Two-stage based Ensemble Optimization for Large-Scale Global Optimization

Yu Wang *Student Member, IEEE*, Bin Li *Member, IEEE*,

***Abstract*— Large-scale global optimization (LSGO) is a very important and challenging task in optimization domain, which is embedded in many scientific and engineering applications. In this paper, a two-stage based ensemble optimization evolutionary algorithm (EOEA) is designed to handle LSGO problems. The performance of EOEA is evaluated on the test functions provided by the LSGO competition of IEEE Congress of Evolutionary Computation (CEC 2010). Compared with some previous LSGO algorithms, EOEA demonstrates better performance.**

I. INTRODUCTION

CONSIDERED as a revolutionary and difficult task, LSGO has attracted increasing research attention in recent years [1][2][3]. Generally speaking, LSGO problems widely exist in engineering applications, such as designing large scale electronic systems; scheduling problems with large dimensional resources; vehicle routing in large scale traffic networks and etc. A function optimization competition on LSGO was held in CEC 2008 [1] to call for more attention. Since the competition of applying LSGO approaches to engineering application has become much more furious, the development of the optimization methods always lead to rapid improvement of the competitive strength.

Historically, a large number of algorithms have been proposed to handle the LSGO problems [2]-[14]. The current EA-based LSGO research mainly focuses on the following two highly concerned directions:

- Developing more effective operators: the successful implementations consist of self-adapting strategies for parameter setting, modification of the classical EA operators, etc. The reason of making these modifications is that the classical operators are usually developed for low-dimensional tasks, but most of them will lose their efficiency on high-dimensional tasks [1]. Recently, this field has attracted increasing attention and the typical approaches include population reduction for differential evolution (DE) [9], dynamic multi-swarm PSO [12] and estimation of distribution algorithm (EDA) with mixed sampling operator [15]. With these approaches, the LSGO problems are optimized as an whole, that is, no divide-and-conquer methods was used. Actually, the implementation of specific operators is attributed to strengthen the algorithm's capability for higher dimensional tasks.
- Since the publication of A. M. Potter and K. A. De Jong's seminal work [4], the cooperative co-evolution (CC) quickly became an important approach for solving LSGO problems. The main idea of CC is to partition the LSGO problem into a number of sub-problems and then 'evolve' them respectively in multiple cycles. Via this divide-and-conquer method, the classical EAs are able to effectively solve many separable problems [4]. Recently, many CC-based EAs have derived many versions and shown excellent performance, such as cooperative co-evolution fast evolutionary programming (FEPCC) [5], differential evolution with CC: DECC-I, DECC-II [6], MLCC [7], DEwSaCC [14]. Particularly, such a CC incorporated approach seems necessary for extremely huge D problems.

From the experimental results obtained by [8]-[15], it is observed that the convergence speed of the EAs without CC is very high for the relatively easy problems, such as the Sphere and Ackley problems. However, their capability of handling the complex problems, such as the Rastrigin and Rochenbrock problems, is very limited. On the contrary, the EAs with CC show better performance on the complex problems, although their convergence speed is slower than those algorithms without CC. For LSGO, the conflict between exploration and exploitation is much more incisive due to the extremely high dimensionality.

In this paper, a two-stage based ensemble optimization evolutionary algorithm (EOEA) is designed to handle LSGO problems, in which the search procedure is divided into two stages: 1) the global shrinking stage and 2) the local exploration stage. The objective of the first stage is to shrink the searching scope to the promising area as quickly as possible, and the objective of the second stage is to explore the limited area extensively to find as better as possible solution.

To achieve the first objective, an EDA based-on mixed Gaussian and Cachy models (MUEDA) [18] is adopted in the first stage. MUEDA has been proven to have high convergence speed in [18]. Since the search spaces of LSGO problems are always very large due to the extremely high dimensionality, MUEDA is used to shrink the search region as quickly as possible to the promising area. It is expected to provide a good starting point for the second stage.

To achieve the second objective, CC-based algorithm is

Manuscript in January, 2010. The authors are with Natural Inspired Computation and Application Laboratory (NICAL), department of electronic science and technology, University of Science and Technology of China (USTC). Phone: +86-551-3601802.

Yu Wang is a PhD candidate. His research interests include evolutionary computation, multi-objective optimization, link biologic concept to evolutionary computation and application of evolutionary computation in power system domain.(wyustc@mail.ustc.edu.cn).

Bin Li is an associate professor. His research interests include evolutionary computation and pattern recognition. (binli@ustc.edu.cn).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

adopted in the second stage. Different from the previous CC-based algorithms, 1) the size of each group, which explores one subspace of solution space, can be adaptively tuned based on a feed back learning technique; 2) the optimizer for each group can be selected stochastically from 3 candidate algorithms according to their previous performance. The fundamental idea is similar to that of [27]; 3) after certain number of iterations, some variables that have more impact on the fitness values are selected to form a new group, and to be optimized.

It has been discussed fully in [7] and [24] that the suitable group size of CC is problem dependent. The large group size is good for unimodal problems, while small group size is better for multimodal problems. [24] proposed a learning based group size adaption strategy, which can adaptively set the group size to be 5, 10, 25, 50 and 100 based on their previous performance. In the second stage of EOEA, a more flexible group size adaption strategy is designed, which can self-adaptively set the group size to be any positive integer via a learning technique.

