Proceedings of the 10th IASTED International Conference
Artifical Intelligence and Applications (AIA 2010)
February 15 - 17, 2010 Innsbruck, Austria

MIGRATING INDIVIDUALS AND PROBABILISTIC MODELS ON DEDAS: A COMPARISON ON CONTINUOUS FUNCTIONS

Santiago Muelas
Dept. of Computer Systems Architecture and Technology
Universidad Politécnica de Madrid, Spain
email: smuelas@fi.upm.es

Alexander Mendiburu
Intelligent Systems Group
Dept. of Computer Architecture and Technology
University of the Basque Country, Spain
email: alexander.mendiburu@ehu.es

Antonio LaTorre
Dept. of Computer Systems Architecture and Technology
Universidad Politécnica de Madrid, Spain
email: atorre@fi.upm.es

José-María Peña
Dept. of Computer Systems Architecture and Technology
Universidad Politécnica de Madrid, Spain
email: jmpena@fi.upm.es

ABSTRACT
One of the most promising areas in which probabilistic graphical models have shown an incipient activity is the field of heuristic optimization and, in particular, in the Estimation of Distribution Algorithms (EDAs). EDAs constitute a well-known family of Evolutionary Computation techniques, similar to Genetic Algorithms. Due to their inherent parallelism, different research lines have been studied trying to improve EDAs from the point of view of execution time and/or accuracy. Among these proposals, we focus on the so-called island-based models. This approach defines several islands (EDA instances) running independently and exchanging information with a given frequency. The information sent by the islands can be a set of individuals or a probabilistic model. This paper presents a comparative study of both information exchanging techniques for a univariate EDA ($UMDA_g$) over a wide set of parameters and problems –the standard benchmark developed for the IEEE Workshop on *Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems* of the ISDA 2009 Conference. The study concludes that the configurations based on migrating individuals obtain better results.

KEY WORDS
Evolutionary Computing, Graphical Models, Estimation of Distribution Algorithms, Island Models

1 Introduction

Estimation of Distribution Algorithms are a set of techniques that belong to the field of Evolutionary Computation. Since they were introduced in the 90s [18, 24], the research community has put a lot of effort in their development, providing powerful algorithms which have been successfully applied to both artificial and real-world problems. In general terms, EDAs are similar to Genetic Algorithms, but their main characteristic is the use of probabilistic models to extract information from the current population (instead of using crossover or mutation operators) in order to create a new and presumably better population. The complexity of the different EDA approaches is usually related to the probabilistic model used, and the ability of that model to identify and represent the (in)dependencies among the variables. Detailed information about the main characteristics of EDAs, as well as the different algorithms that belong to this family can be found in [14, 15, 19, 20].

The main drawback of the most complex EDAs – those that try to consider all the possible (in)dependencies among the variables– is the high computational cost. Due to this, and thanks to the modularity of EDAs, several parallel approaches have been proposed. These proposals can be divided into two groups:

- Direct parallelization (pEDAs): Those whose behavior is exactly the same of the corresponding sequential version. Their main goal is the reduction of the execution time, and the applicability to larger problems.
- Island-based approach (dEDAs): Those that create different subpopulations and exchange information among them, trying to improve the quality of the solutions of the sequential algorithm.

In this work, we pay attention to the second approach. In this scheme, an EDA instance is executed in each island, and some information is exchanged among the islands during the execution. This information can be made up of individuals (as done in other EAs), or probabilistic models (following the rationale that EDAs use them to extract and gather information about the population). Migration of individuals is a classic approach and has proven to obtain successful results [2, 4, 6, 16]. In addition, migration of models was explicitly developed for the distributed estimation of distribution algorithms (dEDAS) [1, 7, 8, 10, 11].

Until now, most of the previous work in dEDAs has been conducted in the discrete domain, and little research has been done in comparing both migration methods (individuals versus models). In particular, in continuous optimization, as far as the authors are aware, only two studies have been done [7, 9]. Although these papers concluded that the migration of models obtains significantly better results than the migration of individuals, the experimental scenario was restricted to a) a limited number of problems with small dimensions and b) a small number of parameters that were analyzed. In this paper, we study empirically both approaches over the standard benchmark developed for the IEEE workshop on *Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems* of the ISDA 2009 Conference. Therefore, our goal is to carry out an extensive study combining a wide set of parameters and using a standard benchmark of problems.

The rest of the paper is organized as follows: Section 2 presents an overview of the previous studies on EDAs and dEDAs. Section 3 describes the proposed experimental scenario. Section 4 presents and comments the results obtained and lists the most relevant facts extracted from this analysis. Finally, Section 5 contains the concluding remarks derived from this study.

