2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

Dynamically updated Region Based Memetic Algorithm for the 2013 CEC Special Session and Competition on Real Parameter Single Objective Optimization

Benjamin Lacroix
Universidad de Granada,
Department of Computer Science and Artificial Intelligence
Email: benjamin@decsai.ugr.es

Daniel Molina
Universidad of Cádiz,
Department of Computer Science and Engineering
Email: daniel.molina@uca.es

Francisco Herrera
Universidad de Granada,
Department of Computer Science and Artificial Intelligence
Email: herrera@decsai.ugr.es

***Abstract*—In this paper, we present a memetic algorithm which combines in a local search chaining framework, a steady-state genetic algorithm as evolutionary algorithm and a CMA-ES as local search method. It is an extension of an already presented algorithm which uses a region-based niching strategy and which has proven to be very efficient on real parameter optimisation problems. In this new version, we propose to dynamically update the niche size in order to make it less dependent to such critical parameter. In addition, we used an automatic configuration tool to optimise its parameters, and show that the optimised version of this algorithm is significantly better than with its default parameters. We tested this algorithm on the Special Session and Competition on Real-Parameter Optimization of the IEEE Congress on Evolutionary 2013 benchmark.**

I. Introduction

One of the main issues when designing an evolutionary algorithm (EA) [1] for real-coded parameter optimisation problems is to offer a good exploration of the search space and, at the same time, to exploit the most promising regions to obtain high quality solutions. Memetic algorithms (MA) were proposed [2] to manage these competing objectives. They 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 combination of an exploration algorithm and a local improvement algorithm.

MAs with an appropriate trade-off between the exploration and exploitation can obtain accurate solutions, improving the search [3], [4]. Therefore, the key issue when designing a MA is to organise both efforts in the most cooperative way.

From those remarks, we propsed in [5] an MA, the Region-Based Memetic Algorithm with LS chaining and CMA-ES (RMA-LSCh-CMA) which includes a novel niching strategy. Niching strategies have been used in EA to either identify various optima in a fitness landscape or to maintain a strong diversity in the EA's population [6]. In our case, we which to maintain a higher diversity in the population in order to let the EA focus on the exploration task by limiting its exploitation power, this task being more efficiently performed by the LS method. Contrarily to most niching strategies where the niches are defined around the solutions of the population, the niches are predefined as divisions of the search space. The search space is divided into equal hypercubes each of which represent one exclusion region, not allowing more than one solution in each one. Also, the LS method is initialised to explore inside these regions. This way, there is no competition between the EA and the LS method.

In this paper, we extend the RMA-LSCh-CMA in order to obtain more robustness by updating dynamically the niche size along the search. The niche size decreases in order to maintain a high diversity in the early stages of the search and reduce it along the process. The resulting algorithm is named DRMA-LSCh-CMA.

Considering the number of parameters implied by such a model, we perform the automatic configuration of this algorithm using IRACE [7] and compare the results obtained with the default parameters. We thus tuned and tested DRMA-LSCh-CMA on the benchmark of the Special Session and Competition on Real Parameter Single Objective Optimization of the IEEE Congress on Evolutionary 2013 (CEC'2013) benchmark.

This paper is structured as follows. Section II describes in details the DRMA-LSCh-CMA and its components. Sectiom III briefly states the experimental setup corresponding to the requirements of the competition. In Section IV, we give a quick introduction on the automatic tuning tools and especially the one we applied here, IRACE as well as a description of the experiments we made with it. Finally, in Section V we present and compare the results obtained by the DRMA-LSCh-CMA with both default and the tuned parameters on the CEC'2013 benchmark.

II. The DRMA-LSCh-CMA

DRMA-LSCh-CMA stands for Dynamically updated Region-based Memetic Algorithm with Local Search Chaining and CMA-ES. This algorithm is an extension of the RMA-LSCh-CMA in which the niche size is dynamically updated along the search.

In this section we describe each concepts and components that forms this model starting by stating the general scheme of a MA with LS chaining. We then describe the concept region-based niching strategy and the different components (EA and LS) of the DRMA-LSCh-CMA and how they are adapted the notion of regions. Finally, we present the modification made to the RMA-LSCh-CMA explaining the motivations of using a dynamic niche size and how it is implemented.

A. General scheme

The general scheme of the DRMA-LSCh-CMA follows the MA with LS chaining as stated in [8]. The concept of LS chaining proposed in that paper is based on the idea that the LS should be applied with higher intensity on the most promising solutions. Promising solutions are identified as the ones maintained the most time in the population for the quality of their fitness.

