A Coral Reefs Optimization Algorithm with Substrate Layers and Local Search for Large Scale Global Optimization

S. Salcedo-Sanz†, C. Camacho-Gómez†, D. Molina‡ and F. Herrera§

†Department of Signal Processing and Communications, Universidad de Alcalá, Madrid, Spain.
‡Department of Computer Science, Universidad de Cádiz, Spain.
§Department of Computer Science and Artificial Intelligence, Universidad de Granada, Spain.
e-mail: sancho.salcedo@uah.es; carlos.camacho@edu.uah.es; daniel.molina@uca.es; herrera@decsai.ugr.es

***Abstract*—This paper presents a new version of the Coral Reefs Optimization (CRO) algorithm to improve its performance in large scale global optimization problems. Specifically, we propose to extend the original CRO with different *substrate layers*, where several exploration operators are defined. This definition allows establishing a competitive co-evolution process within the CRO, which improves the search for optimal solution in large scale optimization problems. The new CRO with substrate layers (CRO-SL) is used in combination with a local search, and the final memetic algorithm obtained has been tested by using a test suite for large scale continuous optimization, showing a robust behavior.**

I. Introduction

In the last years, huge research efforts have been conducted towards solving hard optimization problems that frequently arise in engineering applications. These problems are often characterized by search spaces of nonlinear objective functions with additional requirements, like constraints or a extremely high dimensionality problems. Optimization of problems with a high dimensionality is called Large Scale Global Optimization, LSGO, and it requires a very efficient search, due to the huge increasing of the domain search when the dimensionality increases. In such cases, modern meta-heuristics are excellent options to come up with good solutions within a reasonable computation time.

One important current trend in meta-heuristics design consists of constructing novel hybrid searching mechanisms. These hybrid approaches are based on the mixture of existing meta-heuristic capabilities. Usually, pairs of algorithms are used to form the hybrid approach, such as in [1] where Differential Evolution is mixed with Harmony Search, or in [2] where Differential Evolution and a Simulated Annealing approach are jointly applied to a problem of dynamic economic emissions dispatch. In [3] and [4] Particle Swarm Optimization hybrids with Harmony Search have been described, and their performance analyzed in different optimization problems. In [6] and [5], hybrid evolutionary approaches with harmony search and Particle Swarm Optimization have been described. In [7] a hybrid Genetic Algorithm with Simulated Annealing has been proposed for a specific application of optimal location and rating of distributed energy storage in low voltage smart networks. In a similar context, in [8] a hybrid Harmony Search – Genetic Algorithm is proposed for a problem of finding the optimal operation performance of autonomous micro-grids.

As can be seen in these examples, the idea is to combine different algorithms, with alternative searching capabilities and exploitation procedures, for obtaining new hybrid meta-heuristics which are more powerful than each of the forming algorithms on its own. In this paper we propose a generalization of this idea, based on a modification of a recently proposed evolutionary-type algorithm called Coral Reefs Optimization (CRO) algorithm [9]. The CRO is a meta-heuristic based on simulating the processes that occur in a coral reef, including corals reproduction, fight for the space and larvae setting. Several studies have shown that depending on the type of substrates that form the reef, larvae corals are more or less likely to grow and develop [10]. In this paper we introduce these *substrate layers* in the CRO simulated reef, in order to obtain a general algorithm, which is able to perform co-evolution of different algorithms in just one population. Specifically, each substrate defined in the CRO represents now a different exploration operator (differential evolution search, two-points crossover, harmony search operators, Gaussian mutation, etc.). The larvae formed with these different operators in each substrate go through the entire normal process of the CRO, and fight for the space in the reef. This way, we obtain an algorithm which mixes very different exploration operators within the evolution rules of the CRO, which promotes competitive co-evolution of larvae.

In order to tackle LSGO problems, the proposed CRO with substrate layers (CRO-SL) has been hybridized with a local search method, LS, specially designed for that type of problems, MTS-LS1 [11]. After a certain number of iteration, the best larva obtained by each substrate is improved with the LS method. Because the larvae obtained in each substrate have been developed using a different operation, they have relevant differences that could produce different attraction bases. Also, a reef restarting mechanism has been added to control the diversity collapse in the reef after including the LS.

The performance of the proposed hybrid using CRO-SL and the LS method has been tested by using the LSGO

978-1-5090-0623-6/16/$31.00

test suite proposed in [12], that allows the different authors compare their results within a common benchmark framework. In comparison with the reference algorithm DECC-CG [13], the results obtained by our proposal present interesting improvements, mainly in the hardest functions considered.

The structure of the remainder of the paper is as follows: next section presents the main characteristics of the original CRO, including the different operators and the algorithm's dynamics. Section III-A describes the proposed CRO-SL version, including what we understand for *substrate layer*, and, in this case, how it represents the co-evolution of different searching mechanism with the rules of the CRO. In this section we also describe the hybridization with a local search procedure, that we have included in the algorithm in order to enhance its performance in LSGO. Section IV presents the experimental section of the paper, where the proposed algorithm's performance is evaluated and compared with a reference algorithm. Section V closes the paper by giving some final conclusions and remarks on this research.

