Title	Continuous non-revisiting genetic algorithm with overlapped search sub-region
Author(s)	Chow, CK; Yuen, SY
Citation	The 2012 IEEE Congress on Evolutionary Computation (CEC 2012), Brisbane, QLD., Australia, 10-15 June 2012. In IEEE Transactions on Evolutionary Computation, 2012, p. 1-8
Issued Date	2012
URL	http://hdl.handle.net/10722/196653
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Continuous Non-revisiting Genetic Algorithm with Overlapped Search Sub-Region

Chi Kin Chow
Department of Electronic Engineering
City University of Hong Kong
Hong Kong SAR
E-mail: chowchi@cityu.edu.hk

Shiu Yin Yuen
Department of Electronic Engineering
City University of Hong Kong
Hong Kong SAR
E-mail: kelviny.ee@cityu.edu.hk

***Abstract*— In continuous non-revisiting genetic algorithm (cNrGA), search space is partitioned into sub-regions according to the distribution of evaluated solutions. The partitioned sub-region serves as mutation range such that the corresponding mutation is adaptive and parameter-less. As pointed out by Chow and Yuen, the boundary condition of the mutation in cNrGA is too restricted that the exploitative power of cNrGA is reduced. In this paper, we tackle this structural problem of cNrGA by a new formulation of mutation range. When sub-region is formulated as which certain overlap exists between adjacent sub-regions, this creates a soft boundary and it allows individual move from a sub-region to another with better fitness. This modified cNrGA is named cNrGA with overlapped search sub-region (cNrGA/OL/OGF). By comparing with another work on this problem, Continuous non-revisiting genetic algorithm with randomly re-partitioned BSP tree (cNrGA/RP/OGF), it has an advantage on processing speed. The proposed algorithm is examined on 34 benchmark functions at dimensions ranging from 2 to 40. The results show that the proposed algorithm is superior to the original cNrGA, cNrGA/RP/OGF and covariance matrix adaptation evolutionary strategy (CMA-ES).**

Keywords: continuous non-revisiting genetic algorithm; one-gene-flip mutation; search space re-partitioning; overlapped search sub-region

I. INTRODUCTION

Continuous Non-revisiting Genetic Algorithm (cNrGA) [1] uses a binary space partitioning (BSP) tree archive to record the positions of evaluated solutions. Each node of the tree represents a sub-region in a search space.

Definition 1: The sub-region of **x**

Suppose **x** is a solution in the search space S, i.e. $\mathbf{x} \in S$, and S is partitioned into the sub-region set $H = \cup_i h_i$ by a BSP tree T, we define the sub-region $h \subseteq H$ as the '*sub-region of* **x**' if $\mathbf{x} \in h$ and h is represented by a leaf node of T.

Suppose a parent node **p** has two child nodes **a** and **b**. The child nodes linearly partition the sub-region of **p** into two sub-regions. The corresponding partitioning cuts along the jth dimension where $j = arg\ max|\mathbf{a}(j) - \mathbf{b}(j)|$ at the *decision threshold* δ. In [1], the threshold is chosen to be the mid-point of $\mathbf{a}(j)$ and $\mathbf{b}(j)$.

Fig. 1 shows an example of the space partitioning scheme of cNrGA. Suppose the squares shown in Fig. 1(a) and 1(b) represent the same sub-region of parent node **p**. The letters '**a**' and '**b**' indicate the positions of two evaluated solutions **a** = [0.25, 0.5] and **b** = [0.75, 0.5]. Since **a** and **b** have maximum distance along x_1 axis, the partitioning cut along x_1 axis and δ = 0.5. The gray-filled regions in Fig. 1(a) and Fig. 1(b) represent the sub-regions of **a** and **b** respectively. They are disjoint and their union is the sub-region of **p** (i.e., the child nodes binary divide the parent sub-region). In cNrGA, the sub-region of an individual **x** serves as the range of the possible mutants of **x**. The sub-region (i.e. mutation) size is small if **x** is close to evaluated solution and vice versa. As a result, it establishes a parameter-less and adaptive mutation operator.

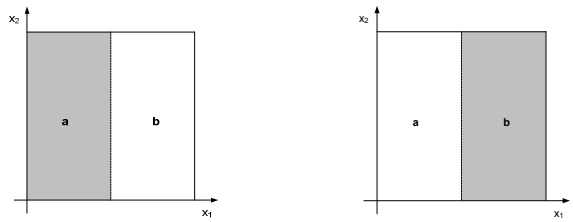

Figure 1. Example of subregions in cNrGA: (a) the grey-filled region represents the subregion of **a** and (b) the grey-filled region represents the subregion of **b**.

In [2], Chow and Yuen pointed out a structural problem of cNrGA: Suppose h is the sub-region of an individual **x**. Since there is no overlap amongst sub-regions, no matter how good (better fitness) the adjacent sub-region of h is, the boundary condition of the mutation in cNrGA restricts which **x** could reach there only by crossover operator. This would reduce the exploitative power of cNrGA. Chow and Yuen in [2] tackled this problem by a modified solution-density estimation. At each iteration, the evaluated solutions are re-shuffled into a random order. Then a new density tree is built from the re-ordered solution sequence. The idea of re-shuffling implements a dynamic-size mutation region. Individual would belong to different sub-regions at different iterations. This creates a soft mutation boundary that individual could cross the soft boundary to a position with better fitness. This modified version of cNrGA is named *cNrGA with randomly re-partitioned density tree* (cNrGA/RP/OGF).

The work described in this paper was supported by a grant from Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 124409].

Though the idea of re-shuffling in [2] significantly enhances the accuracy of the original cNrGA, the improvement is gained by sacrificing its processing speed on re-building the tree at every iteration. In this paper, we solve the mentioned structural problem of cNrGA by an alternate approach: a modified formulation of sub-region. In the new formulation, every sub-region has certain overlap to its adjacent sub-regions. Consequently, it creates soft boundary for mutation in a more efficient way. This enhanced version of cNrGA, is named *cNrGA with overlapped search sub-region* (cNrGA/OL/OGF). It inherits the feature of cNrGA that involves only two parameters: population size and crossover rate. Meanwhile, its parameter-less The proposed algorithm is examined on 34 benchmark test functions at varying dimensions. The experimental results show that cNrGA/OL/OGF is significantly superior to the original cNrGA in all 64 test cases. Moreover, cNrGA/OL/OGF is slightly superior to cNrGA/RP/OGF (it ranks better than cNrGA/RP/OGF in 36 out 64 test cases) and at the same time cNrGA/OL/OGF spends 85% less processing time than cNrGA/RP/PGF to complete an optimization. Moreover, the experimental result also shows that it is also superior to covariance matrix adaptation evolutionary strategy (CMA-ES) [3] in most of the multi-modal test functions.

