Finding Informative Collaborators for Cooperative Co-evolutionary Algorithms Using a Dynamic Multi-population Framework

Xingguang Peng* and Zhe Shi
College of Marine Engineering
Northwestern Polytechnical University, 127 Youyixi Road Xi'an China, 710072
*E-mail: xgpeng.nwpu@gmail.com

***Abstract*—Cooperative co-evolutionary algorithms (CCEAs) conduct high-efficiency problem solving by decomposing a given problem into a number of separate subcomponents, which terms the divide-and-conquer manner. In this paper, the dynamic multi-population framework was incorporated into the CCEAs to continuously search multiple optima of the subcomponents, so as to compensate the lost information induced by problem decomposition and enhance the global optimization capability. These optima are seen as the informative collaborators that can feature the landscapes of the subcomponents. Thus, more accurate fitness evaluation could be conducted by mixing those collaborators. To verify this idea, two dynamic multi- population optimizers were implemented, which results in two dynamic multi-population based CCEAs. Experimental study was carried out on a wide range of benchmark functions. The proposed algorithms was compared with four peer algorithms to verify the effectiveness.**

I. Introduction

In the field of evolutionary computation the concept of co-evolution had been used to propose coevolutionary algorithms for optimization and machine learning.

The CCEA was firstly proposed by Potter and De Jong [1], [2]. The divide-and-conquer evolution scheme was introduced to decompose a complex problem into several relatively simpler subproblems and optimize them separately. This make CCEAs have the potential of high efficiency for some complex optimization problems, such as sensor networks [3], planning [4] and large-scale optimization [5], [6].

However, the divide-and-conquer scheme is also a double-edged sword. A great deal of landscape information may be lost during the decomposition procedure. Without proper landscape information, coevolutionary individuals will be incorrectly assessed and converge to a Nash equilibrium rather than an optimum [7]. It has been theoretically proved that if the information provided by the collaborators is insufficient poor fitness estimation will be inevitable, which could result in some inherent problems that are harmful to global optimization [8]. Relative overgeneralization (RO) may be the most typical inherent problem for CCEAs. In a RO-featured problem, a CCEAs coevolutionary populations are very likely to converge to the larger basin area (rather than a area that the global optimum locates in) where there are many solutions that perform well.

In addition, exchanging more collaborators doesn't mean exchange more collaborative information. In [9] an extreme variant of conventional CCEA which exchanges all individuals between subcomponents still poorly finds the global optimum. Drawn from these theorems and findings the way to develop new CCEAs for global optimization could be inferred: finding informative collaborators to compensate information.

In order to search the informative collaborators, one must consider the dynamic nature of the subcomponents' landscapes. This is because that the assessment of a given individual depends on the collaborators provided by the collaborative subcomponents and those collaborators may change over time during the coevolution. Although this phenomenon has been pointed out in a early fundamental work [7], little work in the literature deals with it. If the dynamics of the landscapes was taken into account when designing a CCEA, collaborators with better informative information could be obtained to enhance the resulting CCEA' performance.

Bearing this idea and motivation in mind, a dynamic multi-population framework is incorporated into the CCEA in this paper to continuously find informative collaborators so as to compensate information. In particularly, two dynamic multi-population approaches are incorporated into the conventional CCEA respectively. In the resultant algorithms multiple optima (local or global) of a given subcomponent are dynamically discovered and maintained. These optima are recognized as informative collaborators that can feature the landscapes of subcomponents. Thus the information compensation is conducted by exchanging multiple optima between subcomponents.

This work is based on our previous work [10] where the idea that finding representatives of subproblems using dynamic evolutionary algorithms was proposed. In this paper, we further verify the proposed idea by designing a cluster-based particle swarm optimization (PSO) algorithm and implementing it as another realization of the dynamic multi-population approach.