II. The Coral Reefs Optimization Algorithm

The CRO is a class of evolutionary algorithm for optimization, recently proposed in [9]. It is based on simulating the corals' reproduction and coral reefs' formation processes. The CRO has been successfully applied to a number of different applications and optimization problems [14]-[19]. Basically, the CRO is a meta-heuristic based on a population of solutions (the reef), which evolve following certain operators inspired in corals' reproduction, such as a modeling of corals' sexual reproduction (broadcast spawning and brooding). After the reproduction stage, the set of formed larvae (namely, newly produced solutions to the problem) attempts to find a place on the reef to develop and further reproduce. This deployment may occur in a free space inside the reef (hole), or in an occupied location, by fighting against the coral currently settled in that place. If larvae are not successful in locating a place to settle after a number of attempts, they are considered as preyed by animals in the reef. Figure 1 shows a pseudo-code of the original CRO algorithm, the interested reader can consult [9] for further details on the basic CRO.

Because of its structure and dynamics, the CRO adopts concepts from Evolutionary Algorithms and Simulated Annealing, but with new variants. The exploration phase of the algorithm is carried out by operators that simulate the reproductive processes of corals. Note that the CRO is defined over an exploitation procedure (the larvae setting), and the type of exploration procedures, i.e. specific mechanism for reproduction (broadcast spawning or brooding) is open. This characteristic opens the possibility of using different types of exploration mechanisms within the CRO, which is the main objective of this paper.

III. Proposal: CRO with substrate layers with Local Search

In this section we present our proposal to extend the CRO and make it stronger for LSGO problems. For this, we combine two components: First, a new CRO algorithm, CRO with substrate layer, that use different procedures to generate the new solutions, increasing the diversity during the search. Then, the best larva obtained in each substrate is improved with a LS method. Finally, a restart mechanism is used.

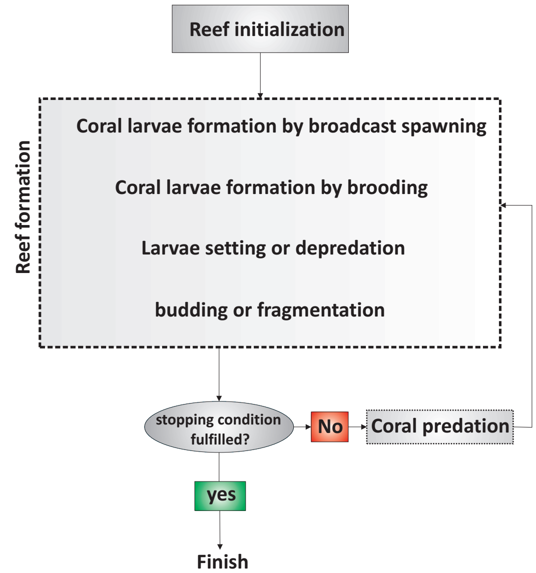

Fig. 1. Original CRO pseudo-code.

A. CRO with substrate layers (CRO-SL)

The original CRO algorithm is based on the main processes of coral reproduction and reef formation that occur in nature. However, there are many more interactions in real reef ecosystem that can be also modelled and incorporated to the CRO approach to improve it. For example, different studies have shown that successful recruitment in coral reefs (i.e., successful settlement and subsequent survival of larvae) depends on the type of substrate on which they fall after the reproduction process [10]. This specific characteristic of the coral reefs was first included in the CRO in [20], in order to solve different instances of the Model Type Selection Problem for energy applications. In [20], different substrate layers were defined in the CRO, in such a way that each layer represents a different model to evaluate the energy demand estimation in Spain, from macro-economic variables. The CRO with substrates is, however, a much more general approach: it can be defined as an algorithm for competitive co-evolution, where each substrate layer represents different processes (different models, operators, parameters, constraints, repairing functions, etc.).

The inclusion of substrate layers in the CRO can be done, in a general way, in a straightforward manner: we redefine the artificial reef considered in the CRO in such a way that each cell of the square grid Ψ representing the reef is now defined by 3 indexes (i, j, t), where i and j stand for the cell location in the grid, and index $t \in T$ defines the substrate layer, by indicating which structure (model, operator, parameter, etc.)

is associated with the cell (i, j). Each coral in the reef is then processed in a different way depending on the specific substrate layer in which it falls after the reproduction process. Note that this modification of the basic algorithm does not imply any change in the corals' encoding.

