2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Influence of regions on the memetic algorithm for the CEC'2014 Special Session on Real-Parameter Single Objective Optimisation

Daniel Molina Benjamin Lacroix Francisco Herrera

Abstract— **Memetic algorithms with an appropriate trade-off between the exploration and exploitation can obtain very good results in continuous optimisation. That implies the evolutionary algorithm component should be focused in exploring the search space while the local search method exploits the achieved solutions. In a previous work, it was proposed a region-based algorithm, RMA-LSCh-CMA, adding to algorithm MA-LSCh-CMA a niching strategy that divides the domain search in equal hypercubes. The experimental results obtained, with the benchmark proposed in the CEC'2014 Special Session on Real-Parameter Single Objective Optimisation, show that the use of these regions allow the algorithm to obtain better results, specially in higher dimensions, and the resulting algorithm is more scalable.**

I. Introduction

In recent years, Evolutionary Algorithms (EA) [1] have arisen as very effective algorithms to solve discrete and real-coded optimisation problems, because there are capable of obtaining quality solutions in several complex problems without specific information about them.

One of the main issues in the design of an EA for an optimisation problem is to offer a good exploration of the search space and, at the same time, to exploit the most promising regions. Thus, a good EA should enforce a in-depth search around the most promising solutions (solutions with the best current fitness) maintaining a good diversity in the population to avoid a premature convergence and exploring well the domain search.

Memetic algorithms (MA) [11] are a hybridisation between EA and local search (LS) algorithms, mixing in one model the exploration power of EA and the exploitative power of the LS. MAs are characterised by the use of an improvement algorithm, responsible of the in-depth search around best solutions, in conjunction with an EA that explores the global domain. There are several MAs [9], and its main idea is to achieve diversity with the EA component obtaining accurate solutions by the LS method. Usually, original EA is modified to enforce its diversity.

One of the most popular technique to enforce diversity into the population is to divide the domain search in regions in which only one (or a small amount) is allowed to be in each one of them. This division can be used using an distance measure, like in *clearing method* [13] or dividing the domain in hypercubes [2]. The main disadvantage of these techniques is to have to set a minimum size: it depends on the problem and has a strong influence over their performance [3], [4].

In a previous work, we have proposed MA-LSCh-CMA [10]. MA-LSCh-CMA is a successful MA that uses as improvement algorithm the powerful CMAES [6], obtaining very good results in continuous optimisation. Originality of MA-LSCh-CMA lies in its ability to apply various times the LS on the same solution, following the local search from the previous state, continuing the LS. This behaviour makes MA-LSCh-CMA very good when a promising solution is achieved, unfortunately it does not guarantee to maintain enough diversity in the population.

In a recent work, we have proposed the algorithm called Region-based MA-LSCh-CMA (RMA-LSCh-CMA) [7], to improve the previous algorithm. The main difference is the concept of region used. This algorithm divides the domain search in regions (or hypercubes) of equals size, and only one solution is allowed in the same region. To enforce a greater exploitation during the final stages of the algorithm, the size of these regions is reduced during the run.

The use of regions followed by RMA-LSCh-CMA allows to maintain a greater diversity in the population, obtaining better results in real continuous problems, in particular in multimodal problems. Also, the use of regions avoids the presence of solutions too close to each one, leading to a more efficient search [7]. Region idea is very similar to maintain a minimum distance between solutions, but it is computationally more efficient.

In this work, we are going to use the benchmark proposed in the *Special Session on Real-Parameter Single Objective Optimization* in WCCI-2014 [8] with MA-LSCh-CMA and RMA-LSCh-CMA. The idea is to show how the concept of regions can improve the search, in error values and in performance. Also, because the proposed experimental conditions include different dimensionality values, we are going to study how this concept allow to increase the scalability.

This paper is structured as follows. In Sections II and III we give brief explanations of MA-LSCh-CMA and RMA-LSCh-CMA, remarking their differences. In Section IV the experimental framework is designed. In Section V, several comparisons are carried out to study how the concept of regions used by RMA-LSCh-CMA allows it to improve the results. Finally, in Section VI, we present the conclusions.