II. The Proposed Algorithm

A dynamic multi-population based CCEA (mCCEA) is proposed in this section. Assume that a given problem is decomposed into N subcomponents, each of which is optimized by a separate subpopulation (co-evolutionary population). The flowchart of the mCCEA for a certain subcomponent is shown in Fig. 1. The subpopulations coevolve in a divide-and-conquer manner by exchanging (ending and receiving) collaborators with each other. That means each subpopulation plays not only as a collaborator sender but also collaborator receiver. As shown in Fig. 1, in the context of sending collaborators, the informative collaborators of a subcomponent are provided to its counterparts (the other subcomponents). To find informative collaborators, a multi-population optimizer (see Section II-A) is used instead of single-population evolutionary algorithms. Several child populations are continuously maintained to simultaneously search current local or global optima which are recognized as the informative collaborators and sent to the other subcomponents occasionally.

In the context of receiving collaborators, the collaborators provided by the other subcomponents are stored in a archive. When evaluating a certain individual, it is mixing with the collaborators to construct a set of complete solutions whose number equals to the number of collaborators. Besides, the historic best individual of each child population is maintained in terms of complete solution whose solutions context is also used as an informative collaborator. That means each historic best individual is maintained together with its best collaborator when achieving up-to-date best fitness value. The collaborative solution context of a historic best individual is also used evaluate the given individual. Given a set of collaborators (received collaborators and collaborative solution context of the historic best individual), *best-of-N* strategy is used to estimate the fitness of the given individual, i.e. the fitness value of the best mixed complete solutions is assigned to the given individual.

In summary, the proposed mCCEA evolves each subcomponent separately with a dynamic multi-population optimizer. It is more practical for real-world problems since no centralized information and randomly-selected collaborators (which may lead to a large amount of fitness evaluation) is needed. Moreover, due to finding and exchanging informative collaborators information compensation could be achieved to prevent CCEAs from inherent problems.

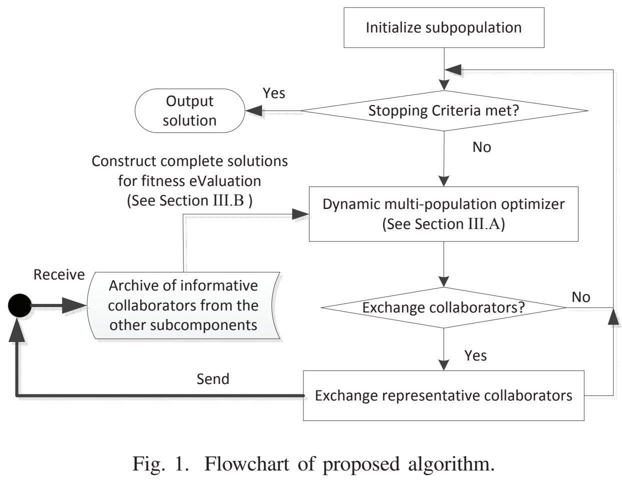

Fig. 1. Flowchart of proposed algorithm.

A. *Dynamic multi-population optimizers*

The key technology of dynamic multi-population approaches is to continuously maintain several child populations to discover and track the moving optima in dynamic landscapes. In general, there are two main categories of multi-population maintaining methods. The first one is to maintain child populations through some explicitly technics according to the searching radius of child populations, such as splitting and merging procedures in the Self-Organizing Scout (SOS) algorithm [11]. The second category employs implicit technics to continuously generate child populations. A number of cluster-based dynamic evolutionary algorithms have been proposed [12]–[14]. The individuals are allocated to different child populations via clustering methods rather than searching radius based technics.

To obtain a generic verification of the proposed algorithm framework (shown in Fig. 1), in this section we will implement two dynamic multi-population optimizers: a modified-SOS and a cluster-based PSO.

1) The modified SOS: In our previous work [10] the SOS was slightly modified and incorporated into the framework of CCEAs. Besides, in each child population a local search technic was employed to track the corresponding optimum. In this paper, we use this algorithm as an implementation of explicit multi-population maintaining method. For saving the space, here we just show the brief flowchart of the modified SOS. Please see the original work for more details.