The rest of this paper is organized as follows: Section II presents the idea of overlapped search sub-region. Section III presents the mechanism of cNrGA/OL/OGF. Section IV reports the experimental results. Section V gives the conclusion.

II. OVERLAPPED SEARCH SUB-REGION

In this section, we present a new formulation of search sub-region for mutation. By simply expanding the sub-region defined in [1] to which adjacent sub-regions have certain overlap to each other, namely *overlapped sub-region*, it creates a path for which individual could move from one sub-region to another with better fitness. Moreover, this modification does not introduce a significant computation load to cNrGA. As a result, the cNrGA that adopts the overlapped sub-region enhances the exploitative power and meanwhile preserves the key feature of cNrGA - adaptive and parameter-less mutation scheme.

The search for the overlapped sub-region h of a solution $\mathbf{x}$ is implemented as a tree node search. The search starts from examining the root node whilst h is initialized as the whole search space. Each time the search moves downwards, h is contracted along a specified direction. It keeps contracting until the search reaches leaf node. Fig. 2 summarizes the procedure to obtain the overlapped sub-region of an individual $\mathbf{x}$. The procedure is similar to that in [1] except for the formulation of δ in step 7 and 10. Rather than the mid-point of $\mathbf{a}(j)$ and $\mathbf{b}(j)$, the value of δ in the overlapped sub-region is contracted to either $\mathbf{a}(j)$ or $\mathbf{b}(j)$.

Input: 1) BSP tree T, 2) solution $\mathbf{z} \in \Re^D$ where D is function dimension and 3) search space S

1. $h = \prod_i [l_i, u_i] := S$
2. **Curr_node** := root node of T
3. **While** (**Curr_node** has two child nodes: left child node **a** and right child node **b**)
4. Comparing dimension j where $|\mathbf{a}(j) - \mathbf{b}(j)| \le |\mathbf{a}(k) - \mathbf{b}(k)|$ for all $k = 1, 2, \ldots, D$
5. **If** $(|\mathbf{a}(j) - \mathbf{z}(j)| \le |\mathbf{b}(j) - \mathbf{z}(j)|)$
6. Curr_node := child node **a**
7. $u_j := \mathbf{b}(j)$
8. **Else**
9. **Curr_node** := child node **b**
10. $l_j := \mathbf{a}(j)$
11. **End**
12. **Loop**

Output: h the overlapped sub-region of $\mathbf{z}$

Figure 2. Psuedo code of overlapped sub-region search.

Facing the same scenario described in Fig. 1, the overlapped sub-regions of **a** and **b**, namely h_a' and h_b', under the proposed search method are defined as the gray-filled regions shown in Fig. 3(a) and Fig. 3(b) respectively. Note that the union of h_a' and h_b' create a channel for which the individual in h_a' could move to h_b' through mutation, and vice versa. Moreover, the sub-region contraction scheme in step 7 and 10 of Fig. 2, guarantees that the resultant sub-region overlaps with all its adjacent sub-regions. Thus, the idea of overlapped sub-region together with mutation allows individuals to gradually approach its optima nearby.

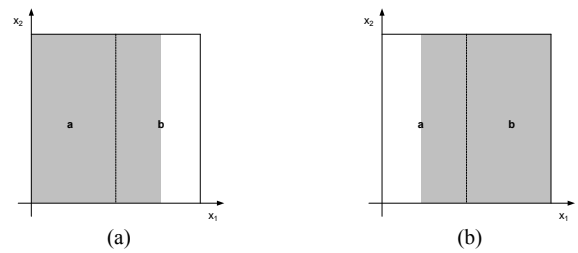

Figure 3. Example of overlapped subregion: (a) the overlapped subregion of **a**; (b) the overlapped subregion of **b**.

Example:

In this example, the BSP tree memorized four two-dimensional evaluated solutions: $\mathbf{z_1}$, $\mathbf{z_2}$, $\mathbf{z_3}$ and $\mathbf{z_4}$. The distribution of the solutions in search space $S = [0,1] \times [0,1]$ is shown in fig. 4(a). The dashed line represents the boundary of the partitioning. Fig. 4(b) shows the corresponding BSP tree. The tree consists of seven nodes: **R**, **A**, **B**, **C**, **D**, **E** and **F**. Node **C** and **D** store the solution $\mathbf{z_2}$ and $\mathbf{z_1}$ whilst node **E** and **F** store the solution $\mathbf{z_4}$ and $\mathbf{z_3}$. To compute the overlapped sub-region h_1 of $\mathbf{z_1}$, the corresponding search starts from the root node **R** (Fig. 5(b)). It is equivalent of initializing h_1 as S, i.e. $h_1 = [0,1] \times [0,1]$ (the gray-filled region in Fig. 5(a)). Afterwards, the search moves to node **A** (Fig. 5(d)) as $z_{1,2}$ is lower than the decision boundary established by $\mathbf{z_1}$ and $\mathbf{z_2}$. h_1 is contracted as the gray filled region shown in Fig. 5(c), i.e. $h_1 = [0,1] \times [0,z_{2,2}]$. The search keeps moving downward to node **D** (Fig. 5(f)) as $z_{1,1}$ is higher than the decision boundary established by $\mathbf{z_1}$ and $\mathbf{z_3}$. The corresponding h_1 is now reduced as the gray filled region shown in Fig. 5(e), i.e. $h_1 = [z_{3,1},1] \times [0,z_{2,2}]$. Since

node D is a leaf node, the search is terminated and the resultant h_1 is $[z_{3,1},1] \times [0,z_{2,2}]$.