II. MA-LSCh-CMA

This section present the MA-LSCh-CMA. You can consult [10] to get a detailed explanation. Because MA-LSCh-CMA and RMA-LSCh-CMA have several components in common,

Daniel Molina is in the Department of Computer Science, University of Cadiz, Francisco Herrera is in Computer Science Department, University of Granada, Spain (emails: daniel.molina@uca.es, {herrera,benjamin}@decsai.ugr.es).

This work was partially supported by the Spanish Ministry of Science and Technology under projects TIN2011-28488 and TIN2012-37930-C02-01.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

we are going to explain more in detail the elements that both algorithms share.

A. Exploration Algorithm, SSGA

It uses a *steady state genetic algorithm*, SSGA [14], as its exploration component, responsible for the global search. In each step for crossover it uses *negative assortative mating*, NAM [5], in which one parent is randomly chosen, and to chose the other one, 3 individuals are randomly selected, and the most distance to first one is chosen. It uses as crossover operator the operator $BLX - \alpha$, with $\alpha = 0.5$ [12], to generate one new solution for each crossover. Finally, the new solution replaces the worst one in the population, if it is better. In MA-LSCh-CMA, there is no mechanism to guarantee that the population maintains a certain diversity.

B. Local Search: CMA-ES

It uses as a LS method the CMA-ES algorithm [6], a well-known algorithm in continuous optimisation. In each iteration it create solutions using a Gaussian distribution with a step size as its standard deviation. Because this step size is adapted during the search, it only requires as parameter the initial step size. In MA-LSCh-CMA, it is chosen as initial step size for each dimension the half of the distance to the nearest solution (in that dimension).

C. Hybridation model

This algorithm applies the following hybridation model, alternating the application of the SSGA to the population with the application of the CMA-ES.

Both algorithms follow the next hybridation model:

1) Apply the SSGA during I_{step} evaluations.
2) Select one solution to be improved by the LS.
3) If this solution was previously selected the LS parameter values from its final stage, to *continue* the search. In other case, select the default LS parameters.
4) Apply the LS during $Istep$ evaluations.
5) Store the final LS parameter values for that solution (in case the new solution will be selected again to improve).
6) If the stopping criterion was not true return to step 1.

In step 2, it is chosen the solution with best fitness that was not previously applied the LS, or it was previously applied and the improvement obtained by its application was better than $\delta_{LS} = 10^{-8}$. The idea is to identify the local optima to avoid wasting time trying to improve it more with the LS.

III. Region Based MA-LSCh-CMA

This section present the RMA-LSCh-CMA. You can consult [7] to get a detailed explanation.

A. The region base MA

Contrarily to most niching strategies where the niches are defined by the area surrounding solutions of the population, we propose here a strategy in which the niches are predefined as divisions of the search space, divided into hypercubes of equal size called here regions. This definition of a niche is illustrated in Figure 1. Each dimension is divided into ND divisions creating a grid of equal hypercubes, that represent exclusive regions which can contain only one solution.

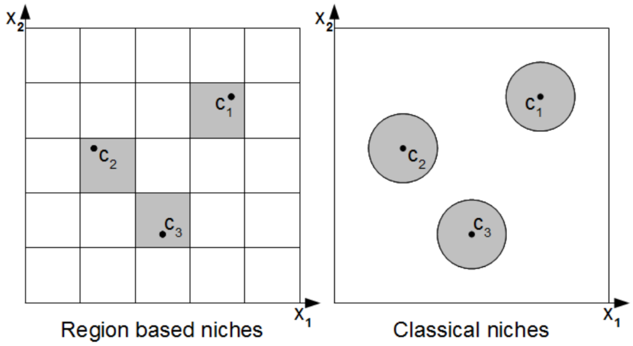

Fig. 1: Different niches strategies

B. The SSGA in a region-based MA

The SSGA is modified to not allow the generation of a solution by the SSGA in a region that is already occupied by another solution in the population if this solution is optimised. In that case, we compare both solutions and, if it is better, the new solution will replace the solution lying in the same region. By optimised, we refer to the fact that the last LS applied to this solution has not brought enough improvements (upper than δ_{LS}). On the other hand, if the solution is not optimised, the EA can replace it with a solution with a better fitness in that region.