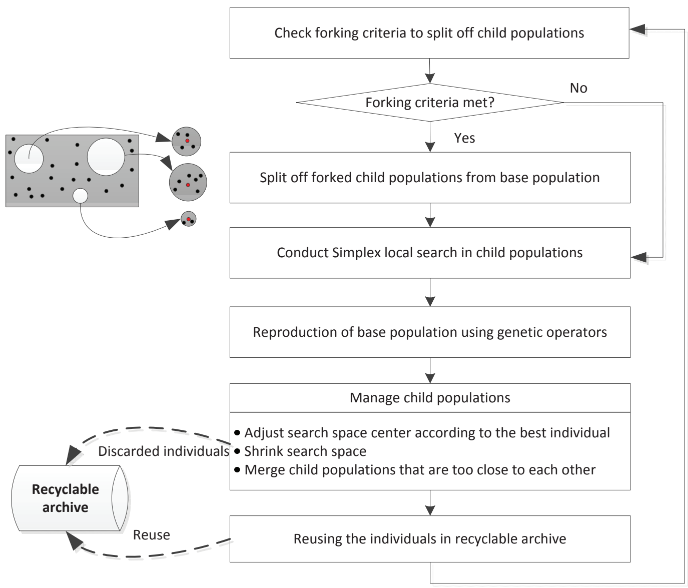

Fig. 2. Brief flowchart of the modified SOS.

As seen in Fig. 2, forking criteria are checked at every generation to determine whether a new child population should be split off from the base population. If so, the corresponding individuals are removed from the base population into a new child population. In the reproduction procedure, all child populations generate their offsprings using the Simplex local search while the base population conducts genetic operators (crossover and mutation) to generate its offsprings. To make the child populations track the local or global optima separately, a management procedure is conducted according to their offsprings. This procedure includes adjusting search space and merging two child populations if any searching center locates in the search area of the other one. Note that, in the search space adjustment and population merging procedure some individuals may be discarded and removed to a recyclable archive. Those individuals are then re-initialized and added to the based population in the next cycle.

2) The cluster-based PSO: In [12] the clustering technic was firstly incorporated into dynamic EAs to continuously maintain several child populations. In that work, a single linkage hierarchical clustering method was proposed to cluster particles into several child populations without preset the number of child populations. Here we borrow this clustering method to generate multiple child populations. Please see the original paper for more details about this clustering method.

The flowchart of the cluster-based PSO is shown in Fig. 3. At the beginning, the randomly initialized particles in a single swarm are clustered into several child swarms. Then each child swarm conducts an independent PSO procedure, the particles of each child swarm will gradually converge to the center (best solution). During this process the historic best solution BI of each child swarm is recorded. If the evolution process is not stuck, the particles in all child swarms will be collected and clustered again at every G_c generations. By doing this, some close child swarms may be merged into a new child swarm. If the evolution process is recognized to be stuck, re-initialization will be conducted. All historic best particles, i.e. $\{BI\}$, are kept in the new single swarm and the remaining particles are re-initialized. Then such single swarm is clustered into several child swarms and the above procedure will be conducted again. Note that, in this paper, the evolution process is recognized to be stuck when the fitness of the best element in BI has not changed for NG_s generations.

