from MOEA/D-DE and MOEA/D-CMAES on the six CMOPs of dimensions $n = 10$ and $n = 30$. The p-value accounts for the Student's t-test results. Bold hypervolume values indicate the better of the two algorithms and bold p-values reflect the statistical significance under confidence level $\alpha = 0.05$.

		MOEA/D		p-value
		DE	CMAES	
$n = 10$	CMOP1	23.52	**24.94**	**2.12E-04**
	CMOP2	284.15	**286.71**	**1.49E-02**
	CMOP3	**19.56**	19.04	**3.38E-02**
	CMOP4	15.02	15.08	1.50E-01
	CMOP5	309.54	**339.56**	**1.85E-06**
	CMOP6	153.12	153.64	3.84E-01
$n = 30$	CMOP1	70.71	**91.16**	**2.25E-20**
	CMOP2	203.31	**206.09**	**8.24E-09**
	CMOP3	24.97	**25.53**	**8.17E-10**
	CMOP4	23.27	**23.96**	**2.31E-15**
	CMOP5	1052.97	**1410.33**	**6.83E-24**
	CMOP6	671.88	**688.22**	**1.62E-03**

number of replacements (n_r) are the general parameters for MOEA/D. The crossover rate (CR), factor of deference vector (F), distribution index (η) and mutation rate (p_m) are the specific parameters for DE. The population size μ and offspring size λ are for the CMAES in MOEA/D-CMAES. The parameter setting follows [2]. Each experiment includes 30 trials of test algorithms.

A. *Hypervolume*

First, we investigate the performance of MOEA/D-CMAES in terms of hypervolume. Table III compares the hypervolume obtained from MOEA/D-DE and MOEA/D-CMAES on the six CMOPs of dimensions $n = 10$ and $n = 30$. On the low dimensional ($n = 10$) problems, the proposed MOEA/D-CMAES gains significantly better hypervolume than MOEA/D-DE does on 3 out of 6 problems (CMOP1, CMOP2 and CMOP5), comparable on 2 problems, and inferior on CMOP3. As the dimension increases to 30, the advantage of MOEA/D-CMAES over MOEA/D-DE becomes more significant: MOEA/D-CMAES outperforms MOEA/D-DE with statistical significance on all the six problems. These satisfactory outcomes validate the effectiveness of MOEA/D-CMAES in solving complex MOPs.

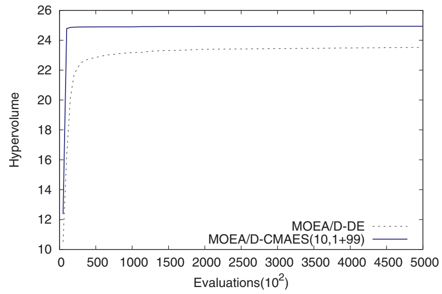
B. *Convergence*

Next, we look into the convergence speed of MOEA/D-CMAES. Figures 1 and 2 show the anytime behavior of MOEA/D-DE and MOEA/D-CMAES in terms of hypervolume against fitness evaluations. On the low dimensional ($n = 10$) problems, MOEA/D-CMAES converges faster and gains better hypervolume on 3 problems, i.e., CMOP1, CMOP5 and CMOP6. The results show that MOEA/D-DE suffers from premature convergence on CMOP2 and CMOP4, but converges faster than MOEA/D-CMAES on CMOP3. Furthermore, MOEA/D-CMAES surpasses MOEA/D-DE in terms of convergence speed and hypervolume on all the six CMOPs with medium dimensional ($n = 30$). The results indicate that MOEA/D-CMAES is more efficient than MOEA/D-DE on the CMOPs, especially those with higher dimensions.