The DRMA-LSCh-CMA is a steady state MA which alternatively applies an EA and an LS method. This hybridisation model allows the same solution to improve several times, creating *LS chains*. Such LS chains are obtained by storing with the solution the final state of the LS parameters after each LS application. This way, the final state of a LS application over a solution can be reused as the initial point of a subsequent LS application over the same solution. The general scheme can be seen in Algorithm 1.

Algorithm 1 Pseudocode of LS chaining MA

1: Generate the initial population
2: **while** not termination-condition **do**
3: Perform the EA with n_{frec} evaluations
4: Build the set S_{LS} of individuals which can be refined by LS
5: Pick the best individual c_{LS} in S_{LS}
6: **if** c_{LS} belongs to an existing LS chain **then**
7: Initialise the LS operator with the LS state stored with c_{LS}
8: **else**
9: Initialise the LS operator with the default LS parameters
10: **end if**
11: Apply the LS algorithm to c_{LS} with I_{str} evaluations, giving c^r_{LS}
12: Replace c_{LS} by c^r_{LS}
13: Store the final LS state with c^r_{LS}
14: **end while**

To select the individual c_{LS} to which the LS will be applied, the following process is used (steps 4 and 5 in Algorithm 1):

- The set S_{LS} is build with the individuals of the population that:
  1) have never been improved by the LS.
  2) have been improved by the LS but with an improvement (in fitness) superior to δ^{min}_{LS}.
- If $|S_{LS}| \neq 0$, the LS is applied on the best individual in S_{LS}. If S_{LS} is empty, the whole population is reinitialised except for the best individual which is maintained in the population.

B. A region-based MA

In this section, we state the notion of the use of a region-based niching strategy in an MA framework.

Niching strategy consists in creating an area around the solutions of an EA's population where no other solution can be present. The main purpose of this tool is to maintain the diversity of the population at a higher level. Maintaining the diversity in a population prevents a fast convergence of the population and allows a better exploration of the search space. This notion is particularly interesting in MA as an EA's first task is to explore, the exploitation of the solutions being done by the LS method. In other words, via this strategy, we offer a clearer separation between the exploration effort done by the EA and the exploitation task of the LS method.

Contrarily to most niching strategies where the niches are defined by the area surrounding solutions of the population, in a region-based niching strategy, 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 (niches) which can contain only one solution.

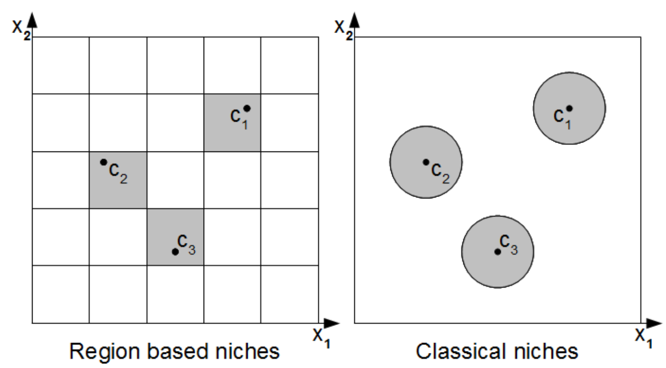

Figure 1. Different niching strategy

C. The components of the DRMA-LSCh-CMA

Here we describe both entities that we use in our algorithm, the EA and the LS and how they have been adapted to the region-based niching strategy. Both components have been kept from the original MA with LS chaining [8].

1) The EA: The EA used is a steady-state genetic algorithm (SSGA) specifically designed to promote high population diversity levels by means of the combination of the $BLX-\alpha$ crossover operator [9] and the *negative assortative mating* strategy (NAM) [10]. Diversity is favoured as well by means of the breeder genetic algorithm (BGA) mutation operator [11].

The replacement strategy used is *Replacement Worst, RW*. The combination *NAM-RW* produces a high selective pressure.

One of the key issues in niching strategies is to decide what to do with a solution generated in the exclusion area of an other solution. The original SSGA has been modified and it is describe in Algorithm 2. It consists in not allowing the generation of a solution in a region that is already occupied by an optimised solution of the population. By optimised, we refer to the fact that it was previously applied the LS over this solution, and the last LS applied has not brought enough improvements (upper than δ_{LS}^{min}). Then, if a solution is optimised, we consider its neighbourhood (and by consequence the region it lies in) has sufficiently been explored. 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. That way, we avoid unnecessary LS evaluations within the region to get a higher quality solutions. This way, we ensure that the population does not hold two solutions in the same region.