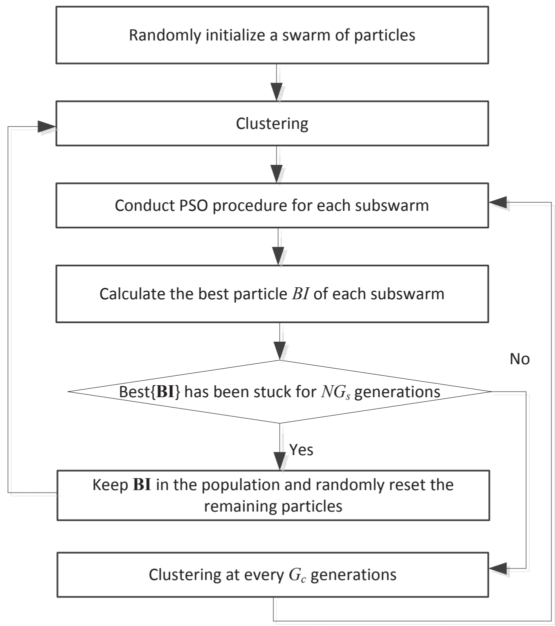

Fig. 3. Flowchart of the cluster-based PSO.

B. Construct complete solutions

Assume that an m-dimensional problem is decomposed into N subcomponents. Every subcomponent is evolved by a co-evolutionary population in a *divide-and-conquer* manner. The ith co-evolutionary is consist of Nc_i child populations $P(j)$, $j = 1, \ldots, Nc_i$. The historic best individual $BI_i(j)$ ($dim(BI_i(j)) = m$) is maintained for $P(j)$ during the run. To conduct co-evolution, each subcomponent provides $Ncom$ informative collaborators.

Algorithm 1 shows how to construct complete solutions in the mCCEA. Two information sources are collected to construct complete solution: 1) the informative collaborators provided by the other subcomponent; 2) historic best individual (together with the whole solution context) of each child population of a certain subcomponent. At every generation, a set of complete solutions are constructed by mixing the collaborative solution context of the collaborators with the corresponding solution context of a certain individual that is under evaluation. Then the *best-of-N* strategy is used to estimate the fitness of the given individual, i.e. the fitness value of the best mixed complete solutions is assigned to the given individual. Note that the historic best individuals are persistently updated during the run after evaluating the child populations.

Algorithm 1 The pseudo code of constructing complete solutions for ith subcomponent.

Shuffle collaborators in each $\{Colb_k\}$, $k \in N, k \neq i$;
for $l = 1, \ldots, Ncom$ **do**
  Construct collaborative solutions $S_i(l) = \{Colb_k(l)\}$, $k \in N, k \neq i$;
end for
for $j = 0, \ldots, Nc_i$ **do**
  Add historic best individual $BI_i(j)$ to $S_i(l)$, i.e. $S_i(Ncom + 1) = Genotype(BI_i(j), g)$, $g \in m$ but gth decision value $\notin$ ith subcomponent;
  for Each individual I in $P(j)$ **do**
    Complete solution $CS(l)$=$\{S(l) \cup I\}$;
    Fitness(I)=*best* $\{F(CS_i(l))\}$, $l \in Ncom + 1$;
    $l_{best} = l$ s.t. $F(CS_i(l))$ is the best;
    if *Fitness*(I) is *better* than *Fitness*($BI_i(j)$) **then**
      $BI_i(j)$=$CS(l_{best})$;
    end if
  end for
end for

III. Experimental Study

Two mCCEA variants are realized under the framework of mCCEA and compared with four peer CCEAs from the literature. The first variant, i.e. mCCEA-1, employs the modified

SOS as the multi-population optimizer while the second one, termed mCCEA-2, employs the cluster-based PSO. A wide range of experiments are carried out on RO-featured problems and ten non-separable continuous test functions.

A. Benchmark Problems

There are two set of test functions. One is RO-featured problems and the other one is selected from the benchmark suite of "learning-based real-parameter single objective optimization" competition in CEC' 2015 .

1) RO-featured problems: As for RO-featured problems, we use the Maximum of Two Quadratics (MTQ) [7] problem domain which has been used to analyze the global optimization capability of CCEAs in a number of previous works. This class of problems include a local suboptimum with a much wider landscape basin which is very likely to misguide the traditional CCEAs.

