Enhancing Covariance Matrix Adaptation Evolution Strategy through Fitness Inheritance

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: {lrt101p, ckting}@cs.ccu.edu.tw

***Abstract*—Evolution strategy (ES) has shown to be effective in many search and optimization problems. In particular, the ES with covariance matrix adaptation (CMAES) achieves great successes and is viewed as a state-of-the-art evolutionary algorithm for complex numerical optimization. The CMAES models the population by a multivariate normal distribution, which requires a considerable amount of fitness evaluation results and thus degrades its efficiency. This paper proposes using fitness inheritance to reduce the computational cost at fitness evaluation. More specifically, the proposed FI-CMAES adopts fitness inheritance to approximate the fitness of offspring. The survivors are selected according to the approximated fitness; thereafter, the survival offspring are evaluated by the original fitness function. By this way, several original fitness evaluations on offspring can be saved. Experiments examine the effectiveness and efficiency of FI-CMAES on the CEC2014 test suite. The results show that FI-CMAES can outperform CMAES in terms of solution quality and convergence speed.**

***Index Terms*—Evolution strategy, CMAES, fitness inheritance, fitness approximation.**

I. Introduction

Evolutionary algorithms (EAs) have shown their great capability in solving difficult optimization problems. Inspired from Darwinian evolution theory [1], EAs mimic the process and operations of nature evolution in the search. In EAs, a candidate solution is treated as an individual or a chromosome. Beyond single-point search, EAs use a population of individuals to search for the optimal solutions. There have been several dialects of EAs, such as evolution strategy, genetic algorithm, and genetic programming, proposed for different search and optimization problems [2–5].

Evolution strategy (ES) is known for its effectiveness in numerical optimization [4, 6]. In particular, the ES with covariance matrix adaptation (CMAES) has achieved many successes in dealing with complex numerical optimization problems [7, 8]. A salient feature of CMAES is the use of multivariate normal distribution to model the population. New candidate solutions, namely offspring, are generated by sampling from the multivariate normal distribution, instead of the crossover operator that is commonly used in EAs. Hence, the distribution plays an important role in CMAES; however, its construction requires a considerable amount of fitness evaluation results, which limits the capability of CMAES. In addition, CMAES generates a larger number λ of offspring than the population size μ, and yet $(\lambda - \mu)$ of the offspring are knocked out by survival selection and thus wasted on the update of population model. For example, CMAES(5,10) produces ten offspring from the distribution, where only five offspring will survive and be used to update the distribution.

This paper proposes the FI-CMAES by introducing fitness inheritance to CMAES to enhance its efficiency. Fitness inheritance is an efficient fitness approximation approach. It estimates the fitness value of an offspring based on its parent's fitness values. Using fitness inheritance in place of the original fitness function to evaluate offspring can save the computational cost at fitness evaluation. Nonetheless, to avoid the cascade of approximation errors, the survivors from the selection will be evaluated with the original fitness function. In this way, the FI-CMAES saves $(\lambda - \mu)$ number of real fitness evaluations and enhances the search efficiency of CMAES. The advantages of FI-CMAES are examined in the experiments in Section IV.

The main contributions of this study are summarized as follows.

- We propose the FI-CMAES by integrating fitness inheritance into CMAES for reduction of computational cost at fitness evaluation.
- We present the multi-parent proportional fitness inheritance, which generalizes the number of inherited parents from 2 to an arbitrary number μ.
- We conduct experiments on the CEC2014 test suite and validate the enhancement of FI-CMAES over CMAES in terms of solution quality and convergence speed.

The rest of this paper is organized as follows. Section II reviews the related work on fitness inheritance methods. Section III describes the proposed FI-CMAES in detail. Section IV presents and discusses the experimental results on the effectiveness and efficiency of the proposed FI-CMAES. Section V concludes the present study.

II. Fitness Inheritance

The concept of fitness inheritance was first introduced in [9]. The fitness inheritance techniques assume the fitness of parents and offspring are correlated. Therefore, each offspring can inherit fitness from parents. There are two common fitness inheritance strategies, i.e., the average fitness inheritance:

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

Algorithm 1 CMAES

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

$$f_{\mathcal{I}}(\hat{x}) = \frac{f(x_1) + f(x_2)}{2},$$

and the proportional fitness inheritance:

$$f_{\mathcal{I}}(\hat{x}) = p(\hat{x}, x_1) f(x_1) + p(\hat{x}, x_2) f(x_2),$$

where $p(\hat{x}, x_i)$ accounts for the similarity between offspring $\hat{x}$ and its i-th parent. Since the proportional fitness inheritance considers the similarity, it is more realistic than average fitness inheritance. Euclidean distance is a common similarity measure for real-valued strings. The similarity to the first parent is calculated by

$$p(\hat{x}, x_1) = \frac{d(\hat{x}, x_2)}{d(\hat{x}, x_1) + d(\hat{x}, x_2)},$$

where $d(\cdot)$ represents Euclidean distance. Note that the similarity is in inverse proportion to Euclidean distance and the sum of similarity is one.

