XIII Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (XIII MAEB)

MAEB 1: Aplicaciones de las Metaheurísticas

Evolving Geographically-embedded Complex Networks using the CRO-SL Algorithm

1st Sancho Salcedo-Sanz
Dept. of Signal Processing and Communications
Universidad de Alcalá
Alcalá de Henares, Spain
sancho.salcedo@uah.es

2nd Lucas Cuadra
Dept. of Signal Processing and Communications
Universidad de Alcalá
Alcalá de Henares, Spain
lucas.cuadra@uah.es

***Abstract*—This paper deals with the problem of evolving geographically-embedded randomly generated complex networks aiming at fulfilling the scale-free property: the fraction of nodes in the network having degree k (k_i = number of links in node n_i) follows a power law probability distribution $P(k) \sim k^{-\gamma}$. Intuitively, this means that most nodes have only a few connections and only a few nodes ("hubs") have a high number of links (or connections). The scale-free property is well-known in very large complex networks (with a huge number of nodes and links) but it has received much less attention for small geographically-embedded networks, in which the study of networks' properties is much more difficult. Regarding this, we explore the feasibility of generating geographically-embedded complex networks even in the case of small networks (those with only hundred of nodes) by means of considering a simple model for network generation based on distances among nodes. We state the problem as an optimization task, in which each node of the network has a link radius assigned to conform its links to other nodes in the network. The idea is to evolve these link radius for all the nodes in the network, aiming at finally fulfilling the scale-free property, when possible. Our machine learning approach for network evolution is based on the recently proposed meta-heuristic called Coral Reefs Optimization algorithm with Substrate Layer (CRO-SL). Our experimental work shows that the proposed model is able to generate geographically (or spatially) embedded networks with the scale-free property. Specifically, we test the performance of the CRO-SL approach in two different, randomly generated, geographically-embedded networks with 200 and 400 nodes, respectively.**

***Index Terms*—Geographically-embedded complex networks; Scale-free networks; Meta-heuristics; CRO-SL.**

I. Introduction

What do systems as different as power grids and ecosystems have in common? Both can be described in terms of graphs: a node represents an entity (generator/load in a power grid, or a species in an ecosystem) that is linked with others (by electrical cables in the power grid or trophic relationships in an ecosystem). These and other dissimilar systems are called complex systems because it is extremely difficult to deduce their emerging collective behavior from only the components of the system [1]. Their topological and dynamical features can be studied using the Complex Network (CN) Science [1]. The interested reader is referred to [1], which clearly explains of CN concepts with a profuse variety of examples in both natural (metabolic networks, gene interactions, food webs, etc.) and artificial systems (the Internet, transport networks, or power grids). In particular, the feasibility of using CN concepts in power grids have been recently discussed in [2] and [3] in combination with evolutionary algorithms in smart grids. More profound technical details about CN can be found in [4], [5], [6], [7] and the references there in.

Most recent studies reveal that many CNs –such as some power grids or the Internet– have a heterogeneous topology [1] as the one represented in Fig. 1 (a). Note that most nodes have only a few connections and only a few nodes ("hubs") have a high number of links. This is why the network is said to have "no scale", so it is called "scale-free" [1]. As shown in Fig. 1 (b), the fraction of nodes having degree k (k_i = number of links in node i) exhibits a power law distribution $P(k) \sim k^{-\gamma}$.

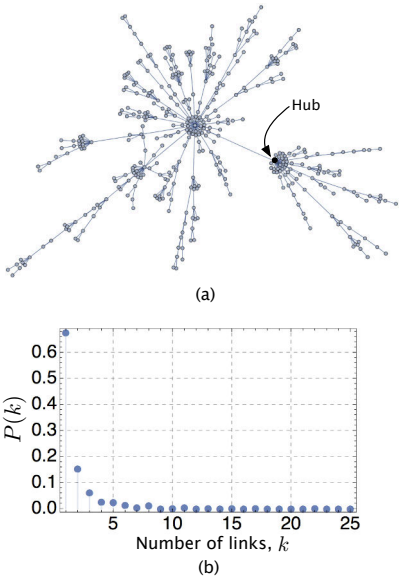


Fig. 1. (a) Example of a scale-free complex network with 400 nodes. (b) Node degree probability density function of a network similar to that represented in (a).

This work has been partially supported by the project TIN2017-85887-C2-2-P of the Spanish Ministerial Commission of Science and Technology (MICYT).

In many of these CNs (for instance, citation networks) the position of nodes in the physical space plays no role at all [8]. However there are other CNs (such as transportation, infrastructure and wireless communication networks) in which nodes and links are embedded in space. In this particular kind of CNs, called "spatial networks" (spatially-embedded or geographically-embedded networks), nodes are located in a space associated to a metric, usually the Euclidean distance [8], [9], [10]. The interested reader is referred to [8] for further details about spatial networks, which can be classified into two categories [8]. The first one, called planar networks, are those that can be drawn in the plane in such a way that their links do not intersect. The second one involves spatial non-planar networks (for instance airline networks, cargo ship networks, or power grids) where links (which can intersect in the plane) have a cost related to their length. Although the scale-free property is well-known in very large, non-spatial complex networks (with a huge number of nodes), however it is not the case in small geographically-embedded networks. This is because, in spatial networks, when geometric constraints are very strong or when the cost associated to the addition of new links is large (water and gas distribution networks, power grids, or communication networks), the appearance of hubs and the scale-free feature become more difficult [8].

In this paper we show that any randomly generated network can be constructed to very approximately follow a scale-free distribution. This result has only been previously proven for geographically-embedded network in a regular lattice [10]. To show this result, we first propose a very simple model for randomly constructing geographically-embedded networks, which consists in assigning a *link radius* to each new node of the network. The proposed model for network construction establishes that each link radius may be different for each node, and it is fully related to the network construction: when a node is randomly generated, it is linked with all other existing nodes in the network which are at a distance smaller than its link radius. In order to show that the network follows a scale-free distribution, we evolve it, i.e. we use an evolutionary-based algorithm in order to assign link radius to all the nodes in the network. The objective is that, eventually, the network follows (approximately) a scale-free distribution. We state this problem as an optimization task, with discrete-based encoding, in which a meta-heuristic search must be applied (since brute-force schemes are discarded due to excessive computational cost). Specifically, we evaluate the performance of the Coral Reefs Optimization algorithm with Substrate Layer (CRO-SL) in this problem of complex networks evolution. We will show that the CRO-SL is able to lead to randomly generated geographically-embedded complex networks fulfilling the scale-free property, and we show it in two cases with randomly generated network of 200 and 400 nodes.

The remainder of the paper has been structured in the following way: next section presents the model we consider to construct geographically-embedded complex networks with randomly-distributed nodes. Section III describes the evolution of the network as an optimization task, defining the encoding, search space and objective function of the problem. Section IV shows the main characteristics of the CRO-SL considered in this paper. Section V describes the experimental part of the paper, with computational results over two randomly generated networks with 200 and 400 nodes. Section VI gives some final conclusions and remarks to close the paper.

II. Growing geographically-embedded complex networks over random-distributed nodes

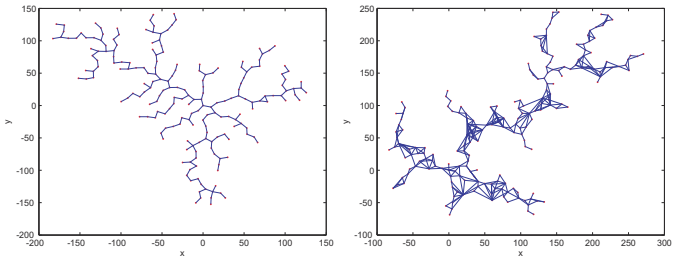
Let us consider a model for growing geographically-embedded complex networks using randomly-distributed nodes. The idea is to grow the network as the random nodes are being generated. Note that since we consider a random location for the new generated nodes, the network is completely constructed from scratch. We can consider many different random ways of generating the network nodes, but in any case, a constraint of maximum distance from a neighbor to others node must be fulfilled. In order to do this, we consider an extremely simple model for nodes generation, in which the new appearing node must be located at a minimum distance from another neighbor node, R_a (attachment radius), to be attached to the network. Otherwise it will be discarded. Note that this radius may be characteristic of the node i currently being generated so that, in this case, we will denote it as R_a^i. However, in the general case, all the nodes in the network will be generated with the same R_a, this simulation parameter being thus equal for all nodes. As previously mentioned, the network will be grown while random nodes are being generated. Aiming at doing this, we propose a simple mechanism for links generation for a new node i: let R_l^i be the *link radius* associated with the recently generated node i, and let $\mathbf{L}$ be the link matrix, in which L_{ij} stands for a binary variable describing whether or not there is a link between node i and an alternative node j. Then, each time a node i is generated, it establishes links to other nodes already attached, in the following way:

$$L_{ij} = \begin{cases} 1, & \text{if } \ d(i,j) < R_l^i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $d(i,j)$ stands for the Euclidean distance (not the geodesic one used in non-spatial networks [8]) between node i and any other existing node (j). It is important to note that the number of links established when the node i is finally generated attached, not discarded) will only depend on R_l^i. Moreover, if we want to ensure that all the nodes are connected with at least one other node in the network, then $R_l^i \geq R_a$.

To illustrate this, let us consider the examples shown in Fig. 2. The first one, in Fig. 2 (a), shows a random network generated with parameters $R_a = 10$ and $R_l^i = 10$ (in this case the same value of R_l^i for all the nodes generated in the network). Note that, since $R_l^i = R_a$, each node will be attached to a very reduced number of other existing nodes in the network. If we keep $R_a = 10$ in the node generation, but R_l^i takes values in $[10, 15, 20, 30]$, depending on the node

generated, then we obtain the network represented in Fig. 2 (b).



(a) (b)

Fig. 2. Example of geographically embedded complex networks generated with the proposed simple model considering R_a and R_l^i; (a) Example of complex network with $R_a = 10$ and $R_l^i = 10$; (b) Example of complex network with $R_a = 10$ and $R_l^i \in [10, 15, 20, 30]$.

III. Quasi-Scale Free geographically-embedded networks with random nodes

Let us consider a random-based geographically-embedded network with N nodes ("network order" $= N$). This means that, after the node generation process, there will be N nodes in the network. Recall that the network is being constructed dynamically, so each time a node i is generated and fulfils the R_a condition, then matrix $\mathbf{L}$ is modified to include the new node links. Let us consider a given R_a for the complete network construction and specific R_l^i radius for each node, and $R_a \geq R_l^i$. Let $\mathbf{r} = [R_l^1, \ldots, R_l^N]$ the link radius associated with the N nodes finally forming the network. The idea is to obtain a vector $\mathbf{r}^*$ which makes the network have a scale-free behavior, i.e., such that it minimizes the following objective (fitness) function:

$$f(\mathbf{r}) = \sum_{k=2}^{N} (p_k(\mathbf{r}) - k^{-\gamma}) \tag{2}$$

where p_k stands for the degree distribution of the random network obtained with a vector $\mathbf{r}$. Note that we aim to find out whether or not there is a $\mathbf{r}^*$ leading to a power law distribution with a given γ.

This problem is therefore stated as an integer optimization problem, in which the final network degree distribution will completely depend on $\mathbf{r}$. The problem is discrete, highly non-linear, and the search space size is huge when the network order N grows, which discards exact solutions via *brute force* algorithms. In these kind of problems meta-heuristics approaches such as Evolutionary Computation-based algorithms are able to obtain very good solutions with a moderate computational complexity. We therefore propose to apply a kind of Evolutionary Algorithm, the aforementioned CRO-SL approach, to solve this optimization problem associated with scale-free random-based networks. The question arising here is whether or not the proposed model for complex network construction over geographically embedded random nodes can generate scale-free networks. Note that in this case the random situation of nodes makes impossible to obtain an exact solution such as the one shown for square lattices in [10]. The approach, therefore, should be stochastic due to the nature of the considered networks, and approximate solutions could arise.

IV. Optimization method: the CRO-SL algorithm

The Coral Reef Optimization algorithm (CRO) [12] (further described in [13]), is an evolutionary-type algorithm based on the behavior of the processes occurring in a coral reef. For an illustrative description of the CRO algorithm, the interested reader is referred to [12], [13]. Additionally, in [14], a new version of the CRO algorithm with Substrate Layer, CRO-SL, has been presented. In the CRO-SL approach, several *substrate layers* (specific parts of the population) have been introduced. In this algorithm, each substrate layer may represent different processes (different models, operators, parameters, constraints, repairing functions, etc.). Specifically, in [15] a version of the CRO-SL algorithm has been recently proposed, in which each substrate layer represents a different search procedure, leading to a co-evolution competitive algorithm. This version of the CRO-SL has been successfully tested in different applications and problems such as micro-grid design [16], vibration cancellation in buildings, both with passive models [17], and active models [18], or in the evaluation of novel non-linear search procedures [19]. This is also the CRO-SL algorithm used in this paper for complex network evolution.

Regarding the algorithm's encoding for the optimization problem at hand, we consider integer vectors as solutions, $\mathbf{x} \in \mathbb{N}$. Note that using this encoding the length of each individual is equal to N. This encoding provides a compact version of the algorithm, and allows using some different searching procedures such as Harmony Search or Differential Evolution. The main problem with a direct encoding of N integer values in the CRO-SL algorithm is that, as N grows, the searching capabilities of the algorithm can be affected, since the search space is huge. It is possible to manage shorter encodings by using a compressed version of the encoding, in such a way that each element of the encoding represents β actual values, such as we proposed in [20]. Fig. 3 shows an example of this compressed encoding, which reduces the current encoding length l to $l' = \frac{l}{\beta}$. Of course, the resolution of the search space is smaller than in the original encoding when the compressed encoding is applied, but on the other hand, it is expected that the CRO-SL algorithm searches for better solutions in this smaller search space.

The considered substrates for solving the stated problem are detailed below. Note that there are general purpose substrates, such as Differential Evolution or Harmony Search-based, and other specific substrates with crossovers and mutations adapted to the chosen encoding. Five different substrates will be described and evaluated later in the experimental section.

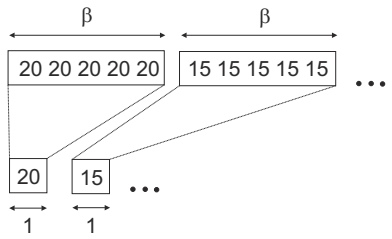
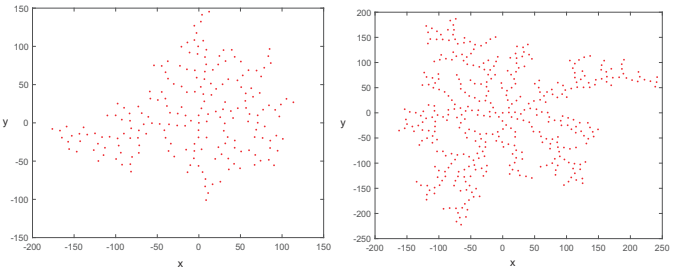


Fig. 3. Compressed encoding example ($\beta = 5$), useful in the evolution of complex networks.

- **Differential Evolution-based operator** (DE): This operator is based on the evolutionary algorithm with that name [22], a method with powerful global search capabilities. DE introduces a differential mechanism for exploring the search space. Hence, new larvae are generated by perturbing the population members using vector differences of individuals. Perturbations are introduced by applying the rule $x'_i = x^1_i + F(x^2_i - x^3_i)$ for each encoded parameter on a random basis, where x' corresponds to the output larva, x^t are the considered parents (chosen uniformly among the population), and F determines the evolution factor weighting the perturbation amplitude.
- **Harmony Search-based operator** (HS): Harmony Search [23] is a population based MH that mimics the improvisation of a music orchestra while its composing a melody. This method integrates concepts such as harmony aesthetics or note pitch as an analogy for the optimization process, resulting in a good exploratory algorithm. HS controls how new larvae are generated in one of the following ways: i) with a probability HMCR $\in [0, 1]$ (Harmony Memory Considering Rate), the value of a component of the new larva is drawn uniformly from the same values of the component in the other corals. ii) with a probability PAR $\in [0, 1]$ (Pitch Adjusting Rate), subtle adjustments are applied to the values of the current larva, replaced with any of its neighboring values (upper or lower, with equal probability).
- **Two points crossover** (2Px): 2PX [21] is considered one of the standard recombination operators in evolutionary algorithms. In the standard version of the operator, two parents from the reef population are provided as input. A recombination operation from two larvae is carried out by randomly choosing two crossover points, interchanging then each part of the corals between those points.
- **Multi-points crossover** (MPx): Similar to the 2PX, but in this case the recombination between the parents is carried out considering a high number of crossover points (M), and a binary template which indicates whether each part of one parent is interchanged with the corresponding of the other parent.
- **Standard integer Mutation** (SM): This operator consists of a standard mutation in integer-based encodings. It consists of mutating an element of a coral with another valid value (different from the previous one). Note that the SM operator links a given coral (possible solution) to a neighborhood of solutions which can be reached by means of a single change is an element of the coral.