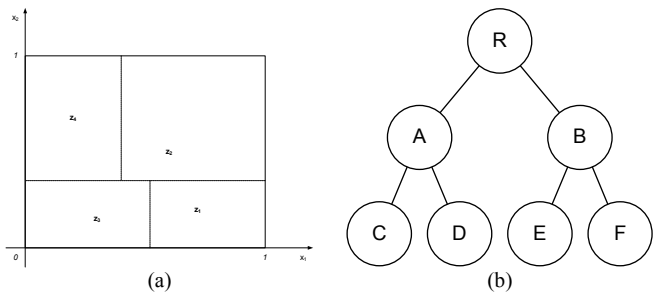

Figure 4. The overlapped subregion of (a) $\mathbf{z_1}$, (b) $\mathbf{z_2}$, (c) $\mathbf{z_3}$ and (d) $\mathbf{z_4}$.

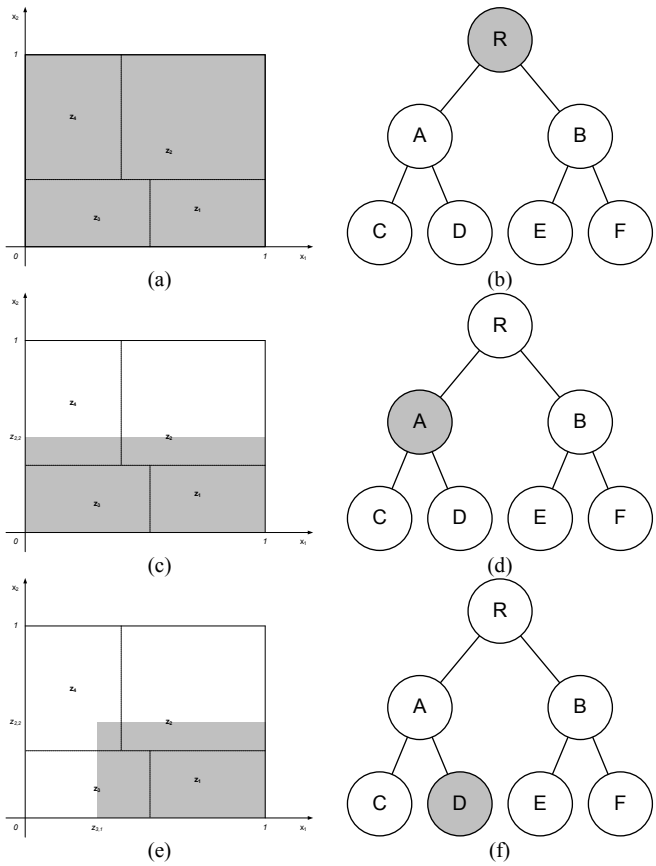

Figure 5. Example of searching the overlapped subregion of $\mathbf{z_1}$.

By repeating the above procedure on $\mathbf{z_2}$ to $\mathbf{z_4}$, the corresponding overlapped sub-regions are determined. They are represented as grey-filled regions in Fig. 6.

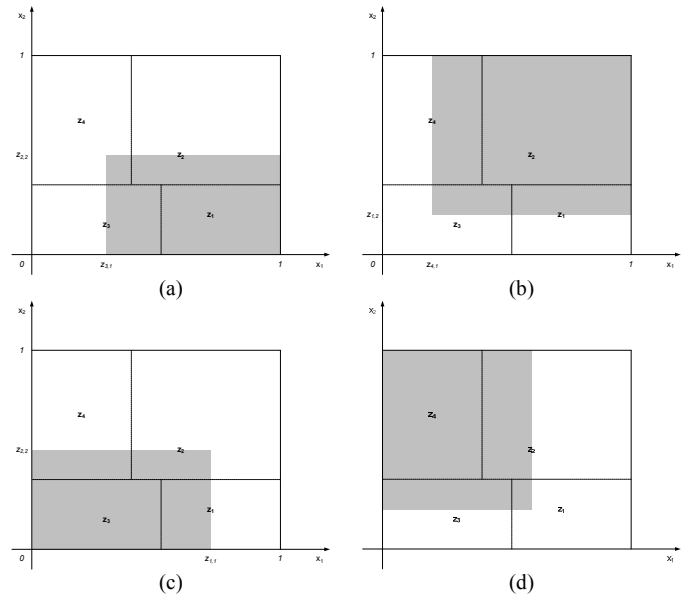

Figure 6. The overlapped subregion of (a) $\mathbf{z_1}$, (b) $\mathbf{z_2}$, (c) $\mathbf{z_3}$ and (d) $\mathbf{z_4}$.

Fig. 7 summarizes the procedure of cNrGA/OL/OGF. Given a D-dimensional minimization problem $F(.)$ with search space $S \subset \Re^D$, the algorithm starts from initializing the current population of μ individuals $\mathbf{P} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_\mu}\}$. Meanwhile, the BSP tree T is initialized to consist of the root node. The population is then evaluated and is recorded by T. Afterwards, for each individual $\mathbf{x_i}$ in $\mathbf{P}$, we generate the corresponding offspring $\mathbf{y_i}$ by the following procedure: given an individual $\mathbf{x_i}$, we randomly select another individual $\mathbf{x_k}$ in $\mathbf{P}$ where $\mathbf{x_k} \neq \mathbf{x_i}$. Uniform crossover with crossover rate γ is performed on $\mathbf{x_i}$ and $\mathbf{x_k}$ to generate $\mathbf{y_i}$. Afterwards, we access the BSP tree to check whether $\mathbf{y_i}$ is a revisit. If $\mathbf{y_i}$ is a revisit, we obtain its overlapped sub-region, and $\mathbf{y_i}$ is replaced by the mutant of itself using One-Gene-Flip (OGF) mutation [2] on that sub-region. Suppose $\mathbf{p}$ is a D-dimensional individual to be mutated and $\prod_{k=1}^{D} [l_k, u_k]$ is the mutation region of $\mathbf{p}$, OGF starts from randomly selecting a dimension $j \in \{1,2,\ldots,D\}$. Then $\mathbf{p}$ is mutated as $\mathbf{p'}$ by replacing the jth element of $\mathbf{p}$ with a random number in the range $[l_j, u_j]$. The values of the genes in the rest of the dimensions are unchanged. After generating the offspring $\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_\mu}$, they would be evaluated and inserted in T. The population $\mathbf{P}$ together with the offspring pool $\{\mathbf{y_i}\}$ is then selected to form the new population. The reproduction and selection processes are repeated until the termination criterion is satisfied.