The fitness inheritance techniques have been investigated with different purposes and goals. As above mentioned, average fitness inheritance and proportional fitness inheritance are most widely used. Some studies exploit the scheme for inheriting fitness [10–13], where a chromosome is divided into several parts called "schemes" and its fitness value is estimated by summing the approximate values of all schemes.

Fitness inheritance has been applied to many EAs, including GA [9], PSO [14], and EDA [15]. Previous studies also show that fitness inheritance can deal with a variety of problem types, such as binary strings [11, 12], integer strings [16, 17], real-valued strings [18], multi-objective optimization problem [19], and real-world applications [9, 14, 16, 17]. These successes suggest that fitness inheritance is promising for enhancement of CMAES.

III. CMAES Using Fitness Inheritance

This study presents the FI-CMAES, a CMAES using fitness inheritance. Specifically, the FI-CMAES adopts proportional fitness inheritance based on Euclidean distance to reduce the number of fitness evaluations needed in CMAES. The following subsections describe CMAES and FI-CMAES in detail.

Algorithm 2 FI-CMAES

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

A. CMAES

CMAES models the population by a multivariate normal distribution:

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

where $\boldsymbol{m}$ is the mean vector and $\boldsymbol{C}$ is the covariance matrix. By updating $\boldsymbol{m}$ and $\boldsymbol{C}$, CMAES can obtain the updated multivariate normal distribution. The mean vector $\boldsymbol{m}$ is updated with the weighted center of population $\mathbf{X} = \{x_1, \ldots, x_\mu\}$, where the weights are determined by the rank of fitness. The covariance matrix $\boldsymbol{C}$ is constructed by eigendecomposition of population $\mathbf{X}$. The details about update of the mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{C}$ refer to [8]. The offspring $\hat{\mathbf{X}} = \{\hat{x}_1, \ldots, \hat{x}_\lambda\}$ are sampled from the multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$. The ratio of population and offspring sizes ($\mu : \lambda$) in CMAES is commonly set to $1 : 2$, whereas that in ES is usually set to 1 : 7.

Algorithm 1 shows the procedure of CMAES. First, CMAES initializes a multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$. Then CMAES samples and evaluates λ offspring from $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ in each generation. The best μ of them are selected to survive and update the multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$. This process continues until the termination criterion is met.

In CMAES, only the best μ individuals are used to update $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$, while the remaining $(\lambda - \mu)$ are discarded. That is, only μ fitness evaluations are in effect. The proportion of fitness evaluations unused in updating $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ increases with the ratio λ/μ. This study aims to address this issue by fitness inheritance. More details about the proposed FI-CMAES are given below.

B. FI-CMAES

FI-CMAES holds two major features: pre-selection by fitness inheritance and multi-parent proportional fitness inheritance. The former selects the best μ out of λ offspring according to the evaluation results from fitness inheritance instead of the original fitness function. As aforementioned, CMAES generates a larger amount λ of offspring than the population size μ. The difference between these two sizes causes a certain number of fitness evaluations unused in the update of population model $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$. The pre-selection by fitness inheritance helps to reduce the number of fitness evaluations from λ to μ in each generation. For the common setting of $\mu : \lambda = 1 : 2$, FI-CMAES can save half of the fitness evaluations in each generation. As the ratio λ/μ increases, FI-CMAES can save even more computational cost at fitness evaluation.

The second feature in FI-CMAES is the multi-parent proportional fitness inheritance. Considering that CMAES uses multiple parents, we propose the multi-parent proportional fitness inheritance by extending the number of inherited parents from 2 to an arbitrary number μ. The multi-parent proportional fitness inheritance approximates the fitness value by

$$f_{\mathcal{I}}(\hat{x}) = \sum_{x_{i \in \{1,\ldots,m\}}} p\left(\hat{x}, x_i\right) f\left(x_i\right),$$

where $x_1, \ldots, x_m$ denote the parents of $\hat{x}$ and $p\left(\hat{x}, x_i\right)$ accounts for the similarity between offspring $\hat{x}$ and its i-th parent x_i. This study adopts Euclidean distance as the similarity measure. The similarity measure is defined by

$$p\left(\hat{x}, x_i\right) = \frac{\sum_{x_{j \in \{1,\ldots,m | j \neq i\}}} d\left(\hat{x}, x_j\right)}{(m-1) \sum_{x_{j \in \{1,\ldots,m\}}} d\left(\hat{x}, x_j\right)}.$$

Like the conventional 2-parent fitness inheritance, the similarity $p\left(\hat{x}, x_i\right)$ for multiple parents is in inverse proportion to the Euclidean distance and the sum of similarity equals to one.