Algorithm 2 Pseudo-code for the region-based SSGA

1: Randomly generate the population
2: **while** not termination-condition **do**
3:   Select two parents in the population
4:   Create an offspring c_n using crossover and mutation
5:   **if** c_n falls in a region containing an individual c_o **then**
6:     **if** c_o is considered optimised **then**
7:       Mutate c_n using the BGA mutation and go back to 5
8:     **end if**
9:   **end if**
10:   **if** c_n falls in a region containing an individual c_o **then**
11:     Replace c_o by c_n if $f(c_o) > f(c_n)$
12:   **else**
13:     Replace the worst individual c_{worst} in the population if $f(c_{worst}) > f(c_n)$
14:   **end if**
15: **end while**

2) The LS: The continuous LS algorithm is CMA-ES [12]. This algorithm is the *state-of-the-art* in continuous optimisation. Thanks to the adaptability of its parameters, its convergence is very fast and obtains very good results. CMA-ES is an algorithm that uses a distribution function to obtain new solutions, and adapt the distribution around the best created solutions.

In CMA-ES, the initial population is generated using the average of the distribution $\vec{m}$ and the initial standard deviation σ. In our case, the solution to optimise c_{LS} is set as $\vec{m}$. As we wish the close surroundings (*i.e.* the region) to be explored firstly, the initial standard deviation σ is set to half the size of a region.

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 an occupied region, the best solution is kept and the other one is replaced by a randomly generated solution.

D. A dynamic number of divisions

In [5], we saw that the number of division (ND) was a critical parameter and significantly influenced the performances of the algorithm.

In more general terms, one of the main issues when implementing a niching strategy is to define the size of the niche. Here, the size of the region is directly dependent on the number of divisions per dimensions ND.

A high number of divisions leads to smaller niches and thus, a poor influence on the search. On the other hand a small number of divisions creates big niches. The diversity will be high but the chances that the local search fails to reach the best solution in its surroundings are higher.

This motivates the use of a dynamic niche size in order to achieve a better robustness of the algorithm. To do so, the number of divisions is increased along the search. With bigger regions at the beginning of the search, a greater diversity is maintained to ensure a strong exploration of the search space. The number of division is then increased in order to allow a better convergence in later stages of the process.

We have decided to use a linear increase of the number of division. At each update, $ND_i = m_u \cdot ND_{i-1}$ where m_u is the update multiplier. If the total number of updates is u, an update occurs every $max\_eval/(u+1)$ where max_eval is the maximum number of fitness evaluation allowed. With this strategy, two parameters appear, ND_0, the initial number of divisions and u, the number of updates.

III. Experimental setup

This algorithm has been tested on the Special Session and Competition on Real Parameter Single Objective Optimization of the IEEE Congress on Evolutionary 2013 (CEC'2013) benchmark. This benchmark is composed of 28 functions defined in 3 dimensions $D = \{10, 30, 50\}$. Following the requirements of the competition, each function is executed 51 times and, for our experiments, we get the average of the function error. Every error value smaller than 10^{-8} is considered as zero.

IV. Automatic Configuration

The automatic configuration of algorithms, which is also called automatic *tuning*, is nowadays receiving a strong attention from the research community. In recent years, several methods for the automatic setting of algorithms parameters, have been proposed. Such methods include for instance ParamILS [13], gender-based genetic algorithms [14], CALIBRA [15], SPO [16], SPO$^+$ [17], iterated race [18], [7], or SMAC [19] which extends earlier research efforts [20], [21].

A. Online and Offline Automatic Configuration

The automatic configuration of parameters is often divided into two different approaches.

Automatic parameter configuration can be performed *online* to set the values of the parameters while the algorithm is running. Sometimes referred to as parameter adaptation, this is typically applied to a small subset of key parameters, since it implies a significant overhead for the algorithm to "learn" good values for the parameters, additionally to the exploration of the search space of the instance to be tackled.

In this paper, we use *offline* automatic configuration. In this case, the purpose is to automatically configure optimization algorithms before they are deployed, that is, to find the best possible setting to be used on future instances.

B. The IRACE software package

In this paper, the automatic configuration tool that we use is the `irace` package. Based on previous works [20], [22], [23], [18], [24], it implements an automatic configuration approach based on *racing* [25]. Statistical tests are used to test for significantly inferior candidate configurations. The `irace` package, implemented as an R [26] package, implements a general *iterated racing* procedure. For more details on this tool, the reader can refer to [7].