Input: 1) a D-dimension minimization problem $F(.)$, 2) search space $S \subset \Re^D$ 4) population size μ and 4) crossover rate γ

1. Initialize the current population $\mathbf{P} = \{\mathbf{x_i} \in S\}$ for $i = 1,2, \ldots, \mu$
2. Initialize BSP tree T to which consists of root node only
3. Evaluate $\mathbf{x_i}$: $f_i = F(\mathbf{x_i})$ for all i
4. Record $\{\mathbf{x_i}\}$ to T
5. **While** terminate criteria is not satisfied
6. **For** $i = 1,2,\ldots,\mu$
7. k := Rand($\{1,2,\ldots,\mu\}/i$)
8. $\mathbf{y_i}$:= the offspring of $\mathbf{x_i}$ and $\mathbf{x_k}$ under uniform crossover with crossover rate γ
9. **If** $\mathbf{y_i}$ is a revisit **then**
10. Search the overlapped sub-region of $\mathbf{y_i}$, h_i

```
            y_i := the mutant of y_i under One-Gene-Flip mutation [2]
        EndIf
    Next i
    P := The elitism of (P ∪ {y_i})
Loop
```

Output: the optimal solution $\mathbf{x_o} \in \mathbf{P}$ where o = arg min$\{F(\mathbf{x_i})\}$

Figure 7. Psuedo code of cNrGA/OL/OGF.

III. EXPERIMENTAL RESULTS

A. Test function set

A real valued function set $\mathbf{F} = \{f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_{34}(\mathbf{x})\}$ consisting of 34 functions are employed to illustrate the performance of cNrGA/OL/OGF. The names of the 34 test functions are listed in Table IV. They are well known benchmark test functions. The first 14 functions and the last function are taken from [4]; $f_{15} - f_{17}$ are taken from [5]; $f_{18} - f_{22}$ and f_{33} are taken from [6] whilst $f_{23} - f_{32}$ are taken from [7].

Seven of them are uni-modal functions; the remaining twenty-seven are multi-modal functions designed with a considerable amount of local minima. Additionally, the function f_6 is a noisy function and the function f_{17} is a hybrid composition function. The dimensions of the first ten and the last twenty functions are adjustable while the dimensions of f_{11} - f_{14} are fixed at two, as they are two-dimensional functions as defined in the original references. The optimal points of the functions f_{24}, f_{27}, f_{28}, f_{31}, f_{33} and f_{34} are not known. Simulations are carried out to find the global minimum of each function.

B. Simulation settings

All test functions with the exception of $f_{11} - f_{14}$, which are two-dimensional, are tested with dimension D = 30 and 40. To provide a fair comparison amongst the test algorithms, the total number of function evaluations of all algorithms is kept a constant: The number of fitness evaluations is 40,000 for $f_{15} - f_{34}$. For $f_{11} - f_{14}$. the number of fitness evaluations is 1,000. Since the test algorithms are stochastic, their performance on each test function is evaluated based on statistics obtained from 100 independent runs. All simulations are done on a PC with 3.2GHz CPU and 1GB memory.

C. Experiment 1

In this section, we compare the performance of cNrGA/OL/OGF with those of cNrGA and cNrGA/RP/OGF. The search spaces of all test algorithms are continuous. The design and settings of cNrGA/RP/OGF and the algorithms for comparison are summarized below.

1. Continuous non-revisiting genetic algorithm with overlapped search sub-region (cNrGA/OL/OGF)
2. Continuous non-revisiting genetic algorithm [1] (cNrGA)
3. Continuous non-revisiting genetic algorithm with randomly re-partitioned BSP tree [2] (cNrGA/RP/OGF).

The population size of cNrGA/OL/OGF is chosen as 20. For cNrGA/RP/OGF and cNrGA, the population sizes are set to 100, which are suggested in [1] [2]. The selection scheme of the test algorithms is elitism selection. The crossover rate γ of the test algorithms is chosen as 0.5. This is the recommended setting in [8, p.48]. The test algorithms are implemented in C language.

The detailed simulation results are reported in Table V. Table I presents a summary of the results. The value inside the table cell indicates the rank of the corresponding algorithm on a particular test function. The shaded cells in the table indicate that the corresponding test algorithm is the best algorithm on a particular test function at a particular function dimension.

TABLE I. RANK OF THE TEST ALGORITHMS. THE CELL WITH GREY COLOR REPRESENTS THAT THE CORRESPONDING TEST ALGORITHM OUTPERFORMS THE OTHERS FOR A PARTICULAR FUNCTION AND A PARTICULAR FUNCTION DIMENSION. (A_1 = CNRGA/OL/OGF; A_2 = CNRGA AND A_3 = CNRGA/RP/OGF) IN SECOND.

	D	A_1	A_2	A_3		D	A_1	A_2	A_3
f_1	30	1	3	2	f_{19}	30	3	1	2
	40	1	3	2		40	3	1	2
f_2	30	2	3	1	f_{20}	30	1	3	2
	40	1	3	2		40	1	3	2
f_3	30	3	2	1	f_{21}	30	2	3	1
	40	2	3	1		40	1	3	2
f_4	30	1	3	2	f_{22}	30	1	3	2
	40	1	3	2		40	1	3	2
f_5	30	1	3	2	f_{23}	30	2	3	1
	40	1	3	2		40	2	3	1
f_6	30	1	2	3	f_{24}	30	2	3	1
	40	1	3	2		40	1	3	2
f_7	30	2	3	1	f_{25}	30	1	3	2
	40	1	3	2		40	1	3	2
f_8	30	1	3	2	f_{26}	30	1	3	2
	40	1	3	2		40	1	3	2
f_9	30	3	2	1	f_{27}	30	3	2	1
	40	3	2	1		40	2	1	3
f_{10}	30	2	3	1	f_{28}	30	1	3	2
	40	1	3	2		40	1	3	2
f_{11}	2	3	1	2	f_{29}	30	1	3	2
f_{12}	2	3	1	2		40	1	3	2
f_{13}	2	3	2	1	f_{30}	30	2	3	1
f_{14}	2	3	2	1		40	2	3	1
f_{15}	30	2	3	1	f_{31}	30	2	3	1
	40	1	3	2		40	2	3	1
f_{16}	30	2	3	1	f_{32}	30	3	2	1
	40	1	3	2		40	3	2	1
f_{17}	30	1	3	2	f_{33}	30	1	3	2
	40	1	3	2		40	1	3	2
f_{18}	30	2	3	1	f_{34}	30	1	3	2
	40	2	3	1		40	1	3	2