Algorithm 2 presents the procedure of FI-CMAES. At initialization, FI-CMAES generates μ individuals from the multivariate normal distribution $\mathcal{N}\left(\boldsymbol{m}, \boldsymbol{C}\right)$ and evaluates them using the original fitness function. These μ individuals provide real fitness evaluates for the first-generation offspring to inherit from. In each generation, λ offspring are sampled from $\mathcal{N}\left(\boldsymbol{m}, \boldsymbol{C}\right)$. According to the fitness values approximated by fitness inheritance, the best μ of λ offspring are selected to survive and further evaluated by the original fitness function. The μ survivors, which have the real fitness values, are used to update the mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{C}$. Noteworthily, there are only μ offspring evaluated by the original fitness function; that is, FI-CMAES saves $(\lambda - \mu)$ real fitness evaluations in each generation.

IV. Experimental Results

This paper conducts a series of experiments to examine the effectiveness and efficiency of the proposed FI-CMAES, in comparison with the original CMAES[1]. Table I lists the parameter setting for CMAES and FI-CMAES in the experiments. Two offspring sizes (10 and 35) are experimented to investigate the effects of $\mu : \lambda$ ratio. The test suite comprises the well-known 30 benchmark functions of CEC2014 [20]. The benchmark functions are classified into four types: unimodal functions (F_1–F_3), multimodal functions (F_4–F_{16}), hybrid functions (F_{17}–F_{22}), and composite functions (F_{23}–F_{30}). All experimental results are obtained from 30 trials of test algorithms.

Table I: Parameter setting

Parameter	Value
Problem size (n)	30
#Evaluations	$10^3 n$
#Trials	30
Population size (μ)	5
Offspring size (λ)	10 and 35

A. Solution Quality

We first compare the solution quality of CMAES and FI-CMAES. This study considers the solution quality at two search phases, i.e., short term (500 fitness evaluations) and long term (30000 fitness evaluations), to investigate the effect of fitness inheritance on different search stages of CMAES.

Regarding the short-term performance, Table II lists the mean best fitness (MBF) obtained from CMAES and FI-CMAES at 500 fitness evaluations. The table also presents the results of Student's t-test. For $\lambda = 10$, FI-CMAES(5,10) achieves significantly better MBF than CMAES(5,10) does on 17 functions and comparably on 10 functions. Considering the function types, FI-CMAES(5,10) betters CMAES(5,10) on unimodal, multimodal, and hybrid functions; but worse on composite functions. For $\lambda = 35$, the improvement of FI-CMAES on CMAES becomes more apparent: FI-CMAES(5,35) significantly outperforms CMAES(5,35) on almost all functions (26 out 30 functions), including unimodal, multimodal, hybrid, and composite functions. In addition, FI-CMAES(5,10) gains higher MBF than FI-CMAES(5,35) does on composite functions, but slightly worse MBF on hybrid functions. The two methods FI-CMAES(5,10) and FI-CMAES(5,35) perform similarly on unimodal and multimodal functions.

As for long-term performance, Table III presents the MBF of CMAES and FI-CMAES at 30000 fitness evaluations. For $\lambda = 10$, there is no significant difference between CMAES(5,10) and FI-CMAES(5,10) on 28 test functions. For $\lambda = 35$, FI-CMAES(5,35) performs significantly better than CMAES(5,35) on 11 functions and comparable on 13 functions. Furthermore, FI-CMAES(5,35) performs better than CMAES(5,35) does on unimodal, hybrid, and composite functions; their performances are comparable on multimodal functions. Comparing FI-CMAES(5,10) and FI-CMAES(5,35), the former performs better on unimodal and multimodal functions, whereas the latter excels on hybrid and composite functions.

B. Convergence Speed

In the experiments, we further compare the convergence speed of CMAES and FI-CMAES to examine their efficiency. Figures 1–4 display the anytime behavior of CMAES and FI-CMAES in terms of MBF and standard error against the number of fitness evaluations. The results show that FI-CMAES(5,10) converges fastest on all the three unimodal

[1]The source code of CMAES is obtained from https://www.lri.fr/~hansen/cmaes_inmatlab.html#C.

Table II: Mean best fitness (MBF) obtained from CMAES and FI-CMAES at 500 fitness evaluations. Bold symbol implies the best performance among all approaches. The comparison results (Win, Lose, and Equal) are determined under $\alpha = 0.01$ confidence level.