The `irace` package has already been extensively tested in several research projects, leading to successful improvement over the state-of-the-art, see for instance [27], [28].

The advantage of this tool is that it handles several parameter types: continuous, integer, categorical, and ordered. Continuous and integer parameters take values within a range specified by the user. Categorical parameters can take any value among a set of possible ones explicitly given by the user. An ordered parameter is a categorical parameter with a pre-defined strict order of its possible values. We also relied on its capability to parallelize the configuration phase in order to reduce considerably the amount of time required for it.

C. Applying automatic configuration to the DRMA-LSCh-CMA

We selected a set of parameters that we considered the most critical to be tuned. Those parameters are listed in Table I. The tuning budget allocated to `irace` is set to 5000. The budget corresponds to number of runs in the conditions defined by the benchmark that `irace` uses to perform the tuning.

Table I
PARAMETERS TO BE TUNED

Parameters	Descriptions	Ranges
I_{str}	LS intensity, number of evaluations allocated to each LS application	$[100, 1000]$
ND_0	Initial number of divisions, defines the size of the niches/regions	$[5, 100]$
u	Number of update to be performed	$[2, 5]$
m_u	Update multiplier	$[1, 5]$
$r_{EA/LS}$	The repartition of the overall effort between the EA and the LS the higher the value the more evaluations allocated to the LS	$[0.1, 0.9]$
NP	Population size of the EA	$[40, 120]$
λ	Parameter to define the CMA-ES population size $p = 4 + \lambda ln(D)$	$[1, 10]$
μ	Defines the parent size for the CMA-ES p/μ	$[1, 5]$
α	Parameter for the $BLX - \alpha$ crossover	$[0.1, 0.9]$

V. RESULTS

In this section, we present the parameters obtained by the automatic configuration and compare the results obtained with the default parameters on the CEC'2013 benchmark. We consider the default parameters the ones used in the static version. Only the number of divisions, static in [5] is replaced here by ND_0, u and m_u. In that case, we intuitively set $ND_0 = 10$, $u = 3$ and $m_u = 2$. That way, the range of number of divisions used along the search covers the different static values tested in [5].

The parameters obtained by the automatic configuration of DRMA-LSCh-CMA are listed in Table II. The detailed results of both configurations are shown in Table III. Note that for the purpose of the competition, the complete results including the best, worst, median, mean and standard deviation of the 51 runs obtained by the best configuration are added in the appendix in Tables V, VI and VII

Table II
PARAMETERS OBTAINED BY TUNING

Parameters	Default	Tuned
I_{str}	500	999
ND_0	10	3
u	3	5
m_u	2	3.982
$r_{EA/LS}$	0.5	0.5486
NP	80	82
λ	3	10
μ	2	3
α	0.5	0.3855

In order to prove the significance of the improvements brought by the automatic configuration of the DRMA-LSCh-CMA, we used the Wilcoxon signed rank test (Table IV). When considering each dimension individually, although we do not obtain a significant difference in dimension 50, we achieve significant improvement using the tuned parameters in dimension 10 (with $\alpha = 0.05$) and in dimension 30 (with $\alpha = 0.01$). Finally, when considering the whole (including every dimensions in the test) we can see that we obtain statistically better results thanks to the automatic configuration of our algorithm.

Table IV
WILCOXON COMPARISON OF THE DRMA-LSCH-CMA WITH THE DEFAULT AND THE TUNED PARAMETERS FOR $D = \{10, 30, 50, *\}$

Dimension	Tuned parameters $R+$	Default parameters $R-$	$p - value$
D10	302.5	103.5	0.0225
D30	334.5	71.5	0.0020
D50	252.5	127	0.1414
D*	2602.5	889	0.0001

VI. CONCLUSION

In this paper, we presented the DRMA-LSCh-CMA, a MA using a combining a LS chaining mechanism and a region-based niching strategy. In that extended version of the RMA-

Table III
MEAN RESULTS OF THE DRMA-LSCH-CMA WITH THE DEFAULT AND THE TUNED PARAMETERS ON EACH FUNCTIONS FOR $D = \{10, 30, 50\}$ (THE BEST RESULTS ARE IN BOLD)

