WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IEEE CEC

Cooperative Coevolution with Global Search for Large Scale Global Optimization

Kaibo Zhang *Student Member,IEEE*
Nature Inspire Computation and Applications Lab(NICAL)
Department of Electronic Science and Technology
School of Information Science and Technology
University of Science and Technology of China
Hefei, Anhui, China

Bin Li *Member,IEEE*
Nature Inspire Computation and Applications Lab(NICAL)
Department of Electronic Science and Technology
School of Information Science and Technology
University of Science and Technology of China
Hefei, Anhui, China

***Abstract*—To improve the performance of EAs on large scale numerical optimization problems, a number of techniques have been invented, among which, Cooperative Coevolution (CC in short) is obviously a promising one. But sometimes CC is easy to lead to premature convergence in large scale global optimization. In this paper, a Cooperative Coevolution Evolutionary Algorithm (CCEA in short) with global search (CCGS) is presented to handle large scale global optimization (LSGO) problems. The performance of CCGS is evaluated on the test functions provided for the CEC 2012 competition and special session on Large Scale Global Optimization. The experiment results show that this technique is more effective than CCEAs without global search.**

I. INTRODUCTION

Evolutionary Algorithms (EAs) have been widely applied to solve many numerical and combinational problems in recent years [1] [2]. However, most of conventional evolutionary algorithms suffer from the "curse of dimensionality" [3], their performance deteriorate rapidly as the dimensionality of the search space increases.

To improve the performance of EAs on problems of large scale (high dimension), a number of techniques have been invented [4] [5] [6], among which, Cooperative Coevolution (CC in short), proposed by Potter and De Jong [4], is obviously a promising one. In CC, the idea "Divide and Conquer" is implemented, which decomposes a large scale problem into several sub-problems with low dimension by separating the whole decision variable set into a number of groups, each group forms a sub-space of solutions, a certain evolutionary algorithm is applied on each sub-space, and after that, they are able to share information with each other.

It has been observed that the convergence speed of the EAs without CC is very high for the relatively easy problems [11], such as the Sphere and Ackley problems. But their capacity of dealing with the complex problems [11], such as the Rastrigin problem is very limited. CCEAs perform better than conventional EAs on the complex problems, but they are still easy to lead to premature convergence in LSGO. For LSGO, the conflict between exploration and exploitation is much more incisive due to the extremely high dimensionality.

To address the problem of premature convergence of CCEAs on LSGO, a CCEA with global search (CCGS) is designed to handle LSGO problems, in which, the searching process is divide into two stages: 1) the subspace exploration stage. 2) the global space exploration stage. The goal of the first stage is to explore each sub-space extensively to find the best possible solution. The second stage aims to explore global searching space to escape from local optimal.

To achieve the first objective, a CC evolutionary algorithm (NCCEA in short) is designed in the first stage. Some important differences between our algorithm and conventional CCEAs are: 1) the size of each group is adaptively changed based on their previous performances. 2) the optimizer for each group can be selected from 3 evolutionary algorithms based on their previous performance, the method adopted is similar as that presented in [17].

In previous literatures, the role of group sizing and optimizer choosing have been discussed much. Generally, the optimizer for each group is same and the size of each group is constant [4] [7] [8] [9]. [10] proposed a multilevel group size adaptive strategy, which design several problem decomposers based on different group sizes to form a decomposer pool and select one decomposer from the decomposer pool based on their performance records in each cycle. [17] proposed a flexible feedback learning strategy, in which the size of each group is adaptively changed based on their previous performance.

To accomplish the second objective, an EDA based-on mixed Gaussian and Cauchy models (MUEDA)[11] is adopted in the second stage. The performance of MUEDA has been studied in previous works [11] [17].