2 Preliminaries

2.1 Estimation of Distribution Algorithms: EDAs

EDAs are non-deterministic, stochastic heuristic search strategies that are part of the Evolutionary Computation paradigm. In EDAs, multiple solutions or individuals are created at every generation, evolving successively until a satisfactory solution is achieved. In brief, the characteristic that clearly differentiates EDAs from other evolutionary search strategies, such as Genetic Algorithms (GAs), is that the evolution from one generation to the next is achieved by estimating the probability distribution of a set of individuals, sampling later the induced model. This avoids the use of crossing or mutation operators, and the number of parameters required by EDAs is considerably reduced. Based on the probabilistic model considered, three main groups of EDAs can be distinguished: univariate models, which assume that variables are marginally independent; bivariate models, which accept dependencies between pairs of variables; and multivariate models, in which there is no limitation on the number of dependencies.

In this study, we focus on the Univariate Marginal Distribution Algorithm for Gaussian Models ($UMDA_g$) [12, 13]. This algorithm considers no dependencies between the variables involved in the problem. It is assumed that the joint density function follows a n-dimensional normal distribution, which is factorized by a product of one-dimensional and independent normal densities.

2.2 Distributed Estimation of Distribution Algorithms: dEDAs

In the distributed Evolutionary Algorithms (dEAs)[1], the whole population is distributed over multiple subpopulations and occasionally allows the migration or exchange of some individuals among the different islands. Therefore, each node executes an independent algorithm on an independent population. An important aspect of the performance of dEAs is the migration strategy. This is configured through different parameters [5]: (i) Migration frequency: How often (number of generations) is information sent?, (ii) Migration rate: How many individuals migrate each time?, (iii) Information selection: What kind of information is exchanged?, (iv) Acceptance policy: How are the incoming and the local information combined?, (v) Migration topology: Which island sends information to which other?

Close scrutiny of migration parameters [21] has proved that, even though EAs with small populations risk being trapped in a local optimum, an appropriate migration strategy can avoid a suboptimal solution from dominating all the populations. A correct configuration can help to obtain better results with fewer evaluations, but configuring these optimal parameters is not a simple issue [3, 17, 23].

Regarding the information exchange among islands, two possible alternatives are available: (i) the straightforward approach of selecting a pull of individuals that will be later sent to the consignees and (ii) the alternative of using the main characteristic of EDAs: the probabilistic models. These probabilistic models will be (or should be) able to represent the (in)dependencies among the variables, and therefore comprise more information than a group of individuals. This second approach opens a new challenge: how should the different probabilistic models be combined? In the simplest case, the combination of the resident model with an immigrant one can be formalized by the following rule:

$$M'_R = \beta M_R + (1 - \beta) M_I \quad (1)$$

where β varies in the range $[0, 1]$ and represents the influence of the immigrant model M_I over the resident model M_R. An extended version of this formula for n immigrant models would be:

$$M'_R = \beta_R M_R + \beta_{I1} M_{I1} + \beta_{I2} M_{I2} + \ldots + \beta_{In} M_{In} \quad (2)$$

In order to compute the value of β two different strategies have been traditionally considered. The simplest one is called *constant value* and it simply assigns to each β a constant value within the interval $[0, 1]$. The second one, called *adaptative value*, computes the β value based on the *quality* of the population associated to each model. For n immigrants, the β value is defined as:

$$\beta_R = \frac{F_R}{F_R + \sum_j^n F_{I_j}}, \beta_{I_i} = \frac{F_{I_i}}{F_R + \sum_j^n F_{I_j}} \quad (3)$$

where F_R represents the mean fitness value of the resident subpopulation and F_{I_i} represents the mean fitness value of the i-th immigrant subpopulation.

For this work we have introduced a new combination model called *uniform combination*. This method does not combine the models, it selects each model component from

[1] also known as coarse-grained, multiple-deme or island models

a model of the global set of the immigrants and resident models. Each model has a probability β of being selected for each of the components of the new model. The β value is computed using the same formula of the adaptative combination method.

3 Experimentation

For the experimentation, the benchmark from the workshop on *Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems - A Scalability Test* to be held at the ISDA 2009 Conference has been considered. This benchmark defines 11 continuous optimization functions. The first 6 functions were originally proposed for the '*Special Session and Competition on Large Scale Global Optimization*" held at the CEC 2008 Congress [22]. The other 5 functions have been specially proposed for the Workshop of the ISDA 2009 Conference. These functions, presented on Table 1, have different degrees of difficulty and can scale to any dimension. Detailed information about the selected benchmark can be found at the web page of the organizers of the workshop[2].