D10			D30			D50		
F	Tuned parameters	Default parameters	F	Tuned parameters	Default parameters	F	Tuned parameters	Default parameters
F1	**1.00E-008**	**1.00E-008**	F1	**1.00E-008**	**1.00E-008**	F1	**1.00E-008**	**1.00E-008**
F2	**1.00E-008**	**1.00E-008**	F2	**1.00E-008**	**1.00E-008**	F2	**1.00E-008**	**1.00E-008**
F3	**1.00E-008**	3.48E-006	F3	**5.91E-004**	1.09E+005	F3	**9.95E+003**	1.21E+006
F4	**1.00E-008**	**1.00E-008**	F4	**1.00E-008**	1.19E+004	F4	**1.21E+002**	1.68E+004
F5	**1.00E-008**	1.35E-008	F5	**9.35E-005**	7.31E-003	F5	**4.95E-004**	2.04E-003
F6	**2.69E+000**	**2.69E+000**	F6	**1.00E-008**	5.18E-001	F6	**4.34E+001**	**4.34E+001**
F7	**1.16E-002**	1.02E-001	F7	**1.54E+000**	2.19E+000	F7	1.54E+001	**3.53E+000**
F8	2.04E+001	**2.03E+001**	F8	**2.09E+001**	2.09E+001	F8	**2.11E+001**	2.11E+001
F9	1.21E+000	**1.04E+000**	F9	**8.79E+000**	1.03E+001	F9	1.76E+001	**1.60E+001**
F10	**3.38E-004**	7.58E-003	F10	**2.56E-003**	1.62E-002	F10	**1.89E-003**	1.86E-002
F11	6.05E-001	**5.66E-001**	F11	4.17E+000	**9.19E-001**	F11	6.13E+000	**1.59E+000**
F12	2.99E+000	**2.59E+000**	F12	**1.38E+001**	1.66E+001	F12	**3.33E+001**	4.22E+001
F13	4.76E+000	**3.55E+000**	F13	**2.84E+001**	3.59E+001	F13	**8.29E+001**	9.34E+001
F14	**2.60E+001**	3.79E+001	F14	**3.12E+002**	3.73E+002	F14	**5.08E+002**	7.39E+002
F15	**2.12E+002**	2.92E+002	F15	**1.56E+003**	2.02E+003	F15	**3.32E+003**	4.31E+003
F16	**3.52E-002**	7.58E-002	F16	**2.10E-002**	4.52E-001	F16	**1.06E-002**	2.32E+000
F17	**1.18E+001**	1.32E+001	F17	**3.86E+001**	5.86E+001	F17	**6.63E+001**	1.02E+002
F18	**1.27E+001**	1.51E+001	F18	**4.38E+001**	1.77E+002	F18	**7.83E+001**	3.83E+002
F19	**5.20E-001**	5.44E-001	F19	**2.01E+000**	2.40E+000	F19	**3.39E+000**	4.10E+000
F20	2.21E+000	**2.06E+000**	F20	**9.70E+000**	1.06E+001	F20	**1.87E+001**	2.00E+001
F21	**4.00E+002**	**4.00E+002**	F21	**3.02E+002**	3.10E+002	F21	6.95E+002	**5.14E+002**
F22	**2.54E+001**	3.98E+001	F22	**1.91E+002**	1.98E+002	F22	**2.53E+002**	3.84E+002
F23	**9.84E+001**	1.55E+002	F23	**1.38E+003**	2.21E+003	F23	**3.30E+003**	4.58E+003
F24	**2.00E+002**	2.00E+002	F24	**2.03E+002**	2.04E+002	F24	2.21E+002	**2.09E+002**
F25	**2.00E+002**	2.00E+002	F25	**2.36E+002**	2.46E+002	F25	**2.94E+002**	2.94E+002
F26	**1.07E+002**	1.20E+002	F26	2.28E+002	**2.15E+002**	F26	2.91E+002	**2.56E+002**
F27	**3.00E+002**	3.02E+002	F27	3.64E+002	**3.45E+002**	F27	6.39E+002	**4.79E+002**
F28	**2.96E+002**	**2.96E+002**	F28	2.96E+002	**2.92E+002**	F28	4.59E+002	**4.00E+002**

LSCh-CMA, the size of the regions (niches) is decreased along the search in order to maintain the a high diversity in the EA's population in the early stages of the search and to allow a better convergence in the latter stages. The aim of this improvement is to make the algorithm less dependent on the region size and thus, more robust.