In recent years, the scheme of combining multiple algorithms in one algorithmic framework has attracted more and more attention in the field of evolutionary computation and heuristic optimization. A number of works [27][28][29][31] have shown that such scheme can effectively improve the algorithms' performance on optimization problems of various complex landscapes. Successful examples were observed on combinatorial optimization problems [28][29][30], global numerical optimization problems [25][26][31][32], and multi-objective optimization problems [33]. Several strategies [27][30][31] have been proposed to tune the contribution of various sub-algorithms adaptively according to their previous performance, and have shown pretty good effect. Considering the complexity of fitness landscapes of LSGO problems, a multi-algorithm scheme is adopted in the second stage of EOEA. A probabilistic method is adopted to select sub-optimizer for each group respectively. The execution probability of each sub-optimizer is learnt gradually with a learning rate according to the previous performance of the sub-optimizer. The fundamental idea of tuning the sub-algorithms' contribution proportionally to their previous performance is similar to that of [27].

It has been discussed in [15] that different variables of LSGO problems may have different effects on the fitness values, in other words, some of the variables may be more important than the others. while in most applications of classical evolutionary algorithms, all variables are treated evenly. In the second stage, we try to evaluate every variable respectively according to its previous performance, and then select a number of most important variables to form a new group to undergo further exploration.

The remainder of this paper is structured as follows: In section II, the EOEA is presented. Section III presents the experimental results and discusses the advantages of EOEA over several previous algorithms. Section VI provides experimental results under the format of CEC 2010 LSGO competition. In section V, a brief conclusion is given.

TABLE I
PROCEDURE OF EOEA

EOEA

Input:
- Optimization task (including the criterion of determining the fitness values and the dimensionality);
- a termination condition;

Output: The best solution found.

Step 0) Initialization: Randomly initialize the population X_0. Set $t = 0$. Conduct the initialized probabilistic model $P(0)$.

Step 1) The first optimization stage: MUEDA.

Step 2) Trigger: Determine whether to trigger the second stage or continue MUEDA. In case of former, go to step 3); otherwise, go back to step 2).

Step 3) The second optimization stage: CC based search stage.

Step 4) Terminate and output.

TABLE II
PROCEDURE OF MUEDA

MUEDA

Step 0) Initialization:
- **Step 0.0)** Set the weight vector W for step 2) (STDC) according to $W(i) = 0.55 - e^{lg(\frac{D}{10^5})}$ for $i = 1, 2, ..., D$.
- **Step 0.1)** Randomly initialize the population X_0. Conduct the probabilistic model P_0 based on X_0.
- **Step 0.2)** Set $t = 0$.

Step 1) Reproduction and update:
- **Step 1.0) Reproduction:** Sample the new candidates based on P_t under mixed distribution N_L.
- **Step 1.1)** Set $t = t + 1$.
- **Step 1.2) Selection:** Select top 20% individuals by truncation strategy.
- **Step 1.3) Update:** Update the model by the selected individuals under update coefficient 1 (the same as UMDAc [17]).

Step 2) Standard Deviation Control Strategy (STDC) [18].

Step 3) If termination condition is not met, goto Step 1); otherwise end MUEDA.

II. ENSEMBLE OPTIMIZATION EVOLUTIONARY ALGORITHM

In order to balance the conflict between exploration and exploitation, a serial two-stage optimization framework is used. In the previous research, many hybrid algorithms have been developed to achieve better balance between exploration and exploitation, such as [25][26] and [32]. Unlike the above three works that simultaneously adopt different new offspring creating strategies to balance exploration and exploitation in one iteration, EOEA divides the search procedure into two stages to fully extract the merits of different search techniques:

- 1) in the first stage, a search technique with high convergence speed is used to shrink the search region to a more promising area.
- 2) in the second stage, a CC based search technique is used to explore the limited area extensively to get better

TABLE III
PROCEDURE OF PROBABILITY BASED CC

Probability based CC
Initialization:
Get $NP/2$ individuals from the first stage and randomly initialize the positions of $NP/2$ individuals to form the population.
Set fitness evaluation size $nfeval = 0$, loop number $LS = 0$.
Evaluate the fitness values $F = (fit_1, fit_2, ..., fit_{ps})$ of X.
Set probabilities of using 3 sub-optimizers $proSO_i = 1/3$, $i = 1, 2, 3$.
Set the learning period $Gs = 2$, learning coefficient $\alpha = 0.15$.
Set accumulators of 3 sub-optimizers $SS_i = 0$, $i = 1, 2, 3$.
Set accumulators of performance of all variables $SV_i = 0$, $i = 1, ...D$.
Set mean group size $meansize = 30$, and standard deviation value $stdsize = 10$.
Optimization procedure:
while $nfeval < maxfeval$
Generate a set of group sizes $GS = \{GS_1, GS_2 ... GS_{gss}\}$ based on $N(meansize, stdsize)$, where gss is the total number of groups and $\sum GS_i = D$, where $N(meansize, stdsize)$ is a Gaussian distribution with mean value $meansize$ and standard deviation $stdsize$.
Select sub-optimizers $SO_1, SO_2 ... SO_{gss}$ from the 3 given sub-optimizers for gss groups by roulette wheel selection based on $proSO_i$, $i = 1, 2, 3$.
for $k = 1 : gss$.
Optimize the kth group by sub-optimizer SO_k.
Calculate the individual performance $P_k = \lvert v - v' \rvert / v$, where v and v' are the current and previous best values.
Add the the individual performance P_k to the accumulator of associated sub-optimizer SS.
Add the the individual performance P_k to the accumulator of associated variables SV.
end for.
Sort the P_k, $k = 1, 2 ... gss$ in descending order $P'_1, P'_2 ... P'_{gss}$.
Select $\lceil gss/2 \rceil$ group sizes with top performance to calculate the new $meansize$ and $stdsize$:
$meansize = \frac{1}{\lceil gss/2 \rceil} \sum_{i=1}^{\lceil gss/2 \rceil} P'_i$, $stdsize = std(P'_1 ... P'_{\lceil gss/2 \rceil})$
for $j = 1 : 3$ // update the execution probabilities
$proSO'_j = (1 - \alpha) proSO_j + \alpha SS_i / \sum SS_i$
$SS_j = 0$
end for.
Calculate the new execution probabilities $proSO_i = proSO'_i / \sum proSO'_i$, $i = 1, 2, 3$.
if $LS\%Gs == 0$ // optimize the variables that have larger effect on fitness value
Sort the SV.
Optimize the $\lceil meansize \rceil$ variables with largest SV values by a randomly selected sub-optimizer.
end for
end while
Output the found optimal solution.