Seen from Table I, cNrGA/OL/OGF is superior to cNrGA. It performs better than cNrGA in 51 out of 64 test cases. Using t tests, 47 of them are with 99.95% significance; 1 of them is with 99% significance and the remaining one is with 97.5% significance). In addition, the performance improvement by cNrGA/OL/OGF is significant. For some of the test functions, the improvements by cNrGA/RP/OGF are even in the order of 10^3 or higher. For example, in D = 40, the averaged optimal fitness of f_1 found by cNrGA/RPOL/OGF is 1.06e-4 and cNrGA is 2.498376; the averaged optimal fitness found of f_{25} by cNrGA/RPOL/OGF is 0.013276 and cNrGA is 126.5674 (the optimal values of f_1 and f_{25} are 0). Thus these results show that, besides search space random re-partitioning, the idea of overlapped search sub-region is another solution to the structural problem of the original cNrGA.

cNrGA/OL/OGF is slightly superior to cNrGA/RP/OGF. It ranks higher than cNrGA/RP/OGF in 36 out of 64 test cases. For these functions, the order of improvement made by cNrGA/OL/OGF is around 10^2. Table II lists the averaged processing time of cNrGA/OL/OGF, cNrGA and cNrGA/RP/OGF. Seen from the table, as expected, the processing time of cNrGA/OL/OGF is similar to that of cNrGA at all test cases as their procedures have no significant difference. On the other hand, the processing time of cNrGA/OL/OGF is significantly faster than that of cNrGA/RP/OGF. Except for the fitness expensive functions f_{16}, f_{17} and f_{34} of which the process is dominated by the evaluation step, cNrGA/OL/OGF spends only 11% to 16% processing time of cNrGA/RP/OGF to complete a search. For f_{26}, the time fraction is even smaller than 7%. For those fitness expensive functions, the corresponding time fractions are around 30% to 50%. In conclusion, cNrGA/OL/OGF outperforms cNrGA/RP/OGF as it spends less computation time to achieve higher accuracy.

TABLE II. AVERAGED PROCESSING TIME OF A_1, A_2 AND A_3 (A_1 = CNRGA/OL/OGF; A_2 = CNRGA AND A_3 = CNRGA/RP/OGF) IN SECOND.

	D	A_1	A_2	A_3		D	A_1	A_2	A_3
f_1	30	1.232	1.015	8.620	f_{19}	30	0.517	0.494	8.181
	40	1.233	1.193	10.144		40	0.660	0.606	8.971
f_2	30	1.261	0.915	8.337	f_{20}	30	1.134	1.048	8.212
	40	1.180	1.007	9.725		40	1.104	1.200	9.251
f_3	30	0.566	0.684	8.320	f_{21}	30	1.508	1.947	9.736
	40	0.721	0.807	9.099		40	1.877	2.348	11.17
f_4	30	0.506	0.253	8.094	f_{22}	30	1.066	1.381	8.826
	40	0.550	0.271	8.801		40	1.391	1.847	10.23
f_5	30	1.166	1.336	9.546	f_{23}	30	1.376	1.146	9.491
	40	1.465	1.637	11.927		40	1.362	1.387	10.37
f_6	30	1.367	0.513	10.749	f_{24}	30	1.202	1.597	8.799
	40	1.471	0.617	11.315		40	1.554	1.871	10.48
f_7	30	1.158	1.288	8.395	f_{25}	30	1.038	1.109	8.642
	40	1.143	1.533	9.906		40	1.096	1.386	10.29
f_8	30	1.270	1.062	8.683	f_{26}	30	0.569	0.866	8.156
	40	1.311	1.271	10.039		40	0.728	0.991	9.349
f_9	30	1.205	1.110	8.530	f_{27}	30	1.213	1.034	7.864
	40	1.159	1.264	9.437		40	1.280	1.259	8.764
f_{10}	30	1.335	1.172	8.516	f_{28}	30	1.509	1.310	7.494
	40	1.401	1.395	9.774		40	1.546	1.568	7.939
f_{11}	2	0.010	0.024	0.015	f_{29}	30	1.341	1.446	8.835
f_{12}	2	0.002	0.005	0.007		40	1.397	1.736	9.748
f_{13}	2	0.002	0.005	0.006	f_{30}	30	2.674	0.605	10.15
f_{14}	2	0.003	0.005	0.007		30	2.273	0.715	11.81
f_{15}	30	0.978	1.174	8.971	f_{31}	40	1.426	1.187	8.706
	40	1.062	1.431	9.530		30	1.434	1.350	9.418
f_{16}	30	10.057	12.339	17.531	f_{32}	40	1.124	0.834	8.411
	40	12.909	16.721	22.002		30	1.077	0.928	9.804
f_{17}	30	21.924	24.765	29.658	f_{33}	30	1.306	1.525	11.30
	40	28.401	30.442	37.427		40	1.400	1.807	12.03
f_{18}	30	1.494	1.273	8.589	f_{34}	30	8.925	8.791	21.71
	40	1.586	1.559	10.038		40	15.00	14.97	28.76

D. Experiment 2

In this section, we compare cNrGA/OL/OGF with a state-of-the-art algorithm: Covariance Matrix Adaptive Evolutionary Strategy CMA-ES [3]. The population size of CMA-ES is chosen by the suggested setting in [3] (i.e. $4 + \lfloor 3\ln D \rfloor$). CMA-ES uses source code version 2008 in [3] and MATLAB version 6.1. The design of cNrGA/OL/OGF is the same in section III.C. The test function set and the corresponding simulation settings are also the same as listed in section III.A and III.B.

Table III lists the averaged best fitness values found by cNrGA/OL/OGF and CMA-ES. The shaded cells in the table indicate that the corresponding test algorithm is the best algorithm on a particular test function at a particular function dimension. Seen from the table, cNrGA/OL/OGF is superior to CMA-ES in 42 out of 64 test cases. Moreover, the success cases are mainly at the multi-modal functions: For the functions $f_7 - f_{34}$ (totally 52 test cases), cNrGA/OL/OGF ranks the first in 38 test cases. The detailed results of CMA-ES are also listed in Table V.