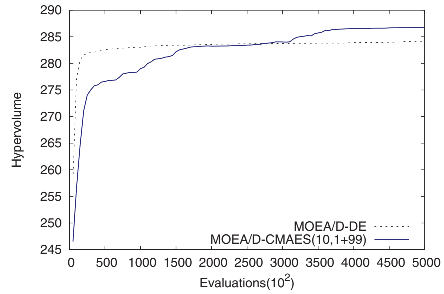
C. *Resultant Fronts*

Figures 3 and 4 further compare the fronts obtained from MOEA/D-DE and MOEA/D-CMAES on the CMOPs. On the low dimensional problems, MOEA/D-CMAES obtains better front than MOEA/D-DE does on CMOP1 and CMOP5. The obtained fronts of MOEA/D-CMAES and MOEA/D-DE on CMOP2 and CMOP6 are comparable. On CMOP4, the front of MOEA/D-CMAES dominates most of the solutions of MOEA/D-DE; however, this situation is not reflected on the hypervolume values. The front of MOEA/D-DE betters that of MOEA/D-CMAES on CMOP3, which is consistent to the results in hypervolume. As for the medium dimensional problems, MOEA/D-CMAES excels MOEA/D-DE on 5 problems except CMOP6. On CMOP6, MOEA/D-CMAES results in a front with wider spread but lower proximity than MOEA/D-DE does.

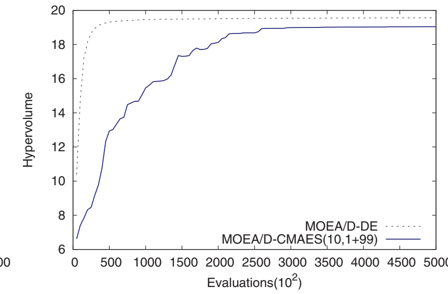
The above results validate the effectiveness and efficiency of MOEA/D-CMAES in dealing with complex multi-objective optimization problems. In comparison to MOEA/D-DE, the



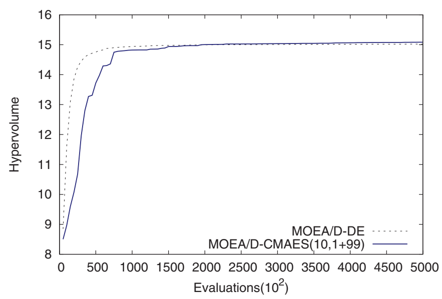
(a) CMOP1



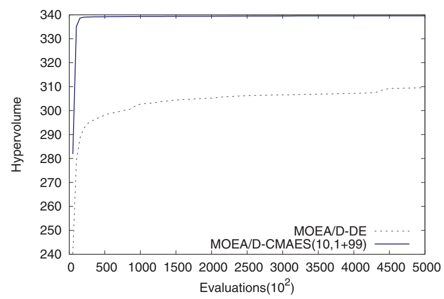
(b) CMOP2



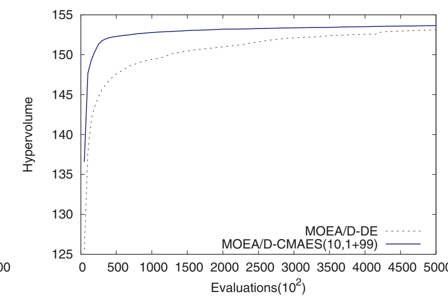
(c) CMOP3



(d) CMOP4

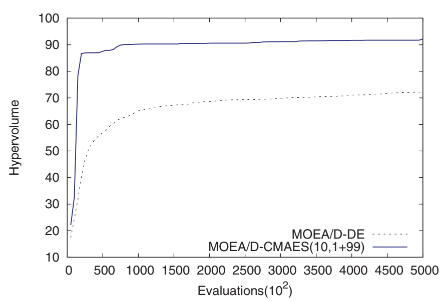


(e) CMOP5

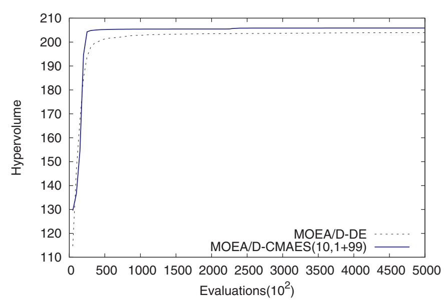


(f) CMOP6

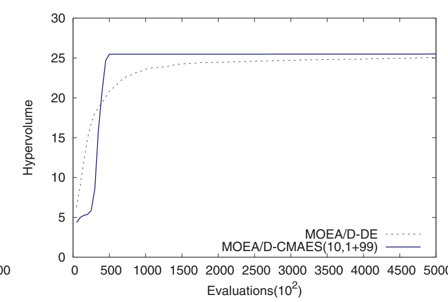
Figure 1: Variation of hypervolume against the number of fitness evaluations for MOEA/D-DE and MOEA/D-CMAES on the six CMOPs with dimension $n = 10$