In this paper, we specify the CRO with Substrate Layers (CRO-SL hereafter), for improving the performance of the basic CRO approach in LSGO. For this, we assign each substrate layer to a different implementation of an exploration procedure. Thus, each coral will be processed in a different way in the reproduction step of the algorithm. Figure 2 shows an example of the CRO-SL, with four different substrate layer. Each one is assigned to a different exploration process, Harmony Search based, Differential Evolution, 1-point crossover or Gaussian mutation. Of course this is only an example and any other distribution of search procedures can be defined in the algorithm. In the specific CRO-SL tested in this paper, each substrate layer only affect to the calculation of the larvae coming from the broadcast spawning process, whereas we have considered the same brooding procedure for all the corals in the reef.

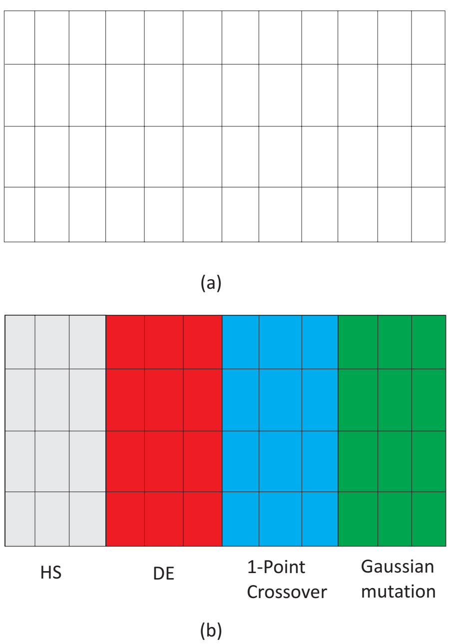

Fig. 2. Example of CRO-SL and comparison with the original reef in the CRO; (a) Reef considered in the original CRO; (b) Reef in the CRO-SL, where four substrate layers associated with the broadcast spawning process have been considered.

There are some important remarks that can be done regarding the CRO-SL proposed. First of all, note that the original CRO is a meta-heuristic based on exploitation of solutions, and leave the specific exploration open (in the same manner as, for example, Simulated Annealing [21]). This way, the CRO-SL can be seen as a generalization of the original CRO, that does not modify the dynamics of the algorithm. The only difference is the specific implementation of the broadcast spawning procedure, which now depends on which layer falls each larva after the reproduction process. Second, as has been previously mentioned, the CRO-SL can be seen as a competitive co-evolution procedure. The CRO-SL is a general procedure to co-evolve different models, operators, parameter values, etc., with the only requisite that there is only one health function defined in the algorithm. Since the CRO is based on a procedure of larvae settlement which involves competition among corals, the substrate layer of the CRO-LS promotes this co-evolution process between corals, without the necessity of defining different populations. In this sense, we say that the CRO-LS makes competitive co-evolution of different models in just one population.

B. Hybridization proposed

Large scale optimization problems require the application of very efficient algorithms. Thus we have complement our co-evolution CRO-SL algorithm with a Local Search method. The designed hybrid model has been designed to obtain a synergy between the two components: the CRO-SL and the LS method. Once each substrate model has created the larvae by applying different exploration strategies, the algorithm stores for each substrate the solution with the best improvement. Then, every $Freq_{LS}$ iterations, an LS method is applied to each one of these stored solutions, improving the just one solution coming from each substrate. The LS used is the MTS-LS1 [11], well-known for being a specific LS for LSGO. This method applies a mutation dimension by dimension, increasing or decreasing a certain value, I_{step} which is adapted during the search (this value is basically decreased after several iterations without improvement in the new solution).

This *improvement phase* is applied between the *reproduction phase* and the *larvae settlement phase* of the CRO-SL. Note that, since the improving phase is not carried out every iteration, the solutions to improve could not come from the last iteration. In this case, the original larva to be improved could be into the population (it should be replaced in the population) or not, (in that case the final solution competes against the current worst individual of the population).

It is also remarkable that the LS is applied in the same iteration for each substrate. The underline idea is to take advantage of the fact that each substrate generates new larvae by using a different strategy. This way we can apply the LS to different solutions, with the idea that distant starting solutions could give to different basins of attraction. However, since not all solutions could be equally promising, the LS method is only applied when the considered solution has a *fitness* value better than the current worst coral of the population.

C. Reef restarting mechanism

Sometimes, when an LS method is incorporated to an evolutionary algorithm, the diversity of the population may suffer a quick reduction during the search, increasing the possibilities of obtain a premature convergence to a local optimum. To prevent this effect in our CRO-SL with LS, we have introduced a reef restarting mechanism: in our case, the reef is restarted depending on two parameters: The measured improvement of the best solution, and the current diversity in the reef. Only when the best solution does not improve during a number of generations, and the diversity is low, the reef is restarted. The restarting procedure is briefly described in Algorithm 1: the reef is restarted when the current best solution is improved less than a threshold value $Restart_{min}$ during $N_{improvement}$ iterations. Jointly, the diversity in the reef is observed, and the reef is restarted if the ratio of difference in fitness between the best coral and the worst one is lower than another threshold $Min_{\Delta Fitness}$.