solutions.

The framework of EOEA is shown in Table I. The details of MUEDA and CC based search stage will be introduced in the following sub-sections.

A. MUEDA

Generally speaking, EDAs employ probabilistic models to describe the promising area in the solution space and use these models to guide the generation of the candidate solutions for the next generation [15][17][18]. In order to reduce the complexity of learning the probabilistic model, a univariate EDA named MUEDA [18] is adopted in EOEA. It is commonly believed that the univariate EDAs are suitable for relatively simple problems where the decision variables are usually independent from each other [15][17]. To enhance the exploration ability of the univariate EDA, a Lévy model is combined with a Gaussian model to guide the generation of the candidate solutions in MUEDA. It is shown in [18] that the MUEDA performs well in both convergence speed and final accuracy on unimodal problems and many simple multimodal problems, but its performance on complex multi-modal problems is unsatisfying. Therefore, in EOEA, we just utilize the merit of the high convergence speed of MUEDA to conduct the first stage search, which usually lasts a short duration. The procedure of MUEDA is shown in Table II, and more details can be found in [15] and [18].

TABLE IV
COMPARISON OF FOUR LSGO ALGORITHMS.

Algorithms	Metric	fun1	fun2	fun3	fun4	fun5	fun6	fun7
MLCC	mean	8.53E-23	4.97E-01	2.01E-12	9.79E+12	3.46E+08	1.37E+07	5.81E+05
	std	4.64E-23	8.46E-01	1.90E-13	3.58E+12	1.18E+08	5.34E+06	4.97E+05
VP-DECC	mean	7.61E-04	**6.60E-06**	3.82E-05	5.44E+14	5.47E+08	1.96E+07	1.37E+11
	std	1.52E-04	**9.17E-07**	2.15E-06	2.11E+14	6.79E+07	1.05E+05	3.72E+10
CPSO-H	mean	1.50E+04	4.15E+00	1.27E-01	1.55E+13	6.30E+08	1.98E+07	2.55E+10
	std	3.07E+04	1.67E+00	5.92E-02	3.42E+12	9.41E+07	3.68E+04	7.35E+09
EOEA	mean	**2.20E-23**	3.62E-01	**1.67E-13**	**2.86E+12**	**2.24E+07**	**3.85E+06**	**1.24E+02**
	std	**2.87E-23**	6.71E-01	**1.13E-14**	**2.91E+12**	**5.91E+06**	**4.97E+05**	**1.55E+02**
Algorithms	Metric	fun8	fun9	fun10	fun11	fun12	fun13	fun14
MLCC	mean	3.84E+07	1.84E+11	3.14E+03	8.56E+12	1.50E+07	9.41E+10	3.17E+08
	std	2.77E+07	6.28E+04	9.15E+02	2.57E+12	2.76E+01	1.21E+04	2.25E+07
VP-DECC	mean	2.60E+12	2.67E+09	6.91E+03	2.01E+02	1.28E+06	1.27E+06	4.03E+09
	std	5.37E+12	3.75E+08	2.78E+02	7.36E+00	4.55E+04	1.60E+05	4.92E+08
CPSO-H	mean	1.21E+08	1.72E+08	7.00E+03	1.98E+02	3.18E+05	2.29E+03	4.15E+08
	std	4.29E+07	3.57E+07	3.81E+02	4.95E-01	4.71E+04	2.37E+03	2.39E+07
EOEA	mean	**1.01E+07**	**4.63E+07**	**1.00E+03**	**3.18E+01**	**2.61E+04**	**1.24E+03**	**1.65E+08**
	std	**1.28E+07**	**4.78E+06**	**6.91E+01**	**6.57E+00**	**1.91E+04**	**4.59E+02**	**8.95E+06**
Algorithms	Metric	fun15	fun16	fun17	fun18	fun19	fun20	
MLCC	mean	6.78E+03	3.86E+02	8.68E+05	1.45E+04	7.74E+07	2.27E+11	
	std	1.85E+03	7.77E-02	2.36E+02	4.01E+03	1.27E+03	8.67E+03	
VP-DECC	mean	1.38E+04	4.11E+02	2.48E+06	1.41E+05	6.84E+06	1.22E+05	
	std	5.28E+02	8.27E+00	1.48E+05	2.61E+04	9.19E+05	1.67E+04	
CPSO-H	mean	1.39E+04	3.97E+02	6.83E+05	5.57E+03	5.19E+07	3.47E+11	
	std	4.78E+02	2.80E-01	9.50E+04	4.84E+03	7.02E+07	4.36E+08	
EOEA	mean	**2.14E+03**	**8.26E+01**	**7.93E+04**	**2.94E+03**	**1.84E+06**	**1.97E+03**	
	std	**1.22E+02**	**1.68E+01**	**8.80E+03**	**6.92E+02**	**9.97E+04**	**2.35E+02**	