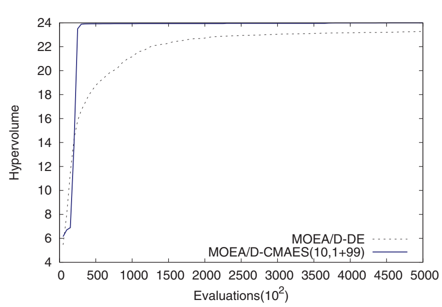
(a) CMOP1



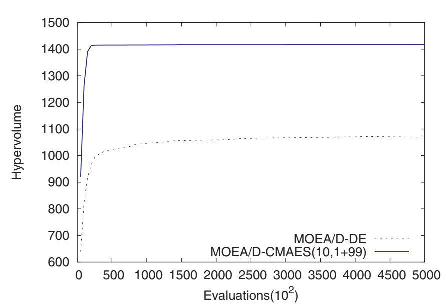
(b) CMOP2



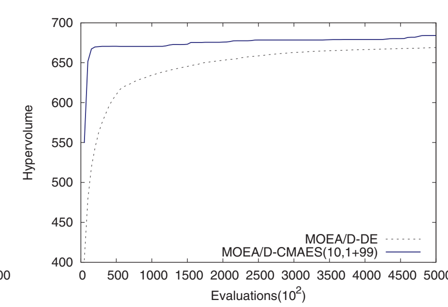
(c) CMOP3



(d) CMOP4



(e) CMOP5



(f) CMOP6

Figure 2: Variation of hypervolume against the number of fitness evaluations for MOEA/D-DE and MOEA/D-CMAES on the six CMOPs with dimension $n = 30$

higher hypervolume of MOEA/D-CMAES shows its effectiveness for CMOPs; in addition, the faster convergence speed exhibits the efficiency of MOEA/D-CMAES. The comparison of resultant fronts further confirms the advantages of MOEA/D-CMAES over MOEA/D-DE.

V. Conclusions

Multi-objective optimization is an important area since many real-world optimization problems contain multiple conflicting objectives. The MOEA/D has proved to be an effective tool for tackling MOPs. On the other hand, CMAES is known as a powerful evolutionary algorithm for complex numerical optimization. This study proposes the MOEA/D-CMAES by integrating CMAES into MOEA/D, aiming for the advantages of MOEA/D in multi-objective optimization and CMAES in complex numerical optimization. In MOEA/D-CMAES, each subproblem is solved by a CMAES. To reduce the number of fitness evaluations required, each CMAES in MOEA/D-CMAES generates only one offspring. By taking the offspring generated in other subproblems into account, the multivariate normal distribution of each CMAES can be updated without the increase in fitness evaluations.

A series of experiments is conducted to examine the performance of MOEA/D-CMAES in comparison to MOEA/D-DE. This study presents six CMOPs based on the complex benchmark functions of the CEC2014 test suite. The experimental results show that MOEA/D-CMAES can significantly outperform MOEA/D-DE in terms of hypervolume and convergence speed, especially on the CMOPs with higher dimensions. In addition, MOEA/D-CMAES can achieve better fronts than MOEA/D-DE does on the six CMOPs. These preferable outcomes validate the effectiveness and efficiency of MOEA/D-CMAES on complex multi-objective optimization problems.

Acknowledgment

This work was supported by the Ministry of Education of Taiwan.

References

[1] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[2] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.

[5] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[6] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards A New Evolutionary Computation*, J. A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, Eds. Springer, 2006, pp. 75–102.

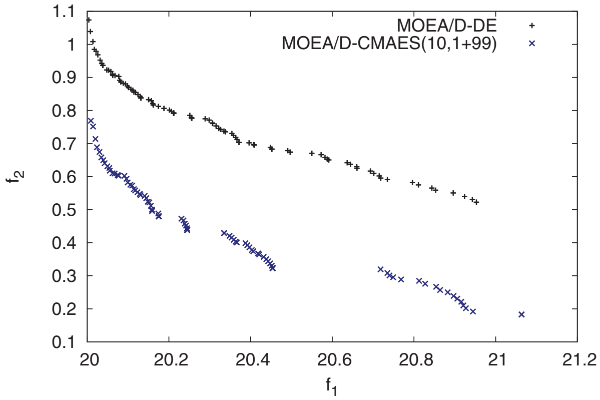
[7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[8] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