Algorithm 1 Pseudo-code for the restarting mechanism.

$Improvement \leftarrow \Delta min(Pop_{Fitness})$ in last $N_{improvement}$ iterations.
if $Improvement < Restart_{min}$ **then**
  $Fitness_{Dif} = \dfrac{max(Pop_{fitness}) - min(Pop_{fitness})}{max(Pop_{fitness})}$
  if $Fitness_{Dif} < Min_{\Delta Fitness}$ **then**
    Restart algorithm
  end if
end if

When the reef is restarted, the best coral is maintained (elitism criterion) and the rest of corals in the reef are randomly initialized.

IV. Experiments and results

In order to analyze the performance of our proposal in LSGO, we have carried out different experiments with the specific benchmark functions and experimental conditions indicated on the Special Session on Large Scale Global Optimisation [12]. This benchmark is composed by 15 continuous optimization functions of dimension 1000 and with different degrees of separability, from completely separable functions to fully non-separable functions:

- Fully separable functions: $f_1 - f_3$.
- Partially separable functions: with a separable subcomponent ($f_4 - f_7$) and without separable subcomponents ($f_8 - f_{11}$).
- Overlapping functions: $f_{12} - f_{14}$.
- Non-separable functions: f_{15}.

A complete information and further description of these functions can be found in [12].

This test suite defines specific experimental conditions, which allow a direct comparison of the optimization results obtained by different algorithms. The proposed CRO-SL algorithm has been run 25 times for each function until the maximum number of fitness evaluations (FEs) is achieved: for every function, MaxFEs = 3.0e+06. Also, the current best fitness obtained is shown for different FEs: 1.2e+05, 3.0e+05, and 6.0e+06.

The experiments have been carried out with the parameters values indicated in Table I. These parameters have been obtained from previous works where they provided good results, though some of them have been updated after some preliminary experimental tests on the objective functions considered. Note that we have not carried out a fully automated tuning of the CRO-SL parameters, so it could possible that the results could improve with different values. We consider this point as a future line to improve the algorithm.

TABLE I
Parameters values used in the hybridization of the CRO-SL with LS.

Parameter	Description	value
Reef	Reef size	240
F_b	Frequency of broadcast spawning	97%
$\mathcal{N}_{att}$	Number of tries of larvae settlement	3
P_d	Probability of depredation	10%
Freq$_{LS}$	Frequency of each LS	5
I_{step}	Initial step size for MTS	20
LS_{evals}	Evaluation in each LS application	2000
$N_{improvement}$	Iterations without improvement	5
$Restart_{min}$	Minimum improvement	10^{-3}
$Min_{\Delta Fitness}$	Ratio of diversity	2%

Six different substrates layers have been used in the experiments carried out:

1) Mutation from the harmony search.
2) Mutation used in differential evolution (with $F = 0.6$).
3) Classic two-points crossover.
4) Gaussian Mutation, with a δ value linearly decreasing during the run, from $0.2 \cdot (A - B)$ to $0.02 \cdot (A - B)$, where $[B, A]$ is the domain search.
5) Multi-point crossover.
6) Gaussian Mutation, with a δ value linearly increasing from $0.2 \cdot (A - B)$ to $0.02 \cdot (A - B)$, where $[B, A]$ is the domain search.

A. *Influence of each substrate in improving the search*

Although the CRO is a simple algorithm, the introduction of different substrates to form the CRO-SL increases its complexity and exploration possibilities, and gives the algorithm a kind of co-evolution capacity. The inclusion of substrate layers opens therefore new questions about the role that each substrate takes in the evolution of the search. In order to analyze the influence of each substrate layer in the algorithm's search capability, we have measured in each iteration which is the substrate that has generated the best coral larva. Figure 3 shows the ratio of times in which each substrate gives the best solution during the search for different objective functions. First, it can be observed that the contribution of each substrate varies during the search. As an example, in the case of F4 and F8 the substrate with the higher ratio is changing during the search, and it strongly depends on the landscape of the function to optimize, as it was expected. At a first glance, it may seem strange the high ratio of good results obtained by the classic 2-point crossover. However, it has to be considered that, in combination with a LS method, it is convenient to have a disruptive mutation, able to maintain at the same the variable values obtained by the LS method.