Table 1. Benchmark Functions

Id	**Name**
f1	Shifted Sphere Function
f2	Shifted Schwefels Problem 2.21
f3	Shifted Rosenbrocks Function
f4	Shifted Rastrigins Function
f5	Shifted Griewanks Function
f6	Shifted Ackleys Function
f7	Schwefels Problem 2.22
f8	Schwefels Problem 1.2
f9	Extended f_{10}
f10	Bohachevsky
f11	Schaffer

Table 2 shows the different parameters used throughout the experiments. In order to analyze the effects of the migration strategies, several island configurations of $UMDA_g$ instances were compared against each other. Some of the parameters have been used in previous studies with dEDAs [8, 11], and additional parameters have been included to obtain a wider view. For each combination, 25 independent executions were carried out. The stopping criterion, as defined in the benchmark, was a fixed number of fitness evaluations (5000 times the dimension of the problem). The performance criterion is the distance (error) between the best individual found and the global optimum in terms of fitness value. A sequential version of the $UMDA_g$ algorithm was also executed with different population sizes (64, 100, 200, 512, 1024 and 2048) in order to have a baseline comparison.

[2]http://sci2s.ugr.es/programacion/workshop/Scalability.html

Table 2. Parameters Values

Common Parameter Values	
Problem Size	50 and 100
Population Size	512, 1024 and 2048
Learning Model	$UMDA_g$
Selected inds. for learning	best 50% of the population
#Islands	8, 16 and 32
Topology	ring2 and all-to-all (a2a)
Migration Period	migrate every 10, 20 and 40 generations
Acceptance policy	best individuals from resident and
	immigrants populations
Particular Parameter Values	
Inds. Migration rate	5%, 10% and 20%
Inds. Emigrants Selection	best or random individuals
Models Combination	adaptative and uniform

4 Analysis of the Results

In order to compare all the configurations across all the functions, the average rank according to the Friedman test was computed for each function and for all the functions. The nWins procedure [17] was also applied to the average ranks per function to perform a global comparative analysis. This procedure carries out a pair-wise statistical comparison over the distribution values of all the available configurations by means of the Wilcoxon signed-rank test with a confidence level of 0.05. With these results, the following analysis is carried out: if one algorithm is significantly better than other ($p - value < 0.05$), the winning algorithm is granted +1 wins and the losing algorithm is penalized with -1 wins. The sum of all the "wins" constitutes the nWins value.

Tables 3 and 4 display a ranking of the configurations on both 50-D and 100-D according to the global average rank, together with the nWins score. Due to the size of these tables (432 rows), only the best and worst 10 configurations as well as the sequential configurations are displayed. From these results, it can be seen that, for the 50 dimensional functions, the best configurations are based on sending individuals, with the lowest population size and number of islands, whereas the worst side of the table is mostly filled with configurations based on sending models with the highest number of islands and the lowest population size. It can also be seen that the ring topology and the selection of the best emigrants are values that can be found in almost all the best configurations. Therefore, it seems that the best configurations are those which are based on sending individuals, have a reasonable population size per island but still small enough to have a considerable number of iterations (due to the restriction of the benchmark), use the ring topology and send their best individuals. On the other hand, the worst configurations have in common the smallest population size per island and have values that quickly decrease the diversity of the populations, i.e highest topology degree and highest migration rate.

The sequential versions of the algorithms are mostly placed around positions near the middle of the table, neither too good nor too bad. The best results are obtained with the configuration with 512 individuals which has a good balance between the population size and the number of iterations. For 100-D the situation is quite similar, confirming the fact that the previous conclusions are stable and can be generalized to a higher number of dimensions. However, it can be seen that the best configurations tend to have greater population sizes and number of islands which implies a greater diversity in the global population. Both the worst and sequential configurations have a similar pattern to the one in 50-D.

The next study consisted in analyzing the performance *on each function* of the configurations based on sending individuals against the equivalent ones based on sending models. For this task, all the configurations were grouped in each of the 8 possible groups (54 configurations per group) for sending individuals or models based on the values of the specific parameters of Table 2. Then, the average rank and the number of wins was obtained for each group (54 values per group) following a procedure similar to the one described before.

Tables 5 and 6 present the results in both 50 and 100 dimensions. For each function, the best average rank is highlighted on both tables. From these results, it can be seen that, in 9 out of 11 functions, the groups based on sending individuals obtained a superior average ranking than the ones based on sending models. These results were confirmed when using the nWins procedure. Only on two functions, f4 and f8, the sending models configurations obtained a better average rank than the sending individuals configurations. Within the individuals configurations, sending the best 5% of individuals, achieves the best performance in most of the functions followed closely by the group which sends the best 10%. On the other hand, sending a 20% of randomly chosen individuals, obtained the worst average results. It seems that sending a small number of the best individuals is the best overall strategy for most of the common configurations. With the average ranks per function for each of the eight groups, a global analysis was also performed. These results are shown in the last row of Tables 5 and 6. It can be seen that the conclusions from the previous analysis are also confirmed in the global one: all the sending individuals groups obtain better values in both average rank and number of wins than the sending models groups and the best results are achieved by the group which sends the best 5% of individuals followed closely by the group which sends the best 10%.