	MBF (at 500 evaluations)				p-value		
Function	CMAES(5,10)	FI-CMAES(5,10)	CMAES(5,35)	FI-CMAES(5,35)	(1) vs. (2)	(3) vs. (4)	(2) vs. (4)
F_1	736890600.00	**313140566.67**	2187699666.67	390245700.00	1.36E-16	6.00E-65	-3.12E-04
F_2	31718580000.00	8023286666.67	90907806666.67	**7184836333.33**	1.43E-16	1.04E-79	7.46E-02
F_3	85630.99	85415.14	86404.36	**84620.12**	3.39E-01	1.65E-03	1.34E-01
F_4	4899.51	1339.62	20863.67	**1244.60**	7.96E-13	3.94E-55	8.19E-02
F_5	**520.96**	521.19	521.17	521.21	-7.90E-05	-1.40E-02	-2.04E-01
F_6	647.20	645.23	648.63	**640.29**	1.26E-05	2.44E-16	2.50E-10
F_7	1045.89	782.16	1651.08	**760.45**	2.30E-16	1.53E-79	1.29E-04
F_8	1011.85	**992.31**	1155.61	994.18	1.32E-08	1.93E-41	-3.26E-01
F_9	1124.27	**1110.29**	1206.17	1118.74	8.23E-07	2.18E-29	-1.31E-02
F_{10}	5975.35	**5442.20**	8120.79	5621.53	2.84E-11	4.39E-26	-6.32E-02
F_{11}	6578.20	**6084.67**	9234.65	6830.82	1.23E-09	2.20E-20	-3.81E-06
F_{12}	1201.73	**1201.63**	1202.39	1202.08	1.66E-01	9.55E-03	-4.85E-04
F_{13}	1305.36	1302.00	1310.11	**1301.48**	9.91E-24	7.00E-45	1.22E-03
F_{14}	1545.07	1429.12	1768.69	**1422.92**	3.08E-20	1.48E-79	3.11E-03
F_{15}	14691.25	**2395.74**	471781.53	2821.68	5.19E-11	2.10E-30	-4.61E-02
F_{16}	**1613.89**	1613.92	1613.93	1613.98	-1.68E-02	-1.08E-03	-1.96E-03
F_{17}	86414176.67	35169216.67	553612133.33	**31921444.00**	8.47E-11	8.14E-52	2.14E-01
F_{18}	688909166.67	**87795636.67**	12427333333.33	191754143.33	1.15E-06	2.47E-51	-3.83E-09
F_{19}	2192.57	2031.96	2553.63	**2023.55**	6.30E-24	2.40E-57	1.66E-01
F_{20}	333009.87	**277540.34**	1414890266.67	351807.10	5.76E-02	5.19E-29	-3.20E-02
F_{21}	33367611.00	20631733.67	1618432000.00	**17053560.67**	1.24E-05	5.39E-39	6.70E-02
F_{22}	6739.81	3578.17	2497294.67	**3570.76**	4.65E-04	1.35E-29	4.52E-01
F_{23}	**2506.88**	2507.25	2515.37	2508.33	-1.77E-01	4.72E-24	-4.34E-03
F_{24}	2602.23	2602.74	**2602.04**	2602.53	-1.90E-07	-2.70E-10	6.46E-03
F_{25}	**2700.11**	2700.11	2700.25	2700.15	-1.51E-01	7.41E-16	-8.61E-04
F_{26}	**2800.00**	2800.01	2800.00	2800.01	-2.02E-07	-3.06E-09	-2.48E-01
F_{27}	**2907.02**	2908.18	2921.66	2911.42	-9.86E-02	4.71E-11	-7.06E-03
F_{28}	**3009.35**	3009.55	3026.33	3015.06	-4.16E-01	4.96E-10	-1.73E-04
F_{29}	**8082322.00**	8873960.67	18180880.00	9799483.67	-6.75E-02	5.59E-20	-8.32E-02
F_{30}	589862.20	**567303.17**	1276621.00	665174.93	2.61E-01	3.53E-17	-2.77E-02
Win					17	26	5
Lose					3	3	9
Equal					10	1	16

functions. As for multimodal functions, FI-CMAES(5,35) achieves the fastest convergence on 6 functions (F_4, F_6, F_7, F_8, F_{10} and F_{16}), while FI-CMAES(5,10) does on F_{11} and F_{12}. Moreover, FI-CMAES(5,10) converges fastest on 5 out of 6 hybrid functions except F_{22}. It also gains the fastest convergence on 6 out of 8 composite functions except F_{24} and F_{26}. In general, FI-CMAES(5,10) converges fastest on unimodal, hybrid and composite functions; FI-CMAES(5,35) converges fastest on multimodal functions.

The above results show the effectiveness and efficiency of FI-CMAES. It can significantly improve CMAES in solution quality, especially under a limited number of fitness evaluations. In addition, FI-CMAES converges faster than CMAES on the four types of test functions. These satisfactory outcomes validate the advantages of FI-CMAES in terms of solution quality and convergence speed.