In addition to that, we used the automatic configuration tool IRACE to optimise the numerous parameters of such an algorithm. We tuned and tested our algorithm on the benchmark proposed for the Special Session and Competition on Real Parameter Single Objective Optimization of the CEC 2013. Thanks to the automatic tuning of parameters, we obtained significantly better results than the same algorithm with default parameters.

VII. ACKNOWLEDGEMENT

This work was supported by Research Projects TIN2012-37930-C02-01 and P08-TIC-04173. Benjamin Lacroix acknowledges the support from the MIBISOC network, an Initial Training Network funded by the European Union, grant PITN-GA-2009-238819.

REFERENCES

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

[2] P. Moscato, "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms," Tech. Rep., 1989.

[3] L. Davis, *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[4] D. E. Goldberg and S. Voessner, "Optimizing global-local search hybrids," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 1999, pp. 220–228.

[5] B. Lacroix, D. Molina, and F. Herrera, "Region based memetic algorithm with LS chaining," in *2012 IEEE Congress on Evolutionary Computation*. Ieee, Jun. 2012, pp. 1474–1479.

[6] S. Das, S. Maity, B.-Y. Qu, and P. Suganthan, "Real-parameter evolutionary multimodal optimization — a survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71 – 88, 2011.

[7] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari, "The irace package, iterated race for automatic algorithm configuration," IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011.

[8] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera, "Memetic Algorithms for Continuous Optimisation Based on Local Search Chains," *Evolutionary Computation*, vol. 18, no. 1, pp. 27–63, Jan. 2010.

[9] L. J. Eshelman and J. D. Schaffer, "Real-coded Genetic Algorithms in Genetic Algorithms by Preventing Incest," *Foundation of Genetic Algorithms 2*, pp. 187–202, 1993.

[10] C. F. Agostinho, C. Fernandes, and A. Rosa, "A Study on Non-random Mating and Varying Population Size in Genetic Algorithms Using a Royal Road Function," in *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*. IEEE Press, 2001, pp. 60–66.

[11] H. Mülenbein and D. Schlierkamp-Voosen, "Predictive models for the breeding genetic algorithm in continuous parameter optimization," *Evolutionary Computation*, vol. 1, pp. 25–49, 1993.

[12] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)," *Evolutionary Computation*, vol. 1, no. 11, pp. 1–18, 2003.

[13] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle, "ParamILS: an automatic algorithm configuration framework," *Journal of Artificial Intelligence Research*, vol. 36, pp. 267–306, Oct. 2009.

[14] C. Ansótegui, M. Sellmann, and K. Tierney, "A gender-based genetic algorithm for the automatic configuration of algorithms," in *Principles and Practice of Constraint Programming, CP 2009*, ser. Lecture Notes

in Computer Science, I. P. Gent, Ed. Springer, Heidelberg, Germany, 2009, vol. 5732, pp. 142–157.

[15] B. Adenso-Díaz and M. Laguna, "Fine-tuning of algorithms using fractional experimental design and local search," *Operations Research*, vol. 54, no. 1, pp. 99–114, 2006.

[16] T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation: The New Experimentalism*. Berlin, Germany: Springer, 2006.

[17] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy, "An experimental investigation of model-based parameter optimisation: SPO and beyond," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2009*, F. Rothlauf, Ed. New York, NY: ACM Press, 2009, pp. 271–278.

[18] P. Balaprakash, M. Birattari, and T. Stützle, "Improvement strategies for the F-race algorithm: Sampling design and iterative refinement," in *Hybrid Metaheuristics*, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, M. J. Blesa, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, Eds. Springer, Heidelberg, Germany, 2007, vol. 4771, pp. 108–122.

[19] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization, 5th International Conference, LION 5*, ser. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, 2011.

[20] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, "A racing algorithm for configuring metaheuristics," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002*, W. B. Langdon *et al.*, Eds. Morgan Kaufmann Publishers, San Francisco, CA, 2002, pp. 11–18.

[21] S. P. Coy, B. L. Golden, G. C. Runger, and E. A. Wasil, "Using experimental design to find effective parameter settings for heuristics," *Journal of Heuristics*, vol. 7, no. 1, pp. 77–97, 2001.

[22] M. Birattari, "The problem of tuning metaheuristics as seen from a machine learning perspective," Ph.D. dissertation, Université Libre de Bruxelles, Brussels, Belgium, 2004.

[23] M. Birattari, *Tuning Metaheuristics: A Machine Learning Perspective*, ser. Studies in Computational Intelligence. Berlin/Heidelberg, Germany: Springer, 2009, vol. 197.