Finally, for each common distributed configuration, i.e., for each selection of population size, number of islands, migration period and topology, the results for each of the eight possible configurations was pairwise compared using the average ranking and nWins procedures. The results for the 50 dimensional functions are presented in Table 7. Since the values are quite similar for the 100 dimensional functions, only the 50 dimensional results are presented but the average of each of the eight groups are presented for both dimensions (Table 8). The best average results are highlighted in both tables. In this analysis, it is also shown that all the configurations which send individuals obtain better results than the sending models ones. Within the groups which send individuals, the configurations which achieve the best global values are the ones which send the best 5% and 10% of the population.

5 Conclusion

This paper presents an extensive comparison of several configurations of dEDAs over a standard benchmark of continuous functions in both 50 and 100 dimensions. Several analysis from different points of view have been carried out and non-parametrical tests have been applied. The attention has been put to which method for exchanging information between dEDAs, the migration of individuals or the migration of probabilistic models, is the best approach for a researcher who would like to apply the $UMDA_g$ dEDAs in a continuous domain. From this perspective, the results from this study clearly express that, for most of the functions, the exchange of individuals obtains significantly better results than the alternative approach of sending models. Furthermore, the question of whether the dEDAs configurations obtain better results than their equivalent sequential versions has also been addressed: the study shows that the best dEDAs configurations outperform the best results of the sequential counterparts. However, it is neccesary to carry out a correct selection of the distributed parameter values in order to achieve these results. Finally, the study also recommends the use of the ring2 topology and the selection of a small percent of emigrants in order to obtain the best results with the dEDAs configurations and discourages the application of configurations which tend to destroy the diversity of the islands population, i.e, the combination of the following values: small population sizes, the all-to-all topology and the exchange of a considerable amount of individuals.

Acknowledgements

The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the Centro de Supercomputación y Visualización de Madrid (CeSViMa) and the Spanish Supercomputing Network. This work was supported by the Madrid Regional Education Ministry and the European Social Fund and financed by the Spanish Ministry of Science TIN2007- 67148.

References

[1] C. W. Ahn, D. E. Goldberg, and R. S. Ramakrishna. Multiple-deme parallel estimation of distribution algorithms: Basic framework and application. *Parallel Processing and Applied Mathematics*, pages 544–551, 2004.

[2] E. Alba. Parallel evolutionary algorithms can achieve super-linear performance. *Information Processing Letters*, 82(1):7–13, 2002.

[3] E. Alba and J. M. Troya. *Applied Intelligence*, 12(3):163–181, 2000.

[4] E. Alba and J. M. Troya. Improving flexibility and efficiency by adding parallelism to genetic algorithms. *Statistics and Computing*, 12(2):91–114, 2002.

[5] E. Cantú-Paz. *Efficient and accurate parallel genetic algorithms*. Kluwer Academic Publishers, 2001.

[6] J. P. Cohoon, S. U. Hegde, W. N. Martin, and D. S. Richards. Punctuated equilibria: A parallel genetic algorithm. In J. J. Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms*, pages 148–154, Cambridge, MA, USA, July 1987.

[7] L. Delaossa, J. A. Gamez, and J. Puerta. Initial approaches to the application of islands-based parallel edas in continuous domains. *Journal of Parallel and Distributed Computing*, 66(8):991–1001, August 2006.

[8] L. Delaossa, J. A. Gámez, and J. M. Puerta. Migration of probability models instead of individuals: An alternative when applying the island model to edas. In *PPSN VIII*, pages 242–252, 2004.

[9] L. Delaossa, J. A. Gamez, and J. M. Puerta. Initial approaches to the application of islands-based parallel edas in continuous domains. In *Int. Conf. Workshops on Parallel Processing*, pages 580–587, 2005.

[10] J. Jaros and J. Schwarz. Parallel bmda with probability model migration. In *Proceeding of the IEEE Congress on Evolutionary Computation*, pages 1059–1066, 2007.

[11] J. Jaros and J. Schwarz. Parallel bmda with an aggregation of probability models. In *Proceeding of the IEEE Congress on Evolutionary Computation*, pages 1683–1690. IEEE Computational Intelligence Society, 2009.

[12] P. Larrañaga, R. Etxeberria, J. Lozano, and J. Peña. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.

[13] P. Larrañaga, R. Etxeberria, J. Lozano, and J. Peña. Optimization in continuous domains by learning and simulation of Gaussian networks. In *Proceedings of the Workshop in Optimization by Building and using Probabilistic Models. A Workshop within the 2000 Genetic and Evolutionary Computation Conference, GECCO 2000*, pages 201–204, Las Vegas, Nevada, USA, 2000.

[14] P. Larrañaga and J. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2002.

[15] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[16] J. Madera, E. Alba, and A. Ochoa. A parallel island model for estimation of distribution algorithms. pages 159–186. 2006.