V. Conclusions

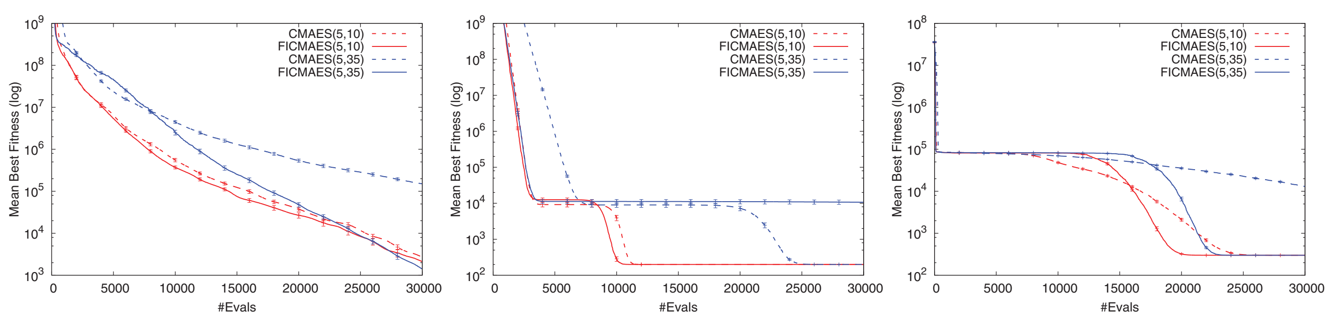
CMAES has achieved great successes and is viewed as a state-of-the-art evolutionary algorithm for many complex numerical optimization problems. It generates a larger amount λ of offspring than the population size μ. The disparity between μ and λ causes some fitness evaluations wasted on the update of population model. This study proposes integrating fitness inheritance into CMAES as the FI-CMAES to address this issue. The FI-CMAES utilizes fitness inheritance to approximate the fitness of λ offspring. Using the approximated fitness, the best μ offspring are selected to survive, and then their fitness values are updated using the original fitness function. Therefore, the needed number of original fitness evaluations in each generation is reduced from λ to μ, which substantially saves the computational cost at fitness evaluation.

Experiments are carried out on 30 CEC2014 benchmark functions. The results show that FI-CMAES achieves significant improvement on CMAES in solution quality, especially under a limited number of fitness evaluations. In addition, FI-CMAES gains faster convergence than CMAES. These preferable outcomes validate the advantages of FI-CMAES in terms of solution quality and convergence speed.

Some extensions can be considered in the future. The ratio $\mu : \lambda$ influences the performance of CMAES and FI-CMAES. Finding a proper setting is a promising work for improving FI-

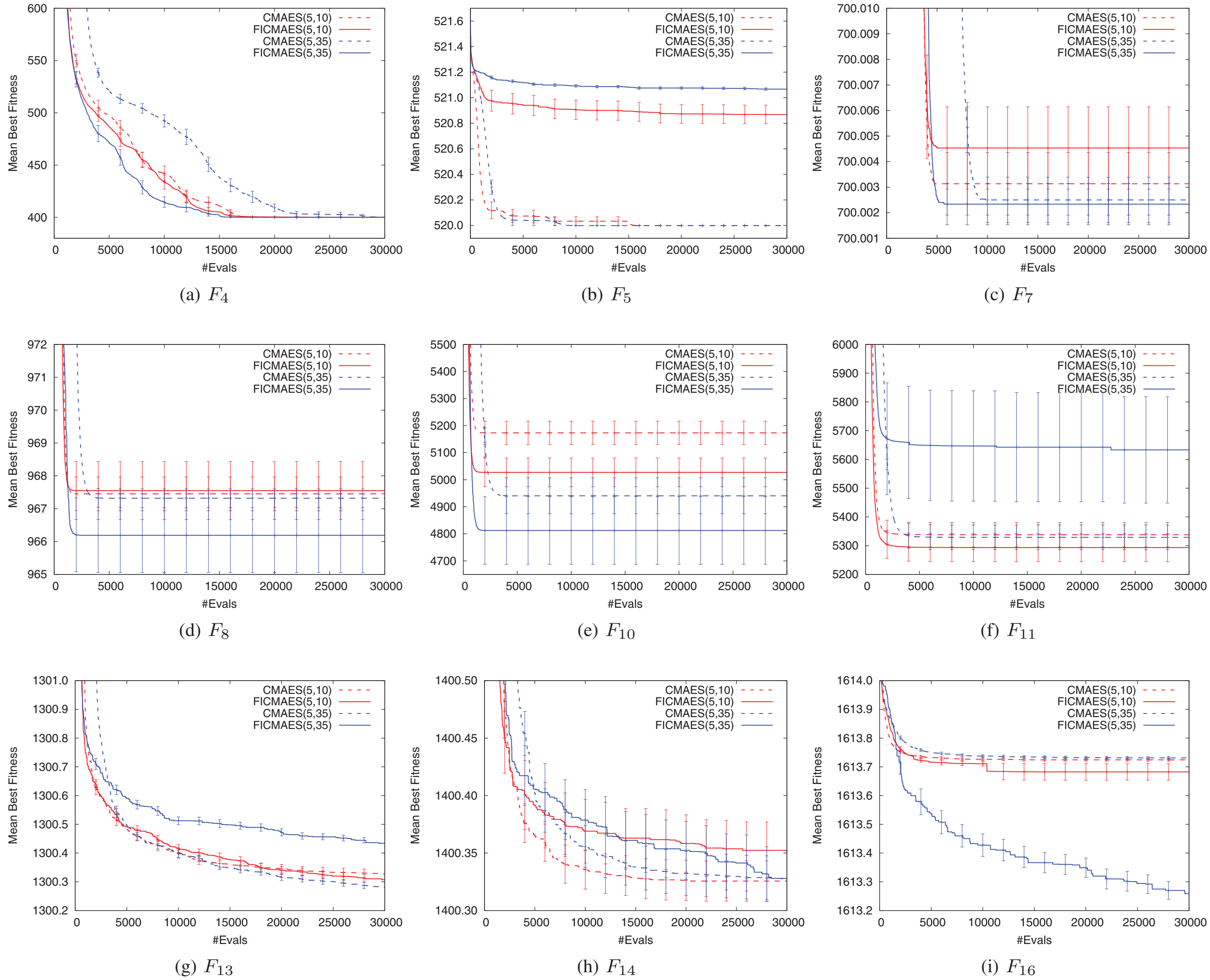
Table III: Mean best fitness (MBF) obtained from CMAES and FI-CMAES at 30000 fitness evaluations. Bold symbol implies the best performance among all approaches. The comparison results (Win, Lose, and Equal) are determined under $\alpha = 0.01$ confidence level.