B. Probability based CC

For LSGO problems, cooperative co-evolution with the following divide-and-conquer strategy is a usual and effective choice [3]-[6]:

- 1 Problem decomposition: Splitting the object vectors into some smaller subcomponents.
- 2 Optimize sub-components: Evolve each subcomponent with a certain optimizer separately.
- 3 Cooperative combination: Combine all subcomponents to form the whole system.

Compared with the classical CC based algorithms, the second optimization stage of EOEA makes the following changes to improve its performance:

1) The group sizes of subcomponents are always problem dependent [10], and are hard to determine for different problems. In the second optimization stage, the group sizes of subcomponents are not constant. Instead, they are self-adaptively learnt by the feedback of previous optimization procedure.

2) The groups are optimized by multiple EA techniques. In EOEA, three sub-optimizers are adopted as candidate optimizers for each group. The execution probabilities of associated sub-optimizers are learnt and updated based on the feedback of the previous optimization procedure. Therefore, the more suitable sub-optimizer is expected to be assigned with larger execution probability. The fundamental idea of tuning the sub-algorithms' contribution proportionally to their previous performance is similar to that of [27].

3) It has been discussed in [15] that the variables of LSGO problems may have different effects on the fitness values, i.e., some of the variables may be more important than the others. In EOEA, the variables that contribute more to the fitness values are selected to form a new group, and to be optimized after some certain loops.

The details of probability based CC are shown in Table III. Generally speaking, many EAs can be used as the subcomponent optimizers. In this paper, we select 3 existing EAs, SaDE [16], GA [20], cooperative GA and DE [21]. The purpose of choosing these three sub-optimizers is that they have been proven to have credit performance and different strengthes based on experience: 1) SaDE has shown excellent performance on uni-modal problems and many multimodal problems [16]; 2) GA is effective in jumping out of the local optima due to the mutation operator; 3) the cooperative GA and DE has good balance between exploration and exploitation. The details of the above three sub-optimizers can be found in [16][20][21].

TABLE V

Experimental Results of EOEA for CEC Competition.

Metric	fun1	fun2	fun3	fun4	fun5	fun6	fun7	fun8	fun9	fun10
FES = 1.2E+05										
Best	1.02E+08	3.70E+03	3.74E+00	7.36E+12	1.79E+07	3.46E+06	2.13E+09	4.76E+09	5.75E+08	6.02E+03
Median	1.44E+08	4.11E+03	4.24E+00	1.18E+13	2.79E+07	4.11E+06	3.23E+09	1.21E+10	6.62E+08	6.35E+03
Worst	2.08E+08	4.30E+03	4.58E+00	2.43E+13	7.79E+07	5.06E+06	4.04E+09	1.95E+10	7.94E+08	7.11E+03
Mean	1.52E+08	4.06E+03	4.22E+00	1.41E+13	3.63E+07	4.27E+06	3.02E+09	1.23E+10	6.76E+08	6.46E+03
Std	3.11E+07	2.25E+02	2.20E-01	5.00E+12	1.77E+07	4.50E+05	6.63E+08	5.18E+09	6.94E+07	3.43E+02
FES = 6.0E+05										
Best	7.56E+01	5.05E+01	3.93E-03	4.06E+12	1.29E+07	2.99E+06	4.48E+06	3.24E+05	1.65E+08	1.02E+03
Median	1.09E+02	8.51E+01	5.93E-03	7.78E+12	2.39E+07	3.97E+06	2.79E+07	1.34E+07	2.37E+08	1.12E+03
Worst	9.49E+02	1.76E+02	1.45E-02	1.25E+13	3.08E+07	4.44E+06	1.54E+08	4.53E+07	2.54E+08	1.25E+03
Mean	2.28E+02	9.10E+01	7.74E-03	7.75E+12	2.24E+07	3.85E+06	5.41E+07	1.45E+07	2.24E+08	1.13E+03
Std	2.77E+02	3.67E+01	3.74E-03	2.60E+12	5.91E+06	4.97E+05	4.97E+07	1.39E+07	3.01E+07	6.54E+01
FES = 3.0E+06										
Best	0.00E+00	3.55E-15	1.53E-13	1.25E+12	1.29E+07	2.99E+06	3.99E+00	9.50E+01	3.75E+07	9.50E+02
Median	1.35E-24	1.78E-14	1.71E-13	2.57E+12	2.39E+07	3.97E+06	5.49E+01	5.51E+06	4.74E+07	1.07E+03
Worst	7.95E-23	1.99E+00	1.81E-13	5.96E+12	3.08E+07	4.44E+06	5.27E+02	4.07E+07	5.31E+07	1.18E+03
Mean	2.20E-23	3.62E-01	1.67E-13	3.09E+12	2.24E+07	3.85E+06	1.24E+02	1.01E+07	4.63E+07	1.08E+03
Std	2.87E-23	6.71E-01	1.13E-14	1.61E+12	5.91E+06	4.97E+05	1.55E+02	1.28E+07	4.78E+06	6.91E+01