[17] S. Muelas, J. M. Peña, V. Robles, A. Latorre, and P. de Miguel. Machine learning to analyze migration parameters in parallel genetic algorithms. In *Innovations in Hybrid Intelligent Systems*, volume 44 of *Advances in Soft Computing*, pages 199–206, November 2007.

[18] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions i. binary parameters. In *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, 1996.

[19] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[20] M. Pelikan, K. Sastry, and E. Cantú-Paz. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications (Studies in Computational Intelligence)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[21] C. Petty and M. Leuze. A theoretical investigation of a parallel genetic algorithm. In *Proceedings of the 3rd International Conference on Genetic algorithms*, pages 398–405, 1989.

[22] K. Tang, X. Yao, P. Suganthan, C. MacNish, Y. Chen, C. Chen, and Z. Yang. Benchmark functions for the cec 2008 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, 2007.

[23] D. Whitley, S. Rana, and R. Heckendorn. The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47, 1999.

[24] A. A. Zhigljavsky. *Theory of Global Random Search*. Kluwer Academic Publishers, 1991.

Table 3. Average Rankings and NWins 50-D

Size	#islands	period	topology	rate	emm. selec	model	avg. ranking	nWins
Best Configurations								
512	8	20	ring2	0.10	best	-	59.09	404
512	8	20	ring2	0.20	best	-	64.27	395
512	8	10	ring2	0.20	best	-	64.59	398
512	8	10	ring2	0.10	best	-	65.45	384
1024	16	10	ring2	0.20	best	-	65.55	372
512	8	20	ring2	0.05	best	-	66.45	381
1024	16	10	ring2	0.10	best	-	70.27	366
512	8	40	ring2	0.05	random	-	71.45	352
512	16	10	ring2	0.10	best	-	74.64	359
512	8	40	a2a	0.05	best	-	74.91	365
Sequential Configurations								
512							147.36	185
256							171.55	132
1024							174.23	89
128							224.18	-21
2048							301.73	-236
64							320.91	-225
Worst Configurations								
512	32	40	a2a	-	-	adaptative	384.36	-370
512	32	10	ring2	-	-	adaptative	384.81	-360
512	32	10	ring2	-	-	adaptative	386.63	-369
512	32	40	a2a	0.20	best	-	388.81	-399
512	32	40	ring2	-	-	adaptative	391.0	-395
512	32	20	a2a	-	-	weightedrandom	391.18	-409
512	32	20	a2a	0.20	best	-	395.54	-396
512	32	20	ring2	-	-	weightedrandom	396.0	-424
512	32	40	a2a	-	-	weightedrandom	402.18	-427
512	32	40	ring2	-	-	weightedrandom	405.90	-431

Table 4. Average Rankings and NWins 100-D

Size	#islands	period	topology	rate	emm. selec	model	avg. ranking	nWins
Best Configurations								
1024	16	10	ring2	0.20	best	-	73.32	400
1024	16	10	ring2	0.10	best	-	75.32	401
512	8	20	ring2	0.05	best	-	80.73	368
512	8	20	ring2	0.10	best	-	80.95	383
1024	16	10	ring2	0.05	best	-	81.05	391
512	8	10	ring2	0.10	best	-	83.91	373
1024	16	20	ring2	0.20	best	-	84.68	385
1024	16	20	ring2	0.10	best	-	85.41	381
1024	8	20	ring2	0.10	best	-	86.50	358
1024	16	10	ring2	0.20	random	-	89.41	361
Sequential Configurations								
512							151.45	151
1024							172.05	73
256							216.18	6
128							243.41	-35
2048							243.45	-120
64							350.00	-264
Worst Configurations								
512	32	10	ring2	-	-	adaptative	389.72	-420
512	32	40	a2a	-	-	adaptative	389.90	-422
512	32	10	a2a	-	-	adaptative	390.18	-422
512	32	20	a2a	-	-	weightedrandom	390.81	-397
512	32	40	ring2	-	-	adaptative	392.18	-426
512	32	20	ring2	-	-	weightedrandom	397.18	-411
512	32	40	a2a	0.20	best	-	401.72	-401
512	32	40	a2a	-	-	weightedrandom	402.81	-422
512	32	20	a2a	0.20	best	-	405.45	-400
512	32	40	ring2	-	-	weightedrandom	408.18	-425