	MBF (at 30000 evaluations)				p-value		
Function	CMAES(5,10)	FI-CMAES(5,10)	CMAES(5,35)	FI-CMAES(5,35)	(1) vs. (2)	(3) vs. (4)	(2) vs. (4)
F_1	2678.44	2114.45	150633.35	**1426.40**	2.13E-01	4.42E-14	1.34E-01
F_2	**200.00**	**200.00**	**200.00**	10695.99	-	-2.70E-08	-2.70E-08
F_3	300.00	**300.00**	13003.15	**300.00**	8.04E-02	1.31E-20	-
F_4	**400.00**	400.13	400.12	**400.00**	-1.63E-01	5.13E-03	1.63E-01
F_5	**520.00**	520.87	**520.00**	521.07	-4.53E-13	-9.45E-41	-5.33E-03
F_6	641.13	638.05	631.65	**611.97**	1.37E-02	7.68E-17	1.09E-24
F_7	700.00	700.00	700.00	**700.00**	-2.46E-01	4.45E-01	1.14E-01
F_8	967.45	967.55	967.32	**966.19**	-4.61E-01	1.71E-01	1.74E-01
F_9	1086.22	1087.55	**1085.03**	1087.15	-5.98E-02	-1.34E-02	3.58E-01
F_{10}	5173.10	5027.24	4940.40	**4812.29**	1.88E-02	1.85E-01	6.09E-02
F_{11}	5338.05	**5293.34**	5328.81	5633.08	2.47E-01	-5.95E-02	-4.25E-02
F_{12}	1200.26	**1200.18**	1200.19	1201.88	1.31E-02	-3.06E-15	-3.13E-15
F_{13}	1300.33	1300.31	**1300.28**	1300.43	1.45E-01	-9.51E-15	-5.88E-11
F_{14}	**1400.33**	1400.35	1400.33	1400.33	-1.98E-01	-4.91E-01	2.47E-01
F_{15}	1504.28	1505.32	**1503.17**	1518.32	-2.39E-02	-1.59E-49	-2.44E-27
F_{16}	1613.72	1613.68	1613.73	**1613.26**	8.00E-02	2.32E-16	6.78E-15
F_{17}	3370.03	3286.39	10570.10	**3143.98**	2.25E-01	1.04E-09	1.14E-01
F_{18}	**2006.61**	2029.30	3603.59	2930.80	-9.34E-02	6.90E-02	-2.54E-03
F_{19}	1915.12	1911.17	1910.95	**1908.42**	8.54E-02	6.42E-07	2.96E-08
F_{20}	2235.54	2265.17	10122.48	**2169.53**	-9.93E-02	5.46E-10	3.32E-06
F_{21}	3363.84	3373.38	14363.78	**3349.83**	-4.56E-01	9.03E-12	4.19E-01
F_{22}	2713.81	2756.78	**2485.28**	2754.76	-2.29E-01	-1.82E-12	4.81E-01
F_{23}	**2500.00**	**2500.00**	**2500.00**	**2500.00**	-	-	-
F_{24}	2600.09	2600.01	2600.12	**2600.01**	5.00E-05	2.54E-12	4.18E-01
F_{25}	**2700.00**	**2700.00**	**2700.00**	**2700.00**	-	-	-
F_{26}	2800.00	2800.00	2800.00	**2763.55**	-	1.53E-04	1.53E-04
F_{27}	**2900.00**	**2900.00**	**2900.00**	**2900.00**	-	-	-
F_{28}	**3000.00**	**3000.00**	**3000.00**	**3000.00**	-	-	-
F_{29}	**3100.00**	**3100.00**	**3100.00**	**3100.00**	-	-	-
F_{30}	**3200.00**	**3200.00**	**3200.00**	**3200.00**	-	-	-
Win					1	11	5
Lose					1	6	6
Equal					28	13	19



(a) F_1 (b) F_2 (c) F_3

Figure 1: Convergence on unimodal functions



(a) F_4 (b) F_5 (c) F_7

(d) F_8 (e) F_{10} (f) F_{11}

(g) F_{13} (h) F_{14} (i) F_{16}

Figure 2: Convergence on multimodal functions

CMAES. In addition, the analysis on its impact to FI-CMAES performance is another important task.