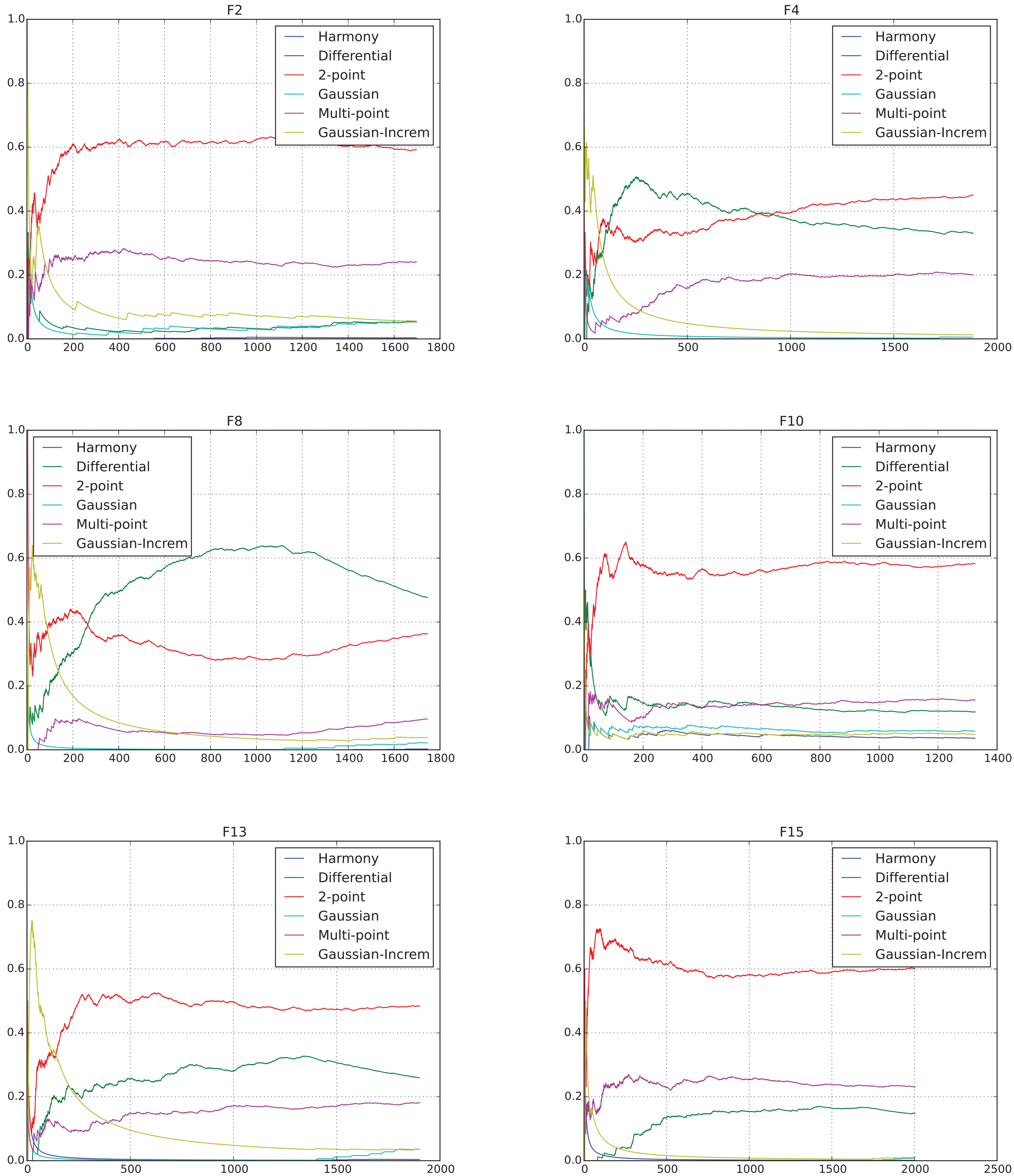

Fig. 3. Ratio of times each substrate gives the best results, for different functions.

B. Experimental Results

The summary of the results obtained by the proposed CRO-SL algorithm are presented in Table II, and in comparison with the reference algorithm (DECC-CG), in Table III. From these table, we can observe that our proposal improve DECC-CG in eight functions, in particular with non-separable and not overlapping like $f_{11}, f_{12}, f_{14}, f_{15}$, the most difficult ones. Also, with the exception of separable functions, where DECC-CG is clearly better (f_1, f_3), in the rest of functions in which DECC-CG obtains better results, results of both algorithms maintain the same level of magnitude. However, for many functions in which CRO improves DECC-CG, the difference is very significant, such as in functions f_5, f_{11}, f_{13}, f_{14} and f_{15}. In this latter case the worst result obtained by CRO-SL is better than the best one obtained by the reference DECC-CG.

It is remarkable that, while DECC-CG tries to improve the problem by decomposing the group of variables in subgroups, CRO-SL with LS tries to optimize all variables at the same time. This is the reason why the DECC-CG obtains better results in separable functions. Note however that the CRO-SL with LS clearly improve the reference algorithm in the most difficult and complex functions, maintaining a robust behavior in the majority of objective functions tested.