Metric	fun11	fun12	fun13	fun14	fun15	fun16	fun17	fun18	fun19	fun20
FES = 1.2E+05										
Best	6.01E+01	7.77E+05	1.63E+06	1.90E+09	6.70E+03	1.08E+02	1.59E+06	3.51E+07	9.35E+06	6.32E+07
Median	6.29E+01	8.08E+05	2.04E+06	2.14E+09	7.35E+03	1.22E+02	1.73E+06	6.73E+07	1.13E+07	9.81E+07
Worst	1.12E+02	9.34E+05	3.36E+06	2.23E+09	7.77E+03	1.53E+02	1.79E+06	9.02E+07	1.25E+07	1.17E+08
Mean	7.26E+01	8.35E+05	2.26E+06	2.13E+09	7.31E+03	1.27E+02	1.71E+06	6.42E+07	1.12E+07	9.12E+07
Std	1.71E+01	5.21E+04	5.77E+05	9.67E+07	2.82E+02	1.55E+01	6.35E+04	1.91E+07	1.05E+06	1.77E+07
FES = 6.0E+05										
Best	2.92E+01	1.89E+05	1.64E+03	6.37E+08	1.97E+03	6.58E+01	5.57E+05	1.80E+04	5.04E+06	3.85E+03
Median	3.19E+01	2.55E+05	2.58E+03	7.53E+08	2.13E+03	7.85E+01	7.14E+05	5.00E+04	5.50E+06	4.48E+03
Worst	8.14E+01	2.99E+05	5.31E+03	9.64E+08	2.33E+03	1.19E+02	8.35E+05	6.55E+04	6.42E+06	5.16E+03
Mean	4.03E+01	2.48E+05	2.83E+03	7.79E+08	2.14E+03	8.33E+01	7.08E+05	4.80E+04	5.56E+06	4.42E+03
Std	1.68E+01	3.24E+04	1.16E+03	8.32E+07	1.22E+02	1.71E+01	7.69E+04	1.35E+04	3.98E+05	4.09E+02
FES = 3.0E+06										
Best	2.68E+01	9.57E+03	7.98E+02	1.55E+08	1.97E+03	6.57E+01	6.29E+04	1.99E+03	1.69E+06	1.65E+03
Median	3.03E+01	1.35E+04	1.12E+03	1.63E+08	2.13E+03	7.85E+01	8.12E+04	2.89E+03	1.82E+06	1.96E+03
Worst	7.87E+01	1.99E+04	2.27E+03	1.85E+08	2.33E+03	1.17E+02	9.61E+04	4.10E+03	2.02E+06	2.34E+03
Mean	3.86E+01	1.37E+04	1.24E+03	1.65E+08	2.14E+03	8.26E+01	7.93E+04	2.94E+03	1.84E+06	1.97E+03
Std	1.65E+01	2.90E+03	4.59E+02	8.95E+06	1.22E+02	1.68E+01	8.80E+03	6.92E+02	9.97E+04	2.35E+02

III. Experiment

The experimental section contains two aspects: 1) in order to benchmark the performance of EOEA, we compare it with three previous effective LSGO optimization algorithms, such as cooperative PSO (CPSO-H) [22], variance priority based cooperative co-evolution differential evolution (VP-DECC) [23], and Multilevel Cooperative Coevolution (MLCC) [24]; 2) the experimental results of EOEA are recorded according to the formal format of CEC 2010 large-scale global optimization competition. In these two experiments, the test functions of CEC 2010 large-scale global optimization competition are used [2]. The detailed parameter settings are as follows: the population size $NP = 30$; learning generations $Gs = 2$; fitness evaluation size 3,000,000; trigger is that the 10% fitness evaluations have been used; the runtime is 25.

The test functions have four types [2]: 1) Separable functions; 2) Partially-separable functions, in which a small number of variables are dependent while all the remaining ones are independent; 3) Partially-separable functions that consist of multiple independent subcomponents, each of which is m-nonseparable; and 4) Fully-nonseparable functions.

A. Comparison with Other LSGO Algorithms

The statistical experimental results of 25 runs are summarized in Table IV. In Table IV, the best result for each function is covered with boldface. It is interesting to observe that EOEA provides the best performance for almost all the