Acknowledgment

This work was supported by the Ministry of Science and Technology of Taiwan, under contract MOST 104-2221-E-194-015-MY3, and the Ministry of Education of Taiwan.

References

[1] C. Darwin, *The Origin of Species*. John Murray, London, 1859.

[2] J. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[4] H.-P. Schwefel, *Evolution and Optimum Seeking*. John Wiley & Sons, Inc, 1993.

[5] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Natural Computing. Springer-Verlag, 2003.

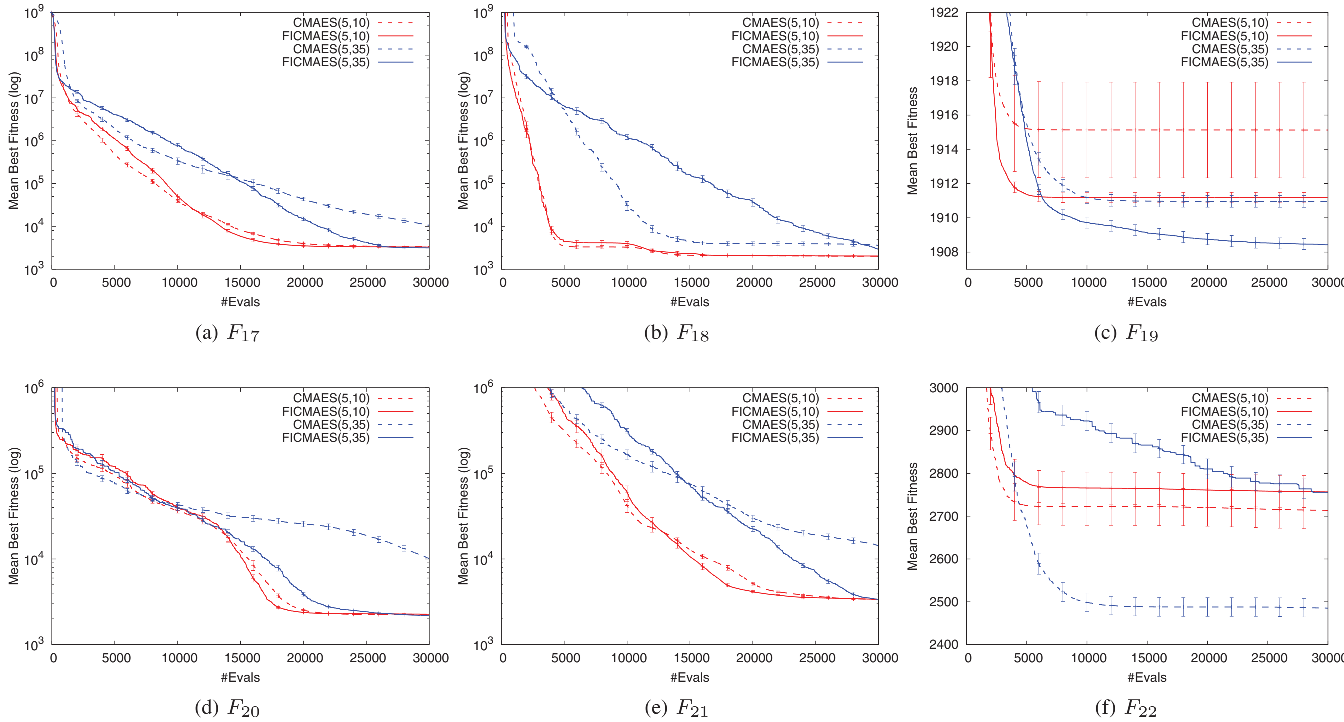
[6] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–A comprehensive introduction," *Natural computing*, vol. 1, no. 1, pp. 3–52, 2002.

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

[8] 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.

[9] R. E. Smith, B. A. Dike, and S. A. Stegmann, "Fitness inheritance in genetic algorithms," in *Proceedings of the ACM symposium on Applied computing*, 1995, pp. 345–350.