EOEA [17] is also a two-stage EA, which adopt MUEDA in the first stage to shrink the search region as quickly as possible to the promising area. Afterward, CC-based algorithm is adopted in the second stage to explore the limited area. Different from EOEA, in CCGS, CC based EA is used to explore the subspaces extensively first, then, MUEDA is adopted to conduct global exploration in the whole space. The idea behind is to approach the promising area by "divide and conquer" scheme at the first stage, then tackle the problem caused by the incorrect partition of the correlated variables via

TABLE II. THE PROCEDURE OF NCCEA

NCCEA
Initialization:
Randomly initialize NP individuals to form the population and evaluate the fitness;
Set fitness evaluation times fevaltimes = 0;
Set the selection probability of three sub-space optimizer SP_i=1/3, i=1, 2, 3
Set accumulators of three sub-space optimizer SA_i=0, i=1, 2, 3. Set learning coefficient α = 0.15
Set mean group size meansize = 30, and standard deviation value stdsize =10.
Optimization:
While fevaltimes < maxfeval
Sample a Gaussian distribution with mean value meansize and standard deviation stdsize to generate the set of group sizes GS = $\{GS_1, GS_2 ... GS_n\}$, where n is the number of groups and $\sum GS_i$=D,D is the dimension of functions
Select different subspace optimizers SO_1, SO_2...SO_n by roulette wheel selection based on SP_i, i = 1, 2, 3.
For k = 1 : n
Optimize the k*th* group by sub-optimizer SO_k.
Calculate the group performance $P_k = \|v - v'\|/v$, where v and v′ are the current and previous best values.
Add group performance P_k to the accumulator of associated sub-optimizer SA_i, i=1, 2, 3.
End for
Sort the P_k, k = 1, 2...n in descending order and get P'_k, k = 1, 2...n, Select [n/2] top performance groups to update meansize and stdsize of Gaussian distribution.
For i =1:3
$SP_i = (1- \alpha) * SP_i + \alpha * SA_i / \sum SA_i$
SA_i = 0;
end for
end while

conducting a global search in the whole space.

The rest of this paper is organized as follows. Section Ⅱ gives a brief introduction to CCGS. Section Ⅲ presents the experimental results. The paper is concluded in Section Ⅳ with a brief statement of future scopes.

II. COOPERATIVE COEVOLUTION WITH GLOBAL SEARCH

In order to overcome the shortcoming of premature convergence, we combine CC with global search to handle LSGO problems. CCGS has two sequential stages with different search techniques:

1) In the first stage, a CC evolutionary algorithm is designed to explore each sub-space extensively to find the best possible solution

2) In the second stage, MUEDA is adopted to explore global searching space to escape from local optimal

The framework of CCGS is presented in Table Ⅰ. The details of the CC evolutionary algorithm (NCCEA) and MUEDA are described in the following sections.

TABLE Ⅰ THE FRAMEWORK OF CCGS

CCGS
Input: Optimization task
Terminal condition
Output: The best solution and optimization information found
● **Step 0: Initialization:** randomly initialize the population and evaluate the fitness
● **Step 1: The first optimization stage:** NCCEA
● **Step 2: Judgment:** If the improvement of solution is lower than threshold, go to step 3; otherwise, back to step 1.
● **Step 3: The second optimization stage:** MUEDA
● **Step 4: Termination and output**

TABLE Ⅲ. THE PROCEDURE OF MUEDA

MUEDA
Initialization: Set the weight vector $W_i=0.55-e^{lg(D/100000)}$, i=1, 2...D, the population X_0 is inherit from the first stage, probabilistic model P_0 is conducted based on X_0, Set t = 0
● **Step 1:** Sample the new individuals based on P_t,
● **Step 2:** Set t=t+1
● **Step 3:** Select top 20% individuals by truncation strategy to update the model
● **Step 5:** Standard Deviation Control Strategy
● **Step 6:** If the termination condition is not meet, go to 1; otherwise, end MUEDA.

A. NCCEA

Conventional Cooperative Co-evolutionary Algorithms (CCEAs) follow three basic steps:

1) Divide the variables of the problem into m low-dimension subcomponents

2) Optimize each of the m subcomponents with a certain EA for a predetermined number of evaluations

3) Terminate the evolutionary process if the halting criterion is satisfied or the maximum number of evaluations is achieved.

Since the publication of A. M. Potter and K. A. De Jong’s seminal work [4], CC quickly become a popular technique in Eas for large scale optimization. In recent years, several variable grouping strategies have been proposed, such as cooperative co-evolution fast evolutionary programming (FEPCC) [7], cooperative approach to particle swarm optimization(CPSO) [9], differential evolution with CC:

DECC-I, DECC-II [23], MLCC [10], CC with delta grouping [24] and CC with variable interaction learning [25].