Finally, we would like to remark that this paper is the first work on the CRO-SL which focus on co-evolving different exploration approaches in the substrate layer. The main objective of the paper is to prove that this meta-heuristic could be competitive with alternative existing methodologies, and further work to improve the algorithm and make it stronger in LSGO problems is needed. We are currently trying different alternative classes of substrates, with novel exploration approaches to be tested in the algorithm. We also would like to point out that the CRO-SL could help directly compare novel exploration operators that can be defined for a given problem, within the same population and in terms of the same objective function. This capability of the CRO-SL will be for sure exploited in the near future.

V. Conclusions

In this paper we have presented a novel modification of the Coral Reefs Optimization algorithm (CRO) to include *substrate layers* on it, obtaining this way a novel hybrid approach called CRO-SL. Each substrate layer represents here a type of exploration (searching) mechanism. This way, the algorithm promotes competitive co-evolution between different searching procedures (substrate layers) within one population (the simulated reef of the CRO), and with the dynamics and rules of the CRO approach. CRO-SL has been hybridized with a LS method and we have applied the hybrid algorithm resulting to tackle Large Scale Global Optimization (LSGO) problems, specifically to a specially designed test suite. The performance of the CRO-SL with LS has been compared to the reference algorithm DECC-CG, proving that our proposal, not been specifically designed for LSGO, obtains better results, specially in the most complex functions: non-separable, and overlapping functions.

Acknowledgements

This work has been partially supported by the projects TIN2014-54583-C2-2-R, TIN2013-47210-P of the Spanish Ministerial Commission of Science and Technology (MICYT), the Research National Projects the Comunidad Autónoma de Madrid, under project number S2013ICE-2933_02, and the Research Regional Projects P08-TIC-04173, P12-TIC-2958.

References

[1] R. Dash, P. K. Dash, R. Bisoi, "A differential harmony search based hybrid interval type2 fuzzy EGARCH model for stock market volatility prediction," *International Journal of Approximate Reasoning*, vol. 59, pp. 81-104, 2015.

[2] H. Zhang, D. Yue, X. Xie, S. Hu and S. Weng, "Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch," *Applied Soft Computing*, vol. 34, pp. 312-323, 2015.

[3] F. Zhao, Y. Liu, C. Zhang and J. Wang, "A self-adaptive harmony PSO search algorithm and its performance analysis," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7436-7455, 2015.

[4] X. Wang and X. Yan, "Global best harmony search algorithm with control parameters co-evolution based on PSO and its application to constrained optimal problems," *Applied Mathematics and Computation*, vol. 219, no. 19, pp. 10059-10072, 2013.

[5] N. Poursalehi, "Development of a new approach, evolutionary harmony search algorithm, for the LPO problem," *Progress in Nuclear Energy*, vol. 81, pp. 78-90, 2015.

[6] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292-305, 2016.

[7] A. F. Crossland, D. Jones and N. S. Wade, "Planning the location and rating of distributed energy storage in LV networks using a genetic algorithm with simulated annealing," *Electrical Power and Energy Systems*, vol. 59, pp. 103-110, 2014.

[8] M. H. Moradi, M. Abedini and S. M. Hosseinian, "Optimal operation of autonomous microgrid using HS-GA," *Electrical Power and Energy Systems*, vol. 77, pp. 210-220, 2016.

[9] S. Salcedo-Sanz, J. Del Ser, I. Landa-Torres, S. Gil-López, and J. A. Portilla-Figueras, "The Coral Reefs Optimization algorithm: a novel metaheuristic for efficiently solving optimization problems," *The Scientific World Journal*, vol. 2014, Article ID: 739768, 2014.

[10] M. J. Vermeij, "Substrate composition and adult distribution determine recruitment patterns in a Caribbean brooding coral," *Marine Ecology Progress Series*, vol. 295, pp. 123-133, 2005.

[11] L. Y. Tseng and C. Chen, "Multiple trajectory search for large scale global optimization," In *Proc. of the IEEE Conference on Evolutionary Computation*, pp. 3052–3059, 2008.

[12] X. Li, K. Tang, M. Omidvar, Z. Yang, and K. Qin. Benchmark functions for the CEC'2013 special session and competition on large scale global optimization. Technical report, Evolutionary Computation and Machine Learning Group, RMIT University, Australia, 2013.

[13] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985 – 2999, 2008.

[14] S. Salcedo-Sanz, D. Gallo-Marazuela, A. Pastor-Sánchez, L. Carro-Calvo, J. A. Portilla-Figueras and L. Prieto, "Offshore wind farm design with the Coral Reefs Optimization algorithm," *Renewable Energy*, vol. 63, pp. 109-115, 2014.

[15] S. Salcedo-Sanz, A. Pastor-Sánchez, J. Del Ser, L. Prieto and Z.W. Geem, "A Coral Reefs Optimization algorithm with Harmony Search operators for accurate wind speed prediction," *Renewable Energy*, vol. 75, pp. 93-101, 2015.