TABLE III. AVERAGED BEST FITNESS FOUND BY A_1 AND A_4 (A_1 = CNRGA/OL/OGF AND A_4 = CMA-ES). THE CELL WITH GREY COLOR REPRESENTS THAT THE CORRESPONDING TEST ALGORITHM OUTPERFORMS THE OTHERS FOR A PARTICULAR FUNCTION AND A PARTICULAR FUNCTION DIMENSION.

	D	A_1	A_4		D	A_1	A_4
f_1	30	0.00021032	0	f_{19}	30	161.321488	0
	40	0.00010624	0		40	289.68322	0
f_2	30	0.040013	0	f_{20}	30	0.002573	2.272353
	40	0.068428	0		40	0.00549	4.705334
f_3	30	5694.893967	0	f_{21}	30	3.921279	6.14771
	40	13800.26735	0		40	5.315093	8.363205
f_4	30	1.558705	99.884616	f_{22}	30	-28.445517	-5.847431
	40	4.58017	100		40	-38.552231	-6.877219
f_5	30	148.225033	3.117448	f_{23}	30	0.015768	0.387157
	40	201.541994	20.34653		40	0.02735	0.66196
f_6	30	0.065156	0.247021	f_{24}	30	-25.564586	-18.075644
	40	0.089021	0.287768		40	-34.622554	-22.809572
f_7	30	1.113463	68.535963	f_{25}	30	0.036832	0.170638
	40	1.026364	107.054915		40	0.013276	0.497917
f_8	30	0.137748	0.000813	f_{26}	30	5141.582	-4930
	40	0.137569	0.001602		40	37868.696	-11440
f_9	30	-12216.086	-7202.3714	f_{27}	30	-0.000019	-0.000356
	40	-16415.539	-9516.91183		40	-0.000002	0
f_{10}	30	0.111353	18.548102	f_{28}	30	-997866.97	-988418.82
	40	0.055231	19.793933		40	-99993680	-99974828
f_{11}	2	7.601263	12.530378	f_{29}	30	1.000014	1
f_{12}	2	-1.018417	-1.023467		40	1.000061	1
f_{13}	2	0.423587	0.602828	f_{30}	30	1.599873	1.435747
f_{14}	2	13.509367	546.83		40	2.134873	2.815275
f_{15}	30	24306.70848	0	f_{31}	30	-1.229E+34	-2.048E+26
	40	9814.777797	5863.99562		40	-2.583E+45	-5.698E+33
f_{16}	30	0.416943	3.009259	f_{32}	30	-2.31704	-0.071431
	40	0.612812	5.341484		40	-2.229647	-0.013356
f_{17}	30	9972.716769	21651.6983	f_{33}	30	-29.532805	-19.062022
	40	1552.854127	16902.2293		40	-39.501225	-22.816724
f_{18}	30	0.00069	0.009898	f_{34}	30	187.961394	334.198898
	40	0.004625	0.058543		40	627.106437	598.102931

IV. CONCLUSION

Continuous non-revisiting genetic algorithm (cNrGA) [1] uses the entire evaluated solutions to compute mutation range. Though the mutation operator of cNrGA is parameter-less and adaptive, the boundary condition of the mutation weakens the exploitative power of cNrGA. A previous work [2] tackled this problem by randomly re-shuffling the solution order at every iteration. This approach sacrifices processing time to achieve better performance. In this paper, we propose an alternative and simpler approach to tackle this problem. In the proposed approach, each sub-region has certain overlap with its adjacent sub-regions. This creates a path between adjacent sub-regions that individual could move from one sub-region to another one with better fitness. The corresponding cNrGA that uses the idea of overlapped search sub-region is denoted cNrGA/OL/OGF (i.e. cNrGA + Overlapped sub-region + One-gene-flip mutation).

cNrGA/OL/OGF is examined on 34 benchmark test functions at 2, 30 and 40 dimension. 64 test cases are involved in total. In the first part of the experiment, we compare its performance with the original cNrGA and a variant of cNrGA

(cNrGA/RP/OGF). The experimental results show that cNrGA/OL/OGF is significantly superior to the original cNrGA in all 64 test cases. Meanwhile, cNrGA/OL/OGF is slightly superior to cNrGA/RP/OGF (it ranks better than cNrGA/RP/OGF in 36 out of 64 test cases). At the same time, cNrGA/OL/OGF spends 85% less processing time than cNrGA/RP/PGF to complete an optimization. Therefore, it empirically illustrates that the idea of overlapped search sub-region keeps as high an accuracy as 'search space re-partitioning' while preserves as high an efficiency as the original cNrGA.

In the second part of the experiment, we compare cNrGA/OL/OGF with a state-of-the-art algorithm: Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [3]. cNrGA/OL/OGF is superior to CMA-ES in 42 out of 64 test cases; and the success cases are mainly at the multi-modal functions. This is worthy of note as EA are mainly targeted to solve the harder multi-modal problems rather than the easier uni-modal problems.

In short, the idea of overlapped search sub-region uses much simplifier procedure to solve the problem of cNrGA. In view of accuracy, the corresponding algorithm, cNrGA/OL/PGF is superior to a state-of-the-art algorithm. In view of processing time, it spends significantly less than an improved version of cNrGA; meanwhile its accuracy is superior to that of the original cNrGA.

REFERENCES

[1] S. Y. Yuen and C. K. Chow, "Continuous non-revisiting genetic algorithm," in *Proc. IEEE Congr. Evol. Comput., 2009*, pp. 1896 – 1903, 2009.

[2] S. Y. Yuen and C. K. Chow, "Continuous non-revisiting genetic algorithm with random search space repartitioning and one-gene-flip mutation," in *Proc. IEEE Congr. Evol. Comput., 2010*, pp. 1– 8, 2010.

[3] N. Hansen, "The CMA Evolutionary Strategy: A Tutorial", Technical Report, 28 June 2011 [Online]. Available: http://www.lri.fr/~hansen/cmatutorial.pdf

[4] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," *IEEE Trans. Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[5] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," *Technical Report*, Nanyang Technological University, Singapore, *KanGAL Report* #2005005, IIT Kanpur, India, 2005.

[6] V. K. Koumousis, C. P. Katsaras, "A sawtooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," *IEEE Trans. on Evolutionary Computation*, vol. 10, no. 1, pp. 19-28, 2006.