Compared with conventional CCEAs, NCCEA has two differences:

1) The group sizes of subspaces are not always constant. They are always problem dependent and determined for different problems. In NCCEA, the group sizes of subspaces are adaptively changed based on a flexible feedback learning strategy.

2) The optimizers for different groups are not necessarily the same, a kind of multiple method ensemble [19] [20] mechanism is adopted. The optimizer for each group can be selected from three evolutionary algorithms based on their previous performance. The selective probability of different evolutionary algorithms are learnt and updated based on previous optimization procedures by the method presented in [17]. PAP [22] also takes multiple algorithms as its constituent algorithms to reduce the risk of falling on problems from different classes.

The detail of NCCEA is presented in Table Ⅱ. We select three different evolutionary algorithms to optimize subspaces. They are SaNSDE [12], GA [13] and CMA-ES [14]. SaNSDE performs well on unimodal problems and many multimodal problems. SaNSDE inherits the self-adapted mutation schemes selection mechanism of SaDE [21], use the same self-adaptive strategy as that in SaDE to adapt the balance between Gaussian and Cauchy operators and follows the strategy in SaDE to dynamically adapt the value of CR (CR is the crossover rate). GA is effective to jump out of local optimum. CMA-ES is effective to overcome problems that are often associated with evolutionary algorithms as follows: 1) Poor performance on badly scaled and/or highly non-separable objective functions. 2) The inherent need to use large population sizes. 3) Premature convergence of the population. The detail of these three Eas can be found in [12] [13] [14].

B. *MUEDA*

The major advantage of EDAs is that they can explicitly learn the promising area of solution space of the optimization problem and then use this structural information to efficiently generate new individuals [18].

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 [11].

In order to tackle the problem possiblely caused by the incorrect partition of correlated variables in CCEA, MUEDA [11] is adopted to conduct the global search in the second stage of CCGS. The procedure of MUEDA is shown in Table Ⅲ, and more details can be found in [11]. The main idea of STDC strategy is to estimate a common threshold of standard deviations for all variables to control their shrinking speeds in the optimization process. The variables that have lower standard deviation values than the corresponding thresholds will be forced to set their standard deviations to be the corresponding thresholds. The weighted mean of the standard deviations of all variables is used as a control standard here.

III. EXPERIMENTAL RESULTS

For each test function, CCGS is run for 25 independent times and all the results of 25 runs are recorded. In this paper, the population size is set to 20, D=1000, the maximum number of evaluations is set to *3.0e6*.

The performance of CCGS is evaluated with CEC'2012 benchmark functions which were proposed on the Special Session and Competition on Large Scale Global Optimization in CEC'2012 [15].These functions are scalable and contain five categories as follows:

1) Separable Functions $\left(f_1 - f_3\right)$
2) Single $-$ group m $-$ nonseparable Functions $\left(f_4 - f_8\right)$
3) $\frac{D}{2m}$-group m-nonseparable Functions (f_9-f_{13})
4) $\frac{D}{m}$-group m-nonseparable Functions (f_9-f_{13})
5) Nonseparable Functions $\left(f_{19} - f_{20}\right)$

A. *Experimental results for CEC2012 LSGO competition*

According to the requirement of CEC 2012 "large-scale global optimization" competition, in Table Ⅳ, the best, worst, median, mean and standard derivation of CCGS are recorded.

B. *Comparison with Other LSGO Algorithms*

The results of CCGS are compared with those of some previous LSGO algorithms, such as DECC-G [16] and MLCC [10] in Table Ⅴ. For separable functions f_1 and f_3, non-separable function f_{10}, MLCC is better than CCGS. DECC-G performs better than CCGS on non-separable function f_{11}. CCGS outperforms DECC-G and MLCC significantly on almost all functions except these four functions. Based on the experimental results obtained on CEC'2012 benchmark functions, we can conclude that CCGS is more suitable for high-dimensional optimization. The reasons can be inferred as follows: 1) the flexible group size adaptive strategy can find the most suitable group sizes in most functions; 2) multi-strategies CC can suit different search space in most functions; 3) MUEDA is effective to search the global searching space

Figs. 1 depict the optimization procedure of median run of CCGS on the following eight selected functions: f_2, f_5, f_8, f_{10}, f_{13}, f_{15}, f_{18}, and f_{20}.