[16] Z. Yang, T. Zhang and D. Zhang, "A novel algorithm with differential evolution and coral reef optimization for extreme learning machine training," *Cognitive Neurodynamics*, in press, 2016.

[17] S. Salcedo-Sanz, C. Casanova-Mateo, A. Pastor-Sánchez and M. Sánchez-Girón, "Daily Global Solar Radiation Prediction based on a Hybrid Coral Reefs Optimization – Extreme Learning Machine Approach", *Solar Energy*, vol. 105, pp. 91-98, 2014.

TABLE II
EXPERIMENTAL RESULTS OBTAINED WITH THE PROPOSED CRO-SL WITH LS.

		F1	F2	F3	F4	F5	F6	F7	F8
1.25E+5	Best	2.67e+07	3.68e+03	2.02e+01	1.81e+11	1.19e+08	1.06e+06	1.82e+09	7.69e+15
	Median	3.36e+07	4.91e+03	2.02e+01	2.71e+11	1.62e+08	1.07e+06	3.75e+09	9.95e+15
	Worst	1.07e+08	6.05e+03	2.03e+01	5.77e+11	3.86e+08	1.07e+06	4.92e+09	1.49e+16
	Mean	**4.67e+07**	**4.81e+03**	**2.03e+01**	**3.18e+11**	**2.23e+08**	**1.07e+06**	**3.59e+09**	**1.02e+16**
	Std	3.37e+07	9.03e+02	2.64e-02	1.51e+11	1.15e+08	1.04e+03	1.32e+09	2.88e+15
6.00E+5	Best	1.22e+06	9.67e+02	2.01e+01	4.45e+10	3.01e+07	1.06e+06	5.26e+08	4.97e+14
	Median	5.91e+06	1.19e+03	2.02e+01	5.92e+10	5.71e+07	1.06e+06	1.07e+09	1.19e+15
	Worst	3.55e+07	1.31e+03	2.02e+01	8.97e+10	7.86e+07	1.07e+06	2.81e+09	8.43e+15
	Mean	**1.16e+07**	**1.18e+03**	**2.02e+01**	**6.43e+10**	**5.49e+07**	**1.06e+06**	**1.46e+09**	**2.56e+15**
	Std	1.41e+07	1.30e+02	3.78e-02	1.71e+10	2.39e+07	1.52e+03	9.29e+08	3.31e+15
3.00E+6	Best	9.60e+05	8.36e+02	2.01e+01	6.18e+09	1.62e+07	1.06e+06	1.41e+08	1.48e+14
	Median	1.28e+06	1.01e+03	2.01e+01	1.32e+10	2.54e+07	1.06e+06	1.70e+08	3.37e+14
	Worst	4.39e+06	1.08e+03	2.02e+01	2.71e+10	3.04e+07	1.06e+06	4.99e+08	9.16e+14
	Mean	**1.84e+06**	**9.84e+02**	**2.01e+01**	**1.55e+10**	**2.38e+07**	**1.06e+06**	**2.78e+08**	**4.56e+14**
	Std	1.43e+06	9.65e+01	2.36e-02	7.90e+09	6.26e+06	1.13e+03	1.65e+08	3.07e+14
		F9	F10	F11	F12	F13	F14	F15	
1.25E+5	Best	9.11e+08	9.45e+07	1.42e+11	3.45e+05	2.63e+10	3.62e+11	1.10e+08	
	Median	9.64e+08	9.48e+07	4.29e+11	1.45e+06	2.93e+10	5.57e+11	1.25e+08	
	Worst	1.15e+09	9.54e+07	9.98e+11	1.33e+07	3.45e+10	6.38e+11	1.69e+08	
	Mean	**9.97e+08**	**9.48e+07**	**4.91e+11**	**3.80e+06**	**2.93e+10**	**5.14e+11**	**1.35e+08**	
	Std	9.62e+07	3.65e+05	3.17e+11	5.48e+06	3.27e+09	1.34e+11	2.36e+07	
6.00E+05	Best	4.66e+08	9.45e+07	5.36e+10	2.56e+03	9.48e+09	8.22e+10	3.09e+07	
	Median	5.63e+08	9.47e+07	1.31e+11	2.85e+03	1.25e+10	1.72e+11	4.08e+07	
	Worst	1.12e+09	9.48e+07	5.15e+11	1.34e+05	1.56e+10	2.34e+11	4.94e+07	
	Mean	**6.52e+08**	**9.46e+07**	**2.06e+11**	**3.03e+04**	**1.26e+10**	**1.74e+11**	**4.06e+07**	
	Std	2.65e+08	1.25e+05	1.89e+11	5.79e+04	2.30e+09	5.81e+10	7.58e+06	
3.00E+6	Best	4.55e+08	9.42e+07	8.57e+09	1.47e+03	4.68e+09	3.03e+10	1.31e+07	
	Median	5.29e+08	9.44e+07	2.35e+10	2.32e+03	5.18e+09	5.39e+10	1.99e+07	
	Worst	5.83e+08	9.48e+07	6.26e+10	9.69e+03	6.31e+09	1.17e+11	2.23e+07	
	Mean	**5.27e+08**	**9.44e+07**	**2.91e+10**	**3.69e+03**	**5.33e+09**	**6.08e+10**	**1.88e+07**	
	Std	5.02e+07	2.70e+05	2.18e+10	3.38e+03	6.23e+08	3.53e+10	3.48e+06	

TABLE III
COMPARISONS OF RESULTS AGAINST THE REFERENCE ALGORITHM (DE-CC-CG) FOR FEs=3.0E+06.