C. The LS in a region-based MA

In the MA-LSCh-CMA, the initial step of the CMA-ES is set between the area limited by its neighbouring solutions. Here the CMA-ES initial step is set according to the size of the region. We want to ensure that the close surrounding of a solution are properly explored by the LS as this task will not be done by the EA. The initial standard deviation is set to half the size of the region. Apart from this modification, in order to allow a proper refinement of the solution, the LS is not influenced by the divisions of the search space. However, if at the end of the LS application, the new solution is in a region occupied, the best one is kept and the other one is replaced by a randomly generated solution.

D. Reduction of regions size

Initially, each dimension is divided in ND parts, to create the regions. Thus, the number of possible regions depends on the dimensionality, is ND^D. During the run, the division in regions is updated U times, where U is called *number of divisions update*. Thus, after a certain number of evaluations, the size of each region is divided by the *multiplier for update*, and the algorithm continues with the same population.

IV. Experimental framework

Experiments are carried out with the benchmark functions and experimental conditions indicated in [8]. Although several of these functions were used in previous benchmarks, like CEC'2013, the results differs because it uses different rotation matrices.

A. Test Functions

The benchmark is composed by 30 functions:

- Tree unimodal functions: $f_1 - f_3$.
- Thirteen simple multimodal functions: $f_4 - f_{16}$.
- Six hybrid functions: $f_{17} - f_{22}$, in which the variables are randomly divided into some subcomponents and then different basic functions are used for each one.
- Seven composition functions, $f_{23} - f_{30}$, that combine the results of evaluate different basic functions.

B. Experimental conditions

All functions have a shifted global optimum. All have the same domain search, [-100, 100], and their shifted global optima is inside it, away from the domain boundaries.

These are the experimental conditions followed:

- All functions are run 51 times.
- Fitness value is the result of the evaluation function.
- Dimensions D=10, 30, 50, and 100.
- The algorithm stops when a certain maximum fitness evaluations MaxFES is achieved, or the error is smaller than 10^{-8}. MaxFEs = $100000 \cdot D$.
- An error value lower than 10^{-8} is considered zero.

V. Experimental results

A. Parameter values

Because both algorithms share the majority of parameters, we have set the same values to them, to make a right comparison. Table I shows the parameter values in common. These values were proposed by the authors [10].

TABLE I: Parameters values in common: RMA-LSCh-CMA and MA-LSCh-CMA

Parameter Name	Value
Population size	100
Crossover Operator	$BLX - 0.5$
N_{NAM}	3
I_{step}	100
δ_{LS}	10^{-8}

TABLE II: Parameters values for RMA-LSCh-MA

Parameter Name	Value
Initial division number (ND)	10
Number of divisions update	4
Multiplier for each update	2

Additionally, RMA-LSCh-CMA requires more parameters. Table II shows their values. These values are the default values proposed in [7], with the exception of the population size, it was 80 individuals in the paper, and 100 in this work. In [7] where also proposed a different set of parameter values obtained by tuning, but in this benchmark the default parameter values obtain better results.

TABLE III: Computational Complexity of MA-LSCh-CMA (in milliseconds)

	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
D = 10	72	198	415	3.01
D = 30	67	607	4061	51.55
D = 50	67	1097	14443	199.19
D = 100	68	3394	80267	1130.48

TABLE IV: Computational Complexity of RMA-LSCh-CMA (in milliseconds)

	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2 - T1)/T0$
D=10	71	205	1660	20.49
D=30	70	654	2914	32.28
D50	70	1350	7097	82.10
D100	69	3821	14312	152.04

B. Algorithms complexity

In Tables III and IV are shown the algorithm complexity of algorithms MA-LSCh-CMA and RMA-LSCh-CMA. We can see that using regions to enforce diversity increases the scalability of the algorithm. For dimension 50, the value for $(\hat{T}2 - T1)/T0$ with RMA-LSCh-CMA is 41% than obtained with MA-LSCh-CMA, and for dimension 100 with RMA-LSCh-CMA is 13% than obtained with MA-LSCh-CMA.

C. RMA-LSCh-CMA vs MA-LSCh-CMA

Table V shows the mean values for each algorithm and function for dimension 50 and 100. We can see that RMA-LSCh-CMA obtains the best average results, and the improvement obtained by the use of regions is clearly increased with the dimensionality.