1) In function f_2, we observe that CCGS, after a small convergence ratio, continues with a quicker ratio.
2) For functions f_5 and f_{20}, the convergence curve has a very steep slope.
3) In function f_8, f_{13} and f_{18}, we observe that the best fitness is improved by steps.
4) In functions f_{10} and f_{15}, after a quick improvement, the fitness value is almost stabilized during the majority of the evaluations.

TABLE IV. EXPERIMENTAL RESULTS OF CCGS

Metric	fun1	fun2	fun3	fun4	fun5	fun6	fun7	fun8	fun9	fun10
FES=1.2E+05										
Best	5.39E+06	3.75E+03	4.02E+00	6.69E+12	1.65E+08	4.84E+06	9.46E+08	9.34E+05	1.32E+09	1.01E+04
Median	7.40E+06	4.30E+03	4.61E+00	2.72E+13	1.94E+08	5.00E+06	6.00E+09	9.29E+07	1.85E+09	1.15E+04
Worst	1.23E+07	5.77E+03	5.48E+00	6.27E+13	6.76E+08	5.24E+06	3.96E+10	1.01E+10	2.35E+09	1.35E+04
Mean	7.66E+06	4.48E+03	4.65E+00	2.97E+13	2.13E+08	5.01E+06	9.27E+09	1.34E+09	1.81E+09	1.16E+04
Std	1.55E+06	5.84E+02	3.19E-01	1.35E+13	9.72E+07	1.21E+05	8.59E+09	3.05E+09	2.73E+08	6.78E+02
FES=6.0E+05										
Best	7.70E-05	2.30E+02	2.19E-05	2.07E+12	1.36E+07	1.95E+06	1.29E+06	1.62E+04	1.74E+08	1.15E+03
Median	4.62E-04	3.62E+02	8.34E-03	6.21E+12	1.89E+07	2.99E+06	3.64E+06	7.40E+07	2.47E+08	6.93E+03
Worst	1.07E-02	6.93E+02	1.67E+00	1.32E+13	3.18E+07	3.73E+06	1.34E+07	9.38E+09	3.06E+08	8.59E+03
Mean	1.39E-03	3.79E+02	4.32E-01	6.64E+12	1.97E+07	2.88E+06	4.66E+06	7.25E+08	2.43E+08	5.62E+03
Std	2.33E-03	1.03E+02	5.15E-01	2.94E+12	4.69E+06	4.87E+05	2.82E+06	2.21E+09	3.42E+07	2.66E+03
FES=3.0E+06										
Best	5.96E-27	9.47E-04	2.06E-13	7.17E+11	1.36E+07	1.95E+06	1.98E+01	3.03E+02	4.05E+07	7.91E+02
Median	6.49E-23	3.00E-03	2.45E-13	1.44E+12	1.89E+07	2.99E+06	1.09E+02	1.12E+07	5.33E+07	6.04E+03
Worst	1.79E-21	1.00E+00	1.38E+00	3.52E+12	3.18E+07	3.73E+06	5.79E+02	8.19E+07	7.56E+07	7.10E+03
Mean	1.83E-22	4.44E-02	1.91E-01	1.79E+12	1.97E+07	2.88E+06	1.37E+02	2.81E+07	5.53E+07	4.74E+03
Std	3.68E-22	1.99E-01	4.49E-01	7.62E+11	4.69E+06	4.87E+05	1.16E+02	3.14E+07	9.60E+06	2.45E+03
Metric	fun11	fun12	fun13	fun14	fun15	fun16	fun17	fun18	fun19	fun20
FES=1.2E+05										
Best	7.01E+01	1.18E+06	2.02E+06	2.92E+09	1.04E+04	2.73E+01	1.88E+06	4.45E+06	5.01E+06	4.76E+07
Median	2.25E+02	1.33E+06	2.70E+06	3.69E+09	1.48E+04	3.89E+01	2.16E+06	2.20E+08	6.60E+06	2.63E+08
Worst	2.26E+02	1.53E+06	4.51E+06	4.95E+09	1.62E+04	4.11E+02	2.74E+06	7.83E+08	7.91E+06	1.84E+09
Mean	1.71E+02	1.34E+06	2.79E+06	3.75E+09	1.46E+04	1.00E+02	2.20E+06	2.93E+08	6.51E+06	3.87E+08
Std	7.47E+01	9.02E+04	5.26E+05	4.72E+08	1.38E+03	1.39E+02	2.13E+05	2.42E+08	6.73E+05	3.81E+08
FES=6.0E+05										
Best	4.83E+01	5.01E+04	1.09E+03	3.55E+08	1.64E+03	2.30E+01	1.69E+05	1.45E+04	1.71E+06	1.34E+03
Median	5.12E+01	5.90E+04	5.07E+03	4.10E+08	1.85E+03	3.33E+01	1.85E+05	5.16E+04	1.90E+06	1.96E+03
Worst	7.47E+01	6.88E+04	2.47E+04	4.45E+08	2.30E+03	4.52E+01	2.20E+05	7.09E+04	2.12E+06	7.91E+03
Mean	5.44E+01	5.87E+04	7.19E+03	4.08E+08	1.87E+03	3.31E+01	1.87E+05	4.86E+04	1.89E+06	2.42E+03
Std	6.23E+00	5.75E+03	5.83E+03	2.32E+07	1.54E+02	5.37E+00	1.30E+04	1.56E+04	1.04E+05	1.38E+03
FES=3.0E+06										
Best	2.64E+01	4.71E+03	5.76E+02	1.17E+08	1.56E+03	2.16E+01	1.26E+04	1.69E+03	5.28E+05	1.05E+03
Median	2.94E+01	5.31E+03	1.26E+03	1.37E+08	1.75E+03	3.29E+01	1.48E+04	3.15E+03	5.92E+05	1.27E+03
Worst	4.72E+01	6.44E+03	2.96E+03	1.54E+08	1.94E+03	4.19E+01	1.67E+04	5.61E+03	6.91E+05	1.90E+03
Mean	2.99E+01	5.35E+03	1.51E+03	1.35E+08	1.74E+03	3.11E+01	1.48E+04	3.13E+03	5.93E+05	1.31E+03
Std	3.98E+00	4.39E+02	6.94E+02	9.05E+06	8.94E+01	5.22E+00	1.02E+03	1.01E+03	4.21E+04	2.14E+02