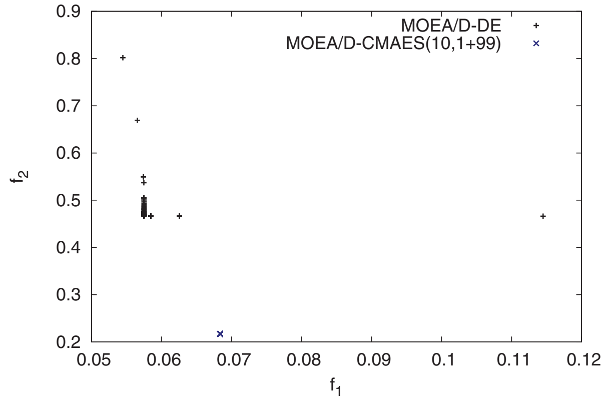
[9] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[10] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

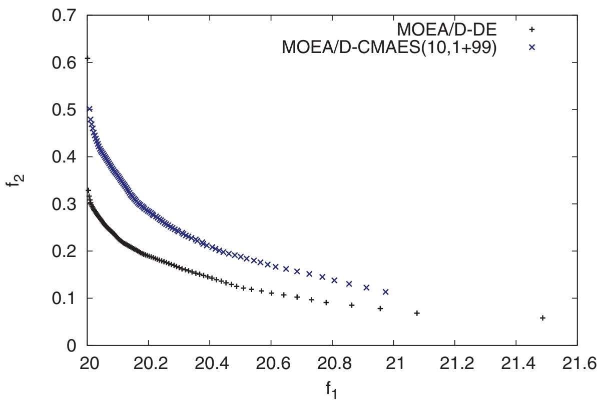
[11] J. J. Liang, B.-Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," Nanyang Technological University, Tech. Rep., 2013.