Comparisons by category, RMA-LSCh-CMA scales specially well in unimodal functions (in multimodal are very similar) and in hybrid functions.

Table VI resumes different measures comparing MA-LSCh-CMA and RMA-LSCh-CMAin relation with the dimensionality. We can observe than RMA-LSCh-CMA in average is better for each measure, and the improvement obtained increases with the dimensionality. Thus, RMA-LSCh-CMA is better than MA-LSCh-CMA, both in error values and in performance, and this improvement increases with the dimensionality.

D. Results obtained by RMA-LSCh-CMA

Due to space limitation, we are going to put only the results obtained by RMA-LSCh-CMA, the best algorithm, to make them available to other researchers. Tables VII, VIII, IX, X show the results obtained by RMA-LSCh-CMA for dimension 10, 30, 50, and 100.

We can see that the scalability of the algorithm is good, because it maintain good results in higher dimensions. Also, the differences between median and mean are reasonable. In summarize, experiments show that RMA-LSCh-CMA is a robust algorithm, with an interesting scalable behaviour.

TABLE V: Median error for each dimension

(a) Dimension 50

Fun	RMA-LSCh-CMA	MA-LSCh-CMA
1	0.000000E+00	0.000000E+00
2	0.000000E+00	0.000000E+00
3	1.771750E+01	**0.000000E+00**
4	**0.000000E+00**	5.454993E+01
5	1.999994E+01	1.999999E+01
6	6.121990E+00	**1.229027E+00**
7	0.000000E+00	0.000000E+00
8	1.981726E-05	**0.000000E+00**
9	**3.681346E+01**	4.083218E+01
10	2.424665E+02	**7.600832E+00**
11	**3.623177E+03**	5.554667E+03
12	**1.438951E-02**	9.540046E-02
13	**2.192767E-01**	2.640195E-01
14	**2.467165E-01**	2.613563E-01
15	**4.686589E+00**	5.426558E+00
16	**1.872406E+01**	2.009951E+01
17	1.943020E+03	**1.899226E+03**
18	4.790563E+02	**3.347629E+02**
19	**1.394770E+01**	3.761487E+01
20	**7.641425E+02**	8.096744E+02
21	**1.249797E+03**	1.770009E+03
22	3.107434E+02	**2.697483E+02**
23	3.440050E+02	3.440045E+02
24	**2.589911E+02**	2.591884E+02
25	**2.099881E+02**	2.138974E+02
26	**1.001966E+02**	1.147241E+02
27	4.884886E+02	**3.927057E+02**
28	1.308381E+03	**1.227101E+03**
29	1.622501E+03	**1.091956E+03**
30	**9.589913E+03**	9.729267E+03

(b) Dimension 100

Fun	RMA-LSCh-CMA	MA-LSCh-CMA
1	**1.338985e-01**	3.088033e+04
2	**0.000000e+00**	3.493682e+02
3	**1.918758e+03**	6.231324e+03
4	1.055354e+02	**9.048479e+01**
5	1.999996e+01	1.999998e+01
6	2.686612e+01	**8.382388e+00**
7	4.327280e-04	**2.017546e-09**
8	4.696169e-05	**8.238439e-10**
9	**9.539947e+01**	1.257094e+02
10	9.839455e+02	7.674920e+02
11	**8.549727e+03**	1.304217e+04
12	**1.355434e-02**	1.353486e-01
13	**3.065968e-01**	3.774176e-01
14	1.240539e-01	**1.199853e-01**
15	**1.060084e+01**	2.245631e+01
16	**4.268038e+01**	4.318112e+01
17	**5.183200e+03**	3.937255e+04
18	9.787416e+02	**5.117417e+02**
19	9.951786e+01	**8.489232e+01**
20	**2.975314e+03**	4.736397e+03
21	**3.516743e+03**	2.356521e+04
22	**7.793208e+02**	1.319897e+03
23	3.483129e+02	**3.482351e+02**
24	**3.590149e+02**	3.617879e+02
25	**2.368728e+02**	2.549403e+02
26	2.000692e+02	2.000792e+02
27	9.112618e+02	**3.942113e+02**
28	3.288493e+03	**2.311201e+03**
29	3.165691e+03	**1.724867e+03**
30	**9.247364e+03**	9.389607e+03