[7] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky, "A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems," *Journal of Global Optimization*, vol. 31, no. 4, pp. 635-672, 2005.

[8] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*, Springer 2003.

TABLE IV. TESTED FUNCTION SET

1. Sphere function	8. Generalized Griewank function	15. High conditioned elliptic function	22. Inverted cosine wave function (Masters)	29. Periodic problem
2. Schwefel's problem 2.22	9. Generalized Schwefel's problem 2.26	16. Weierstrass's function	23. Inverted cosine mixture problem	30. Salomon problem
3. Schwefel's problem 1.2	10. Ackley function	17. Hybrid Composition function	24. Epistatic Michalewicz problem	31. Shubert problem
4. Schwefel's problem 2.21	11. Shekel's Foxholes function	18. Levy function	25. Levy and Montalvo 2 problem	32. Sinusoidal problem
5. Generalized Rosenbrock function	12. Six-Hump Camel-Back function	19. Zakharov function	26. Neumaier 3 problem	33. Michalewicz function
6. Quartic function	13. Branin function	20. Alpine function	27. Odd Square problem	34. Whitely's function
7. Generalized Rastrigin function	14. Goldstein-Price function	21. Pathological function	28. Paviani problem	

TABLE V. AVERAGE λ, STANDARD DEVIATION σ AND CONFIDENCE LEVEL C OF THE BEST FITNESS VALUES FOUND BY CNRGA/OL/OGF, CNRGA, CNRGA/RP/OGF AND CMA-ES

		f_1		f_2		f_3		f_4		f_5	
D		30	40	30	40	30	40	30	40	30	40
cNrGA/OL/OGF	λ	0.00021	0.000106	0.040013	0.068428	5694.894	13800.27	1.558705	4.58017	148.225	201.542
	σ	(0.00128)	(0.000419)	(0.073447)	(0.139908)	(1649.97)	(3008.32)	(0.45764)	(1.590288)	(398.28)	(346.953)
cNrGA	λ	1.318867	2.498376	2.116662	4.864231	4449.206	15137.31	41.29776	52.47628	17151.32	38713.86
	σ	(1.266875)	(2.301997)	(1.206012)	(2.000995)	(1882.65)	(4415.3)	(7.554621)	(6.154092)	(33513.1)	(59449.6)
	C	99.95%	99.95%	99.95%	99.95%	99.95%	99%	99.95%	99.95%	99.95%	99.95%
cNrGA/RP/OGF	λ	0.0016	0.0265	0.0276	0.1821	3682.445	12641.86	4.8974	9.9131	270.6306	500.2206
	σ	(0.0006)	(0.0087)	(0.0061)	(0.0325)	(1264.80)	(3244.59)	(0.8637)	(1.3557)	(467.283)	(466.0214)
	C	99.95%	99.95%	95%	99.95%	99.95%	99%	99.95%	99.95%	97.5%	99.95%
CMA-ES	λ	0	0	0	0	0	0	99.88462	100	3.117448	20.34653
	σ	(0)	(0)	(0)	(0)	(0)	(0)	(0.928884)	(0)	(8.794756)	(15.0540)
	C	90%	99%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%

		f_6		f_7		f_8		f_9		f_{10}	
D		30	40	30	40	30	40	30	40	30	40
cNrGA/OL/OGF	λ	0.065156	0.089021	1.113463	1.026364	0.137748	0.137569	-12216.1	-16415.5	0.111353	0.055231
	σ	(0.160437)	(0.067383)	(1.204938)	(1.261428)	(0.185331)	(0.112326)	(243.864)	(278.425)	(0.278819)	(0.067646)
cNrGA	λ	8.758332	13.21974	28.3382	44.36638	1.898842	3.468228	-12892.1	-16885.4	4.267127	5.804797
	σ	(0.487066)	(0.687212)	(6.720075)	(8.13394)	(1.33541)	(2.122299)	(274.100)	(343.189)	(1.196357)	(1.471469)
	C	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%
cNrGA/RP/OGF	λ	8.89	13.1477	0.4012	4.1583	0.2427	0.9833	-13780.3	-18337.1	0.0693	0.336
	σ	(0.4702)	(0.6021)	(0.2685)	(1.3899)	(0.0868)	(0.0488)	(3.1115)	(12.9207)	(0.0146)	(0.066)
	C	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	90%	99.95%
CMA-ES	λ	0.247021	0.287768	68.53596	107.0549	0.000813	0.001602	-7202.37	-9516.91	18.5481	19.79393
	σ	(0.084432)	(0.096449)	(29.536683)	(40.197509)	(0.00293)	(0.004235)	(665.7429)	(792.020)	(5.150792)	(2.000001)
	C	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%

		f_{11}	f_{12}	f_{13}	f_{14}	f_{15}		f_{16}		f_{17}	
D		2	2	2	2	30	40	30	40	30	40
cNrGA/OL/OGF	λ	7.601263	-1.01842	0.423587	13.50937	24306.71	9814.778	0.416943	0.612812	9972.717	1552.854
	σ	(6.315254)	(0.090734)	(0.190097)	(20.466481)	(181211.4)	(36939.5)	(0.423109)	(0.436035)	(20819.89)	(9332.06)
cNrGA	λ	1.030169	-1.03161	0.397926	3.04896	2816973	8643722	5.315799	10.0091	99428.37	150157.1
	σ	(0.24037)	(0)	(0)	(0.255734)	(2683420)	(7718188)	(1.862029)	(2.185494)	(31395.44)	(30256.9)
	C	99.95%	90%	90%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%
cNrGA/RP/OGF	λ	1.5602	-1.0316	0.3979	3.0126	16064.63	86334.47	0.1972	0.7808	18197.69	26416.1
	σ	(1.139)	(0.0001)	(0.0001)	(0.0363)	(11628.83)	(53995.52)	(0.0297)	(0.0822)	(4716.6581)	(3011.6877)
	C	99.95%	90%	90%	99.95%	< 50%	99.95%	99.95%	99.95%	99.95%	99.95%
CMA-ES	λ	12.53038	-1.02347	0.602828	546.83	0	5863.996	3.009259	5.341484	21651.7	16902.23
	σ	(5.626609)	(0.081616)	(2.04941)	(5359.25)	(0)	(2245.255)	(1.895191)	(2.548713)	(29737.29)	(28064.53)
	C	99.95%	< 50%	80%	80%	90%	80%	99.95%	99.95%	99.9%	99.95%