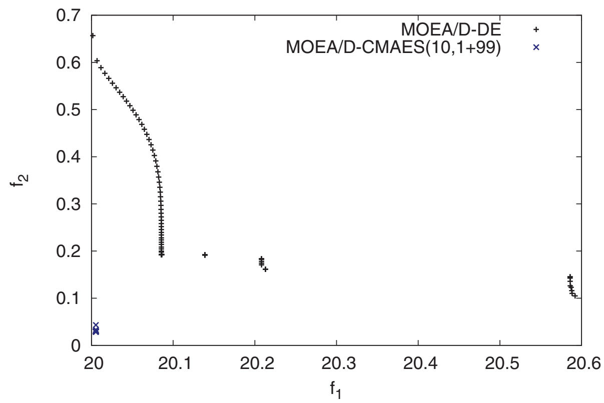
(a) CMOP1



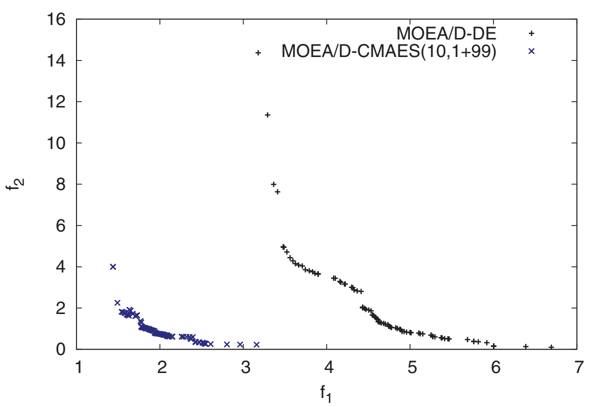
(b) CMOP2



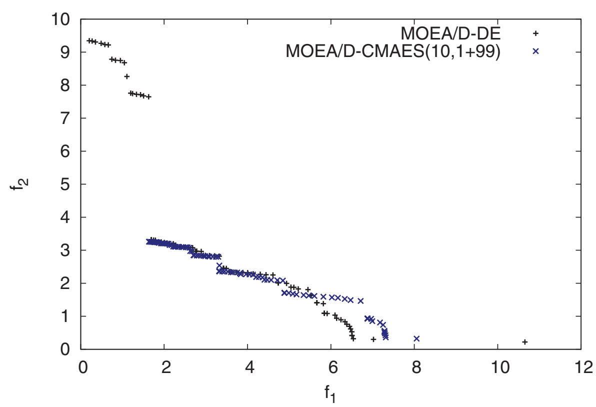
(c) CMOP3



(d) CMOP4

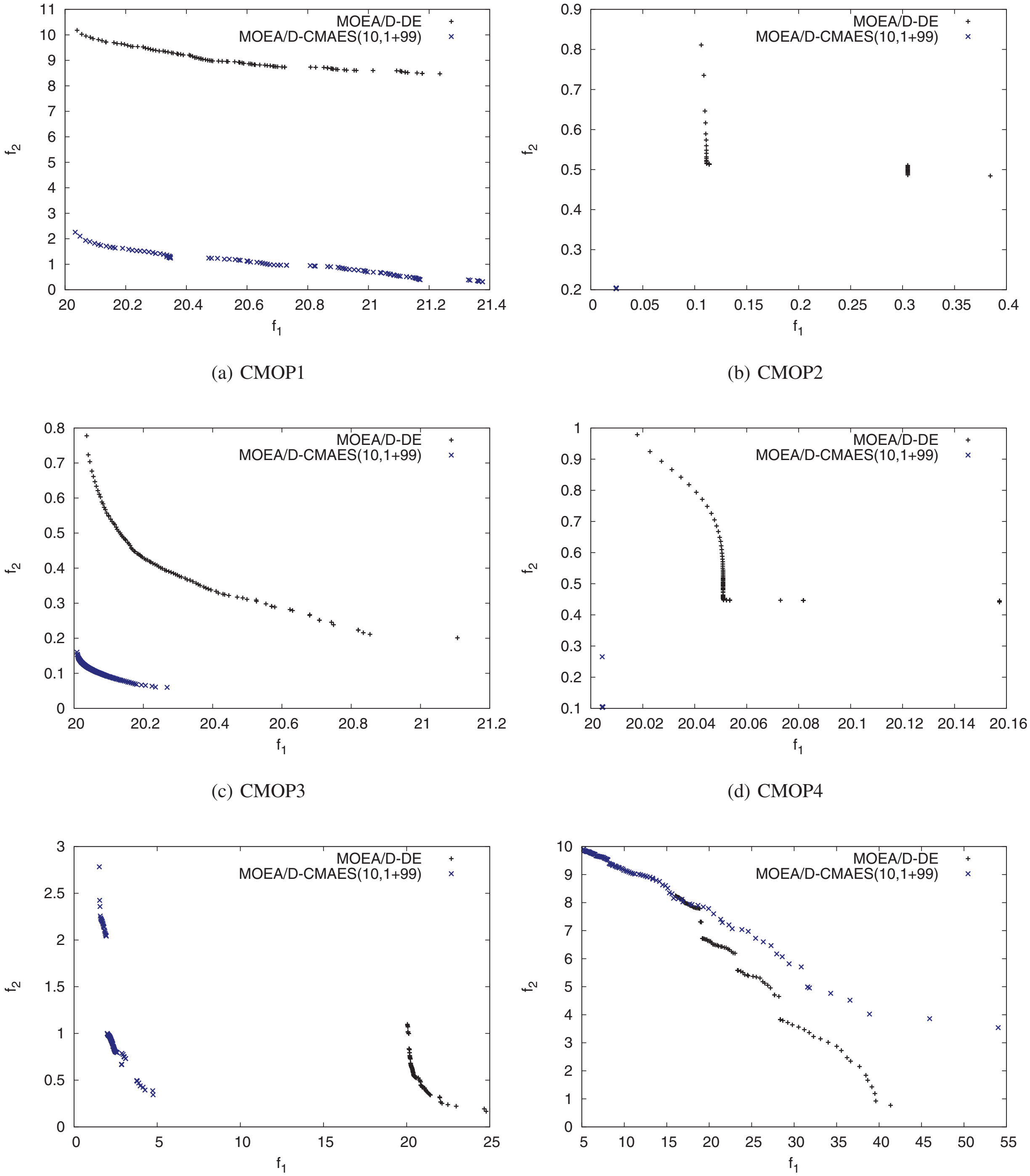


(e) CMOP5



(f) CMOP6

Figure 3: Fronts obtained from MOEA/D-DE and MOEA/D-CMAES on the six MOPs with dimension $n = 10$



(a) CMOP1 (b) CMOP2

(c) CMOP3 (d) CMOP4

(e) CMOP5 (f) CMOP6

Figure 4: Fronts obtained from MOEA/D-DE and MOEA/D-CMAES on the six MOPs with dimension $n = 30$