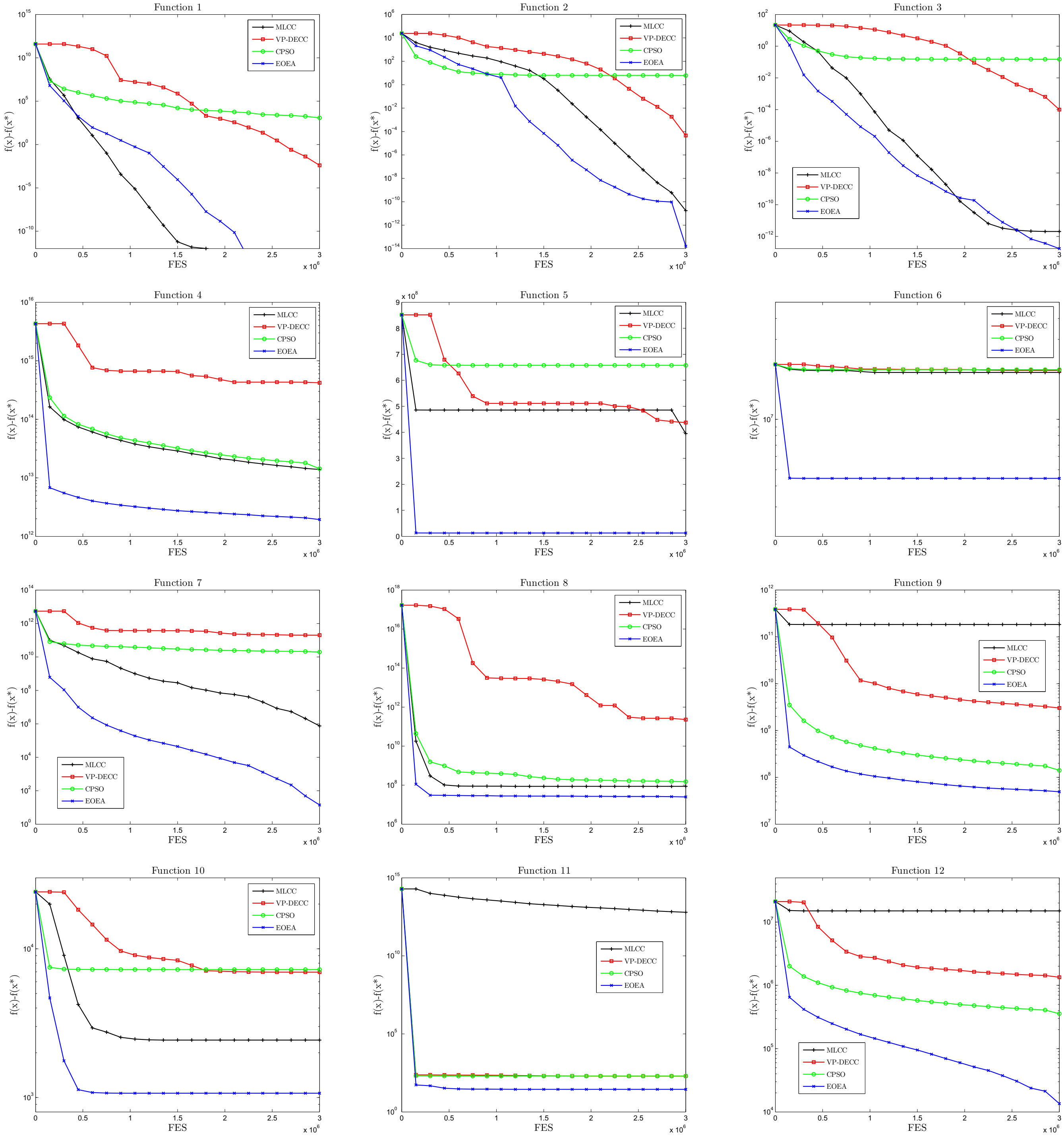

Fig. 1. Optimization curves of functions 1-12.

problems only except function 2 (separable Rastrigin's problem). For the separable functions 1 and 3, only MLCC can obtain comparable results. As to the non-separable functions, which are more difficult due to the complicated relationships among variables, EOEA remarkably outperforms the other algorithms. Especially, on the completely non-separable function Rosenbrock's problem, EOEA outperforms the other algorithms by at least two magnitudes. The reason can be summarized as follows: 1) the first optimization stage can locate the population to promising region; 2) the multiple search techniques ensemble can strengthen the searching ability [19]; and 3) the adaptive group size learning based strategy enhance that the second optimization stage can find the most suitable group size for different problems. In