V. Experiments and Results

In this section we show different computational results obtained with the CRO-SL in the evolution of two different random networks with 200 and 400 nodes, respectively. The resulting randomly generated nodes have been represented in Fig. 4 without the corresponding links which form the network, for the sake of clarity. In both cases, a common $R_a = 10$ value has been considered, whereas the link radius to be assigned by the CRO-SL has been forced to fulfill the property $10 \leq R^i_l \leq 100$. Table I shows the corresponding values for the CRO-SL parameters considered in the experiments carried out.



(a) (b)

Fig. 4. Randomly-generated nodes for the $N = 200$ and $N = 400$ networks (represented without links, and with $R_a = 10$); (a) $N = 200$; (b) $N = 400$.

TABLE I
Parameters of the CRO-SL used in the evolution of the networks considered. See [12], [13] for further details about the parameters.

CRO-SL	Parameters
Initialization	Reef size = 50×40, $\rho_0 = 0.9$
External sexual reproduction	$F_b = 0.80$
Substrates	$\mathcal{T} = 5$ substrates: HS, DE, 2Px, MPx, SM
Internal sexual reproduction	$1 - F_b = 0.20$
Larvae setting	$\kappa = 3$
Asexual reproduction	$F_a = 0.05$
Depredation	$F_d = 0.15$, $P_d = 0.05$
Stop criterion	$k_{max} = 500$ iterations

First, we have tackled the evolution of the $N = 200$ network, from scratch by using the CRO-SL algorithm. A compressed encoding with $\beta = 5$ has been considered so that the corals length is in this case $l' = \frac{200}{5} = 40$. Fig. 5 shows the results obtained by the CRO-SL in the evolution of this network. Fig. 5 (a) shows the network obtained after the optimization process, which has obtained an excellent

agreement of the network distribution node degree with a power law distribution $k^{-1.55}$ (Fig. 5 (b)). Note that, in this case, we have explored 12 values of the node degree k in the network, ranging from 2 to 12, while the rest bring in upper values of k. The best solution $\mathbf{r}^*$ found by the CRO-SL algorithm has been represented in Fig. 5 (c), note the runs of $\beta = 5$ equal values in the solution. Fig. 5 (d) shows the fitness evolution of the best coral in the reef. As can be seen, the CRO-SL is able to converge almost up to optimality in just 500 generations, showing a fast and robust behaviour in this problem.

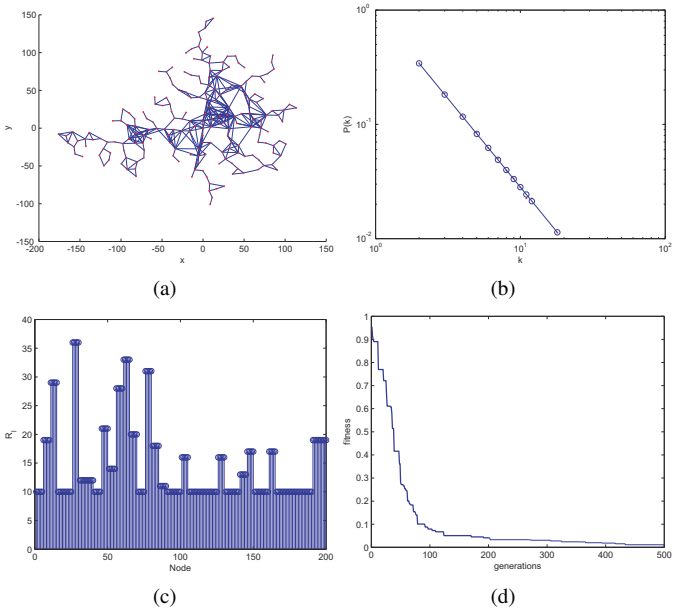


Fig. 5. Example of geographically-embedded complex network with $N = 200$ nodes, evolved with the CRO-SL algorithm; (a) Resulting spatial network obtained; (b) Node degree distribution for the network represented in (a): blue circles stand for the power law distribution $k^{-1.55}$, and red points for the actual degree distribution of the obtained network; (c) Best solution obtained with the CRO-SL; (d) CRO-SL fitness evolution.

Fig. 6 shows the results obtained by the CRO-SL in the evolution of the second network considered, with a network order of $N = 400$ nodes. In this case, we have considered a compressed encoding with $\beta = 10$, which leads to a $l' = \frac{400}{10} = 40$, similar to the $N = 200$ case. We have found that this compressed encoding provides the best results. Fig. 6 (a) shows the resulting network generated by the CRO-SL algorithm, which is constructed to very approximately follow a power law distribution $k^{-1.59}$. In this case we have explored 15 values of the degree k, from 2 to 15 and a rest in upper values of k. The best solution $\mathbf{r}^*$ found by the CRO-SL algorithm has been displayed in Fig. 6 (c). Note the runs of $\beta = 10$ equal values in the solution. Fig. 7 shows the network evolution process in 6 steps for the best solution found by the CRO-SL. In this figure it is possible to see the process of network construction as the nodes are being attached. It is important to take into account that the spatial network construction depends on the position of the randomly generated nodes (we have considered geographically-embedded networks), controlled by R_a and also in the values of R_l^i, which are evolved by the CRO-SL algorithm.

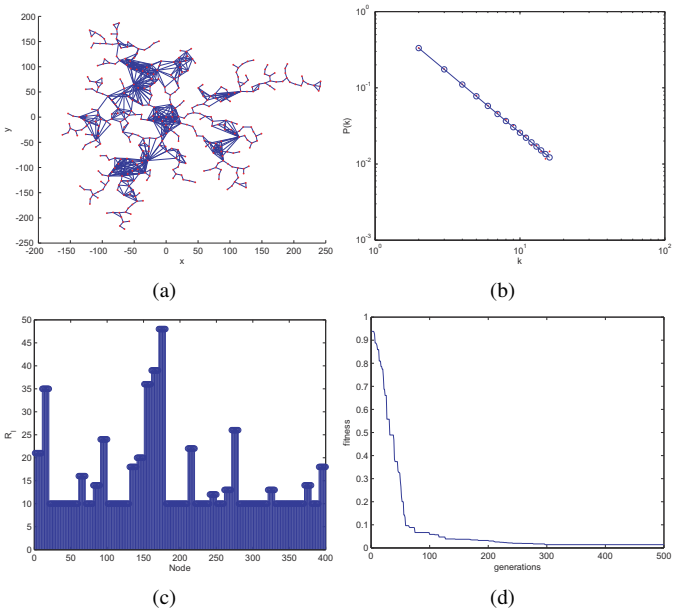


Fig. 6. Example of geographically-embedded complex network with $N = 400$ nodes, evolved with the CRO-SL algorithm; (a) Resulting geographically-embedded network ; (b) Node degree distribution for the network represented in (a): blue circles stand for the power law distribution $k^{-1.59}$, and red points for the actual degree distribution of the obtained network; (c) Best solution obtained with the CRO-SL; (d) CRO-SL fitness evolution.

As can be seen in the results obtained, it is possible to obtain quasi-scale-free geographically-embedded random networks, considering a very simple model of distances between nodes. It is necessary to solve an optimization problem, which is hard, since it must optimize the link radius of all the randomly generated nodes which form the network. We have shown how the CRO-SL algorithm is able to successfully solve this task, finding near optimal solutions to the optimization problem.

VI. Summary and Conclusions

In this paper we have shown that random geographically-embedded networks can be constructed, in such a way that they fulfil the scale-free property, i.e. the fraction of nodes in the network having degree k (k_i = number of links in node n_i) follows a power law probability distribution $P(k) \sim k^{-\gamma}$. Up until now, the scale-free property in geographically-embedded network has only been studied for regular networks in a mesh. We have considered completely randomly generated nodes for the networks, and we have established the on-line construction of the network, following a very simple model which only depends on the distances between new generated nodes and existing nodes in the network (R_l^i). We

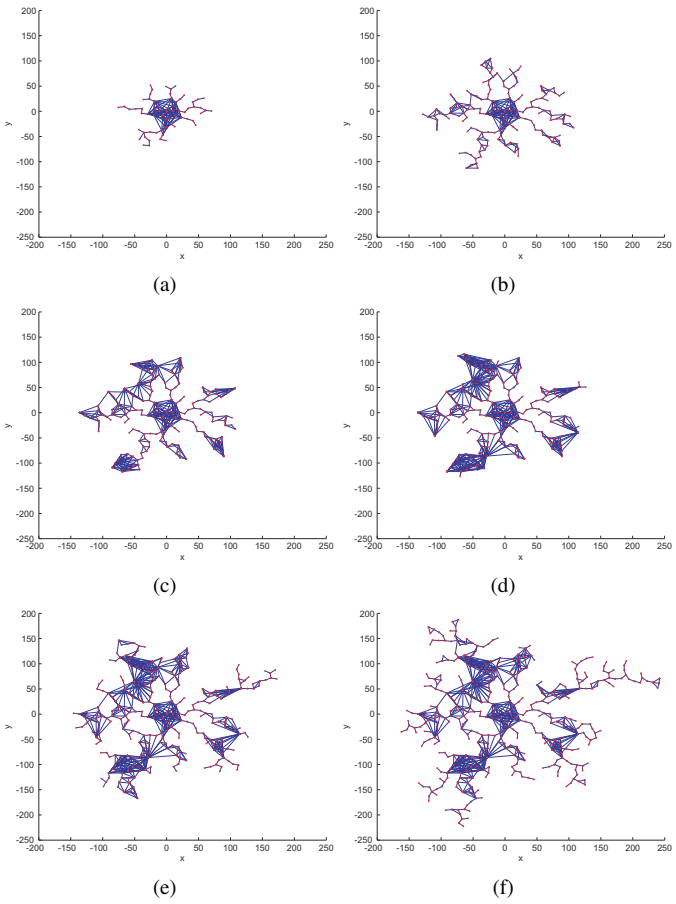


Fig. 7. Network evolution process in 6 steps for the $N = 400$ case (best solution obtained with the CRO-SL algorithm).

propose then the evolution of the network with the objective of fulfilling the scale-free property: we have described this problem as an optimization task, consisting on assigning a given link radius R_l^i to each node of the network, as soon as it is randomly generated. The optimal assignment of these link radius leads to an evolution of the network to be quasi-scale-free when it is completely constructed. We have applied the modern meta-heuristic Coral Reefs Optimization with Substrate Layers (CRO-SL), which is able to combine different searching procedures within a single-population algorithm. A discussion on the optimal problem's encoding with different lengths using a compression procedure is also carried out. We have successfully tested the CRO-SL in two randomly generated networks of 200 and 400 nodes, where we have shown that the CRO-SL is able to obtain quasi-scale free geographically-embedded networks when it is applied.

References

[1] A.L. Barabási and M. and Pósfai, Network Science. Cambridge University Press, Cambridge, UK, 2016.

[2] L. Cuadra, M. del Pino, J. C. Nieto-Borge, and S. Salcedo-Sanz, "A critical review of robustness in power grids using complex networks concepts," *Energies*, vol. 8, no. 9, pp. 9211-9265, 2015.

[3] L. Cuadra, M. del Pino, J. C. Nieto-Borge, and S. Salcedo-Sanz, "Optimizing the Structure of Distribution Smart Grids with Renewable Generation against Abnormal Conditions: A Complex Networks Approach with Evolutionary Algorithms," *Energies*, vol. 10, no. 8, pp. 1097, 2017.

[4] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268, 2001.

[5] R. Albert and A. L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, pp. 47, 2002.

[6] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[7] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 45, no. 7291, pp. 1025, 2010.

[8] M. Barthélemy, "Spatial networks", *Physics Reports*, vol. 499, no. 1-3, pp. 1-101, 2011.

[9] M. Barthélemy, *Morphogenesis of Spatial Network*, Springer, 2017.

[10] K. Kosmidis, S. Havlin and A. Bunde, "Structural properties of spatially embedded networks," *Europhysics Letters*, vol. 82, no. 4, pp. 1-5, 2008.

[11] A. L. Barabási and R. Albert, "Emergence of scaling in random networks" *Science*, vol. 286, pp. 509-512, 1999.

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

[13] S. Salcedo-Sanz, "A review on the coral reefs optimization algorithm: new development lines and current applications," *Progress in Artificial Intelligence*, vol. 6, pp. 1-15, 2017.

[14] 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*, vol. 10, no. 3, pp. 145-158, 2017.

[15] S. Salcedo-Sanz, C. Camacho-Gómez, D. Molina and F. Herrera, "A Coral Reefs Optimization algorithm with substrate layers and local search for large scale global optimization," *In Proc. of the IEEE World Congress on Computational Intelligence*, Vancouver, Canada, July, 2016.

[16] S. Salcedo-Sanz, C. Camacho-Gómez, R. Mallol-Poyato, S. Jiménez-Fernández and J. del Ser, "A novel Coral Reefs Optimization algorithm with substrate layers for optimal battery scheduling optimization in micro-grids," *Soft Computing*, vol. 20, pp. 4287-4300, 2016.

[17] S. Salcedo-Sanz, C. Camacho-Gómez, A. Magdaleno, E. Pereira and A. Lorenzana, "Structures vibration control via tuned mass dampers using a co-evolution coral reefs optimization algorithm," *Journal of Sound and Vibration*, vol. 393, pp. 62-75, 2017.

[18] C. Camacho-Gómez, X. Wang, I. Díaz, E. Pereira and S. Salcedo-Sanz, "Active vibration control design using the Coral Reefs Optimization with Substrate Layer algorithm," *Computers & Structures*, in press, 2017.

[19] S. Salcedo-Sanz, "Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures," *Physics Reports*, vol. 655, 1-70, 2016.

[20] S. Salcedo-Sanz, A. Gallardo-Antolín, J. M. Leiva-Murillo and C. Bousoño-Calzón, "Off-line speaker segmentation using genetic algorithms and mutual information," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 2, pp. 175-186, 2006.

[21] A. E. Eiben and J. E. Smith. Introduction to evolutionary computing. Springer-Verlag, Natural Computing Series 1st edition, 2003.

[22] R. Storn and K. Price, "Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* vol. 11, pp. 341-359, 1997.

[23] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60-68, 2001.

Metaheurísticas para calibración de modelos basados en agentes en dinámicas de adopción premium

Ignacio Moya*, Manuel Chica*,William Rand†, Oscar Cordón*
*Instituto Andaluz Interuniversitario DaSCI (Data Science and Computational Intelligence), Universidad de Granada, España
† Poole College of Management, North Carolina State University, NC, United States
Emails: imoya@ugr.es, manuelchica@ugr.es, wmrand@ncsu.edu, ocordon@decsai.ugr.es

***Resumen*—Las aplicaciones *freemium* están creando nuevos escenarios de marketing, incentivando la adopción de servicios mediante la interacción entre usuarios. Para entender las dinámicas de estas aplicaciones son útiles los modelos basados en agentes, pero deben calibrarse con datos reales para poder ajustar su comportamiento a la realidad. Las metaheurísticas son métodos de optimización frecuentemente usados para calibración de modelos, dado que pueden ajustar los parámetros del modelo. En este artículo comparamos distintas metaheurísticas para calibrar un modelo basado en agentes que replica las dinámicas de adopción de contenido *premium* usando el modelo de Bass. Aplicamos estas metaheurísticas a cuatro datasets y llevamos a cabo un análisis de sensibilidad sobre los parámetros de las soluciones encontradas por los algoritmos. Nuestros experimentos muestran que CMA-ES encuentra mejores soluciones que los demás algoritmos en los distintos datasets de adopción *premium*. Nuestro análisis de sensibilidad muestra un amplio rango de valores para los coeficientes de imitación e innovación del modelo de Bass.**

***Index Terms*—calibración de modelos, metaheurísticas, modelado basado en agentes, modelo de negocio freemium**

I. Introducción

El modelo de negocio *freemium* combina la oferta de un producto o servicio sin coste con contenido *premium* que el usuario puede adquirir de manera opcional [1]. Este modelo se está extendiendo en aplicaciones *online* [2], donde están creando nuevos escenarios de marketing dado que este contexto propicia la adopción de productos a través de la interacción entre usuarios [3]. Con estos escenarios también surgen distintos problemas al planear campañas de marketing: qué individuos seleccionar para conseguir una campaña de marketing viral, cómo lidiar con *influencers*, o cómo incentivar a los usuarios a atraer a otros mandando invitaciones a través de sus redes [4]–[7]. Entender estos y otros efectos sociales detrás de las compras de contenido *premium* puede ser más fácil usando un modelo basado en agentes (ABM), puesto que permiten probar recompensas, incentivos, y políticas de *targeting* que incrementen el número de usuarios premium.

Un ejemplo es el ABM desarrollado para *Creature Party* [8]. Esta aplicación es un juego social multiplataforma para niños donde el usuario interactúa *online* con otros usuarios, que pueden ser *premium* o básicos. Un usuario básico tiene acceso total al juego pero un usuario *premium* recibe beneficios adicionales como una cantidad semanal de monedas del juego, la habilidad de adoptar mascotas virtuales, acceso a todos los avatares, y contenido reservado para usuarios *premium* como grupos y aventuras. La metodología ABM [9], [10] utiliza una población de entidades autónomas llamadas agentes que se comportan de acuerdo a reglas simples e interactúan unos con otros. Agregar estas reglas con la interacción entre los agentes nos permite representar dinámicas complejas y emergentes, así como definir escenarios *what-if* y predecir escenarios hipotéticos [11]. Este enfoque encaja con los patrones de crecimiento del mercado que resultan de la interacción de muchos usuarios, que son más complejos que cualquier adopción individual [12]. Gracias a sus aplicaciones exitosas, se ha incrementado el número de trabajos que emplean ABMs, particularmente para analizar la adopción de nuevos productos, políticas de adopción, y estrategias de *targeting* [13], [14].