Table 5. Average ranking and nWins per function on 50-D functions

Function	0.05-best	0.05-random	0.10-best	0.10-random	0.20-best	0.20-random	adaptative	uniform
f1	**148.28**/ 7	159.14/ 5	167.52/ 3	171.68/ 1	189.30/-1	191.56/-3	392.98/-7	311.55/-5
f2	186.02/ 2	180.56/ 2	178.91/ 2	**176.63**/ 2	190.33/ 2	192.22/ 2	344.74/-7	282.59/-5
f3	152.24/ 4	**148.46**/ 5	167.85/ 3	162.20/ 4	194.17/-2	188.72/-2	390.33/-7	328.02/-5
f4	226.54/-1	242.52/-1	224.33/-1	240.13/-1	223.85/-1	232.57/-1	**147.33**/ 6	194.72/ 0
f5	**155.43**/ 5	172.28/ 3	168.63/ 4	169.20/ 3	183.19/ 0	199.00/-3	390.67/-7	293.61/-5
f6	**164.03**/ 5	172.13/ 2	171.04/ 4	177.35/ 3	189.16/ 0	194.36/-2	384.09/-7	279.84/-5
f7	178.97/ 3	200.03/ 0	**174.39**/ 4	193.23/ 2	189.84/ 3	204.58/ 0	299.67/-6	291.29/-6
f8	245.02/-1	236.91/-1	244.61/-1	237.04/-1	238.07/-1	238.26/-1	**38.35**/ 7	253.74/-1
f9	195.28/ 2	203.39/ 2	**188.74**/ 2	196.91/ 2	201.38/ 2	207.47/ 1	282.61/-6	256.22/-5
f10	**179.08**/ 4	191.60/ 2	179.37/ 2	183.19/ 2	200.49/ 1	197.31/ 1	316.43/-7	284.53/-5
f11	194.30/ 2	202.44/ 2	**189.81**/ 2	198.29/ 2	200.19/ 2	207.07/ 1	283.50/-6	256.40/-5
All functions	**187.03**/6	195.23/2	189.95/4	194.95/3	203.41/0	208.86/-3	269.55/-6	282.97/-6

Table 6. Average ranking and nWins per function on 100-D functions

Function	0.05-best	0.05-random	0.10-best	0.10-random	0.20-best	0.20-random	adaptative	uniform
f1	**149.31**/ 7	158.60/ 5	170.72/ 1	171.85/ 3	185.56/-2	191.32/-2	395.28/-7	309.36/-5
f2	189.17/ 2	178.65/ 2	183.52/ 2	**177.93**/ 3	194.87/ 2	191.96/ 1	341.76/-7	274.15/-5
f3	151.15/ 5	**138.41**/ 6	171.50/ 2	157.11/ 3	198.06/-2	191.31/-2	392.76/-7	331.70/-5
f4	220.57/-1	240.44/-1	222.94/-1	229.76/-1	225.07/-1	220.70/-1	**167.11**/ 6	205.39/ 0
f5	159.45/ 5	**157.99**/ 5	171.53/ 4	170.94/ 2	189.39/-2	185.95/-2	395.17/-7	301.57/-5
f6	**165.31**/ 4	169.90/ 4	166.02/ 4	170.59/ 4	194.46/-2	193.75/-2	391.48/-7	280.48/-5
f7	173.10/ 3	180.47/ 2	**163.78**/ 3	176.88/ 2	189.40/ 3	197.87/-1	348.37/-7	302.13/-5
f8	234.91/ 0	258.20/-2	238.33/ 0	246.91/-1	231.57/ 0	235.30/ 1	**27.50**/ 7	259.28/-5
f9	**180.44**/ 3	200.31/ 1	181.66/ 2	198.02/ 2	198.04/ 2	202.07/ 2	330.22/-7	241.23/-5
f10	**176.44**/ 3	179.05/ 3	185.53/ 2	187.92/ 3	199.70/ 2	200.10/-1	333.58/-7	269.68/-5
f11	**182.57**/ 2	198.69/ 2	182.81/ 2	196.46/ 2	199.33/ 2	201.28/ 2	330.35/-7	240.50/-5
All functions	**185.55**/5	193.41/2	190.62/4	195.34/3	206.26/-1	207.09/-1	269.72/-6	283.97/-6