TABLE VI: Results for MA-LSCh-CMA and RMA-LSCh-CMA for dimension (average for function)

Dimension	Algorithm	Best	Worst	Median	Mean	Std
10	MA-LSCh-CMA	5.851346E+01	1.325460E+02	8.024321E+01	7.961767E+01	2.015156E+01
	RMA-LSCh-CMA	**4.683987E+01**	**1.264823E+02**	**7.645782E+01**	**7.649017E+01**	2.032637E+01
30	MA-LSCh-CMA	1.773469E+02	1.366445E+03	3.485442E+02	3.151299E+02	2.007272E+02
	RMA-LSCh-CMA	**1.225540E+02**	**5.752820E+02**	**3.031459E+02**	**2.871957E+02**	1.026800E+02
50	MA-LSCh-CMA	5.728093E+02	3.976336E+03	1.038021E+03	8.066302E+02	6.773991E+02
	RMA-LSCh-CMA	**4.911126E+02**	**1.330508E+03**	**7.876310E+02**	**7.551120E+02**	1.817873E+02
100	MA-LSCh-CMA	1.042859E+03	2.086971e+04	3.023720e+03	4.538572e+03	4.185062e+03
	RMA-LSCh-CMA	**8.619702E+02**	**2.218368e+03**	**1.418016e+03**	**1.434800e+03**	2.869823e+02

TABLE VII: Results for 10D

Function	Best	Worst	Mean	Median	Std
1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
3	0.000000E+00	5.229974E-06	1.025485E-07	0.000000E+00	7.251275E-07
4	0.000000E+00	4.335407E+00	8.500798E-02	0.000000E+00	6.010972E-01
5	0.000000E+00	2.000000E+01	1.365196E+01	1.999868E+01	9.235427E+00
6	0.000000E+00	1.807498E-03	1.478613E-04	0.000000E+00	3.869960E-04
7	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
8	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
9	9.949591E-01	6.964708E+00	3.316530E+00	2.984877E+00	1.632705E+00
10	6.245444E-02	1.219157E+02	7.677946E+00	3.414961E+00	2.324503E+01
11	1.873633E-01	1.303928E+02	2.013497E+01	1.182953E+01	3.230385E+01
12	0.000000E+00	1.331651E-01	1.646457E-02	1.124002E-02	2.378510E-02
13	9.069848E-03	7.038013E-02	3.292333E-02	2.886623E-02	1.567129E-02
14	6.164570E-02	1.988720E-01	1.264898E-01	1.207352E-01	3.356309E-02
15	3.103627E-01	7.114398E-01	4.714944E-01	4.476892E-01	9.598157E-02
16	2.341056E-01	2.093216E+00	1.054166E+00	1.123691E+00	4.701500E-01
17	6.273603E-01	3.971328E+02	7.833846E+01	4.022057E+01	9.486355E+01
18	6.543135E-02	2.222729E+01	5.220721E+00	3.346637E+00	4.586417E+00
19	7.849730E-03	2.099368E-01	7.660733E-02	6.187272E-02	4.399772E-02
20	4.635259E-03	6.313467E+01	8.056691E+00	3.121138E+00	1.262373E+01
21	1.607504E-01	2.754776E+02	4.928617E+01	1.736043E+01	7.047081E+01
22	3.628534E-02	3.928931E+01	8.474625E+00	6.321621E-01	1.120228E+01
23	3.294575E+02	3.294575E+02	3.294575E+02	3.294575E+02	2.842171E-13
24	1.000000E+02	1.164711E+02	1.084430E+02	1.087211E+02	2.946063E+00
25	1.107912E+02	2.013828E+02	1.750708E+02	1.977504E+02	3.232990E+01
26	1.000136E+02	1.000673E+02	1.000364E+02	1.000324E+02	1.402073E-02
27	1.224286E+00	4.002907E+02	1.847796E+02	3.000632E+02	1.547445E+02
28	1.000856E+02	4.944884E+02	3.887168E+02	3.600612E+02	8.073141E+01
29	1.616316E+02	2.456375E+02	2.270654E+02	2.288594E+02	1.275042E+01
30	4.992300E+02	8.223850E+02	5.851143E+02	5.640864E+02	6.482633E+01