Concretamente el ABM para *Creature Party* predice el número de usuarios básicos que adquieren contenido *premium* simulando periodos de tiempo específicos y monitorizando el comportamiento de los agentes. Este ABM utiliza información real de la red de usuarios para generar una red artificial [15] con una estructura similar en la que replicar el proceso social de adopción mediante el modelo de Bass [16]. También simula el comportamiento de usuarios siguiendo diferentes patrones estacionales, lo que es muy relevante para las aplicaciones que muestran diferente actividad entre semana que durante fines de semana.

En general, para usar un ABM es necesario validar su comportamiento calibrando sus parámetros. Llevamos acabo este procedimiento utilizando *data-driven automated calibration*, que consiste en modificar los parámetros del modelo de manera automática y así ajustar su salida a los datos reales. Este proceso es muy importante durante la validación del modelo [12], [17], [18]. Implementamos nuestro proceso de calibración empleando metaheurísticas [19] como algoritmo de búsqueda de la mejor configuración de parámetros del modelo. En este artículo presentamos una comparación de las siguientes metaheurísticas: *hill climbing* [19], algoritmos genéticos [20], *differential evolution* [21], *PSO* [22], y CMA-ES [23]. Elegimos estas metaheurísticas de modo que el grupo de métodos sea heterogéneo en cuanto a su complejidad, permitiendo apreciar la ganancia de rendimiento que consiguen métodos más avanzados. El rendimiento de las

metaheurísticas presentadas se basará en lo bien que el modelo calibrado replique los datos reales de *Creature Party*, que se presentan como la evolución de usuarios *premium* a lo largo del tiempo. Estos datos se presentan en distintos periodos de tiempo componiendo los distintos datasets que utilizaremos para validar las metaheurísticas para el problema. Una vez calibrado el modelo para *Creature Party* desarrollamos un análisis de sensibilidad sobre las soluciones generadas enfocándonos en los coeficientes de adopción del modelo de Bass y los parámetros de estacionalidad del modelo, utilizando técnicas de visualización de datos (*scatter plots*) para los valores obtenidos.

En la Sección II describiremos el ABM de adopción de contenido *premium* de *Creature Party*. La Sección III presenta las distintas metaheurísticas empleadas en el proceso de calibración. Mostramos la experimentación y el análisis de sensibilidad en la Sección IV. Por último, remarcamos nuestras conclusiones en la Sección V.

II. Descripción del modelo

La Sección II-A presenta la estructura general del modelo. En la Sección II-B se describen los mecanismos y el comportamiento de los agentes, incluyendo sus interacciones sociales en la aplicación. Finalmente presentamos el proceso de adopción seguido por el modelo en la Sección II-C.

A. Estructura general

El modelo simula la evolución del número de usuarios *premium* durante un periodo de tiempo determinado usando un *time step* diario. De este modo la salida del modelo consiste en los nuevos usuarios *premium* para cada paso de la simulación, que es el indicador principal de este tipo de mercados [24]. Observando la evolución de adopciones *premium* diarias podemos medir como de bien se ajusta el modelo a los datos históricos de las distintas instancias. El modelo ABM para *Creature Party* modela los usuarios existentes como agentes con un factor de escala de 2:1, por lo que 20000 agentes representan los 40000 usuarios activos de la aplicación. Esta escala se define para reducir el coste computacional manteniendo un buen nivel de representatividad.

Los agentes representan usuarios básicos o *premium* del total de usuarios de la aplicación, dependiendo del ratio inicial de usuarios *premium* α sobre el total de usuarios. Durante la inicialización del modelo, cada agente es inicializado como usuario básico o *premium* aleatoriamente. Debido a la aleatoriedad del modelo cada ejecución puede resultar en distintos patrones de difusión y resultados finales, por lo que usamos simulaciones de Monte-Carlo (MC) y ejecutamos múltiples veces la simulación. Cada agente toma decisiones asíncronamente, es decir, sin mecanismos de sincronización con el resto de agentes, dado que así el modelo se asemeja más a un modelo de simulación continuo, más cercano a la realidad [12].

B. Actualización del estado de los agentes y de su red

Los agentes tienen distintas variables de estado para representar la transición de usuario básico a *premium*. En el modelo un usuario pasa de básico a *premium* cuando compra en la tienda *in-game*. No consideraremos la transición inversa (esto es, dejar de ser *premium* para ser básico, *churn*), dado que no hay datos disponibles y no podríamos calibrar este comportamiento.

Cada agente tiene un conjunto de enlaces que representan las relaciones sociales entre usuarios de la aplicación. Estas relaciones son enlaces unidireccionales que habilitan el intercambio de información y canalizan influencias entre ellos. Los enlaces de la red social se generan usando un algoritmo basado en un grado de distribución dado [15], [25], con un grado medio de $48{,}19$ usuarios y una densidad de $0{,}0024$.

En cada paso del modelo, cada agente primero decide si jugar o no siguiendo una probabilidad que depende del día de la semana. Modelamos este comportamiento usando dos parámetros: (a) la probabilidad de jugar entre semana ($d \in [0,1]$) y (b) la probabilidad de jugar en fin de semana ($f \in [0,1]$). Si un agente juega a la aplicación, podrá decidir posteriormente si adoptar contenido *premium* en base a la influencia social de sus contactos (imitación) o por su propia iniciativa (innovación). Este proceso de adopción sigue el modelo de Bass [12], [16], descrito en la Sección II-C. La probabilidad de jugar de cada agente sigue la Ecuación 1. El agente A decide jugar el día t si la función $jugar_t^A(r)$ devuelve 1, donde $r \in [0,1)$ es un número aleatorio generado usando una distribución uniforme.

$$jugar_t^A(r) = \begin{cases} 1, & \text{si } r < d \wedge t \in entre{-}semana, \\ 1, & \text{si } r < f \wedge t \in fin{-}semana, \\ 0, & \text{e.o.c.} \end{cases} \quad (1)$$

C. Modelo de adopción

El modelo de adopción elegido es la versión ABM del modelo de Bass [8], [16]. En el modelo de adopción de Bass, un agente básico puede hacerse *premium* con una probabilidad que incluye tanto el efecto de la publicidad (externalidad) como el de la influencia social o boca a boca. La probabilidad de que un agente se haga *premium* debida a la interacción con otros agentes se regula por la fracción de sus vecinos que se han hecho *premium* en los pasos anteriores. Por tanto, en cada paso un agente básico i puede hacerse *premium* debido a dos circunstancias:

1. Innovación: con probabilidad $\hat{p}$ (coeficiente de innovación), $\hat{p} \in (0,1)$, un agente básico puede hacerse *premium* debido a información externa a la red.
2. Imitación: con probabilidad $\hat{q}$ (coeficiente de imitación), $\hat{q} \in (0,1)$, un agente básico puede hacerse *premium* observando el estado de sus vecinos, donde f es la fracción de vecinos *premium*.

En estas circunstancias podemos modelar el proceso de adopción *premium* siguiendo la Ecuación 2 [12]. Un agente A puede hacerse *premium* en un step t si $adoptar$ devuelve 1, donde $r, s \in [0,1)$ son números aleatorios generados usando una distribución uniforme y f_A es la fracción de vecinos *premium* de A.

$$adoptar_t^A(r,s) = \begin{cases} 1, & \text{si } r < \hat{p} \vee s < f_A\hat{q}, \\ 0, & \text{e.o.c.} \end{cases} \quad (2)$$

III. Metaheurísticas para calibración

En esta sección presentamos nuestra propuesta para calibrar ABM usando métodos automáticos basados en metaheurísticas. La calibración automática es un proceso intensivo que usa una medida de error para comparar la salida del modelo con datos reales, modificando sus parámetros para encontrar la configuración que más se ajuste a los datos reales [17], [18]. Medimos la calidad del modelo ejecutando una simulación y comparando su salida con los datos reales. En este trabajo hemos considerado metaheurísticas para calibración dado que los parámetros del ABM no muestran interacciones lineales y la mejor opción es usar un algoritmo de optimización no-lineal que pueda gestionar un espacio de búsqueda muy amplio [17], [26], [27]. Aun así, elegir la metaheurística más adecuada para calibrar ABM no es trivial dado que hay que tener en cuenta dos criterios opuestos: la exploración del espacio de búsqueda (diversificación) y la explotación de las soluciones más prometedoras (intensificación).

En general, las metaheurísticas elegidas modificarán el conjunto de parámetros del modelo para ajustarse a los datos reales de las distintas instancias. Calibramos cuatro parámetros reales del modelo anteriormente descrito: los dos parámetros que regulan la estacionalidad $d, f \in [0,1]$, el coeficiente de innovación $\hat{p} \in [0,1]$, y el coeficiente de imitación $\hat{q} \in [0,1]$. Las metaheurísticas seleccionadas usan la misma función de *fitness*, que determina la calidad del modelo con relación a los datos reales de usuarios *premium*. Para evaluar el ajuste de un modelo utilizamos cada *dataset* R y lo dividimos en R_{train} y R_{test} para seguir un enfoque de validación *hold-out* [27]. Ambos conjuntos tienen sus variables de entorno correspondientes E_{train} y E_{test}, que definen las condiciones como el estado de la red social y el número de usuarios *premium*. Utilizaremos 15 ejecuciones de MC, siendo la estimación de error $\epsilon(R_{train}, M(P^*, E_{train}))$ la media del ajuste en estas 15 ejecuciones de la simulación. En nuestro caso elegimos L^2 o distancia euclídea para calcular el error ϵ:

$$L^2 = \sqrt{\sum_{i=0}^{n-1} |V_M(i) - V_R(i)|^2},$$

donde n es el número de puntos del histórico y V_M y V_R son los vectores de salida (nuevos usuarios *premium*) del modelo y de la instancia respectivamente. Por último, hemos implementado las metaheurísticas en Java utilizando el *framework* ECJ [28]. Las secciones siguientes presentan los detalles de las metaheurísticas elegidas: *hill climbing* [19], algoritmos genéticos [20], *differential evolution* [21], *PSO* [22], y CMA-ES [23].

A. HC

Hemos definido una búsqueda local para comparar su rendimiento con las metaheurísticas poblacionales. Nuestro procedimiento de búsqueda local implementa una estrategia tipo *hill climbing* [19]. Partiendo de una solución aleatoria, refina la calidad de la solución incrementando o decrementando el valor de uno de los parámetros de manera iterativa. Nuestra búsqueda local considera la variable $incremento$, que regula la variación aplicada a los parámetros de la solución. En cada iteración, la búsqueda local se moverá al primer vecino que mejore su *fitness* por un $umbral$ dado.

B. GA

Diseñamos nuestro algoritmo genético (GA) siguiendo una estrategia generacional donde cada nueva generación reemplaza a la anterior [20]. Fijamos el tamaño de población del algoritmo en 100 invididuos y el el número de generaciones en 200. Nuestra implementación usa selección por torneo [19], con k siendo el tamaño del torneo. También incluimos elitismo débil, por lo que el mejor padre siempre pasa a la siguiente generación. Como estrategia de cruce elegimos un cruce multipunto *PMX* con probabilidad p_c para cada individuo. Respecto a la estrategia de mutación, cada individuo tendrá una probabilidad p_m de reiniciar uno de sus parámetros (genes), sustituyendo su valor por un número aleatorio usando una distribución uniforme en el intervalo $[0,1]$.

C. PSO

Para nuestra implementación de *Particle Swarm Optimization* (PSO) [22] consideramos también una población de 100 individuos y 200 iteraciones. Cada individuo considera 5 vecinos que se asignan aleatoriamente durante la inicialización del algoritmo. En cada iteración, cada individuo se mueve (modifica sus valores) considerando su mejor configuración conocida, el mejor de sus vecinos, y el mejor global. Este movimiento se calcula utilizando cuatro parámetros que actúan como pesos: velocidad (v_c), mejor personal (p_c), mejor vecino (i_c), y mejor global (g_c).

D. DE

El diseño que hemos elegido para *differential evolution* (DE) [21] considera una población de 100 invididuos y se ejecuta durante 200 generaciones. La variante utilizada es *DE/rand/2*, que genera el vector donante a partir de la siguiente ecuación: $x_i(G+1) = x_{r1}(G) + F(x_{r2}(G) - x_{r3}(G))$, donde x_i es el vector generado y $r1$, $r2$, y $r3$ son soluciones diferentes. F es un parámetro del algoritmo que actúa de amplificador de la mutación. Esta variante genera nuevos individuos usando el vector donante con una probabilidad definida por el parámetro CR.

E. CMA-ES

Covariance matrix adaptation evolution strategy (CMA-ES) [23] es otro algoritmo bio-inspirado que desarrolla la mutación de su población (o recombinación) de acuerdo a una matriz de covarianza. CMA-ES suele funcionar muy bien con problemas similares al nuestro, ya que los parámetros son valores reales y el tamaño del conjunto a calibrar es pequeño. Hemos elegido la variante $(\mu/\mu_I, \lambda)$-CMA-ES, donde en cada iteración se combinan los mejores μ candidatos con λ nuevas soluciones.

IV. Experimentación

A. Configuración

La Tabla I muestra los parámetros que hemos usado para nuestros experimentos. Nuestro proceso de calibración considera datos reales de *Creature Party* divididos en cuatro datasets diferentes. Estos datasets contienen el número de usuarios premium por día para periodos de tiempo distintos que a su vez dividimos en *training* y *test*. El primer dataset contiene 91 días, con 60 para *training* y 31 para *test*. El segundo y el tercer dataset consideran 46 días, divididos en 31 días para *training* y 15 para *test*. El cuarto y último dataset contiene 45 días, con 30 para *training* y 15 para *test*.

Tabla I: Valores de los parámetros de las metaheurísticas

General		**GA**		**PSO**	
Nombre	Valor	Nombre	Valor	Nombre	Valor
Evaluaciones	20,000	p_c	1.0	v_c	0.3
Individuos	100	p_m	0.2	p_c	0.04
Generaciones	200	k	3	i_c	0.3
Monte-Carlo	15			g_c	0.3
HC		**DE**		**CMA-ES**	
Nombre	Valor	Nombre	Valor	Nombre	Valor
$incremento$	0.01	F	0.6	λ	8
$umbral$	0.01	CR	0.3	μ	4

B. Resultados de calibración

Mostramos los resultados de calibración obtenidos por las metaheurísticas en la Tabla II. Estos resultados muestran el ajuste de la mejor solución después de 15 ejecuciones con distintas semillas. Este ajuste se presenta como el *fitness* medio y la desviación típica de las 15 simulaciones de MC para *training* y *test*. Como se puede observar, el *fitness* de la mayoría de metaheurísticas es mejor para los datasets más pequeños (2, 3, y 4), lo cual parece coherente dado que su comportamiento debería ser más fácil de ajustar. Por otro lado, los valores de desviación típica son similares. Con respecto al ajuste de las soluciones, estros resultados muestran que CMA-ES supera a los otros algoritmos por un pequeño margen. Además de valores de fitness bajos, CMA-ES consigue unos valores de desviación típica reducidos, por lo que se muestra estable con distintas semillas.

Mostramos el rendimiento de los algoritmos poblacionales durante el proceso de calibración usando el dataset 3 en la Figura 1. Esta gráfica muestra la evolución del *fitness* de la mejor solución en cada generación, por lo que el progreso de los algoritmos se muestra cada 100 evaluaciones. Estos resultados muestran que los algoritmos alcanzan valores de error reducidos rápidamente y sugieren que aumentar el número de evaluaciones no mejoraría el resultado final de manera significativa.

En la Figura 2 mostramos una comparación entre los datos históricos y la salida del modelo calibrado para la mejor solución encontrada por CMA-ES para el segmento *training* del dataset 2. Estos resultados muestran que el proceso de calibración obtiene buenos resultados y que el modelo calibrado

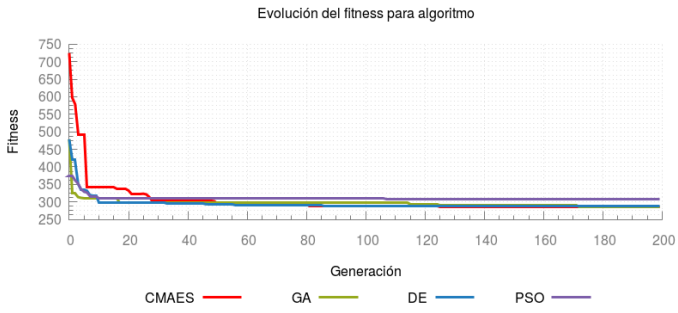


Figura 1: Evolución del *fitness* de los algoritmos poblacionales para el dataset 3.