[24] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle, "F-race and iterated F-race: An overview," in *Experimental Methods for the Analysis of Optimization Algorithms*, T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, Eds. Berlin, Germany: Springer, 2010, pp. 311–336.

[25] O. Maron and A. W. Moore, "The racing algorithm: Model selection for lazy learners," *Artificial Intelligence Research*, vol. 11, no. 1–5, pp. 193–225, 1997.

[26] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2008, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org

[27] J. Dubois-Lacoste, M. López-Ibáñez, and T. Stützle, "A hybrid TP+PLS algorithm for bi-objective flow-shop scheduling problems," *Computers & Operations Research*, vol. 38, no. 8, pp. 1219–1236, 2011.

[28] J. Dubois-Lacoste, M. López-Ibáñez, and T. Stützle, "Automatic configuration of state-of-the-art multi-objective optimizers using the TP+PLS framework," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2011*, N. Krasnogor and P. L. Lanzi, Eds. New York, NY: ACM Press, 2011, pp. 2019–2026.

APPENDIX
DETAILED RESULTS FOR THE COMPETITION OF THE DRMA-LSCH-CMA

Table V
RESULTS FOR THE DRMA-LSCH-CMA WITH AUTOMATICALLY CONFIGURED PARAMETERS FOR 10D

	Best	Worst	Median	Mean	Std
F1	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F2	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F3	0.00E+000	1.23E-007	0.00E+000	5.07E-009	2.41E-008
F4	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F5	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F6	0.00E+000	9.81E+000	0.00E+000	2.69E+000	4.42E+000
F7	0.00E+000	1.65E-001	5.88E-004	1.16E-002	3.43E-002
F8	2.00E+001	2.05E+001	2.04E+001	2.04E+001	8.36E-002
F9	0.00E+000	3.16E+000	1.08E+000	1.21E+000	8.15E-001
F10	0.00E+000	9.86E-003	0.00E+000	3.38E-004	1.71E-003
F11	0.00E+000	2.98E+000	9.95E-001	6.05E-001	6.92E-001
F12	0.00E+000	5.97E+000	2.98E+000	2.99E+000	1.21E+000
F13	9.95E-001	1.29E+001	4.28E+000	4.76E+000	3.00E+000
F14	1.25E-001	1.59E+002	6.89E+000	2.60E+001	4.43E+001
F15	1.03E+001	4.90E+002	1.86E+002	2.12E+002	1.22E+002
F16	0.00E+000	8.73E-002	3.56E-002	3.52E-002	2.59E-002
F17	1.02E+001	1.39E+001	1.17E+001	1.18E+001	7.90E-001
F18	1.07E+001	1.54E+001	1.24E+001	1.27E+001	1.09E+000
F19	9.88E-003	8.59E-001	5.24E-001	5.20E-001	1.34E-001
F20	1.20E+000	3.12E+000	2.19E+000	2.21E+000	4.46E-001
F21	4.00E+002	4.00E+002	4.00E+002	4.00E+002	5.74E-014
F22	5.10E+000	1.04E+002	1.87E+001	2.54E+001	2.05E+001
F23	1.28E+001	4.42E+002	6.17E+001	9.84E+001	9.44E+001
F24	2.00E+002	2.07E+002	2.00E+002	2.00E+002	1.04E+000
F25	2.00E+002	2.01E+002	2.00E+002	2.00E+002	1.26E-001
F26	1.00E+002	2.00E+002	1.03E+002	1.07E+002	1.91E+001
F27	3.00E+002	3.00E+002	3.00E+002	3.00E+002	2.73E-004
F28	1.00E+002	3.00E+002	3.00E+002	2.96E+002	2.80E+001