Table 7. Common Configurations Analysis 50-D

Size	#is.	per.	top.	0.05-best	0.05-random	0.10-best	0.10-random	0.20-best	0.20-random	adaptative	uniform
512	8	10	a2a	**94.00**/ 4	118.50/ 2	136.73/ 0	117.95/ 3	158.68/-3	96.64/ 2	343.18/-7	151.86/-1
512	8	10	ring2	75.32/ 3	108.82/-3	**63.55**/ 3	80.59/ 2	63.82/ 4	76.64/ 2	290.27/-7	161.32/-4
512	8	20	a2a	102.18/ 5	**96.55**/ 2	132.77/ 2	123.14/ 0	146.91/ 0	143.41/ 0	289.64/-7	180.00/-2
512	8	20	ring2	64.36/ 2	111.64/-1	**57.64**/ 5	103.73/-1	62.82/ 4	90.95/ 3	257.91/-7	192.05/-5
512	8	40	a2a	**72.82**/ 6	92.64/ 4	139.82/ 2	155.68/ 0	175.77/ 2	189.64/-2	265.64/-7	216.23/-5
512	8	40	ring2	101.09/ 0	**68.82**/ 3	73.27/ 3	76.91/ 2	88.64/ 2	105.36/ 2	228.77/-6	215.68/-6
512	16	10	a2a	165.23/ 3	**121.14**/ 6	222.64/-1	174.45/ 4	259.23/-4	193.55/ 0	363.64/-7	241.00/-1
512	16	10	ring2	128.86/ 0	119.86/-2	**72.73**/ 4	78.91/ 3	83.64/ 5	94.91/ 2	354.91/-7	280.82/-5
512	16	20	a2a	151.91/ 2	**97.36**/ 7	222.09/-1	168.64/ 4	247.59/-2	228.64/ 1	351.18/-7	308.27/-4
512	16	20	ring2	102.91/ 2	134.95/-2	**81.64**/ 5	90.77/ 3	89.27/ 3	113.14/ 1	347.64/-7	336.18/-5
512	16	40	a2a	160.00/ 4	**116.09**/ 7	243.27/-1	198.09/ 4	308.18/-5	278.45/ 0	352.55/-5	352.91/-4
512	16	40	ring2	158.82/ 0	177.82/ 0	**133.00**/ 6	133.27/ 6	169.82/ 0	175.00/ 0	351.82/-6	354.09/-6
512	32	10	a2a	288.14/ 1	**255.86**/ 6	288.68/ 2	258.23/ 6	358.45/-4	350.64/-2	378.09/-7	338.00/-2
512	32	10	ring2	181.18/ 4	210.73/-1	186.18/ 4	206.82/ 0	**180.64**/ 4	196.82/ 1	379.55/-7	376.00/-5
512	32	20	a2a	263.55/ 1	**218.45**/ 6	276.77/ 0	235.32/ 5	390.00/-4	360.45/ 1	374.64/-4	385.64/-5
512	32	20	ring2	236.41/ 4	257.09/ 0	**236.09**/ 4	261.00/ 0	261.73/ 2	268.18/ 2	381.36/-6	390.45/-6
512	32	40	a2a	317.50/ 2	**295.41**/ 4	315.86/ 2	297.55/ 6	383.55/-2	374.64/ 0	379.09/-6	396.64/-6
512	32	40	ring2	**319.73**/ 4	324.45/ 1	321.45/ 2	323.18/ 1	327.45/ 2	329.36/ 2	385.73/-6	400.36/-6
1024	16	10	a2a	104.82/ 2	**104.00**/ 2	104.55/ 2	100.82/ 2	105.18/ 2	107.00/ 2	345.00/-7	158.00/-5
1024	16	10	ring2	82.36/ 1	118.05/-1	68.55/ 5	111.23/-1	**64.55**/ 5	91.36/ 3	281.91/-6	200.82/-6
1024	16	20	a2a	115.36/ 2	125.55/ 2	**113.09**/ 2	121.91/ 2	122.82/ 2	120.55/ 2	295.09/-7	210.27/-5
1024	16	20	ring2	112.77/ 0	114.32/ 1	96.50/ 3	110.41/ 2	**88.18**/ 5	114.32/ 0	235.59/-5	234.82/-6
1024	16	40	a2a	**113.23**/ 4	142.32/ 1	155.86/ 1	136.41/ 2	195.91/-1	204.73/-2	261.64/-2	239.36/-3
1024	16	40	ring2	154.64/-3	133.05/ 1	130.23/ 2	**118.59**/ 2	136.32/ 1	127.32/ 2	230.14/-1	246.45/-4
1024	32	10	a2a	153.27/ 3	**103.36**/ 5	199.36/ 0	129.45/ 5	216.27/-4	178.36/ 1	361.36/-7	255.00/-3
1024	32	10	ring2	175.95/-1	219.45/-3	109.09/ 4	144.32/ 1	**92.77**/ 7	130.95/ 4	351.55/-7	297.18/-5
1024	32	20	a2a	133.05/ 3	**131.45**/ 3	195.64/ 3	168.05/ 3	261.59/-5	204.64/ 3	346.18/-6	312.27/-4
1024	32	20	ring2	225.18/-1	241.73/-3	**174.32**/ 5	184.68/ 3	179.86/ 4	182.50/ 4	345.09/-7	343.00/-5