VI. Conclusions

In this paper, we have analysed MA-LSCh-CMA and RMA-LSCh-CMA using the new benchmark proposed in the *Special Session on Real-Parameter Single Objective Optimization*. Both of them are memetic algorithms that use the CMA-ES algorithm to best individuals to make a in-depth local search. The main difference between RMA-LSCh-CMA and MA-LSCh-CMA is that the first one divides the domain search in regions of equals size and it only allows one solution on each region, to maintain diversity into the population. During the search these regions are reduced to increase the degree of exploitation of the solutions. The experimental results show that the use of regions to maintain diversity improves the results, and that improvement increases with the dimensionality. Also, at the same time, it reduces a lot the algorithm complexity when the dimensionality increase. Thus, RMA-LSCh-CMA is a robust and scalable algorithm.

Acknowledgment

This work was partially supported by the Spanish Ministry of Science and Technology under projects TIN2011-28488 and TIN2012-37930-C02-01.

References

[1] T Bäck, D B Fogel, and Z Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, 1997.

[2] Ying-Ping Chen and Chao-Hong Chen. Enabling the extended compact genetic algorithm for real-parameter optimization by using adaptive discretization. *Evolutionary Computation*, 18(2):199–228, 2010.

[3] Antonio Dello Cioppa, Claudio De Stefano, and Angelo Marcelli. On the Role of Population Size and Niche Radius in Fitness Sharing. *IEEE Transactions on Evolutionary Computation*, 8(5), 2004.

[4] Swagatam Das, Sayan Maity, Bo-Yang Qu, and P.N. Suganthan. Real-parameter evolutionary multimodal optimization A survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1(2):71–88, June 2011.

[5] C. Fernandes and A. Rosa. A Study of non-Random Matching and Varying Population Size in Genetic Algorithm using a Royal Road Function. *Proc. of the 2001 Congress on Evolutionary Computation*, pages 60–66, 2001.

[6] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159–195, 2001.

TABLE VIII: Results for 30D

Function	Best	Worst	Mean	Median	Std
1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
3	0.000000E+00	3.527440E+02	2.619492E+01	0.000000E+00	6.591081E+01
4	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
5	1.999871E+01	1.999999E+01	1.999971E+01	1.999979E+01	2.525192E-04
6	4.617794E-03	3.415131E+00	1.135849E+00	1.042735E+00	1.002887E+00
7	0.000000E+00	9.857285E-03	1.932801E-04	0.000000E+00	1.366697E-03
8	0.000000E+00	9.958320E-01	1.953497E-02	0.000000E+00	1.380693E-01
9	8.954632E+00	2.586893E+01	1.792877E+01	1.790926E+01	3.972464E+00
10	1.346573E+00	3.612410E+02	8.124660E+01	8.580182E+00	1.016626E+02
11	3.463690E+02	2.757179E+03	1.549521E+03	1.610068E+03	5.819634E+02
12	5.100968E-03	3.771581E-02	1.597474E-02	1.323069E-02	8.122876E-03
13	9.761344E-02	1.802711E-01	1.376759E-01	1.365498E-01	2.045458E-02
14	1.373580E-01	3.057716E-01	2.216354E-01	2.274716E-01	3.444128E-02
15	1.332803E+00	3.384309E+00	2.450783E+00	2.472471E+00	4.281959E-01
16	8.010510E+00	1.176225E+01	9.647416E+00	9.659988E+00	9.328604E-01
17	3.184446E+01	1.674199E+03	6.978608E+02	7.524421E+02	3.226947E+02
18	4.173044E+01	2.579907E+03	5.669142E+02	2.333781E+02	6.910744E+02
19	2.964660E+00	8.441385E+00	5.822519E+00	5.824098E+00	1.339573E+00
20	8.161756E+01	6.012153E+02	1.988780E+02	1.837007E+02	8.390115E+01
21	1.261600E+02	1.075745E+03	5.732908E+02	5.407018E+02	2.634281E+02
22	2.734139E+01	3.779415E+02	1.588346E+02	1.463604E+02	6.165145E+01
23	3.152441E+02	3.152444E+02	3.152442E+02	3.152442E+02	8.012869E-05
24	2.000000E+02	2.257560E+02	2.222347E+02	2.240828E+02	6.341092E+00
25	2.026154E+02	2.109567E+02	2.056072E+02	2.049766E+02	2.532965E+00
26	1.000829E+02	2.000127E+02	1.020888E+02	1.001303E+02	1.384855E+01
27	3.002385E+02	4.013025E+02	3.284447E+02	3.099193E+02	3.377058E+01
28	3.000000E+02	9.277563E+02	8.233830E+02	8.466024E+02	1.129816E+02
29	8.066873E+02	2.164003E+03	1.146583E+03	1.098135E+03	2.377853E+02
30	7.538351E+02	2.958855E+03	2.040669E+03	1.984265E+03	4.929734E+02