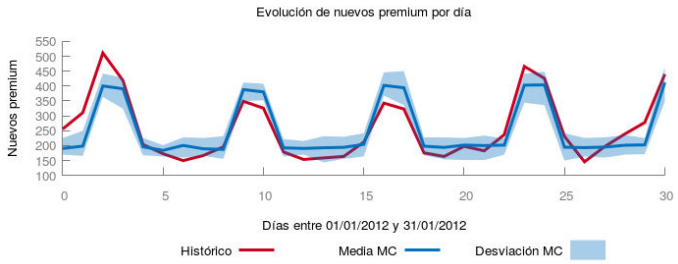


Figura 2: Comparación entre la salida del modelo calibrado y los datos históricos de *training* del dataset 2 (31 días).

se ajusta a los datos históricos. Por ejemplo en el step 4, el número de usuarios *premium* fue 204 y la simulación predice 196. Aunque hay excepciones, como el día 2 donde el número de usuarios *premium* fue 511 pero el modelo predice 404.

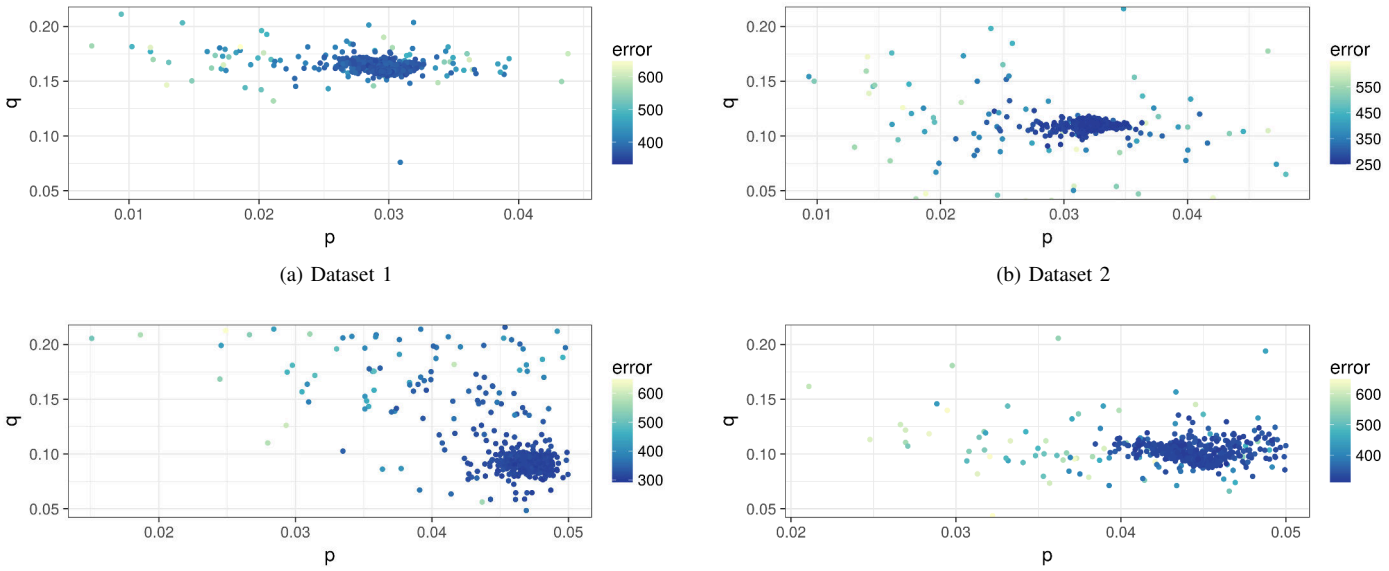
C. Análisis de sensibilidad

Nuestros experimentos señalan CMA-ES como el mejor método, consiguiendo los mejores resultados en tres conjuntos de *training* y uno de *test*. En esta sección hacemos un análisis de sensibilidad a las soluciones generadas por CMA-ES para cada dataset. Los *scatter plots* de las Figuras 3 y 4 muestran los valores de los coeficientes de Bass y los parámetros de estacionalidad para los soluciones generadas por CMA-ES durante el proceso de calibración. Estas soluciones provienen de una única ejecución de la metaheurística, recogidas desde el principio de la calibración hasta que el algoritmo termina, filtrando las soluciones que superen 500 de *fitness*. Este filtro permite concentrar el análisis en las mejores soluciones.

Podemos observar que los valores de los coeficientes de Bass de las soluciones para los distintos dataset son similares. Por ejemplo, para el primer dataset (Figura 3a) las mejores soluciones rondan $\hat{p} = 0{,}03$ y $\hat{q} = 0{,}16$. En cambio el coeficiente de imitación es ligeramente más bajo en el resto de datasets ($\hat{q} = 0{,}1$). En el caso del coeficiente de innovación, los dos primeros dataset (Figura 3a y 3b) muestran sus mejores valores alrededor de $\hat{p} = 0{,}03$ y los dos siguientes (Figura 3c y 3d)) alrededor de $\hat{p} = 0{,}045$. En estos últimos observamos también mayor dispersión en sus soluciones con

Tabla II: Fitness medio y desviación típica de cada metaheurística para cada dataset

	Dataset 1		Dataset 2		Dataset 3		Dataset 4	
	TRA	TEST	TRA	TEST	TRA	TEST	TRA	TEST
HC	365.6 (18.7)	349.2 (21.9)	256.0 (17.2)	211.0 (20.3)	301.1 (19.8)	351.7 (15.7)	336.1 (30.3)	**222.1 (20.1)**
GA	351.9 (19.6)	333.7 (22.4)	247.6 (24.8)	182.1 (19.1)	286.0 (22.4)	331.4 (20.0)	304.5 (26.0)	283.7 (31.5)
DE	351.8 (18.9)	335.3 (20.4)	**245.3 (22.0)**	194.8 (20.5)	286.8 (21.6)	348.0 (17,0)	**301.6 (17.6)**	283.5 (34.2)
PSO	347.2 (15.0)	332.5 (20.8)	246.1 (13.8)	191.7 (19.5)	308.0 (19.7)	**261.5 (22.7)**	316.5 (31.5)	365.2 (20.4)
CMA-ES	**326.9 (18.2)**	**296.4 (15.6)**	248.1 (13.9)	**164.8 (17.0)**	**286.0 (16.9)**	318.1 (22.3)	303.3 (24.4)	267.2 (26.4)



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

Figura 3: *Scatter plots* para el análisis de sensibilidad sobre los coeficientes de Bass (p/q).

mejor *fitness*, lo que podría indicar que estos dataset tienen mayor tolerancia a distintos valores de estos parámetros. Globalmente, se puede observar que las soluciones muestran mayor dispersión para el coeficiente de innovación ($\hat{p}$), lo que indica que es un parámetro más sensible. En comparación con otras configuraciones del modelo de Bass para otros mercados publicados anteriormente en la literatura de marketing [12], los modelos calibrados presentan un valor alto para el coeficiente de innovación ($\hat{p}$) y un valor bajo para el coeficiente de imitación ($\hat{q}$).

Por último, los *scatter plots* de la Figura 4 muestran los valores de estacionalidad para los soluciones encontradas para los distintos datasets. Estas soluciones muestran valores constantes en los diferentes datasets, con valores que rondan d=0,1 para la probabilidad de jugar entre semana y f=0,2 para la probabilidad de jugar en fin de semana. Estos parámetros también muestran mayor concentración en el parámetro (d), lo que sugiere que el ABM de *Creature Party* es más sensible a los cambios en la probabilidad de jugar entre semana que en para la probabilidad de jugar en el fin de semana.

V. Conclusiones

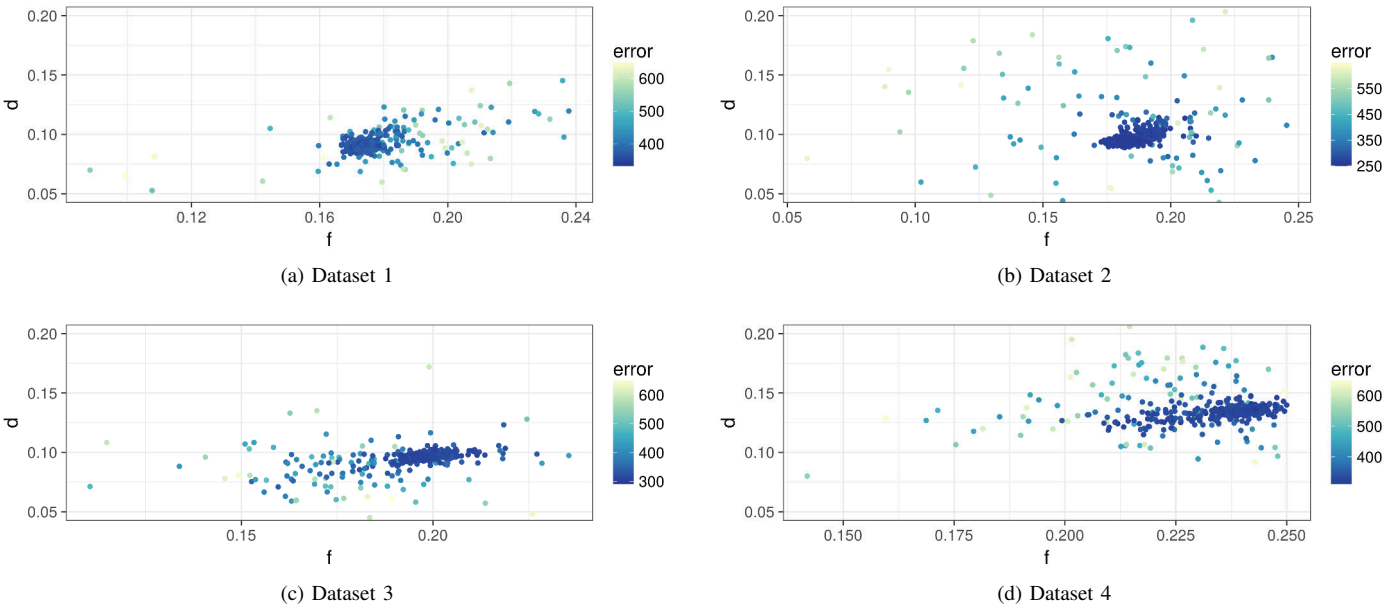
En este artículo hemos aplicado distintas metaheurísticas a la calibración de un modelo ABM usando datos reales. Después de comparar el rendimiento de los algoritmos, hemos elegido las soluciones calibradas por CMA-ES, la mejor metaheurística según nuestra experimentación, para desarrollar un análisis de sensibilidad sobre sus parámetros. Usando estas soluciones, hemos visualizado los valores de los coeficientes del modelo de Bass y los parámetros de estacionalidad.

Los resultados de calibración de las distintas metaheurísticas muestran valores de *fitness* cercanos. Sin embargo, CMA-ES consigue mejores soluciones por lo que lo distinguimos como el mejor método, lo cual es coherente con publicaciones anteriores sobre optimización continua utilizando metaheurísticas.

El análisis de sensibilidad aplicado sobre las soluciones calibradas por CMA-ES muestra que sus parámetros para los distintos dataset son similares. Además estos valores son diferentes de otros modelos similares [12], lo que podría indicar que las aplicaciones *freemium* como *Creature Party* tienen un componente de innovación mayor. Por último, dado que nuestro análisis no tiene en cuenta las características específicas del diseño del ABM subyacente, pensamos que nuestros resultados podrían ser generalizables a otros modelos.

Agradecimientos

Este trabajo está financiado por el Ministerio de Economía y Competitividad bajo el proyecto NEWSOCO (ref. TIN2015-67661-P), incluyendo Fondos Europeos de Desarrollo Regional (ERDF).



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

(d) Dataset 4

Figura 4: *Scatter plots* con los valores de los parámetros de estacionalidad: entre semana (d) y fin de semana (f).

Referencias

[1] V. Kumar, "Making freemium work: Many start-ups fail to recognize the challenges of this popular business model." *Harvard Business Review*, vol. 92, no. 5, pp. 27–29, May 2014. [Online]. Available: https://hbr.org/2014/05/making-freemium-work

[2] M. Trusov, R. E. Bucklin, and K. Pauwels, "Effects of word-of-mouth versus traditional marketing: Findings from an internet social networking site," *Journal of Marketing*, vol. 73, no. 5, pp. 90–102, 2009.

[3] B. Libai, R. Bolton, M. S. Bügel, K. De Ruyter, O. Götz, H. Risselada, and A. T. Stephen, "Customer-to-customer interactions: broadening the scope of word of mouth research," *Journal of Service Research*, vol. 13, no. 3, pp. 267–282, 2010.

[4] O. Hinz, B. Skiera, C. Barrot, and J. U. Becker, "Seeding strategies for viral marketing: An empirical comparison," *Journal of Marketing*, vol. 75, no. 6, pp. 55–71, 2011.

[5] R. Van der Lans, G. Van Bruggen, J. Eliashberg, and B. Wierenga, "A viral branching model for predicting the spread of electronic word of mouth," *Marketing Science*, vol. 29, no. 2, pp. 348–365, 2010.

[6] D. J. Watts and P. S. Dodds, "Influentials, networks, and public opinion formation," *Journal of Consumer Research*, vol. 34, no. 4, pp. 441–458, 2007.

[7] P. Schmitt, B. Skiera, and C. Van den Bulte, "Referral programs and customer value," *Journal of Marketing*, vol. 75, no. 1, pp. 46–59, 2011.

[8] M. Chica and W. Rand, "Building agent-based decision support systems for word-of-mouth programs: A freemium application," *Journal of Marketing Research*, vol. 54, no. 5, pp. 752–767, 2017. [Online]. Available: https://doi.org/10.1509/jmr.15.0443

[9] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the 37th conference on Winter simulation*. ACM, 2005, pp. 2–15.

[10] J. M. Epstein, *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, 2006.

[11] M. A. Janssen and E. Ostrom, "Empirically based, agent-based models," *Ecology and Society*, vol. 11, no. 2, p. 37, 2006.

[12] W. Rand and R. T. Rust, "Agent-based modeling in marketing: Guidelines for rigor," *International Journal of Research in Marketing*, vol. 28, no. 3, pp. 181–193, 2011.

[13] M. Trusov, W. Rand, and Y. V. Joshi, "Improving prelaunch diffusion forecasts: Using synthetic networks as simulated priors," *Journal of Marketing Research*, vol. 50, no. 6, pp. 675–690, 2013.

[14] M. Haenlein and B. Libai, "Targeting revenue leaders for a new product," *Journal of Marketing*, vol. 77, no. 3, pp. 65–80, 2013.

[15] F. Viger and M. Latapy, "Efficient and simple generation of random simple connected graphs with prescribed degree sequence," in *Lecture Notes in Computer Science. Computing and Combinatorics*. Springer, 2005, vol. 3595, pp. 440–449.

[16] F. M. Bass, "A new product growth model for consumer durables," *Management Science*, vol. 36, no. 9, pp. 1057–1079, 1969.

[17] R. Oliva, "Model calibration as a testing strategy for system dynamics models," *European Journal of Operational Research*, vol. 151, no. 3, pp. 552–568, 2003.

[18] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings of the 37th conference on Winter simulation*, 2005, pp. 130–143.

[19] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[20] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of evolutionary computation*. Bristol (UK): IOP Publishing Ltd., 1997.

[21] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[23] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[24] P. W. Farris, N. T. Bendle, P. E. Pfeifer, and D. J. Reibstein, *Marketing metrics: The definitive guide to measuring marketing performance*, 2nd ed. Wharton School Publishing, 2010.

[25] R. Milo, N. Kashtan, S. Itzkovitz, M. Newman, and U. Alon, "On the uniform generation of random graphs with prescribed degree sequences," *arXiv preprint http://arxiv.org/abs/cond-mat/*, 2004.

[26] J. H. Miller, "Active nonlinear tests (ANTs) of complex simulation models," *Management Science*, vol. 44, no. 6, pp. 820–830, 1998.

[27] F. Stonedahl and W. Rand, "When does simulated data match real data? Comparing model calibration functions using genetic algorithms," in *Advances in Computational Social Science*, ser. Agent-Based Social Systems. Springer, Japan, 2014, vol. 11, pp. 297–313.

[28] S. Luke, L. Panait, G. Balan, S. Paus, Z. Skolicki, J. Bassett, R. Hubley, and A. Chircop, "Ecj: A java-based evolutionary computation research system," *Downloadable versions and documentation can be found at the following url: http://cs. gmu. edu/eclab/projects/ecj*, 2006.