1024	32	40	a2a	222.41/ 1	177.77/ 7	266.41/ 1	**230.05**/ 3	301.82/-3	289.55/-1	347.18/-3	353.27/-5
1024	32	40	ring2	267.73/-2	276.73/-2	**248.64**/ 7	253.73/ 4	258.55/ 4	261.18/ 1	350.00/-6	360.09/-6
1024	8	10	a2a	**116.18**/ 1	137.64/ 1	140.27/ 1	139.91/ 1	136.36/ 1	128.09/ 1	297.36/-7	154.32/ 1
1024	8	10	ring2	113.64/ 2	148.86/-3	103.82/ 2	135.95/ 0	**92.36**/ 2	117.09/ 0	201.73/ 0	156.59/-3
1024	8	20	a2a	**122.91**/ 2	143.05/ 1	132.18/ 2	154.95/-1	141.18/-2	139.36/ 0	196.00/ 0	161.68/-2
1024	8	20	ring2	123.64/-1	122.23/ 2	107.18/ 0	123.68/ 1	**106.18**/ 3	134.50/-3	204.64/ 0	181.27/-2
1024	8	40	a2a	**115.82**/ 2	138.05/ 0	129.00/ 2	143.59/ 1	140.09/ 1	158.95/-2	198.09/ 0	195.27/-4
1024	8	40	ring2	134.32/-1	129.50/ 0	127.95/ 0	**123.68**/ 0	124.45/ 1	126.77/ 2	196.73/ 0	207.09/-2
2048	16	10	a2a	191.14/ 1	233.36/-2	188.23/ 6	217.45/ 0	**187.14**/ 6	208.36/-1	336.64/-4	263.14/-6
2048	16	10	ring2	219.55/ 2	258.18/-4	204.09/ 4	243.59/-1	**192.91**/ 6	236.91/-1	254.05/ 0	282.05/-6
2048	16	20	a2a	**203.55**/ 4	245.41/-2	209.55/ 4	243.18/-2	219.91/ 1	231.45/ 0	248.64/ 0	284.27/-5
2048	16	20	ring2	260.45/-2	266.45/-3	243.18/ 2	255.18/-3	**236.45**/ 6	249.00/ 1	267.82/ 0	292.73/-1
2048	16	40	a2a	**243.95**/ 3	254.64/ 0	253.36/ 2	257.82/ 0	264.27/ 2	277.77/-2	254.09/ 0	296.77/-5
2048	16	40	ring2	279.18/-3	275.18/-1	**266.77**/ 3	269.55/ 1	269.73/ 1	271.55/ 3	272.27/ 0	303.64/-4
2048	32	10	a2a	195.23/ 2	208.18/ 1	197.14/ 2	**191.45**/ 3	206.36/ 2	202.95/ 2	347.18/-6	287.45/-6
2048	32	10	ring2	235.27/ 1	284.64/-4	208.05/ 4	261.18/-2	**195.09**/ 6	249.32/ 1	289.64/ 0	315.64/-6
2048	32	20	a2a	**202.23**/ 4	246.27/ 0	211.91/ 4	245.82/-2	222.45/ 0	231.95/ 0	307.64/ 0	316.64/-6
2048	32	20	ring2	284.73/-2	291.45/ 0	268.95/ 1	277.36/ 2	**259.86**/ 4	273.64/ 1	278.18/ 0	326.91/-6
2048	32	40	a2a	**259.36**/ 4	276.09/-1	284.18/ 1	289.09/-1	291.23/ 0	302.45/-2	266.64/ 0	329.36/-1
2048	32	40	ring2	314.18/-3	303.36/ 1	308.09/ 0	**297.55**/ 1	306.00/ 2	305.00/ 1	298.18/ 0	332.55/-2
2048	8	10	a2a	207.82/ 3	261.09/-3	203.77/ 4	260.36/-4	**201.45**/ 5	241.91/-1	282.36/ 0	275.64/-4
2048	8	10	ring2	250.45/-1	258.36/-2	**220.55**/ 4	261.18/-2	208.09/ 6	248.00/ 1	255.95/ 0	280.64/-6
2048	8	20	a2a	247.82/ 2	267.91/-4	240.23/ 3	262.86/-3	**236.41**/ 3	251.91/ 3	236.59/ 0	280.09/-4
2048	8	20	ring2	254.50/ 0	263.00/ 0	256.59/ 0	258.27/ 0	**247.45**/ 2	252.55/ 3	257.27/ 0	286.41/-5
2048	8	40	a2a	**253.36**/ 2	262.18/-1	257.23/ 2	264.32/ 1	269.59/ 1	274.91/-2	262.50/ 0	291.50/-3
2048	8	40	ring2	267.73/ 0	266.45/ 1	**260.45**/ 2	265.91/ 1	264.82/ 1	267.64/ 1	261.68/ 1	297.18/-7

Table 8. Sum of the Results from the Common Configuration Analysis

Dimensions	0.05-best	0.05-random	0.10-best	0.10-random	0.20-best	0.20-random	adaptative	uniform
50	**184.11** /1.44	191.77 /0.54	186.84/2.43	191.44 /1.24	200.00/1.50	204.83 /0.74	297.34/-3.63	275.68/-4.26
100	**180.22**/2.09	187.34/0.94	185.30/2.39	189.49/1.57	200.50/1.04	201.06/0.63	313.96/-4.39	274.13/-4.28