There are a set of controlling parameters in the MTQ function to feature the landscape of the problem domain. In this paper, we use the default values to set most of the parameters except H_2 which is used to feature the hight of the global optimal 'peak'. By assigning different H_2 values problem difficulty could be controlled (the smaller the H_2 is the more difficult the resultant problem is).

2) Non-separable continuous benchmark test functions: To further verify the performance of mCCEA-1 and mCCEA-2, we introduce ten benchmark test functions for 'learning-based real-parameter single objective optimization' competition [15] in CEC' 2015. Then 2D test functions ($F_1 \sim F_5$, F_9, $F_{11} \sim F_{12}$, and $F_{14} \sim F15$) are selected from the original benchmark suite and renamed as $F_1 \sim F10$ respectively in this paper.

B. Algorithms for Comparison

The following typical CC algorithms are used as peer algorithms to compare with the MMO-CC.

1) Traditional CCEA (tCCEA) [16]: tCCEA conducts *best-of-N fitness* evaluation with 4 randomly chosen and the current best individuals of the other subcomponent.

2) Biased CCEA (bCCEA) [17]: The fitness value of a given individual is partly biased according to the fitness value obtained like tCCEA. The remaining part is based on collaborating with the historical best collaborator. A an algorithmic parameter δ is used to control the biasing rate.

3) Complete CCEA (cCCEA): It is an extreme variant of the tCCEA. An individual has to access the whole population to conduct *best-of-N* fitness evaluation.

4) Cooperative Coevolutionary Differential Evolution (CCDE): It employs SaNSDE [18] as the subcomponent optimizer. It has been used as the optimizer of several successful CC algorithms on large-scale optimization problems.

C. General Experimental Parameter Settings

GA toolbox [19] with the default settings is used to implement the mCCEA-1, tCCEA, bCCEA and cCCEA. In mCCEA-2 *standardPSO*[1] is used with default settings and $NG_s = 10$, $G_c = 5$. In all compared algorithms, the size of each subpopulation is 50. Each algorithm terminates when the number of generation exceeds 1000 and the performance is obtained according to 50 independent runs.

D. Comparison on MTQ problems

H_2 is set to 70, 150 and 300 respectively to obtain three MTQ problems. Table I shows the results of convergence rates, statistical comparison and average number of fitness evaluations. The statistical comparison is conducted according to Wilcoxon rank sum test. Note that an algorithm is recognized to successfully converge to the global optimum when the fitness value of its output solution differs from that of the global optimum within a small value of 0.1.

As can be seen in Table I, according to the convergence ratios and significantly statistical comparison, CCEA-1 or mCCEA-2 shows better performance than that of the compared algorithms in all test cases. This verifies that the dynamic multi-population optimizer works effectively under the CCEA framework. The relative high fitness values and good diversity of the identified optima endow the informative collaborators with high-quality representative capability of the corresponding subcomponent. Therefore, benefit by dynamically finding and exchanging informative collaborators, mCCEA-1 and mCCEA-2 can achieve a better cooperative co-evolution.

TABLE I
COMPARISON RESULTS ON 3 MTQ PROBLEMS WITH DIFFERENT DIFFICULTY. "S+" DENOTES SIGNIFICANT BETTER.

H_2	Alg.	Rate	mCCEA-1 / mCCEA-2 vs.
70	mCCEA-1	98%	N/A
	mCCEA-2	90%	N/A
	bCCEA	72%	s+ / s+
	cCCEA	64%	s+ / s+
	tCCEA	2%	s+ / s+
	CCDE	22%	s+ / s+
150	mCCEA-1	98%	N/A
	mCCEA-2	98%	N/A
	bCCEA	78%	s+ / s+
	cCCEA	52%	s+ / s+
	tCCEA	8%	s+ / s+
	CCDE	26%	s+ / s+
300	mCCEA-1	100%	N/A
	mCCEA-2	100%	N/A
	bCCEA	98%	s+ / s+
	cCCEA	64%	s+ / s+
	tCCEA	4%	s+ / s+
	CCDE	40%	s+ / s+