An empirical validation of a new memetic CRO algorithm for the approximation of time series*

*Note: The full contents of this paper have been published in the volume *Lecture Notes in Artificial Intelligence 11160* (LNAI 11160)

Antonio M. Durán-Rosal
Dept. of Computer Science and Numerical Analysis
University of Córdoba
Córdoba, Spain
aduran@uco.es

Pedro A. Gutiérrez
Dept. of Computer Science and Numerical Analysis
University of Córdoba
Córdoba, Spain
pagutierrez@uco.es

Sancho Salcedo-Sanz
Department of Signal Processing and Communications
Universidad de Alcalá
Madrid, Spain
sancho.salcedo@uah.es

César Hervás-Martínez
Dept. of Computer Science and Numerical Analysis
University of Córdoba
Córdoba, Spain
chervas@uco.es

***Abstract*—The exponential increase of available temporal data encourages the development of new automatic techniques to reduce the number of points of time series. In this paper, we propose a novel modification of the coral reefs optimization algorithm (CRO) to reduce the size of the time series with the minimum error of approximation. During the evolution, the solutions are locally optimised and reintroduced in the optimization process. The hybridization is performed using two well-known state-of-the-art algorithms, namely Bottom-Up and Top-Down. The resulting algorithm, called memetic CRO (MCRO), is compared against standard CRO, its statistically driven version (SCRO) and their hybrid versions (HCRO and HSCRO, respectively). The methodology is tested in 15 time series collected from different sources, including financial problems, oceanography data, and cardiology signals, among others, showing that the best results are obtained by MCRO.**

***Index Terms*—Time series size reduction, segmentation, coral reefs optimization, memetic algorithms**

Modelling Mandible Articulation for Skull-Face Overlay in Forensic Identification[1]

E. Bermejo, C. Campomanes-Álvarez,
A. Valsecchi, O. Ibáñez, and S.Damas
*Andalusian Institute on Data Science and Computational Intelligence (DaSCI), University of Granada
Email: enrique.bermejo@decsai.ugr.es, sdamas@ugr.es

O. Cordón*
Research Center on Information and Communication Technologies
University of Granada
Email: ocordon@decsai.ugr.es

I. Introduction

Forensic anthropology is a sub-field of physical anthropology that involves applying scientific knowledge to the collection and analysis of medico-legal information. It includes the recovery, identification, and description of human skeletal remains [2]. Currently, several forensic identification techniques are available, i.e., DNA samples, fingerprint recognition, or dental records. When none of the previous methods can be applied, the analysis of skeletal remains becomes the last resort of forensic identification. One of the alternative techniques is craniofacial superimposition (CFS) [3], which involves the superimposition of a complete skull (or a skull model) with a number of *ante mortem* (AM) images of an individual. CFS is thus a technique used when only skeletal information is available for the forensic assessment and other techniques such as DNA or dental record analysis are not possible or conclusive.

Traditional CFS techniques are tedious and based on a 'trial and error' process requiring several hours of manual processing to obtain a correct superimposition. Therefore, there is a strong interest on designing automatic methods to assist the CFS identification procedure [4]. The process requires a forensic expert to position the skull in the same pose as the face in the photograph. This process is known as Skull-Face Overlay (SFO).

Up to now, all computer-based SFO methods have considered the mandible as a rigid part of the skull. These methods usually follow one of the following approaches to approximate the mandible aperture [5]: i) Before capturing the 3D model, the mandible was manually located relative to the cranium so that the model resembled the facial expression of the photograph under study; and ii) Once the mandible and the cranium were scanned, the 3D models were positioned according to the relative aperture in the photograph using 3D modeling software.

Such a simplification (*anatomically incorrect*) causes a negative impact on the accuracy of the automatic SFO method. As the AM images used to perform CFS are typically provided by relatives, the missing person usually appears in relaxed situations, most of them smiling or with the mouth slightly open. Generally, cases with grimaces or forced poses are discarded due to the fact that the mandible is in an exaggerated position and these kinds of facial expressions distort the soft tissue of the face. Additionally, each individual comparison should involve the analysis of one skull against more than one AM photograph of the same person to significantly increase the reliability and accuracy of the method [6]. Overall, this is a very time-consuming task even using an automatic SFO method.

[1] This article is a summary of the work published in Information Sciences [1], to be considered as a part of the CAEPIA'18 Key Works. The motivation, the main contributions and some conclusions are briefly summarized.

II. Motivation, Proposal, and Conclusions

Those photographs where subjects appear with their mouths open reduce the confidence of the identification. Therefore, it is essential to model the articulation of the mandible in order to improve CFS reliability, considering that we only have skeletal information available to infer its movement. In our contribution, we have modeled and integrated the mandible articulation within the SFO optimization algorithm.

In particular, we considered a simple model [7] to estimate and parameterize the mandible aperture movement using the aperture percentage. Moreover, we proposed different design alternatives to integrate the estimation of the mandible aperture within the scheme of the current state-of-the-art SFO optimization algorithm [8], namely RCGA. Specifically, RCGA is a real-coded, elitist genetic algorithm that performs the registration of the 3D skull with the 2D AM photograph.

Our proposal involves three different design alternatives to balance the exploration-exploitation trade-off during the optimization. We have performed a thorough experimental study to analyze the suitability of the proposed articulation model to the SFO technique. Additionally, we have designed a ground-truth database to allow an objective evaluation of the reliability of our proposal.

According to the results of this experimentation (Figure 1), the application of a simple mandible aperture model has proven its effectiveness, significantly improving the accuracy and the versatility of the state-of-the-art automatic SFO method. Such an outstanding performance facilitates the use of facial photos where the individual either smiles or opens the mouth partially. Such photos have been usually discarded

in real identification scenarios. The availability of new AM evidence is a crucial consequence of this work. Indeed, the use of multiple facial photos of the same individual is essential to increase the reliability of the identification based in CFS [9]

In our proposal, the expert is only required to pinpoint the incisors, and the mandibular and cranial condyles, a much simpler task to carry out. This task is only performed once, no matter which number of AM photos the skull is compared with. Thereby, our articulated SFO approach avoids entirely the time-consuming and error-prone positioning of the mandible, besides adding versatility to the procedure as it can adapt its pose to different mouth openings in the photographs.

Acknowledgment

This publication is supported by Spanish Ministerio de Economía y Competitividad under the NEWSOCO project (ref. TIN2015-67661-P) and the Andalusian Dept. of Innovación, Ciencia y Empresa under project TIC2011-7745, both including European Regional Development Funds (ERDF). Dr. C. Campomanes Álvarez's work has been supported by Spanish MECD FPU grant AP-2012-4285. Dr. Ibáñez's work has been supported by Spanish MINECO Juan de la Cierva Fellowship JCI-2012-15359.

References

[1] E. Bermejo, C. Campomanes-Álvarez, A. Valsecchi, O. Ibáñez, S. Damas, and O. Cordón, "Genetic algorithms for skull-face overlay including mandible articulation," *Information Sciences*, vol. 420, pp. 200 – 217, 2017.

[2] M. Yoshino and S. Seta, "Skull-photo superimposition," *Encyclopedia of Forensic Sciences*, vol. 2, pp. 807–815, 2000.

[3] M. Yoshino, *Craniofacial superimposition*, in: C. Wilkinson and C. Rynn, Eds. Craniofacial Identification, University Press, Cambridge, 2012.

[4] S. Damas, O. Cordón, O. Ibáñez, J. Santamaría, I. Alemán, M. Botella, and F. Navarro, "Forensic identification by computer-aided craniofacial superimposition," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–27, 2011.

[5] M. I. Huete, O. Ibáñez, C. Wilkinson, and T. Kahana, "Past, present, and future of craniofacial superimposition: Literature and international surveys," *Legal Medicine*, vol. 17, no. 4, pp. 267–278, 2015.

[6] S. Damas, C. Wilkinson, T. Kahana, E. Veselovskaya, A. Abramov, R. Jankauskas, P. T. Jayaprakash, E. Ruiz, F. Navarro, M. I. Huete, E. Cunha, F. Cavalli, J. Clement, P. Lestón, F. Molinero, T. Briers, F. Viegas, K. Imaizumi, D. Humpire, and O. Ibáñez, *Study on the performance of different craniofacial superimposition approaches (II): Best practices proposal*. Forensic Science International, 2015, vol. 257.

[7] J. J. Lemoine, J. J. Xia, C. R. Andersen, J. Gateno, W. Buford, and M. A. K. Liebschner, "Geometry-Based Algorithm for the Prediction of Nonpathologic Mandibular Movement," *Journal of Oral and Maxillofacial Surgery*, vol. 65, no. 12, pp. 2411–2417, 2007.

[8] B. R. Campomanes-Alvarez, O. Ibanez, C. Campomanes-Alvarez, S. Damas, and O. Cordón, "Modeling Facial Soft Tissue Thickness for Automatic Skull-Face Overlay," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2057–2070, oct 2015.

[9] D. Austin-Smith and W. R. Maples, "The reliability of skull/photograph superimposition in individual identification," *Journal of Forensic Sciences*, vol. 39, no. 2, pp. 446–55, mar 1994.

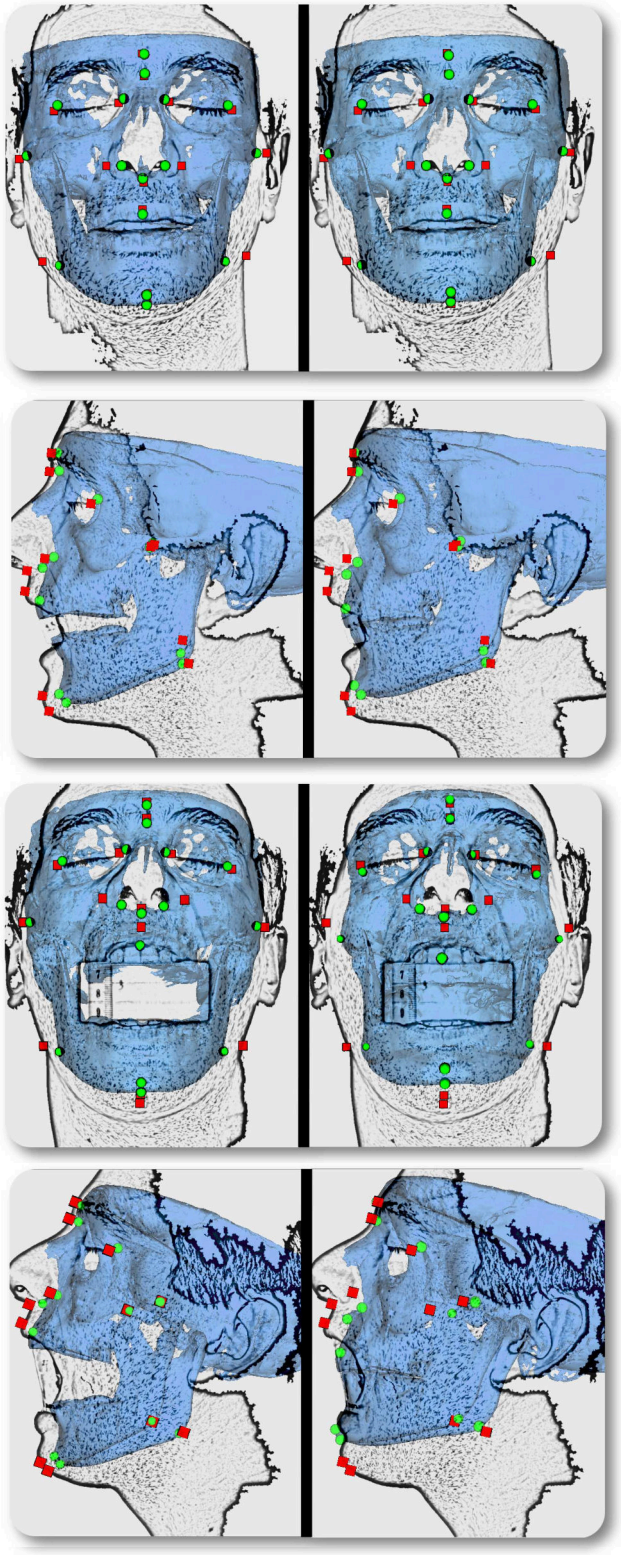


Fig. 1. Visual SFO results: A1 (left images) and RCGA (right images) for different frontal and lateral instances (Smile, 15, 30, and 40).

Propuestas de mejora para la evaluación del arte evolutivo.

Francisco Fernández de Vega[1], Cayetano Cruz[1], Patricia Hernández[2]

I. Resumen

En este trabajo se discute el modo de evaluar adecuadamente el arte evolutivo, no tanto desde el punto de vista de la función de fitness, que también, sino considerando el resultado final al que se llega. En la aplicación tradicional de los algoritmos evolutivos a la resolución de problemas el resultado obtenido es fácilmente evaluable y comparable con otras metodologías, pues basta obtener una medida cuantitativa de la calidad final de la solución. Sin embargo, en los procesos asociados al arte y diseño, estas medidas objetivas no son tan sencillas. Este trabajo, que se plantea como un posicionamiento sobre la cuestión de cómo evaluar apropiadamente el arte evolutivo, revisa alguna de las propuestas que se han producido en este ámbito, pasando por la evaluación interactiva, el test de turing para el arte así como algunas competiciones internacionales en el dominio de los algoritmos evolutivos, para discutir finalmente el papel que otros actores fundamentales, y frecuentemente olvidados en el mundo del arte, están teniendo en este arte generativo basado en evolución: el público, la crítica, galerías de arte, museos, etc. Una vez discutida la cuestión, y planteada la solución propuesta, se describen algunas experiencias en las que algunos, si no todos, de estos actores han sido incluidos en el proceso, y los resultados que se han obtenido. Entendemos que estas experiencias pueden servir de ejemplo para otras iniciativas artísticas que puedan surgir en este ámbito de aplicación de los algoritmos evolutivos, y a la vez la propuesta sirva para clarificar el papel del arte evolutivo en el contexto artístico en general.

II. Introducción

La aplicación de los algoritmos evolutivos al arte y diseño no es nuevo [11]. Desde la década de los noventa, son muchos los trabajos que han utilizado diferentes variaciones de los algoritmos evolutivos para generar lo que los autores consideran *arte*. De hecho, el arte generativo se ha estado utilizando durante el siglo XX para, basado en medios físicos, químicos, mecánicos o en computadores, generar un nuevo tipo de arte. Ya en los años 60, se organizó en Londres una llamativa exposición en la que arte gráfico, poesía, música, etc, era generada mediante programas de computador [10].

El término *arte* es algo que depende de la sociedad, y ha variado con frecuencia a lo largo de los siglos. Como caso paradigmático de esta variación, cabe destacar el rechazo que el salón de arte de París tuvo hacia los nuevos pintores denominados impresionistas, que decidieron abrir un salón alternativo donde mostrar su obra [6]. En sus inicios, el impresionismo era rechazado por la crítica -compuesta fundamentalmente por los eruditos y artistas más tradicionales de la época, aunque fue recibiendo aceptación del público de forma gradual, y llegó con el tiempo a ser considerado una nueva forma de expresión artística al mismo nivel que el arte más *clásico*. Al impresionismo le siguieron otras muchas tendencias en el siglo XX, como el expresionismo, cubismo, futurismo, dadaismo, surrealismo, etc. Y es que las formas artísticas trazan líneas que navegan arrastrando esa cadena de colisiones haciendo del arte un lugar activo, positivo, lúcido, versátil y maleable.

Del mismo modo, en el mundo de la música *seria*, los comienzos del siglo XX en Estados Unidos de América vieron nacer una nueva forma de expresión musical fruto del mestizaje de culturas y tradiciones provenientes de varios continentes. La música Jazz, que inicialmente era despreciada como música de *raza* (eufemismo para hablar de personas de color), fue conquistando los gustos del público, y posteriormente, también la crítica musical entendió sus fundamentos e importancia, de tal modo que alguno de los compositores y músicos Jazz de color, son hoy día considerados como alguno de los más grandes artistas y compositores de la historia, incluyendo a Luis Amstrong -trompetista- y Duke Ellington -

[1]Universidad de Extremadura, 06800 Mérida, España. fcofdez@unex.es, ccruz@unex.es
[2]Universidad de Sevilla, España patriciapahr@gmail.com

pianista-, reconocido actualmente este último como uno de los más grandes compositores americanos de la historia [9].

También lo contrario ha sucedido con frecuencia: tendencias artísticas consideradas desde el inicio como *arte formal* o académico no han sido aceptadas por el público de forma mayoritaria, tal como sucede con el serialismo y dodecafonismo de Schoemberg, que aunque de utilidad innegable asociado a otras formas de arte, como el cine, sigue despertando poco interés entre el público en general [2], y pocas son las compañías, si es que hubiera alguna, que puedan sobrevivir interpretando solamente obras de este tipo sin ayudas del sector público. Quizá sea necesaria una reflexión adicional sobre qué tipo de forma de arte se adapta mejor a la naturaleza intrínseca del ser humano, que es finalmente el destinatario final de la misma, y sobre si el término *arte* se ha desvirtuado o no en el siglo XX y convertido en *cajón de sastre* para incluir elementos que quizá debieran categorizarse de otro modo. Como en otros ámbitos sociales, estirar en demasía el significado de un término puedo provocar que el mismo pierda su contenido semántico. Pero más allá de esta reflexión, conviene aquí considerar el arte evolutivo (y generativo en términos más amplios) en relación al arte humano.

Podemos resumir diciendo, que en la evolución y cambio que el término *arte* ha tenido a lo largo de los siglos, se han visto implicados no sólo los eruditos, que podríamos describir como la *crítica*, sino también el público que con su favor impulsa o no a ciertos artistas, y los museos que acogen y exponen lo que se considera *arte*.

Sin embargo, cuando nos fijamos en los resultados obtenidos mediante procesos evolutivos en el mundo del arte [11] -y centrándonos aquí en el arte plástico- no todos los actores anteriores han sido tenidos en cuenta, como veremos a continuación.