TABLE V. COMPARISON OF THREE ALGORITHMS

Test Function	CCGS Mean	DECC-G Mean	MLCC Mean	CCGS/DECC-G t-test	CCGS/MLCC t-test
fun1	1.83E-22	2.93E-07	**1.53E-27**	-15.2011+	2.5799
fun2	**4.44E-02**	1.31E+03	5.57E-01	-179.6979+	-0.9709+
fun3	1.91E-01	1.39E+00	**9.88E-13**	-53.5206+	2.7266
fun4	**1.79E+12**	1.70E+13	9.61E+12	-12.6084+	-10.1043+
fun5	**1.97E+07**	2.63E+08	3.84E+08	-12.9057+	-23.5263+
fun6	**2.88E+06**	4.96E+06	1.62E+07	-11.3021+	-11.9375+
fun7	**1.37E+02**	1.63E+08	6.89E+05	-5.3209+	-4.1801+
fun8	**2.81E+07**	6.44E+07	4.38E+07	-5.4782+	-1.9187+
fun9	**5.53E+07**	3.21E+08	1.23E+08	-35.4416+	-23.4921+
fun10	4.74E+03	1.06E+04	**3.43E+03**	-82.5407+	8.8483
fun11	2.99E+01	**2.34E+01**	1.98E+02	16.3649	-107.69+
fun12	**5.35E+03**	8.93E+04	3.49E+04	-54.6001+	-26.7924+
fun13	**1.51E+03**	5.12E+03	2.08E+03	-4.1693+	-3.9527+
fun14	**1.35E+08**	8.08E+08	3.16E+08	-49.7917+	-29.6775+
fun15	**1.74E+03**	1.22E+04	7.11E+03	-52.1817+	-17.9431+
fun16	**3.11E+01**	7.66E+01	3.76E+02	-24.7603+	-32.7071+
fun17	**1.48E+04**	2.87E+05	1.59E+05	-61.5430+	-45.1831+
fun18	**3.13E+03**	2.46E+04	7.09E+03	-9.1184+	-3.6554+
fun19	**5.93E+05**	1.11E+06	1.36E+06	-44.5754+	-46.4445+
fun20	**1.31E+03**	4.06E+03	2.05E+03	-33.1104+	-17.3856+

+The value of t-test is significant at $\alpha = 0.05$ by a two-tailed test.