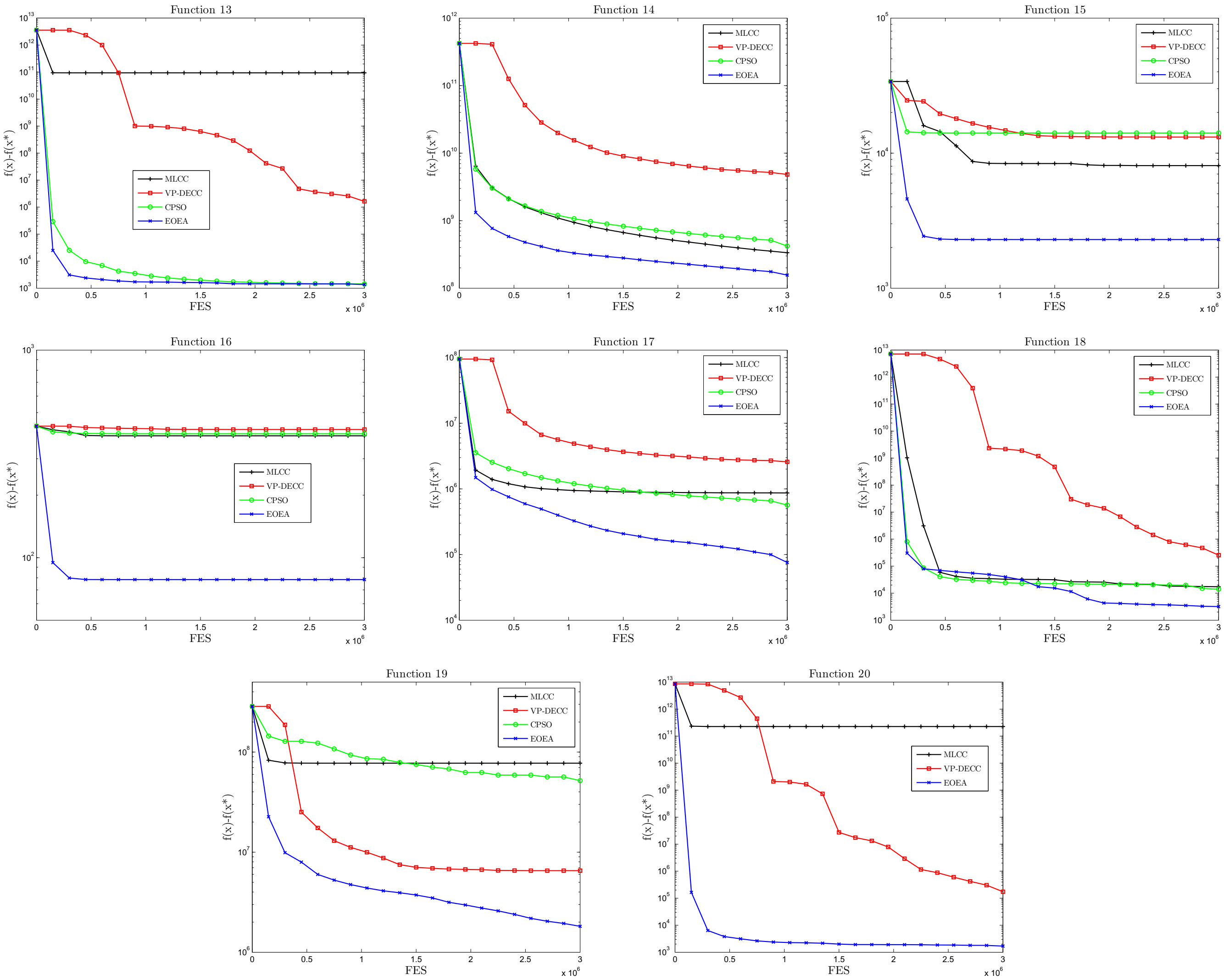

Fig. 2. Optimization curves of functions 13-20.

summary, the excellent performance of EOEA comes from the cooperation of all the incorporated strategies, while the missing of any strategy results in worse performance.

Figs. 1 and 2 depict the optimization procedure of median run of the four compared algorithms. For the separable functions 1-3, the optimization curves of MLCC and EOEA, which can find satisfactory solutions (the best fitness is lower than $1e-5$ [2]), are similar while the other two algorithms perform much worse. For the other 17 non-separable problems, EOEA always provides the best solution from the very beginning. It is not surprising that the first optimization stage using MUEDA can find a promising region within a little optimization cost.

B. *Experimental Results for CEC 2010 Competition*

According to the requirement of CEC 2010 "large-scale global optimization" competition, the experimental results are recorded in Table V. For comparison, please refer to other papers submitted to the same special session.

IV. Conclusion

Due to the diverse applications, it is important to propose effective algorithms for LSGO problems. In this paper, we design a two-stage based ensemble optimization evolutionary algorithm (EOEA) for LSGO. The motivation of EOEA is to combine multiple EA techniques in a proper way to tackle multiple challenges of LSGO problems.

Experimental evidence is provided to show how these incorporated strategies improve the performance. It is shown that EOEA is definitely fit for complex LSGO problems. Besides, the formal formatted experimental results are recorded for CEC 2010 "large-scale global optimization" competition.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China (Grants No. U0835002); the Chinese Academy of Science (CAS) Special Grant for Postgraduate Research, Innovation and Practice; and Graduate

Innovation Fund of University of Science and Technology of China.

The authors are grateful to Prof. Suganthan and Zhenyu Yang for providing the source code of their algorithms.

REFERENCES

[1] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, Z. Yang, "Benchmark Functions for the CEC2008 Special Session and Competition on Large Scale Global Optimization", Technical Report for CEC 2008 special issue, Nov. 2007.

[2] K. Tang, Xiaodong Li, P. N. Suganthan, Z. Yang and T. Weise, "Benchmark Functions for the CEC'2010 Special Session and Competition on Large Scale Global Optimization," Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, http://nical.ustc.edu.cn/cec10ss.php, Nanyang Technological University, 2009

[3] "Large-Scale Nonlinear Optimization ", in series: Nonconvex Optimization and Its Applications, vol. 83, D. Pillo, Gianni; Roma, Massimo (Eds.), DOI 10.1007/0-387-30065-1, Springer Netherlands, 2006.

[4] A. M. Potter and K. A. D. Jong, "A cooperative coevolutionary approach to function optimization", *Proc. the Third International Conference on Parallel Problem Solving from Nature*, pp. 249-257. Springer-Verlag, 1994.

[5] Y. Liu, X. Yao, Q. Zhao and T. Higuchi, "Scaling Up Fast Evolutionary Porgramming with Cooperative Co-evolution," *Proc. IEEE Congress on Evolutionary Computation,* pp. 1101-1108, 2001.

[6] Z. Yang, K. Tang and X. Yao, "Differential Evolution for High-Dimensional Function Optimization", *Proc. IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 3523-3530.

[7] Z. Yang, K. Tang and X. Yao: "Large Scale Evolutionary Optimization Using Cooperative Coevolution," *Information Sciences,* vol. 178, no. 15, pp. 2985-2999, 2008.

[8] C. MaCnish and X. Yao, "Direction Matters in High-dimensional Optimisation," *Proc. IEEE Congress on Evolutionary Computation*, pp. 2377-2384, Hongkong, 2008.

[9] J. Brest, A. Zamuda, B. Boskovic, M. S. Maucec and V. Zumer, "High-dimensional Real-parameter Optimization Using Self-adaptive Differential Evolution Algorithm with Population Size Reduction," *Proc. IEEE Congress on Evolutionary Computation*, pp. 2032-2039, Hongkong, 2008.