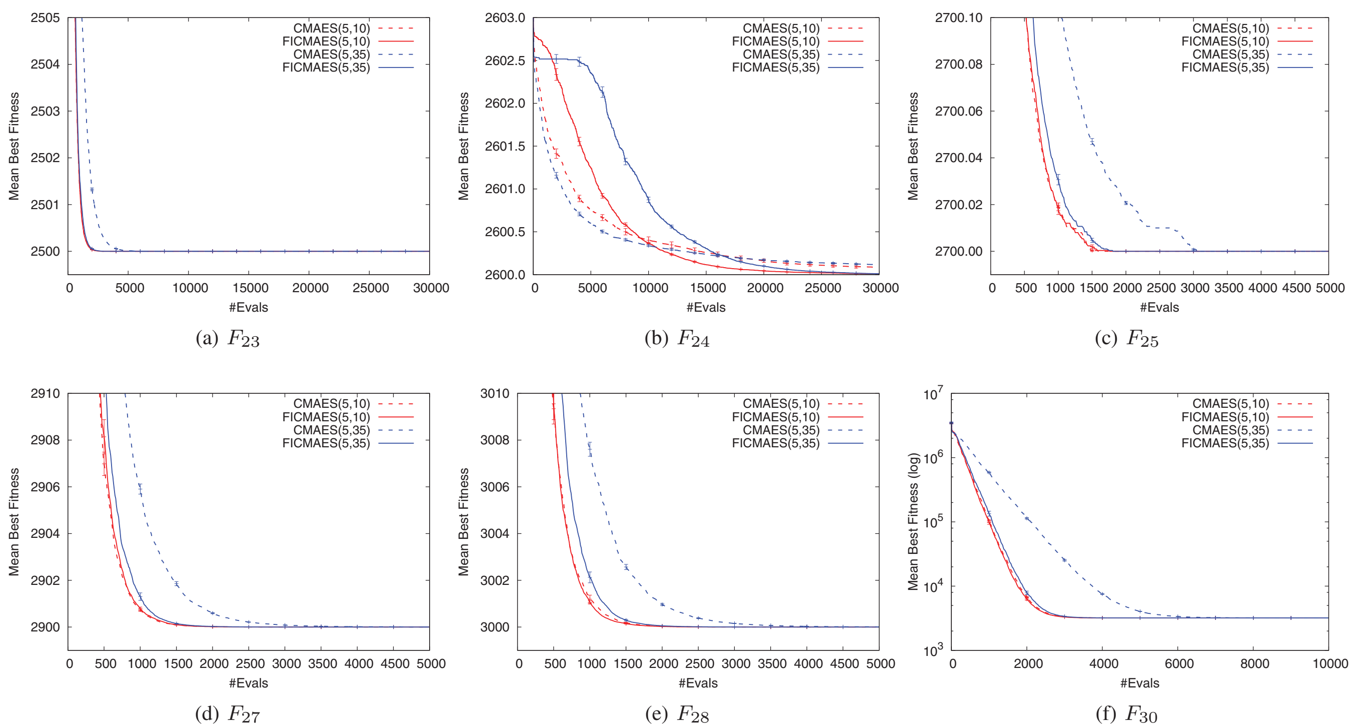
[10] K. Sastry, D. E. Goldberg, and M. Pelikan, "Don't



(a) F_{17} (b) F_{18} (c) F_{19}

(d) F_{20} (e) F_{21} (f) F_{22}

Figure 3: Convergence on hybrid functions



(a) F_{23} (b) F_{24} (c) F_{25}

(d) F_{27} (e) F_{28} (f) F_{30}

Figure 4: Convergence on composite functions

evaluate, inherit," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 551–558.

[11] K. Sastry, M. Pelikan, and D. E. Goldberg, "Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation," in *Proceedins of IEEE Congress on Evolutionary Computation*, vol. 1, 2004, pp. 720–727.

[12] K. Sastry, D. E. Goldberg, and M. Pelikan, "Efficiency enhancement of probabilistic model building genetic algorithms," IlliGAL, Report 200420, February 2004.

[13] K. Sastry, C. F. Lima, and D. E. Goldberg, "Evaluation relaxation using substructural information and linear estimation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 419–426.

[14] M. Ebrahimi, M. R. Farmani, and J. Roshanian, "Multidisciplinary design of a small satellite launch vehicle using particle swarm optimization," *Structural and Multidisciplinary Optimization*, vol. 44, no. 6, pp. 773–784, 2011.

[15] M. Pelikan and K. Sastry, "Fitness inheritance in the bayesian optimization algorithm," in *Proceedings of Genetic and Evolutionary Computation Conference*, 2004, pp. 48–59.

[16] C. Pilato, A. Tumeo, G. Palermo, F. Ferrandi, P. L. Lanzi, and D. Sciuto, "Improving evolutionary exploration to area-time optimization of fpga designs," *Journal of Systems Architecture*, vol. 54, no. 11, pp. 1046–1057, 2008.

[17] C. Pilato, D. Loiacono, A. Tumeo, F. Ferrandi, P. L. Lanzi, and D. Sciuto, "Speeding-up expensive evaluations in high-level synthesis using solution modeling and fitness inheritance." in *Proceedings of Computational Intelligence in Expensive Optimization Problems*, 2010, pp. 701–723.

[18] L. G. Fonseca, A. C. Lemonge, and H. J. Barbosa, "A study on fitness inheritance for enhanced efficiency in real-coded genetic algorithms," in *Proceedings of IEEE Congress on Evolutionary Computatio*, 2012, pp. 1–8.

[19] M. Reyes-Sierra and C. A. C. Coello, "Dynamic fitness inheritance proportion for multi-objective particle swarm optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 89–90.

[20] 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.