[7] Benjamin Lacroix, Daniel Molina, and Francisco Herrera. Region based memetic algorithm for real-parameter optimisation. *Information Sciences*, 261:15–31, 2014.

[8] J.J. Liang, B.Y. Qu, P.N. Suganthan. ”Problem Definitions, Evaluation Criteria for the CEC 2014 Special Session, and Competition on Single Objective Real-Parameter Numerical Optimization”. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, December 2013. Technical report.

[9] P. Merz and Bernd Freisleben. *Fitness Landscapes and Memetic Algorithm Design*. D. Corne, M. Dorigo and F. Glower, editors. McGraw-Hill, London, 1999.

[10] Daniel Molina, Manuel Lozano, Carlos García-Martínez, and Francisco Herrera. Memetic Algorithms for Continuous Optimization Based on Local Search Chains. *Evolutionary Computation*, 18(1):27–63, 2010.

[11] P. A. Moscato. *Memetic Algorithms: a Short Introduction*. D. Corne, M. Dorigo and F. Glower, editors. McGraw-Hill, London, 1999.

[12] T. Nomura and K. Shimohara. An Analysis of Two-parent Recombinations for Real Valued Chromosomes in an Infinite Population. *Evolutionary Computation Journal*, 3(9):283–308, 2001.

[13] A. Petrowski. A clearing procedure as a niching method for genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 798–803, May.

[14] G. Syswerda. Uniform Crossover in Genetic Algorithms. In J. David Schaffer, editor, *Proc. of the Thrid Int. Conf. on Genetic Algorithms*, pages 2–9. Morgan Kaufmann Publishers, San Mateo, 1989.

TABLE IX: Results for 50D

Function	Best	Worst	Mean	Median	Std
1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
3	0.000000E+00	1.322879E+03	2.344839E+02	1.771750E+01	3.389534E+02
4	0.000000E+00	9.810311E+01	1.866664E+01	0.000000E+00	3.799332E+01
5	1.999922E+01	1.999999E+01	1.999991E+01	1.999994E+01	1.196615e-04
6	1.230035E+00	1.126953E+01	6.341466E+00	6.121990E+00	2.302482E+00
7	0.000000E+00	1.232099e-02	5.316288e-04	0.000000E+00	2.199802e-03
8	0.000000E+00	5.806045e-05	1.935430e-05	1.981726e-05	1.409291e-05
9	2.188920E+01	5.074315E+01	3.564299E+01	3.681346E+01	6.519047E+00
10	3.560551E+00	6.020461E+02	2.239043E+02	2.424665E+02	1.479644E+02
11	1.226740E+03	4.709829E+03	3.457559E+03	3.623177E+03	6.444440E+02
12	5.513460e-03	2.653887e-02	1.558060e-02	1.438951e-02	5.331830e-03
13	1.630036e-01	2.777435e-01	2.207196e-01	2.192767e-01	2.348579e-02
14	1.992321e-01	2.954067e-01	2.459198e-01	2.467165e-01	2.176523e-02
15	3.175081E+00	6.885428E+00	4.798497E+00	4.686589E+00	7.357523e-01
16	1.664508E+01	2.101696E+01	1.874715E+01	1.872406E+01	1.113784E+00
17	6.848466E+02	3.580324E+03	1.988571E+03	1.943020E+03	6.507009E+02
18	9.370334E+01	2.515978E+03	7.011082E+02	4.790563E+02	5.980367E+02
19	1.043051E+01	1.895826E+01	1.398320E+01	1.394770E+01	1.928223E+00
20	2.884071E+02	4.152443E+03	1.031268E+03	7.641425E+02	7.514658E+02
21	5.209328E+02	2.060567E+03	1.275634E+03	1.249797E+03	4.111007E+02
22	2.793264E+01	8.175153E+02	3.325134E+02	3.107434E+02	1.752374E+02
23	3.440045E+02	3.440056E+02	3.440050E+02	3.440050E+02	2.551582e-04
24	2.548420E+02	2.744139E+02	2.638528E+02	2.589911E+02	7.121144E+00
25	2.052070E+02	2.256620E+02	2.126556E+02	2.099881E+02	6.896996E+00
26	1.001511E+02	1.002611E+02	1.001966E+02	1.001966E+02	2.873123e-02
27	3.079453E+02	6.534402E+02	4.811120E+02	4.884886E+02	7.218094E+01
28	1.096069E+03	2.423497E+03	1.370109E+03	1.308381E+03	2.477899E+02
29	9.983074E+02	2.751970E+03	1.632789E+03	1.622501E+03	3.665546E+02
30	8.506993E+03	1.315283E+04	9.860507E+03	9.589913E+03	9.844969E+02