[10] Z. Yang, K. Tang and X. Yao, "Multilevel Cooperative Coevolution for Large Scale Optimization," *Proc. IEEE Congress on Evolutionary Computation*, pp. 1663-1670, Hongkong, 2008.

[11] S. Hsieh, T. Sun C. Liu and S. Tsai, "Solving Large Scale Global Optimization Using Improved Particle Swarm Optimizer," *Proc. IEEE Congress on Evolutionary Computation*, pp. 1777-1784, Hongkong, 2008.

[12] S. Zhao, J. J. Liang, P. N. Suganthan, M. F. Tasgetiren, "Dynamic Multi-swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization," *Proc. IEEE Congress on Evolutionary Computation*, pp. 3845-3852, Hongkong, 2008.

[13] L. Tseng and C. Chen, "Multiple Trajectory Search for Large Scale Global Optimization," *Proc. IEEE Congress on Evolutionary Computation*, pp. 3052-3059, Hongkong, 2008.

[14] A. Zamuda, Janez Brest, B. Boskovic and V. Zumer, "Large Scale Global Optimization Using Differential Evolution with Self Adaptation and Cooperative Co-evolution," *Proc. IEEE Congress on Evolutionary Computation*, pp. 3719-3726, Hongkong, 2008.

[15] Y. Wang and B. Li, "A Restart Uni-variable Estimation of Distribution Algorithms: Sampling Under Mixed Gaussian and Levy Probability Distribution," *Proc. IEEE Congress on Evolutionary Computation*, pp. 3218-3925, Hongkong, 2008.

[16] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," *Proc. IEEE Congress on Evolutionary Computation,* pp. 1785-1791, 2005.

[17] P. Larranga, R. Etxeberria, J. A. Lozano, and J. M. Pena, "Optimization in continuous domains by learning and simulation of Gaussian networks," *Proc. . 2000 Genetic and Evolutionary Computation Conf. Workshop Program,* pp. 201-204, 2000.

[18] Y. Wang, B. Li, "A Self-adaptive Mixed Distribution Based Univariate Estimation of Distribution Algorithm for Large Scale Global Optimization," in book: Nature-Inspired Algorithms for Optimization, in series: Studies in Computational Intelligence, Raymond Chiong(Ed), vol. 193, Chiong, Raymond (Ed.), pp. 171-198, Hardcover ISBN: 978-3-642-00266-3, Springer-Verlag.

[19] Y. Wang, B. Li, "Cooperative Estimation of Distribution Based Differential Evolution for Economic Dispatch Optimization of Power System," Information Sciences, accepted.

[20] K. Deb, http://www.iitk.ac.in/kangal/code/new nsga/nsga2code.tar.

[21] Y. Wang and B. Li, "Multi-strategy Ensemble Evolutionary Algorithm for Dynamic Multi-objective Optimization," *Memetic Computing,* vol. 2, no. 1, pp. 1-22, September 2009.

[22] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225-239, June 2004.

[23] Y. Wang, B. Li and X. X. Lai, "Variance Priority based Cooperative Co-evolution Differential Evolution for Large Scale Global Optimization," *Proc. the 2009 IEEE Congress on Evolutionary Computation (CEC2009),* Norway, 18th-21st May, 2009, pp. 1232-1239.

[24] Z. Y. Yang, K. Tang and X. Yao, "Multilevel Cooperative Coevolution for Large Scale Optimization," *Proc. the 2008 IEEE Congress on Evolutionary Computation (CEC2008),* Hongkong, China, 2008, pp. 1663-1670.

[25] V. Robles, Jose M. Peña, Pedro Larrañaga and María S. Pírez, "GA-EDA: A New Hybrid Cooperative Search Evolutionary Algorithm," in book: Towards a New Evolutionary Computation, in series: Studies in Computational Intelligence, J.A. Lozano, P. Larrañaga, I. Inza and E. Bengoetxea (Eds.), vol. 192, Springer-Verlag, pp. 187-219, 2006.

[26] J. Sun, Q. F. Zhang and E.Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Information Sciences,* vol. 169, pp. 249-262, 2005.

[27] J. A. Vrugt, B. A. Robinson and J. M. Hyman, "Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 243-259, 2009.

[28] P. Cowling P, K. Chakhlevitch, "Hyperheuristics for managing a large collection of low level heuristics to schedule personnel," *Proc. IEEE Congress on Evolutionary Computation (CEC2003),* Canberra, Australia, pp. 1214-1221, 2003.

[29] P. Cowling, G. Kendall, E. Soubeiga, "A hyperheuristic approach for scheduling a sales summit," *In: Selected Papers of the Third International Conference on the Practice And Theory of Automated Timetabling, PATAT 2000,* Springer, Konstanz, Germany, Lecture Notes in Computer Science, 176-190, 2000.

[30] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, R. Qu, "A survey of hyper-heuristics," Tech. Rep. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham (2009).

[31] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *IEEE Trans. Evol. Comput.*13(2) (2009) 398-417.

[32] R. Mallipeddi, S. Mallipeddi, P. N. Suganthan, "Ensemble strategies with adaptive evolutionary programming," *Information Sciences,* vol. 180, pp. 1571-1581, 2010.

[33] J. A. Vrugt and B. A. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search," *Proc Natl Acad Sci U S A.,* vol. 104, no. 3, pp. 708-711, January 2007.