		f_{18}		f_{19}		f_{20}		f_{21}		f_{22}	
D		30	40	30	40	30	40	30	40	30	40
cNrGA/OL/OGF	λ	0.00069	0.004625	161.3215	289.6832	0.002573	0.00549	3.921279	5.315093	-28.4455	-38.5522
	σ	(0.004296)	(0.02719)	(53.026579)	(67.262435)	(0.004343)	(0.007858)	(0.660458)	(0.87138)	(1.328865)	(1.217775)
cNrGA	λ	0.878762	1.804246	62.5495	196.0133	1.844601	3.439882	7.308501	10.76009	-21.3948	-27.8322
	σ	(1.116297)	(1.299734)	(16.316091)	(35.577081)	(0.845755)	(1.184665)	(0.720814)	(0.747244)	(0.868425)	(1.182437)
	C	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%
cNrGA/RP/OGF	λ	0.0001	0.0014	81.3997	218.7599	0.0445	0.1964	3.8777	5.8393	-23.7051	-31.1301
	σ	(0)	(0.0005)	(16.2368)	(32.7561)	(0.0283)	(0.0914)	(0.5783)	(0.7001)	(1.4997)	(1.3427)
	C	90%	80%	99.95%	99.95%	99.95%	99.95%		99.95%	99.95%	99.95%
CMA-ES	λ	0.009898	0.058543	0	0	2.272353	4.705334	6.14771	8.363205	-5.84743	-6.87722
	σ	(0.048732)	(0.261247)	(0)	(0)	(3.222127)	(4.881311)	(1.030838)	(1.033479)	(1.86469)	(2.16221)
	C	95%	97.5%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%

		f_{23}		f_{24}		f_{25}		f_{26}		f_{27}	
D		30	40	30	40	30	40	30	40	30	40
cNrGA/OL/OGF	λ	0.015768	0.02735	-25.5646	-34.6226	0.036832	0.013276	5141.582	37868.7	-1.9E-05	-2E-06
	σ	(0.050435)	(0.0654)	(1.069375)	(1.199331)	(0.171848)	(0.049985)	(7577.837)	(33186.48)	(0.000018)	(0.000002)
cNrGA	λ	0.340207	0.646515	-21.7762	-26.5304	77.64581	126.5674	26253.7	164855.7	-0.00019	-1.7E-05
	σ	(0.176879)	(0.274409)	(1.05818)	(2.022216)	(25.9464)	(36.060326)	(18828.05)	(85898.48)	(0.000247)	(0)
	C	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%
cNrGA/RP/OGF	λ	0.0001	0.0019	-26.1732	-34.1372	33.3733	63.1649	8250.21	57358.37	-0.0003	0
	σ	(0)	(0.0005)	(0.8743)	(0.945)	(20.3531)	(24.4071)	(6069.6638)	(30891.53)	(0.0004)	(0.0001)
	C	99.75%	99.95%	99.95%	99.75%	99.95%	99.95%	99.9%	99.95%	99.95%	< 50%
CMA-ES	λ	0.387157	0.66196	-18.0756	-22.8096	0.170638	0.497917	-4930	-11440	-0.00036	0
	σ	(0.235472)	(0.257462)	(0.89084)	(1.754068)	(1.596769)	(2.742278)	(0)	(0)	(0.003526)	(0)
	C	99.95%	99.95%	99.95%	99.95%	75%	95%	99.95%	99.95%	80%	99.95%

		f_{28}		f_{29}		f_{30}		f_{31}		f_{32}	
D		30	40	30	40	30	40	30	40	30	40
cNrGA/OL/OGF	λ	-997867	-1E+08	1.000014	1.000061	1.599873	2.134873	-1.2E+34	-2.6E+45	-2.31704	-2.22965
	σ	(2.13502)	(159.9661)	(0.000031)	(0.00007)	(0.43345)	(0.419325)	(4.49E+33)	(1.05E+45)	(1.201983)	(1.196075)
cNrGA	λ	-997559	-1E+08	1.025141	1.059445	2.042637	4.331422	-6.3E+33	-3.8E+44	-2.66063	-2.32352
	σ	(1491.09)	(533212)	(0.026007)	(0.046436)	(0.740081)	(1.136662)	(5.38E+33)	(6.80E+44)	(0.873017)	(0.862687)
	C	97.5%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	97.5%	< 50%
cNrGA/RP/OGF	λ	-997867	-1E+08	1.009	1.0175	0.7239	1.3119	-3.1E+34	-6.3E+45	-3.1212	-3.1103
	σ	(0.2097)	(2983.9163)	(0.0022)	(0.0031)	(0.1027)	(0.1682)	(4.80E+33)	(1.78E+45)	(0.9049)	(0.9089)
	C	< 50%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%	99.95%
CMA-ES	λ	-988419	-1E+08	1	1	1.435747	2.815275	-2E+26	-5.7E+33	-0.07143	-0.01336
	σ	(2740.34)	(1482.29)	(0)	(0)	(1.34398)	(3.345315)	(1.75E+27)	(5.23E+34)	(0.14138)	(0.039456)
	C	99.95%	99.95%	99.95%	99.95%	80%	97.5%	99.95%	99.95%	99.95%	99.95%

		f_{33}		f_{34}	
D		40	40	30	40
cNrGA/OL/OGF	λ	-29.5328	-39.5012	187.9614	627.1064
	σ	(0.09898)	(0.072626)	(110.1502)	(947.4825)
cNrGA	λ	-26.7722	-35.0015	20101.75	48081.11
	σ	(0.627312)	(1.033578)	(48409.66)	(62336.87)
	C	99.95%	99.95%	99.95%	99.95%
cNrGA/RP/OGF	λ	-29.126	-38.5107	557.8686	1462.317
	σ	(0.1323)	(0.1544)	(75.278)	(56.8124)
	C	99.95%	99.95%	99.95%	99.95%
CMA-ES	λ	-19.062	-22.8167	334.1989	598.1029
	σ	(3.155965)	(6.001745)	(115.0855)	(201.4437)
	C	99.95%	99.95%	99.95%	< 50%