TABLE X: Results for 100D

Function	Best	Worst	Mean	Median	Std
1	0.000000E+00	8.294900e-01	2.096037e-07	1.338985e-01	2.148479e-01
2	0.000000E+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
3	2.297558E+02	3.602810e+03	2.047680e+03	1.918758e+03	8.327843e+02
4	0.000000E+00	2.280353e+02	1.056709e+02	1.055354e+02	7.303576e+01
5	1.999985E+01	1.999999e+01	1.999996e+01	1.999996e+01	3.002242e-05
6	1.653533E+01	3.966517e+01	2.612602e+01	2.686612e+01	5.038756e+00
7	0.000000E+00	7.396045e-03	0.000000e+00	4.327280e-04	1.420433e-03
8	7.310115e-06	1.081272e-04	4.418525e-05	4.696169e-05	1.717164e-05
9	6.367749E+01	1.293446e+02	9.452115e+01	9.539947e+01	1.426078e+01
10	4.811494E+02	1.788446e+03	9.563256e+02	9.839455e+02	2.804663e+02
11	6.599748E+03	9.792360e+03	8.529045e+03	8.549727e+03	7.114195e+02
12	5.454382e-03	2.259985e-02	1.303121e-02	1.355434e-02	3.805194e-03
13	2.407476e-01	3.769922e-01	3.006908e-01	3.065968e-01	2.713187e-02
14	9.845731e-02	1.442743e-01	1.248828e-01	1.240539e-01	9.999619e-03
15	8.129618E+00	1.501749e+01	1.042165e+01	1.060084e+01	1.545311e+00
16	3.980867E+01	4.452669e+01	4.286827e+01	4.268038e+01	1.160471e+00
17	3.345809E+03	6.801327e+03	5.194470e+03	5.183200e+03	7.980310e+02
18	2.665561E+02	8.757247e+03	5.891675e+02	9.787416e+02	1.273552e+03
19	6.127934E+01	1.175469e+02	1.036788e+02	9.951786e+01	1.373116e+01
20	1.312009E+03	5.271713e+03	2.714441e+03	2.975314e+03	1.101745e+03
21	1.937954E+03	5.266148e+03	3.516025e+03	3.516743e+03	7.521412e+02
22	1.689190E+02	1.680732e+03	7.056086e+02	7.793208e+02	3.230215e+02
23	3.482842E+02	3.483444e+02	3.483111e+02	3.483129e+02	1.373773e-02
24	3.322769E+02	3.768691e+02	3.592935e+02	3.590149e+02	7.535504e+00
25	2.182126E+02	2.728012e+02	2.341050e+02	2.368728e+02	1.413945e+01
26	2.000000E+02	2.000828e+02	2.000780e+02	2.000692e+02	2.536428e-02
27	5.991066E+02	1.140925e+03	9.033352e+02	9.112618e+02	1.235201e+02
28	2.319712E+03	4.705286e+03	3.179931e+03	3.288493e+03	5.832432e+02
29	1.318750E+03	4.821635e+03	3.275740e+03	3.165691e+03	7.553812e+02
30	5.971089E+03	1.112881e+04	9.383213e+03	9.247364e+03	9.434189e+02