E. Comparison on ten non-separable problems

To save the space the cCCEA, as an an extreme varint of tCCEA, is not considered in the following experiments. Each algorithm terminates when the number of generations exceeds 200 and the performance is obtained according to 50 independent runs. More particularly, the values of best,

[1]https://www.researchgate.net/publication/259643342_Source_code_for_an_implementation_of_Standard_Particle_Swarm_Optimization_revised

worst, median, mean and standard deviation of each algorithm are given together with two statistical comparisons (Wilcoxon signed rank test and average Friedman ranks) are given in Table II.

As for the Wilcoxon signed rank test results, the mCCEA-1's performance is significantly better than that of the tCCEA, bCCEA and CCDE on all test functions. mCCEA-2 also shows very competitive performance on most of the functions. As for the average Friedman ranks results, mCCEA-1 wins the first place on all test functions. The mCCEA-2' performance is also very competitive since it ranks the second place on 80% of the test functions.

IV. Conclusions

The global optimization performance of conventional CCEAs may be dramatically affected by the quality of the collaborators when the problem decomposition error is present, some inherent problems like the RO may lead to sub-optimization. To address this, we extend our previous work to incorporate a dynamic multi-population framework as the optimizer of the subcomponents. Such optimizers dynamically locate and track global and local optima which are seen as informative collaborators and exchanged between subcomponents for the purpose of information compensation.

To obtain a generic verification of the thought, two dynamic multi-population approaches (modified SOS and a simple cluster-based PSO) have been implemented. The resultant algorithms named mCCEA-1 and mCCEA-2 have been compared with four other peer CCEAs on MTQ problems and ten selected test functions in the benchmark suite for "learning-based real-parameter single objective optimization" competition in CEC' 2015. Statistical results obtained via Friedman ranks and Wilcoxon rank sum test show the effectiveness of the mCCEA variants: the mCCEA-1 outperforms the peer algorithms in all situations and the mCCEA-2 also shows competitive performance on most of the test functions.

Acknowledgment

This work has been supported by NSFC under Grant No. 61473233 and Fundamental Research Funds for the Central Universities of China under Grant No. 3102016ZY007.

References

[1] M. A. Potter, "The design and analysis of a computational model of cooperative coevolution," Ph.D. dissertation, 1997.

[2] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.

[3] A. S. Ruela, L. L. Aquino, and F. G. Guimares, "A cooperative coevolutionary algorithm for the design of wireless sensor networks," in *13th annual conference companion on Genetic and evolutionary computation*. ACM, 2011, pp. 607–614.

[4] M. Colby, M. Knudson, and K. Tumer, "Multiagent flight control in dynamic environments with cooperative coevolutionary algorithms," in *In Association for the Advancement of Artificial Intelligence, Modeling in Human Machine Systems: Challenges for Formal Verification 2014*. AAAI, 2014.

[5] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.

[6] Y. Mei, M. N. OMIDVAR, X. LI, and X. YAO, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Transactions on Mathematical Software*, vol. 42, no. 2, pp. 13:1–24, 2016.

[7] R. P. Wiegand, "An analysis of cooperative coevolutionary algorithms," Ph.D. dissertation, 2003.

[8] L. Panait, "Theoretical convergence guarantees for cooperative coevolutionary algorithms," *Evolutionary Computation*, vol. 18, no. 4, pp. 581–615, 2010.

[9] L. Panait, S. Luke, and J. F. Harrison, "Archive-based cooperative coevolutionary algorithms," in *2006 Genetic and Evolutionary Computation Conference*, 2006, pp. 345–352.

[10] X. Peng, K. Liu, and Y. Jin, "A dynamic optimization approach to the design of cooperative co-evolutionary algorithms," *Knowledge-Based Systems*, vol. 109, pp. 174–186, 2016.