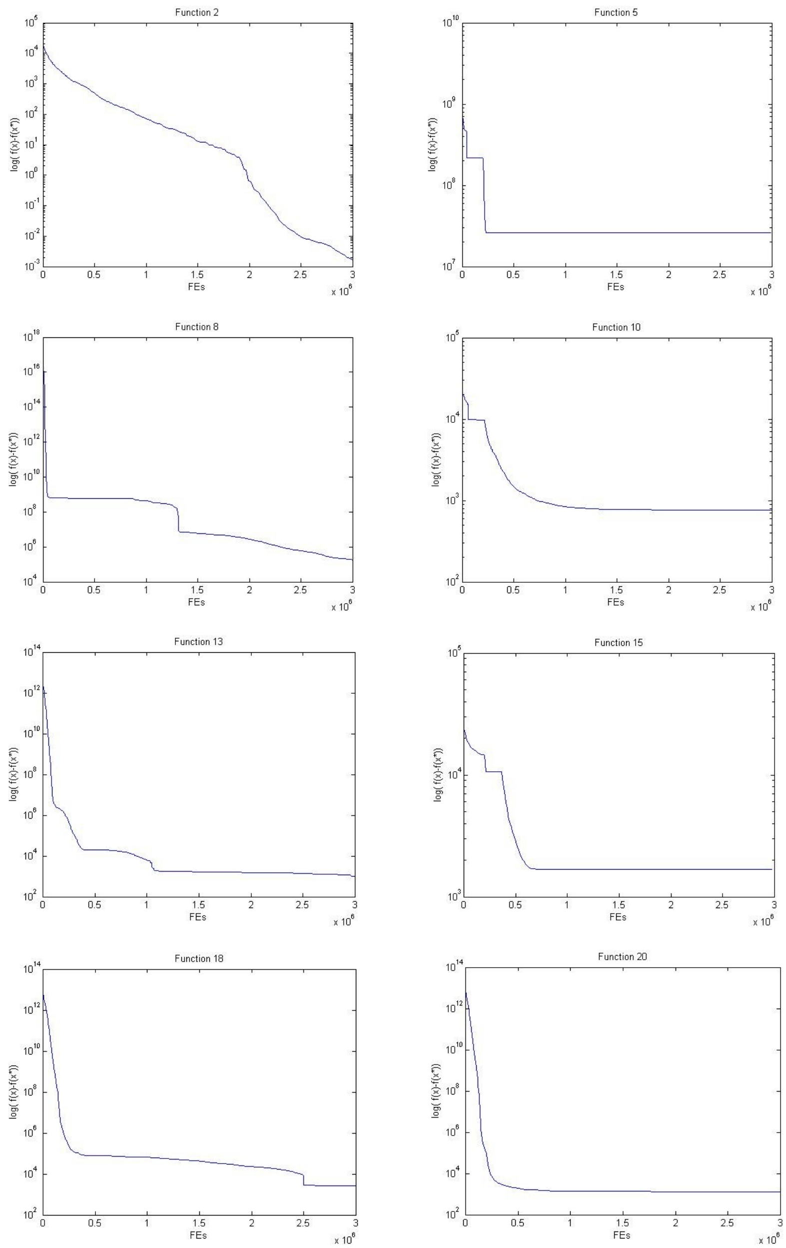

Figure 1. Convergence plot of functions f_2,f_5,f_8,f_{10},f_{13},f_{15},f_{18} and f_{20}

IV. CONCLUSION

In this paper, CCGS is presented to solve the large scale global optimization problems. In CCGS, a two-stage framework is adopted to combine the merits of CCEA and global search EA. In the first stage, a CCEA is designed to explorer sub-space effectively, and in the second stage, MUEDA is adopted to explore the global searching space to help the algorithm escape from local optimal. The results of CCGS show that it is effective to solve LSGO problems.