El arte contiene una parte de naturaleza transitoria y circunstancial propia de un determinado espacio tiempo y en este trabajo se pretende revisar la importancia de cada uno de esos actores en el arte evolutivo, y alguno de los resultados más llamativos obtenidos cuando todos los actores participan de algún modo en el proceso.

Este artículo por tanto pretende mostrar el camino hacia una correcta evaluación de trabajos artísticos generados mediante procedimientos basados en algoritmos evolutivos, y describe experiencias recientes que tienen en cuenta la propuesta planteada.

El resto del artículo se estructura del siguiente modo: La sección III describe las dificultades para evaluar adecuadamente el arte evolutivo. En la sección IV presentamos nuestro posicionamiento al respecto, con ideas sobre como mejorar. La sección V presenta algunos resultados obtenidos mediante la propuesta descrita, y finalmente la sección VI resume nuestras conclusiones.

III. Estado del arte

Cuando los investigadores decidieron por primera vez aplicar los algoritmos evolutivos en procesos de creación artística, se encontraron un problema, que es la clave en la evaluación del arte a lo largo de la historia: ¿Cómo evaluar correctamente la calidad estética de una imagen surgida de la evolución?

La solución tradicional ha sido permitir que sean los humanos quienes evalúen la calidad, produciendo así un cambio profundo en los algoritmos evolutivos tradicionales, dando lugar a lo que se conocen como Algoritmos Evolutivos Interactivos, con multitud de aplicaciones hoy día [13].

Así, las herramientas que se han desarrollado y están disponibles para el arte evolutivo, tal como PicBreeder, se basan en el buen criterio de las personas a la hora de elegir lo que se considere estéticamente más razonable [12].

Ha habido intentos de encerrar en fórmulas matemáticas la calidad estética de una obra plástica, pero el resultado no ha sido muy afortunado [8]. También se ha intentado evaluar cual es la opinión del público sobre el arte plástico, cuales son sus preferencias, y con un procedimiento basado en encuestas, se llegó a conclusiones poco satisfactorias [3].

Trabajos como este último ponen el dedo en la llaga: la participación de un alto número de usuarios diferentes dando su opinión sobre qué es más apreciado en una obra de arte, cuyo ejemplo más notable es Picbreeder, no tiene necesariamente que conducir a un resultado adecuado.

Pero entonces, si la participación de usuarios no es suficiente para garantizar la calidad del resultado, ¿qué otros elementos sería necesario incluir?

En algunos trabajos recientes se habla de la posibilidad de generar un test de turing para el arte basado en computador [1], que podría aplicarse por tanto al arte evolutivo. Pero precisamente en este modelo hay varios elementos no triviales:

- ¿Quiénes deben actuar como jurado?
- ¿Con qué arte humano comparamos, con el producido por artistas profesionales, o cualquier persona en general?

- El hecho de que un jurado no sepa si una obra ha sido generada por ordenador o no, ¿Le otorga automáticamente un sello de calidad?

Estas son las tres preguntas básicas a las que nos vamos a enfrentar en la siguiente sección, describiendo alguna de las propuestas en las que hemos trabajado en los últimos años, y los resultados que hemos conseguido para intentar proponer modelos más adecuados para el arte evolutivo.

IV. Metodología: Sobre la forma de evaluar el arte evolutivo

Hasta la fecha, las buenas intenciones han dominado el arte evolutivo. En general, un algoritmo interactivo -que es el utilizado de forma tradicional en este contexto- funciona del siguiente modo: todas las operaciones del algoritmos son llevadas a cabo de modo estándar salvo la evaluación del fitness. Así, los usuarios se encargan de ir decidiendo la calidad del ”producto en evolución”. Pero en este proceso, el cansancio del usuario afecta al resultado del proceso, como ya ha sido reconocido con anterioridad [4]. Aunque existen propuestas que tratan de evitar este cansancio utilizando dispositivos no intrusivos que analizan el comportamiento del usuario [3], el problema de quién evalúa la calidad -el usuario en este caso- sigue presente.

Una vez que el algoritmo termina, el resultado obtenido en la última generación sería el *producto* generado. Pero en el arte evolutivo, la *obra de arte* producida, si se pretende que emule en calidad la contraparte humana, debería ser mostrada al público y evaluada de forma similar al resto de obras de arte. En realidad no es esto lo que sucede en la mayoría de los casos; típicamente, los resultados obtenidos mediante algoritmos evolutivos han sido presentados a la comunidad en sus propios foros, que incluye revistas dónde los algoritmos son descritos y congresos especializados, tal como EvoMusart, o más generalistas que acogen cualquier resultado notable en el área. Pero en pocas ocasiones este proceso de exposición y muestra es utilizado para obtener información de lo que el público opina.

Este modo de proceder, es bastante lejano a cómo en realidad el mundo del arte funciona. Examinaremos con detalle los tres puntos destacados anteriormente, y propondremos en cada caso cómo podríamos acercarnos más al circuito artístico internacional, para llegar a que el arte evolutivo pueda ocupar el puesto que merece.

IV-A. El Test de Turing para el arte

Como se describía más arriba, la dificultad para medir la calidad estética de las obras, ha llevado a proponer la necesidad de un Test de Turing para el arte: si un jurado humano no puede distinguir una obra generada por un computador de una elaborada por un humano, el trabajo, se diría, supera el Test de Turing.

Hay tres razones para cuestionar la utilidad de un test como el anterior. En primer lugar, la indecidibilidad sobre el origen de una obra -humana o creada por computador- no tiene porqué otorgarle de forma automática un sello de calidad. En el arte tradicional, la importancia del reconocimiento del artista detrás de la obra se debe principalmente a cuestiones históricas, de consideraciones académicas, favor del público, y, por último, del mercado del arte, que más que calidad, lo que evalúa es la cotización del artista firmante [14].

Pero en realidad hay una segunda razón que permite cuestionar el mencionado test tenga sentido: algunas tendencias artísticas nacidas en el siglo XX, generan resultados que fácilmente podrían confundirse con el producto de un programa de ordenador. En música, el serialismo y la atonalidad se basa en procesos con un componente estocásticos, y que aunque son llevados a cabo por personas, son sencillamente emulables por computador (mucho más que la música tonal de tradición clásica occidental, en la que el alto número de reglas que la define implica la construcción de sistemas basados en reglas, o con aprendizaje sobre casos conocidos, de alta complejidad [15]). Igualmente en el arte plástico, el uso de herramientas digitales por parte de los artistas, podría también inducir a error a los jurados, confundiendo arte humano, en este caso, por arte generado evolutivamente.

En resumen, el Test de Turing no parece la herramienta necesaria para calificar la calidad de un resultado generado mediante evolución. Quizá, pudiera ser más reveladora la apreciación de especialistas con criterio artístico, como los curadores. No obstante, más allá de la capacidad técnica para emular la manera de hacer una obra, está la capacidad para elaborar un lenguaje diferenciado y reconocible, así como el discurso de trasfondo que subyace en la obra y su relación con factores relacionados con las tendencias y oportunidades de interés particular en cada momento histórico.

IV-B. Sobre los artistas.

En los ejemplos más conocidos de arte evolutivo, tal como el que pueda producirse con herramientas similares a Picbreeder, mediante un proceso que podríamos llamar *crowd-painting*, se invita a usuarios particulares, distribuidos a través de internet, para que de modo colectivo evolucionen imágenes. ¿Es éste el procedimiento adecuado para generar un producto artístico de calidad? Creemos que no. Si queremos que el producto final sea de calidad, es necesario incluir en el algoritmo los elementos necesarios -aunque puede que no suficientes- para conseguirlo. En nuestra opinión, esto requiere la presencia de artistas humanos, de modo que sean artistas experimentados los que puedan hacer la evaluación estética de los resultados parciales, y que de este modo el algoritmo pueda progresar, dado que hasta el momento no se ha encontrado la forma de evaluar automáticamente la calidad estética. No basta incluir usuarios en la interacción, deben ser usuarios especiales, artistas, los encargados de la evaluación de la calidad en el proceso evolutivo. Sólo cuando en el futuro los procesos de aprendizaje máquina consigan modelar a los artistas adecuadamente, podrán ser sustituidos, pero mientras tanto los artistas deben estar en el centro del proceso de evaluación.

En algunas de nuestras experiencias recientes, optamos por esta línea de trabajo [16]. No obstante, aunque creemos necesaria la presencia de artistas en el algoritmo evolutivo, esto no tiene porqué ser suficiente. Para certificar la calidad de un trabajo, debe haber una evaluación final que permita discernir sobre la calidad del algoritmo ejecutado, del mismo modo que después de la ejecución de un algoritmo evolutivo estándar, se evalúa la calidad final para saber si la solución obtenida es factible o, en caso contrario, ejecutar de nuevo el algoritmo para intentar de nuevo obtener una solución válida.

En el mundo del arte tradicional, son las competiciones, museos y galerías quienes aplican el filtro final, además, por supuesto, del público que es quién visita las exposiciones y, finalmente, adquiere las obras.

En el dominio de los algoritmos evolutivos, algunos congresos de relevancia, han organizado en ocasiones competiciones internacionales. Cabe destacar el Gecco Evolutionary Art, Design and Creativity competition [7], que ha permitido a los artistas evolutivos enviar sus trabajos para que un jurado internacional eligiera el ganador de la competición.

Este tipo de competiciones, permiten avanzar en alguno de los puntos controvertidos destacados en la sección previa: el establecimiento de un jurado y una competición permite que sea un grupo especializado el que dictamine. En esta línea, podemos referirnos a alguno de nuestros trabajos previos en este dominio, como XY que fue enviado a la competición en el año 2013 y resultó ganador (ver figura 1) [7]. Hay que destacar que aunque el trabajo se apartaba del modelo interactivo tradicional de algoritmo evolutivo, y en su lugar utilizaba la propuesta coocida como Algoritmo Evolutivo Desconectado [17], el resultado fue favorablemente evaluado por el jurado.

Figura 1. XY: ACM Gecco 2013 Evolutionary Art, Design and Creativity competition winner. Four out of the sixty works produced are displayed.

No obstante, la competición se desarrolla en un congreso específico de algoritmos evolutivos, ACM GECCO 2013 en este caso, y el jurado está compuesto por investigadores del área, algunos de ellos siendo también artistas. Pero por lo anterior, tanto el ámbito de la competición como el tipo de jueces, hace que exista una ”desviación o tendencia”que favorece el tipo de trabajos presentados.

Entendemos que es necesario dar un paso más para conseguir posicionar el arte evolutivo dónde le corresponde.

IV-C. Sobre los jurados.

La composición del jurado es crucial para calificar la calidad de un trabajo. En el mundo del arte, los curadores (comisarios, - curator en inglés) de las exposiciones son los que deciden los artistas cuyo trabajo merecen la pena. Por otro lado, en las competiciones artísticas, las selección de un jurado adecuado es lo que otorga calidad a la misma. Tanto competiciones como exposiciones se llevan a cabo en museos y galerías, que actuando como *crítica* permiten filtrar y definir en cada momento histórico lo que merece la pena del panorama artístico.

Por todo lo dicho anteriormente, entendemos que la forma más adecuada de evaluación del arte evolutivo,

es que compita con el arte humano en igualdad de condiciones en el contexto concreto del arte tradicional: competiciones de arte, galerías y museos.

Describimos a continuación la experiencia obtenida con las obras que hemos desarrollado en los últimos cinco años, y que nos ha permitido afrontar el proceso de selección de calidad descrito anteriormente, en un recorrido por ciudades y galerías de todo el mundo. En las experiencias que describimos, se incluyen tanto galerías de arte, como otras competiciones internacionales alejadas del círculo de los algoritmos evolutivos.

V. Experimentos y Resultados

Desde el año 2012, utilizando el Algoritmo Evolutivo Desconectado -que avanza en la dirección del algoritmo evolutivo interactivo, permitiendo al usuario ejecutar todos y cada uno de los pasos del algoritmo evolutivo [16]- hemos ido recorriendo una serie de etapas que nos han permitido mejorar los procesos de evaluación de los resultados artísticos, avanzando en el camino que creemos es más adecuado para posicionar el Arte Evolutivo en el mundo del arte plástico, llegando a obtener algunos de los objetivos perseguidos y, como se verá más adelante, tan sólo pendiente de la decisión sobre la ubicación definitiva de las obras generadas, ya sea en alguna colección particular o en algún museo de arte contemporaneo. Tal como se describe anteriormente en la metodología, hemos tratado de afrontar procesos de evaluación de calidad cada vez más cerca del mundo del arte, y más lejos del área de los algoritmos evolutivos.

V-A. Galerías de Arte y su público

Han sido tres las obras colectivas de arte evolutivo que hemos producido en estos últimos cinco años: *XY*, *XYZ* y finalmente *The horizon project* [5]. Desde el principio se consideró la necesidad de exponer las obras y obtener realimentación de la opinión del público.

En cuanto a la primera obra, XY, se decidió mostrarla en congresos del área. Las exposiciones en Cancún, Madrid y Mérida, en congresos como IEEE CEC o MAEB, permitieron obtener información de público *afín*, y detectamos los problemas asociados con la utilización de encuestas a los usuarios, que implican un cansancio similar al de los usuarios que participan de modo interactivo con los algoritmos evolutivos.

Este mismo problema, que se ha repetido con las obras posteriores, está ya siendo objeto de mejora. Recientemente hemos trabajado en un modelo de exposición interactiva, que permite analizar al usuario en la visita, mediante dispositivos Kinnect. Los resultados muestran que los datos obtenidos son similares a los obtenidos mediante encuesta, con la ventaja de ser métodos no intrusivos [3]. No obstante, lo que más importa en la propuesta que hacemos, es el modo de evaluación por los actores fundamentales del mundo del arte.

En una segunda etapa, y considerando que el lugar natural para un artista y su obra es una galería de arte, decidimos apostar por este ámbito expositivo. Así, en 2015 y 2017, organizamos exposiciones temporales de la obra XYZ en la Galería de Arte ”Back Ballery Project”, en Vancouver. Conseguimos gestionar la exposición e inauguración del evento en la galería la misma semana que tuvo lugar el congreso IEEE CEC en la ciudad. En el congreso se mostraron algunas reproducciones de la obra, y se redirigió a los asistentes a la galería para la visita. Hay que enfatizar el hecho de que no había ninguna conexión previa entre la galería de arte, dedicada exclusivamente al mundo del arte, y el congreso mencionado.

De forma parecida, se gestionó otra exposición en la Gallerie Louchard de París, en Octubre de 2017, aunque en este caso, la galería si ofrecía a los organizadores del congreso Artificial Evolution 2017 un espacio en el que mostrar arte evolutivo enviado al congreso.

Lo anterior muestra que las galerías de arte pueden estar interesadas en el arte evolutivo igual que en cualquier otro tipo de arte contemporáneo, y que la aceptación de la obra para ser expuesta por parte de galerías de arte, como en los ejemplos descritos anteriormente, son un indicativo, y por tanto un modo de evaluación, de la calidad de las obras.

V-B. Evaluación definitiva: Competición Internacional de Arte

No obstante, y tal como indicábamos en secciones previas, la mejor manera de evaluar un trabajo artístico, independientemente del procedimiento con que se ha generado, es presentarlo en alguna competición puramente artística, a nivel internacional. Esto es lo que se hizo con la tercera obra, The Horizon Project [5] (ver figura 2). Se decidió presentarlo a la competición Show Your World 2017 [1], celebrada en

[1] http://www.reartiste.com/juried-exhibition-show-your-world-2017/

New York, y cuyas obras finalistas serían expuestas en la galería *Gallery MC*, en Manhattan. La obra fue seleccionada por un jurado internacional como obra finalista, siendo la única obra Española finalista en esta competición, y por supuesto la única generada mediante procesos evolutivos.

Hasta dónde sabemos, esta es la primera vez que una obra de estas características es evaluada -y seleccionada como finalista- en una competición internacional abierta en el ámbito del mundo del arte, con un jurado especialista, y expuesto en una exposición dirigida por un curador profesional.

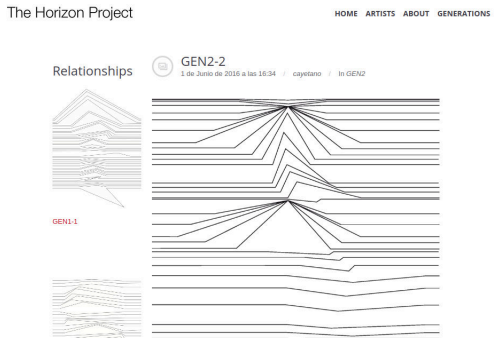


Figura 2. The Horizon Project, available at: http://merida.herokuapp.com/

En resumen, aunque en los problemas de optimización la calidad del resultado es fácil de evaluar considerando si el resultado obtenido da solución al problema planteado, creemos que el camino mostrado para evaluar el arte evolutivo en este trabajo, que incluye al público y la crítica, representada por galerías de arte y competiciones internacionales, es el único que puede otorgar un sello de calidad a un resultado artístico.

VI. Conclusiones

Este artículo presenta nuestra posición sobre lo que consideramos más adecuado para la correcta evaluación de un proyecto de arte evolutivo.