[11] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Pub, 2002.

[12] S. Yang and C. Li, "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments," *IEEE Transaction on Evolutionary Computation*, vol. 14, no. 6, pp. 959–974, 2010.

[13] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang, "An adaptive multipopulation framework for locating and tracking multiple optima," *IEEE Transaction on Evolutionary Computation*, vol. 20, no. 4, pp. 590–605, 2015.

[14] U. Halder, S. Das, and D. Maity, "A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 881–897, 2013.

[15] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization," Computational Intelligence Laboratory, Zhengzhou University and Nanyang Technological University, Tech. Rep. 201411A, 2014.

[16] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Parallel problem solving from naturePPSN III*. Springer, 1994, pp. 249–257.

[17] L. Panait, S. Luke, and R. P. Wiegand, "Biasing coevolutionary search for optimal multiagent behaviors," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 629–645, 2006.

[18] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *IEEE Congress on Evolution Computation*. IEEE, 2008, pp. 1110–1116.

[19] K. Sastry, "Single and multiobjective genetic algorithm toolbox in c++," IlliGAL, Tech. Rep., 2007.

TABLE II

STATISTICAL PERFORMANCE COMPARISON ON MINIMIZATION PROBLEMS $F_1 \sim F_{10}$ OVER 50 RUNS. THE VALUE OF THE GLOBAL OPTIMUM OF EACH FUNCTION IS GIVEN UNDER THE FUNCTION NUMBER. SIGNIFICANCE LEVEL FOR PAIR-WISE WILCOXON SIGNED RANKS TEST IS 0.05. S+, S- AND $\sim$ MEANS SIGNIFICANT BETTER, SIGNIFICANT WORSE AND INSIGNIFICANT DIFFERENT RESPECTIVELY. A SMALLER AVERAGED FRIEDMAN RANKS MEANS A BETTER PERFORMANCE.