ACKNOWLEDGMENT

The work was partially supported by the National Natural Science Foundation of China (No. 61071024, U0835002), and the USTC Innovation Fund for Young Researchers.

REFERENCES

[1] T. Bäck. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. ser. Dover Books on Mathematics. Oxford University Press, 1996.

[2] Sarker. R, Mohammadian. M, Yao. X: Evolutionary Optimization. Kluwer Academic Publishers Norwell, MA, USA (2002)

[3] R. E. Bellman. *Dynamic Programming*. ser. Dover Books on Mathematics. Princeton University Press, 1957.

[4] A. M. Potter and K. A. De Jong, "A Cooperative Co-evolutionary Approach to Function Optimization", *Proc. of the Third International Conference on Parallel Problem Solving from Nature,* pp. 249-257,Springer-Verlag, 1994

[5] M. Weber, F. Neri, V. Tirronen, Shuffle or update parallel differential evolution for large-scale optimization, Soft Computing—A Fusion of Foundations, Methodologies and Applications 15 (11) (2011) 2089–2107.

[6] HU Cheng-yu, WANG Bo. Particle Swarm Optimization with Dynamic Dimension Crossover for High Dimensional Problems[J];Computing Technology and Automation;2009-01

[7] Y. Liu, X. Yao, Q. Zhao and T. Higuchi, "Scaling Up Fast Evolutionary Programming with Cooperative Coevolution," *Proc. of the Congress on Evolutionary Computation*, pp. 1101–1108, 2001

[8] D. Sofge, K. A. De Jong, and A. Schultz, "A blended population approach to cooperative coevolution for decomposition of complex problems," Proc. of the Congress on Evolutionary Computation, pp.413–418, 2002

[9] Bergh, F.v.d. and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," IEEE Transactions On Evolutionary Computation, vol. 3 pp. 225–239, 2004.

[10] Z. Yang, K. Tang, and X. Yao. Multilevel cooperative coevolution for large scale optimization. In 2008 IEEE Congress on Evolutionary Computation, pages 1663–1670, 2008.

[11] 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.

[12] Z. Yang, K. Tang, X. Yao, Self-adaptive differential evolution with neighborhood search, in: Proceedings of the 2008 Congress on Evolutionary Computation, in press

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

[14] Hansen, N. (2005). The CMA evolution strategy: A tutorial. Working Paper

[15] 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://staff.ustc.edu.cn/~ketang/cec2012/lsgo_competition.htm

[16] Zhenyu Yang, Ke Tang and X. Yao: "Large Scale Evolutionary Optimization Using Cooperative Coevolution", Information Sciences, 178(15):2985-2999, August 2008.

[17] Y. Wang, B. Li, "Two-stage based Ensemble Optimization for Large-Scale Global Optimization," Proc. the 2010 IEEE Congress on Evolutionary Computation(CEC2010), Barcelona, 2010, pp. 4488-4495

[18] B. Yuan, Marcus Gallagher "Experimental Results for the Special Session on Real-Parameter Optimization at CEC 2005: A Simple, Continuous EDA", Proc. Conference on Evolutionary Computation, special issue on Real-Parameter Optimization, 2005.

[19] Jasper A. Vrugt, Bruce A. Robinson, and James 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.

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

[21] 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

[22] F. Peng, K. Tang, G. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization", IEEE Transactions on Evolutionary Computation, vol. 14, no. 5, 2010.

[23] 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.

[24] Omidvar, M.N., Li, X. and Yao, X. (2010), "Cooperative Co-evolution with Delta Grouping for Large Scale Non-separable Function Optimization", in Proceedings of Congress of Evolutionary Computation (CEC 2010), IEEE, p.1762 - 1769."

[25] Wenxiang Chen, Thomas Weise, Zhenyu Yang, Ke Tang: Large-Scale Global Optimization Using Cooperative Coevolution with Variable Interaction Learning. PPSN (2) 2010: 300-309"