Table VI
RESULTS FOR THE DRMA-LSCH-CMA WITH AUTOMATICALLY CONFIGURED PARAMETERS FOR 30D

	Best	Worst	Median	Mean	Std
F1	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F2	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F3	0.00E+000	2.10E-002	0.00E+000	5.91E-004	3.18E-003
F4	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F5	2.37E-008	5.71E-004	4.06E-005	9.35E-005	1.34E-004
F6	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F7	5.58E-002	1.06E+001	8.20E-001	1.54E+000	2.16E+000
F8	2.09E+001	2.10E+001	2.10E+001	2.09E+001	4.15E-002
F9	4.24E+000	1.45E+001	9.21E+000	8.79E+000	2.08E+000
F10	0.00E+000	2.22E-002	0.00E+000	2.56E-003	4.70E-003
F11	9.95E-001	6.96E+000	3.98E+000	4.17E+000	1.39E+000
F12	6.96E+000	1.99E+001	1.39E+001	1.38E+001	2.94E+000
F13	1.46E+001	4.69E+001	2.80E+001	2.84E+001	7.82E+000
F14	1.85E+001	6.51E+002	2.83E+002	3.12E+002	1.67E+002
F15	5.46E+002	2.64E+003	1.52E+003	1.56E+003	4.53E+002
F16	6.14E-003	5.90E-002	1.95E-002	2.10E-002	1.07E-002
F17	3.41E+001	4.57E+001	3.74E+001	3.86E+001	3.18E+000
F18	3.68E+001	5.60E+001	4.32E+001	4.38E+001	3.48E+000
F19	1.42E+000	2.68E+000	1.96E+000	2.01E+000	3.04E-001
F20	7.65E+000	1.13E+001	9.71E+000	9.70E+000	8.46E-001
F21	2.00E+002	4.44E+002	3.00E+002	3.02E+002	7.81E+001
F22	6.12E+001	4.05E+002	1.67E+002	1.91E+002	6.85E+001
F23	3.98E+002	2.66E+003	1.42E+003	1.38E+003	4.43E+002
F24	2.00E+002	2.29E+002	2.00E+002	2.03E+002	5.33E+000
F25	2.00E+002	2.66E+002	2.48E+002	2.36E+002	2.40E+001
F26	2.00E+002	3.25E+002	2.00E+002	2.28E+002	4.77E+001
F27	3.00E+002	5.56E+002	3.23E+002	3.64E+002	8.40E+001
F28	1.00E+002	3.00E+002	3.00E+002	2.96E+002	2.80E+001

Table VII
RESULTS FOR THE DRMA-LSCH-CMA WITH AUTOMATICALLY CONFIGURED PARAMETERS FOR 50D

	Best	Worst	Median	Mean	Std
F1	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F2	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F3	1.60E-008	2.70E+005	2.86E-002	9.95E+003	3.97E+004
F4	0.00E+000	2.87E+003	0.00E+000	1.21E+002	5.34E+002
F5	1.66E-004	1.76E-003	4.49E-004	4.95E-004	2.31E-004
F6	4.34E+001	4.34E+001	4.34E+001	4.34E+001	3.51E-009
F7	2.07E-001	4.02E+001	1.06E+001	1.54E+001	1.37E+001
F8	2.11E+001	2.12E+001	2.11E+001	2.11E+001	3.45E-002
F9	1.04E+001	2.36E+001	1.77E+001	1.76E+001	2.81E+000
F10	0.00E+000	7.40E-003	0.00E+000	1.89E-003	3.26E-003
F11	1.99E+000	1.29E+001	5.97E+000	6.13E+000	2.29E+000
F12	1.79E+001	5.47E+001	3.28E+001	3.33E+001	7.06E+000
F13	5.54E+001	1.20E+002	7.95E+001	8.29E+001	1.91E+001
F14	1.45E+002	1.12E+003	4.97E+002	5.08E+002	2.19E+002
F15	1.71E+003	5.08E+003	3.38E+003	3.32E+003	7.58E+002
F16	4.05E-003	1.93E-002	1.07E-002	1.06E-002	3.32E-003
F17	5.64E+001	7.93E+001	6.56E+001	6.63E+001	4.76E+000
F18	6.94E+001	8.95E+001	7.82E+001	7.83E+001	4.64E+000
F19	2.25E+000	4.31E+000	3.39E+000	3.39E+000	3.80E-001
F20	1.73E+001	2.04E+001	1.88E+001	1.87E+001	8.18E-001
F21	2.00E+002	1.12E+003	8.36E+002	6.95E+002	4.01E+002
F22	5.17E+001	7.31E+002	2.27E+002	2.53E+002	1.44E+002
F23	1.70E+003	4.88E+003	3.35E+003	3.30E+003	7.24E+002
F24	2.00E+002	2.66E+002	2.18E+002	2.21E+002	1.41E+001
F25	2.72E+002	3.12E+002	2.93E+002	2.94E+002	8.71E+000
F26	2.00E+002	3.52E+002	3.25E+002	2.91E+002	6.28E+001
F27	3.12E+002	8.72E+002	6.83E+002	6.39E+002	1.56E+002
F28	4.00E+002	3.40E+003	4.00E+002	4.59E+002	4.20E+002