Nuestra propuesta, que renuncia al test de turing por su incapacidad para otorgar un sello de calidad a un trabajo, se basa en involucrar a actores fundamentales del mundo del arte: la crítica y al público.

El artículo muestra como ejemplo el recorrido de un proyecto artístico colectivo que comenzó en 2012, y su proyección en los diferentes foros artísticos, ha permitido dotar de sentido diferenciado a la obra, más allá de la distición por ser arte evolutivo.

Agradecimientos

Agradecemos el apoyo del Ministerio de Economía y Competitividad proyecto TIN2017-85727-C4-{2,4}-P, Junta de Extremadura, Consejería de Comercio y Economía, proyecto IB16035 a través del Fondo Europeo de Desarrollo Regional, “Una manera de hacer Europa”.

Referencias

[1] Boden, M.A.: The turing test and artistic creativity. Kybernetes **39**(3), 409–413 (2010)
[2] Botstein, L.: Schoenberg and the audience: modernism, music, and politics in the twentieth century. Schoenberg and his World pp. 19–54 (1999)
[3] Fernández, F., García, M., Merelo, J., Aguilar, G., Cruz, C., Hernández, P.: Analyzing evolutionary art audience interaction by means of a kinect based non-intrusive method. In: Studies in Computational Intelligence series, vol. 785, pp. 321–347 (2018)
[4] Frade, M., Fernández de Vega, F., Cotta, C.: Evolution of artificial terrains for video games based on accessibility. In: European Conference on the Applications of Evolutionary Computation, pp. 90–99. Springer (2010)
[5] Hernández, P., Fernández de Vega, F., Cruz, C., Albarrán, V., García, M., Navarro, L., Gallego, T., García, I.: The horizon project: Emotion in lines. Journal Art and Science **1**(1), 1–9 (2017)
[6] House, J.: Impressionism: Paint and Politics. Yale University Press (2004)
[7] Loiacono, D.: Gecco-2013 competitions. ACM SIGEVOlution **6**(2), 27–28 (2014)
[8] McCormack, J.: Open problems in evolutionary music and art. In: Workshops on Applications of Evolutionary Computation, pp. 428–436. Springer (2005)
[9] Peress, M.: Dvorak to Duke Ellington: A Conductor Explores America’s Music and Its African American Roots. Oxford University Press on Demand (2004)
[10] Reichardt, J.: Cybernetic serendipity: the computer and the arts. Praeger (1969)
[11] Romero, J.J.: The art of artificial evolution: A handbook on evolutionary art and music. Springer Science & Business Media (2008)
[12] Secretan, J., Beato, N., D Ambrosio, D.B., Rodriguez, A., Campbell, A., Stanley, K.O.: Picbreeder: evolving pictures collaboratively online. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1759–1768. ACM (2008)
[13] Takagi, H.: Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. Proceedings of the IEEE **89**(9), 1275–1296 (2001)
[14] Thompson, D.: The $12 million stuffed shark: The curious economics of contemporary art. Macmillan (2010)
[15] Fernández de Vega, F.: Revisiting the 4-part harmonization problem with gas: A critical review and proposals for improving. In: Evolutionary Computation (CEC), 2017 IEEE Congress on, pp. 1271–1278. IEEE (2017)
[16] Fernández de Vega, F., Cruz, C., Navarro, L., Hernández, P., Gallego, T., Espada, L.: Unplugging evolutionary algorithms: an experiment on human-algorithmic creativity. Genetic Programming and Evolvable Machines **15**(4), 379–402 (2014)
[17] Fernández de Vega, F., Navarro, L., Cruz, C., Chavez, F., Espada, L., Hernandez, P., Gallego, T.: Unplugging evolutionary algorithms: on the sources of novelty and creativity. In: Evolutionary Computation (CEC), 2013 IEEE Congress on, pp. 2856–2863. IEEE (2013)

An Evolutionary Approach to Metroidvania Videogame Design

Álvaro Gutiérrez Rodríguez, Carlos Cotta, Antonio J. Fernández Leiva
ETSI Informática, Campus de Teatinos, Universidad de Malaga, 29071 Málaga – Spain
Email:{alvarogr,ccottap,afdez}@lcc.uma.es

***Abstract*—Game design is a fundamental and critical part of the videogame development process, demanding a high cost in terms of time and effort from the team of designers. The availability of tools for assisting in this task is therefore of the foremost interest. These can not just speed up the process and reduce costs, but also improve the overall quality of the results by providing useful suggestions and hints. A conceptual system to approach the construction of this kind of tools is presented in this work. By using a learning component, the preferences and expertise of the designers can be modelled and to some extent simulated. This model is subsequently exploited by an optimization component that tries to create adequate game designs. A proof of concept of the system is provided in the context of level design in Metroidvania games. It is shown that the system can produce quality solutions and hints to the designer.**

I. Introduction

The development of a videogame encompasses different stages/phases and usually involves teams specialized on particular areas. Among all these phases, the design phase is crucial for the ultimate fate of the game: it is in this stage where it is decided how the game will be, what the requirements for subsequent development phases will be, and most importantly, which the source of fun in the game will be.

Unlike other development stages, design is not so commonly assisted by AI tools. This contrasts with the pervasive use of such tools in content generation. For example, Non-Player Characters (NPCs) are agents controlled by the computer whose behaviour must be believable (i.e., in accordance with the rol of that character) to keep the player's inmersion in the game. Developers also use procedural generation techniques to generate new content (e.g., maps[1], weapons[2], stories[3], etc.) during gameplay in order to diminish monotony. The use of content generation tools reduces the workload of designers and artists, and produce results that are generally well-received by the end users.

As anticipated before, the field of intelligent tools for game design is still nascent, but it does not mean that it has not yet been explored [1]–[6]. Indeed, previous works provide interesting and explanatory results, paving the way for further developments. While some of them focus on specific videogame genres and in stablished mechanics, a broader perspective is possible. In this sense, this work is directed to propose a tool aimed to create a complete game (leaving art and sound aside) for any genre, creating the mechanics, game rules, game elements, NPC behaviors and levels. For this, we propose the use of bioinspired algorithms for learning and optimization. More precisely, we pose the use of evolutionary algorithms to generate game contents (game rules, mechanics, etc.) and machine learning tools such as artificial neural networks to mimic the way the designer thinks. After briefly outlining others related works in Section II, this system is described for a general point of view in Section III. As an initial proof of concept, we have picked the case of Metroidvania games [7]. The deployment of the system on this context is then detailed in Section IV, and the results of an empirical examination are provided in Section V. We close this work with a summary of findings and an outline of future work in Section VI.

[1]http://spelunkyworld.com/index.html
[2]https://borderlandsthegame.com/
[3]https://www.shadowofwar.com/es/

II. Background

AI-assisted game design refers to the development of AI-powered tools supporting the game design and development process [6]. Combining these tools with Procedural Content Generation techniques (PCG) is a good approach to aid the game designer.

Liapis *et al.* [4], [8] generate new designs using genetic algorithms (GAs) as a PCG technique. Designers have a map design on which they are working, and the application derives new designs using this former as a seed. The GA uses different criteria to evolve solutions, such as game pace and player balance (see also [9]). The playability of the design is defined by considering if all resources and bases are reachable from any other base or resource. The application shows twelve derivations, six made with the evolution system commented and other six with novelty search. An important part of the experiments involved having professional designers use the tool. The feedback was positive, indicating the system was capable of providing interesting suggestions.

ANGELINA is a cooperative coevolutionary system for automating the process of videogame design. There have been several different versions of ANGELINA in the past [10]–[12]. Focusing on the last version [5], it can create every content of a game: mechanics, game rules, programming, sound, art and levels. The creation of a game starts with a semantic derivation of a sentence or word; after this derivation, free contents (e.g., models, textures, sounds, etc) in agreement with the derivation are searched in the web. Later, mechanics, game

rules and, finally, levels in agreement with those mechanics and game rules are created. During the process, a game is represented as a map which defines passable and non-passable areas in a two-dimensional grid, a layout which describes the arrangement of player and non-player characters in the map, and a ruleset which describes the effects of collisions between game entities, as well as movement types for the NPCs, and time and score limits for the game [12]. A collection of EAs run concurrently, each of them aimed to optimize a different component; in order to have a complete vision and optimize their individual objectives, they share information on the game (board state, rules, mechanics) during the evolutionary process and use the fitness of individual components to increase the overall fitness of the finished artifact[4].

III. An Automated System for Game Design

The design process is difficult due to the different abstract facets it encompasses. Consider for example the design process of a character: the designer receives the story of the character and some features of their personality (e.g., heroic, brave, etc.); then, the designer creates several different designs, all of which can be functional, well drawn, and using an appropriate color palette. But which of them is the best design? The designer could arrange all different designs in a screenshot of the movie or videogame to see which one fits better but how is this decision taken? Designers need to use their experience and creativity to choose the fittest design.

If we now think about level design in a Metroidvania videogame, the main issue is the same: designers knows what kind of experience they want to create with certain mechanics, game rules and level elements specified but, what is the best combination and order of all the elements? Some designs could be created, tested and then the most convincing design (according to the designer's creative mind and subjective opinion) could be refined.

The examples above tackle different elements of a game but they are solved in the same way: through the designer's expertise. In both cases the different designs are evaluated by the mind of the designer, so we need to recreate this cognitive process. To this end, we propose a framework that orchestrates the use of bioinspired algorithms for learning and optimization – see Fig. 1. A typical configuration would involve the use of artificial neural networks (ANNs) for learning and evolutionary algorithms (EAs) for optimization.

Within this framework, EAs will be used to recreate the different ideas that a designer can have during the design process. This is done by trying to optimize an objective function that mimics the designer's goals and preferences. This function is provided by the learning component of the framework. This component is initially seeded with a collection of learning cases, namely examples that can be either positive (goals, appealing features of solutions, etc.) or negative (undesirable features of solutions, traps to be avoided, etc.). Using these, a first model of the designer is built and used by the EA to generate potentially admissible solutions.

[4]Two games produced by this system were presented in the Ludumdare 28 GameJam – http://ldjam.com/ .

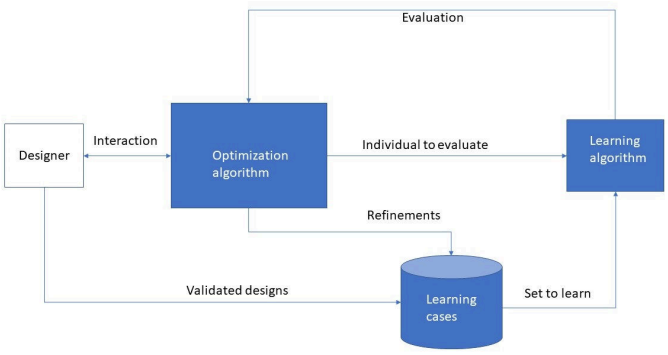


Figure 1: System scheme

During the process, the designer can interact with the system by providing additional examples or clues to refine the model and hence bias the search process in specific directions.

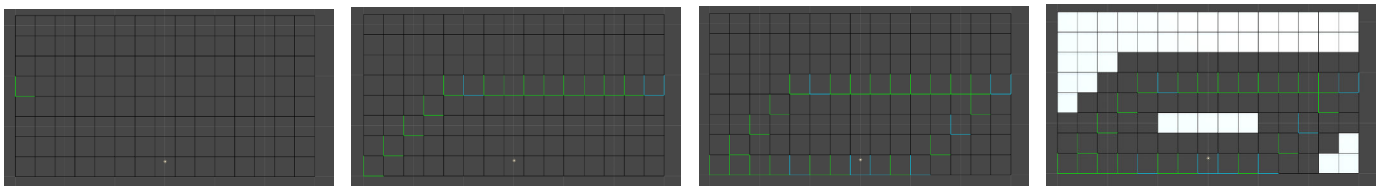
In line with the work by Sorenson and Pasquier [13], in which the fitness function calculate the feasibility and fun of the design, we aim to customize this function to each designer. To this end, the learning component needs to learn to think like the designer and will validate the different individuals (ideas) of the EA like the designer would. A system such as this one can be used to create a complete game of any genre. As an initial step, in this work we have focused on Metroidvania games, whose complexity is more amenable for a study under controlled conditions. This is detailed in next section.

IV. A Case Study: Metroidvania Game Design

When a game designer starts the design of a new game, one of the starting points is selecting the genre of the game. Each genre has several predefined mechanics that define it. When the designer select the genre, new mechanics are created and mixed. In our case, we focus on the Metroidvania genre [14] due to its diversity of mechanics. This genre is famous for mixing the Metroid and Castlevania games series. Games in this genre feature a large interconnected map through which the player can move, having to obtain objects, weapons or abilities to unlock the different locked areas. The map is composed of different areas, each of which is in turn composed of different rooms (including secret rooms). Rooms are where the different enemies, objects, new abilities, are placed.

Typically, a Metroidvania game is a side-scrolling platform game whose creation can be accomplished by adequately defining the following elements [7]:

- **Mechanics:** all actions that the player or character can do, changing the state of the game as a consequence.
- **Game rules:** these indicate the results of an action, i.e., how an action has modified the state of the game. Rules are composed by a set of previous states of game elements, a set of actions and a set of new states (not necessarily the states of the elements involved in the action). For example: stateOf(boxA) + action(moveBoxAToPositionX) = stateOf(doorA, open).
- **Level elements:** all objects included in a certain level (phase, stage, or screen) of the game. We distinguish



(a) (b) (c) (d)

Figure 2: (a) Space partition (b) Main path of the level (c) Joining the main path and the secondary one (d) Complete design of the level.

between two types: interactive and non-interactive. Interactive elements are those whose state can be changed by an action. Conversely, non-interactive elements are those whose state is not affected by player's actions.

- **Objects:** every game object that the player collect, changing the player's state as a result. Such a modification will be defined by the game rules, and can be a new ability or updated stats, just to give two examples.
- **Characters:** these are agents controlled by the computer whose goals are defined by the game rules and whose behaviors are defined by game mechanics.
- **Levels:** This is the scenario in which players use their mechanics and the different enemies, objects and level elements created for the game are placed. The game can have one or several levels but the goal in all of them is defined by the game rules.

We are specifically going to focus on the generation of levels. To do so, for each level to be created we pick a width and height, and generate an empty space of these dimensions, discretized into cells whose size doubles the player's size. Each cell is a possible point of the path to complete the level. Once we have this grid of cells, the designer (or the automated system) creates the path and the optimization algorithm finds a combination of mechanics to use in the so-defined path. The cells that are not part of the path are merely "atrezzo" of the level (e.g., solid walls, pipings, skeletons, etc.). See Fig. 2 for an example of a level created by the system.

The path was calculated by choosing two random points on the left and right sides of the grid and connecting them. Subsequently, a secondary path is created using the same procedure, picking two randoms points in the grid at Manhattan distance $\geqslant 2$ of the main path.

A. Learning process

As stated in the previous section, the learning component is responsible of modelling and simulating the cognitive process of the designer regarding how good a particular level design is. In this work, we have opted for using a neural network for this purpose, although arguably other machine learning methods could be used as well. The role of this component is to serve as a judge capable of assessing the creations of the optimization component in the same way that the designer would do. To this end, we need a reference set composed of good and bad solutions, as dictated by the preferences/knowledge of the designer. We shall use in the following the notation Ref_{good} and Ref_{bad} to refer to the collection of good and bad solutions

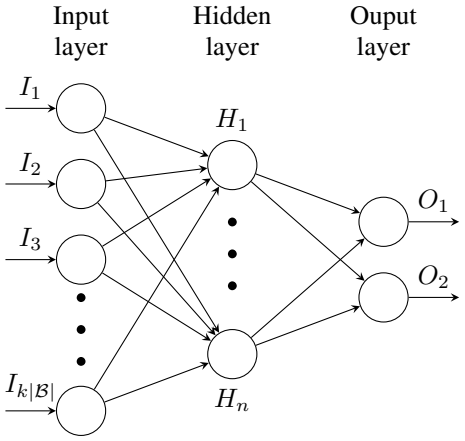


Figure 3: ANN structure used to evaluate motifs

respectively. This is the initial input to the system, and will be used the get the ball rolling in the automated design process.

In order to model the designer's preferences, the next step is extracting the necessary information from the reference sets. For this purpose, the solutions (i.e., game designs) contained in these are scanned in order to identify particular motifs, which are subsequently classified as good or bad, depending on their presence in good/bad solutions. Notice that this can lead to contradictory inputs to the classifier, since a given motif can appear in both good or bad solutions. The particular learning algorithm used must therefore be able to discern whether the motif is really a significant indicator of goodness/badness or it is simply an irrelevant piece of information.

In the particular context we are considering in this work, solutions are the description of the main path of the level (the same process can of course be applied to the secondary path). We assume we have a collection $\mathcal{B}$ of possible building blocks for this path, each of them representing the contents of a certain grid cell (i.e., a corridor, a staircase, a moving platform, etc.). The whole path is therefore a sequence of blocks. Let m be the length of this sequence. In order to extract motifs, we have opted for a very simple strategy which amounts to identifying contiguous subsequences of a certain length k. Thus, if we have a solution $[b_1, b_2, \ldots, b_m]$, we can extract $m - k + 1$ motifs from it: $\langle b_1, \ldots, b_k \rangle$, $\langle b_2, \ldots, b_{k+1} \rangle$, up to $\langle b_{m-k+1}, \ldots, b_m \rangle$. This is repeated for each solution in either reference set, and each motif is assigned a label *good/bad* depending on its provenance.