		F1	F2	F3	F4	F5	F6	F7	F8
DECC-CG	Best	**1.75e-13**	9.90e+02	**2.63e-10**	7.58e+09	7.28e+14	**6.96e-08**	1.96e+08	**1.43e+14**
	Median	**2.00e-13**	1.03e+03	**2.85e-10**	2.12e+10	7.28e+14	**6.08e+04**	4.27e+08	**3.88e+14**
	Worst	**2.45e-13**	1.07e+03	**3.16e-10**	6.99e+10	7.28e+14	**1.10e+05**	1.78e+09	**7.75e+14**
	Mean	**2.03e-13**	1.03e+03	**2.87e-10**	2.60e+10	7.28e+14	**4.85e+04**	6.07e+08	**4.26e+14**
	Std	1.78e-14	2.26e+01	1.38e-11	1.47e+10	1.51e+05	3.98e+04	4.09e+08	1.53e+14
CRO-SL with LS	Best	9.60e+05	**8.36e+02**	2.01e+01	**6.18e+09**	**1.62e+07**	1.06e+06	**1.41e+08**	1.48e+14
	Median	1.28e+06	**1.01e+03**	2.01e+01	**1.32e+10**	**2.54e+07**	1.06e+06	**1.70e+08**	3.37e+14
	Worst	4.39e+06	**1.08e+03**	2.02e+01	**2.71e+10**	**3.04e+07**	1.06e+06	**4.99e+08**	9.16e+14
	Mean	1.84e+06	**9.84e+02**	2.01e+01	**1.55e+10**	**2.38e+07**	1.06e+06	**2.78e+08**	4.56e+14
	Std	1.43e+06	9.65e+01	2.36e-02	7.90e+09	6.26e+06	1.13e+03	1.65e+08	3.07e+14
		F9	F10	F11	F12	F13	F14	F15	
DECC-CG	Best	**2.20e+08**	**9.29e+04**	4.68e+10	**9.80e+02**	2.09e+10	1.91e+11	4.63e+07	
	Median	**4.17e+08**	**1.19e+07**	1.60e+11	**1.03e+03**	3.36e+10	6.27e+11	6.01e+07	
	Worst	**6.55e+08**	**1.73e+07**	7.16e+11	**1.20e+03**	4.64e+10	1.04e+12	7.15e+07	
	Mean	**4.27e+08**	**1.10e+07**	2.46e+11	**1.04e+03**	3.42e+10	6.08e+11	6.05e+07	
	Std	9.89e+07	4.00e+06	2.03e+11	5.76e+01	6.41e+09	2.06e+11	6.45e+06	
CRO-SL with LS	Best	4.55e+08	9.42e+07	**8.57e+09**	1.47e+03	**4.68e+09**	**3.03e+10**	**1.31e+07**	
	Median	5.29e+08	9.44e+07	**2.35e+10**	2.32e+03	**5.18e+09**	**5.39e+10**	**1.99e+07**	
	Worst	5.83e+08	9.48e+07	**6.26e+10**	9.69e+03	**6.31e+09**	**1.17e+11**	**2.23e+07**	
	Mean	5.27e+08	9.44e+07	**2.91e+10**	3.69e+03	**5.33e+09**	**6.08e+10**	**1.88e+07**	
	Std	5.02e+07	2.70e+05	2.18e+10	3.38e+03	6.23e+08	3.53e+10	3.48e+06	

[18] I. G. Medeiros, J. C. Xavier-Júnior and A. M. Canuto, “Applying the Coral Reefs Optimization algorithm to clustering problems,” In *Proc. of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, 2015.

[19] M. Li, C. Miao and C. Leung, “A Coral Reef Algorithm Based on Learning Automata for the Coverage Control Problem of Heterogeneous Directional Sensor Networks,” *Sensors*, vol. 15, pp. 30617-30635, 2015.

[20] S. Salcedo-Sanz, J. Muñoz-Bulnes and M. Vermeij, “New Coral Reefs-based Approaches for the Model Type Selection Problem: A Novel Method to Predict a Nation’s Future Energy Demand,” *International Journal of Bio-inspired Computation*, in press, 2016.

[21] D. Kirpatrick, C. D. Gerlatt and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671-680, 1983.