Function	Algorithm	Best	worst	Median	Mean	Std	mCCEA-1 / mCCEA-2 vs. (Wilconxon)	Friedman (mean ranks)
	mCCEA-1	100.00138	740.61482	169.81371	230.83397	160.62826	N/A / $\sim$	1.74
	mCCEA-2	100.356	368.813	135.874	167.23	67.7221	$\sim$ / N/A	1.6
F1	bCCEA	103.96952	912811000	5389.92245	43562100	155480000	s+ / s+	4.36
(100)	tCCEA	215.56448	191665000	3298.75112	3888020	27099800	s+ / s+	4.14
	CCDE	101.3109	10391.47	922.68505	1772.84031	2137.56023	s+ / s+	3.16
	mCCEA-1	200.00193	811.41426	242.36952	293.76796	136.60789	N/A / $\sim$	1.78
	mCCEA-2	200.156	600.204	228.723	247.815	66.1697	$\sim$ / N/A	1.54
F2	bCCEA	239.38716	2280030000	5440.25574	145833000	476101000	s+ / s+	4.44
(200)	tCCEA	212.27575	19181600	2499.05091	425506.475	2719040	s+ / s+	4
	CCDE	201.7248	10568.17	688.79575	1883.25167	2433.23969	s+ / s+	3.24
	mCCEA-1	300.00019	300.01825	300.00303	300.00416	0.00386	N/A / s-	1
	mCCEA-2	300.014	302.694	300.086	300.428	0.6025	s+ / N/A	2.74
F3	bCCEA	300.06996	304.04046	302.10599	302.02212	0.96074	s+ / s+	4.52
(300)	tCCEA	300.06883	304.52032	301.09079	301.48692	1.14052	s+ / s+	4.02
	CCDE	300.0182	301.5586	300.2489	300.33848	0.31951	s+ / $\sim$	2.72
	mCCEA-1	400	400.00002	400	400	5.1587E-06	N/A / s-	1.09
	mCCEA-2	400	401.002	400.002	400.239	0.3957	s+ / N/A	2.39
F4	bCCEA	400.00273	426.19617	401.00023	401.31581	3.64282	s+ / s+	3.72
(400)	tCCEA	400.00029	430.81825	401.06866	404.05262	7.74149	s+ / s+	4.44
	CCDE	400.0023	401.0743	400.99535	400.57163	0.49534	s+ / s+	3.36
	mCCEA-1	500	500.31236	500.00004	500.06249	0.12613	N/A / s-	1.14
	mCCEA-2	500	504.511	500.315	500.508	0.6524	s+ / N/A	2.64
F5	bCCEA	500.03048	932.50045	501.11187	514.98612	64.69657	s+ / s+	3.92
(500)	tCCEA	500.05101	1586.87763	503.01843	675.82202	332.30734	s+ / s+	4.26
	CCDE	500.0019	517.9151	500.40395	502.14202	5.12343	s+ / $\sim$	3.04
	mCCEA-1	900.00016	1000.01676	900.00254	903.00602	15.67502	N/A / s-	1.2
	mCCEA-2	900.023	908.063	900.0155	900.5718	1.2146	s+ / N/A	2.46
F6	bCCEA	900.02175	1001.04911	902.58654	933.67696	46.25903	s+ / s+	3.9
(900)	tCCEA	900.05655	1608.67222	1000.07495	980.4748	123.46551	s+ / s+	4.26
	CCDE	900.0077	907.3563	900.9119	901.22338	1.33708	s+	3.18
	mCCEA-1	1100.02128	1109.95425	1100.1897	1100.4379	1.38351	N/A / s-	1.1
	mCCEA-2	1100.59	1111.88	1100.8	1103.42	2.5251	s+ / N/A	2.64
F7	bCCEA	1101.58548	1208.00622	1108.56948	1110.83037	15.47258	s+ / s+	3.94
(1100)	tCCEA	1102.06202	1199.35029	1108.45602	1113.12269	18.68598	s+ / s+	4.24
	CCDE	1101.185	1160.954	1104.0545	1105.66846	8.38916	s+ / s+	3.08
	mCCEA-1	1200.00552	1308.64465	1200.03737	1216.3004	37.65394	N/A / s-	1.44
	mCCEA-2	1200.03	1246.62	1200.4	1202.7	6.48024	s+ / N/A	2.86
F8	bCCEA	1201.06302	1853.06193	1352.44552	1333.77416	97.96627	s+ / s+	4.86
(1200)	tCCEA	1200.28921	1323.37231	1208.17407	1244.42548	50.10516	s+ / s+	3.84
	CCDE	1200.072	1201.923	1200.472	1200.51794	0.34991	s+ / s-	2
	mCCEA-1	1400.47889	1401.47327	1401.05814	1401.04564	0.29018	N/A / s-	1
	mCCEA-2	1403.85	1450.65	1403.4	1414.58	10.5635	s+ / N/A	2.76
F9	bCCEA	1406.0355	7796.52367	1428.55908	1589.63281	897.95043	s+ / s+	3.74
(1400)	tCCEA	1402.75172	14191.9005	1538.0799	2959.40782	3066.01293	s+ / s+	4.26
	CCDE	1402.413	1711.535	1414.7575	1470.32188	84.29948	s+ / s+	3.24
	mCCEA-1	1500.04472	1600.02181	1600.00479	1591.19934	27.72482	N/A / s-	1.22
	mCCEA-2	1509.7	1600.3	1600.1	1594.12	17.7863	s+ / N/A	1.78
F10	bCCEA	1600.05782	1602.05421	1600.48044	1600.61062	0.42575	s+ / s+	3.7
(1500)	tCCEA	1600.10496	4335.61778	1600.53231	1661.34023	387.92735	s+ / s+	3.9
	CCDE	1600.199	1603.054	1600.943	1601.08556	0.64886	s+ / s+	4.4