Once the set of motifs is available, it is fed to an artificial neural network such as the one depicted in Fig. 3. The input level has $k|\mathcal{B}|$ neurons, meaning that each motif is encoded as the concatenation of k bitstrings, each of them representing a block in the motif. To be precise, each of these bitstrings has as many bits as block types are, and only one bit is set to 1 (the bit corresponding to the block in the corresponding position of the motif). As to the output level, it has two neurons, corresponding to each of two classes considered (*good*/*bad*). After training the ANN, any given motif can be assessed numerically with a value ranging from -1 to 1 by subtracting the output (which is between 0 and 1) of these two output neurons. We use this procedure in order to handle contradictory information, which can be present as stated before.

B. Optimization process

The optimization process has the role of generating tentative solutions. As stated before, each solutions indicates the different mechanics to be used in the main path of the level. We represent them with an integer array of length m. Each position in the array represents a cell in the main path, and each integer value $0, \ldots, |\mathcal{B}| - 1$ indicates a certain type of block. In order to evaluate a given solution, it is scanned and the motifs it contains are extracted, much like it was described in the previous subsection. Given that we are considering substrings of length k as motifs, it is convenient to define a function

$$\mathcal{M}^{(k)} : \mathcal{B}^m \rightarrow \mathbb{N}^{|\mathcal{B}|^k} \tag{1}$$

that computes how many times each motif appears in a certain solution (note that each solution contains $m - k + 1$ motifs, some of them possibly repeated, among $|\mathcal{B}|^k$ different potential motifs).

Now, we resort to the assessment of the ANN in order to evaluate a particular collection of motifs. More precisely, let S be a solution (a length-m sequence of blocks). Then, $\mathcal{M}^{(k)}(S)$ would be an array with the frequency of each motif in S. We will use the notation $\mathcal{M}^{(k)}(S)_{i_1 \ldots i_k}$ to refer to the element in this frequency array corresponding to motif $\langle i_1 \ldots i_k \rangle$, i.e., the count number of this specific motif in this solution S. The quality value $f(S)$ of this solution would be then:

$$f(S) = \sum_{i_1 \ldots i_k \in \{1,\ldots,|\mathcal{B}|\}} T^{\text{ANN}}_{i_1 \ldots i_k} \mathcal{M}^{(k)}(S)_{i_1 \ldots i_k} \tag{2}$$

where $T^{\text{ANN}}_{i_1 \ldots i_k}$ is the assessment provided by the ANN for that specific motif $\langle i_1 \ldots i_k \rangle$. Thus, the objective function would compute the sum of the value attributed by the ANN to each motif (which ranges from 1 for very desirable motifs down to -1 for highly undesirable solutions, with all the range of intermediate values for motifs of more or less uncertain desirability) present in the solution, weighted by the corresponding frequency of the motif.

The search engine used is an elitist genetic algorithm (GA) with binary tournament selection, uniform crossover and random mutation. The objective of the search would be providing designers different suggestions that mimic their preferences, which could be in turn refined online by tagging particular solutions as good or bad and retraining the ANN.

V. Experiments and Results

The system described in the previous sections has been put to test in order to obtain a proof of concept of its functioning. The results obtained will be described later on in this section. Previously, let us detail the configuration of the experiments.

A. Experimental setting

The game design task considered in this work consists of constructing the level of a Metroidvania game using a collection $\mathcal{B}$ of 7 different blocks. The length of the main path is in this case $m = 50$, thus resulting in a search space whose size $|\mathcal{B}|^m \geqslant 10^{42}$ makes a brute force exploration be out of question. The motifs considered are substrings of length $k = 3$. Hence, the ANN utilized in the learning phase has an input layer of $k|\mathcal{B}| = 21$ neurons. We have chosen $n = 63$ neurons in the intermediate layer (that is, thrice the size of the input layer), and have 2 output neurons as described in Section IV-A. All neurons have a sigmoid activation function, and the ANN is trained using backpropagation (learning rate $\delta = 0.2$, momentum $\alpha = 0.33$; run until the error is less than 0.1 or a maximum of 100,000 learning epochs is reached). As to the GA utilized in the optimization component has a population size of $\mu = 50$ individuals, crossover probability $p_c = 0.9$, mutation probability $p_m = 1/m = 0.05$, and is run for *maxevals*$= 5000$ evaluations (we created different sets of parameters, tested them with the same training set and those mentioned before obtained the best results).

We have considered three test cases to evaluate the behavior of the system. Each of these test cases is constructed by creating an initial payoff table for each of the motifs. Once adequate reference sets Ref_{bad} and Ref_{good} are defined (by providing solutions that aim to maximize or minimize the resulting sum of payoffs according to the initial table), the whole system tries to (i) discover the usefulness of each motif (which is quantitatively unknown to the system, and can only be inferred via the appearance of these motifs in good or bad solutions) and (ii) construct good solutions by combining appropriately these motifs (not straightforward in general, given the fact that all motifs in a solution are in partial overlap with each other).

The first test case is given by a payoff matrix

$$T_{xyz} = \begin{cases} 1 & (x = y) \vee (x = z) \vee (y = z) \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

i.e., motifs with 3 different elements are undesirable, whereas motifs with 2 or 3 identical elements are good. This is an easy test case that can nevertheless provide interesting information in terms of the bias of the optimizer towards particular directions. The second test case is actually a variant of the previous one, and is given by the following payoff table:

$$T_{xyz} = \begin{cases} 0 & x = y = z \\ 1 & x \neq y \neq z \\ 3 & \text{otherwise} \end{cases} \tag{4}$$

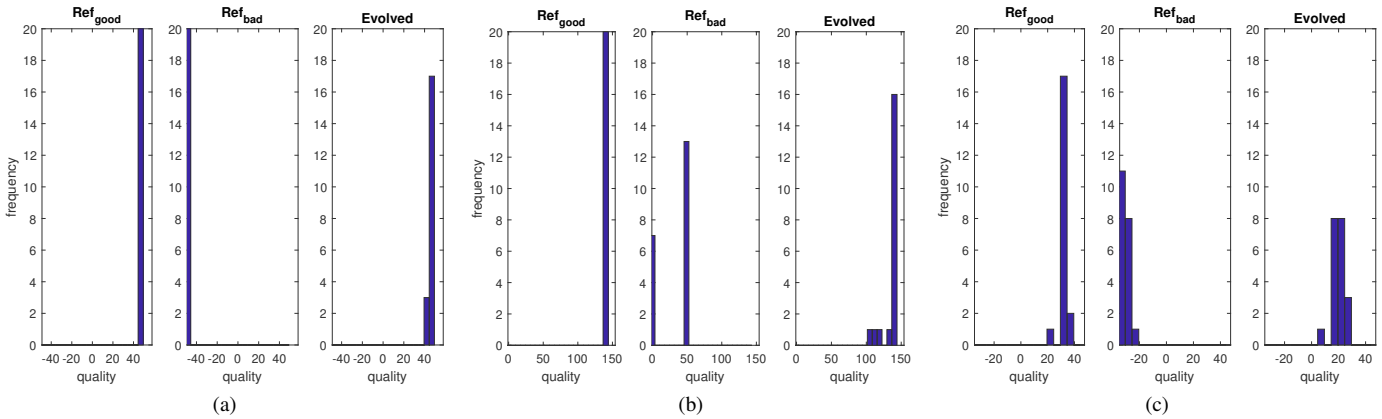


Figure 4: Histogram of objective values according to the underlying payoff table for reference solutions (Ref_{good} and Ref_{bad}) and solutions evolved by the system. (a) Test case #1 (b) Test case #2 (c) Test case #3

In this case, motifs with 3 identical elements are highly undesirable. motifs with 3 different elements are better, yet suboptimal; the best motifs are those with exactly 2 identical elements. Finally, the third test case is defined via random payoffs: T_{xyz} is uniformly drawn at random from the interval $[-1, 1]$ for each particular motif $\langle x, y, z \rangle$.

In the first two test cases, perfect reference sets are constructed by picking at random solutions only comprising good or bad motifs (in the second test case, suboptimal motifs are just included in bad solutions). In the third case, we have resorted to an GA analogous to that used in the optimization component aiming to maximize/minimize the objective function defined by the corresponding payoff table. In all cases, both Ref_{bad} and Ref_{good} comprise 20 solutions each.

For each test case, the ANN has been trained using the motifs extracted from the reference sets and the GA has been run 20 times, keeping the best solution found in each run. All algorithms have been implemented on the videogame engine Unity using C# as programming language. The AForge library[5] has been used to support the ANN.

B. Results

A first glimpse of the experimental results obtained is provided in Fig. 4. Therein, the objective values of both reference solutions and evolved solutions is shown for each test case. As indicated in previous subsection, the reference solutions for the first two test cases are perfect, in the sense they only contain desirable or undesirable motifs. Thus, the mass of objective values is concentrated on the left end of the histogram for bad solutions and on the right end for good solutions. The reference solutions for the third test case were empirically obtained and hence there is more variety of objective values (note that this may to some extent constitute a more realistic setting, in which solutions provided by the designer are not ideal flawless prototypes but can rather have an inherently noisy structure). In either case, notice how the solutions provided by the EA also tend to cluster towards the right end of the histogram, indicating that they are objectively good according to the hidden criterion used for defining goodness/badness.

[5]http://www.aforgenet.com/

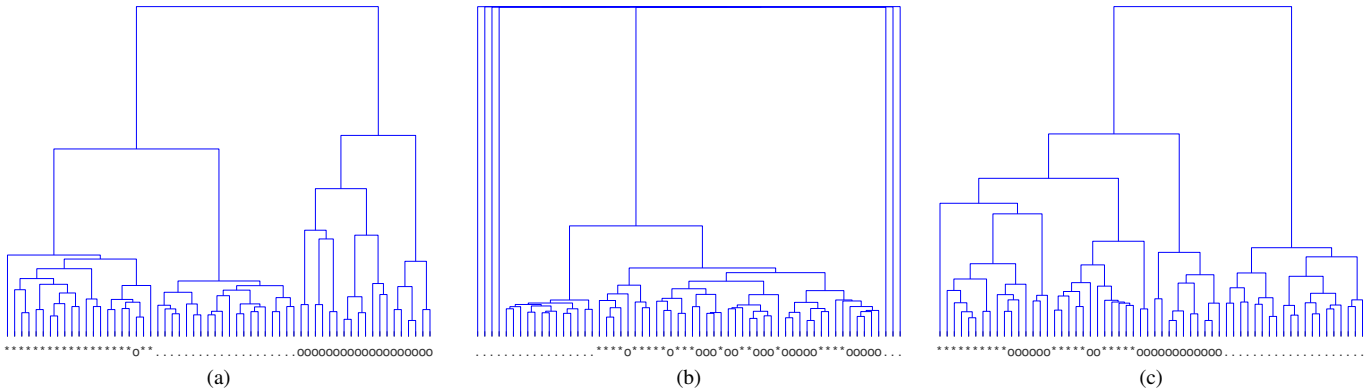
Table I: Structural difference between solutions in the test case #1. The number of motifs of each type (based on the number b of identical blocks in it) is shown for each data set

	b=2	**b=3**	**b=0**
Ref_{good}	897	63	0
Ref_{bad}	0	0	960
Evolved	533	414	13

Some further insight is obtained if we take a look at the structure of evolved solutions and try to compare these with solutions in either reference set. To this end, we compute for each solution S the associated motif-frequency array $\mathcal{M}^{(k)}(S)$. Subsequently, we can compute the structural distance $D(S, S')$ between two solutions S and S' as the Euclidean distance between the corresponding motif-frequency arrays, i.e.,

$$D(S, S') = \sqrt{\sum_{i_1 \ldots i_k} \left(\mathcal{M}^{(k)}(S)_{i_1 \ldots i_k} - \mathcal{M}^{(k)}(S')_{i_1 \ldots i_k}\right)^2} \quad (5)$$

Once this is done for each pair of solutions, the resulting distance matrix can be used to perform a hierarchical cluster analysis. We have done this on the solutions available for each test case, using Ward's minimum variance method to guide the agglomerative clustering process [15]. As can be seen in Fig. 5, in both test cases #2 and #3, evolved solutions have some affinity among them but tend nevertheless to mix with clusters of solutions in Ref_{good}. Together with the good objective quality of these solutions discussed before, this indicates that the system did faithfully capture the features of interest in solutions and could combine them appropriately. Test case #1 is also interesting: evolved solutions are objectively good, but they tend to cluster together, mostly apart from solutions in either reference set. If we perform a quick structural analysis, we obtain the results shown in Table I. Note that a few bad motifs (composed of three different blocks; $b = 0$) have slipped into these solutions, although this is not very significant. It is far more interesting to note that evolved solutions tend to have much more motifs composed of identical blocks. This may be an artifact of the learning process, that



(a) (b) (c)

Figure 5: Clustering of solutions in the reference sets and evolved by the system. A dot ('.') represents solutions in Ref_{bad}, a star ('*') solutions in Ref_{good}, and a circle ('o') evolved solutions. (a) Test case #1 (b) Test case #2 (c) Test case #3

might marginally favor some particular motif, or a byproduct of the search dynamics of the EA, whereby it may be easier for it to create good solutions by exploiting this kind of motifs. Of course, at this point the loop could be closed by having the designer inspect these solutions, possibly providing complementary preferences in order to support or discard this kind of solutions.

VI. Conclusions and Future Work

We have presented the concept of an AI-assisted videogame design system, aimed to help designers create games by suggesting new ideas based on a model of their preferences and knowledge. To this end, we propose the orchestrated use of machine learning techniques and optimization methods, the former to capture the designers' expertise, and the latter to exploit this expertise in a systematic (and hopefully creative) way. The experimental proof of concept has shown that not only quality solutions analogous to those provided as reference can be generated, but also different non-anticipated biases can appear, providing interesting hints to the designer.

As future work, we plan to deploy the system on a more realistic environment to test its capabilities. Needless to say, closing the loop and having the designer introduce dynamic preferences is another improvement of the foremost interest.

Acknowledgments

This work is supported by Spanish Ministerio de Economía, Industria y Competitividad under projects EphemeCH (TIN2014-56494-C4-1-P) and DeepBIO (TIN2017-85727-C4-1-P), and by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

References

[1] R. Hunicke, M. Leblanc, and R. Zubek, "MDA : A formal approach to game design and game research," in *AAAI Workshop on Challenges in Game AI*, vol. 4. San Jose: Press, 2004, pp. 1–5.

[2] N. Sorenson and P. Pasquier, "The evolution of fun: Automatic level design through challenge modeling," in *International Conference on Computational Creativity*, D. Ventura, A. Pease, R. P. y Pérez, G. Ritchie, and T. Veale, Eds. Lisbon, Portugal: Department of Informatics Engineering, University of Coimbra, 2010, pp. 258–267.

[3] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for personalized game content creation tools," in *The 9th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Boston, USA: AI Access Foundation, 2013, pp. 11–16.

[4] ——, "Sentient sketchbook: Computer-aided game level authoring," in *ACM Conference on Foundations of Digital Games*, G. N. Yannakakis, E. Aarseth, K. Jørgensen, and J. C. Lester, Eds. Chania, Crete: Society for the Advancement of the Science of Digital Games, 2013, pp. 213–220.

[5] M. Cook and S. Colton, "Ludus ex machina: Building a 3d game designer that competes alongside humans," in *International Conference on Computational Creativity*, S. Colton, D. Ventura, N. Lavrač, and M. Cook, Eds. Ljubljana, Slovenia: computationalcreativity.net, 2014, pp. 54–62.

[6] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 4, pp. 317–335, 2015.

[7] K. Pearson, "Guide to making metroidvania styled games – part 1," PlatformerPower, 2017, https://platformerpower.com/post/160130225259/guide-to-making-metroidvania-styled-games-part-1 [accessed 28-Jun-18].

[8] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for sentient sketchbook," in *2014 IEEE Conference on Computational Intelligence and Games*. Dortmund, Germany: IEEE, 2014, pp. 1–8.

[9] R. Lara-Cabrera, C. Cotta, and A. J. Fernández-Leiva, "On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms," *Natural Computing*, vol. 13, no. 2, pp. 157–168, Jun 2014.

[10] M. Cook and S. Colton, "Multi-faceted evolution of simple arcade games," in *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, S. Cho, S. M. Lucas, and P. Hingston, Eds. Seoul, South Korea: IEEE, 2011, pp. 289–296.

[11] M. Cook, S. Colton, and A. Pease, "Aesthetic considerations for automated platformer design," in *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, M. Riedl and G. Sukthankar, Eds. California, USA: The AAAI Press, 2012, pp. 124–129.

[12] M. Cook, S. Colton, and J. Gow, "Initial results from co-operative co-evolution for automated platformer design," in *Applications of Evolutionary Computation – EvoApplications 2012*, ser. Lecture Notes in Computer Science, C. Di Chio *et al.*, Eds., vol. 7248. Berlin, Heidelberg: Springer Verlag, 2012, pp. 194–203.

[13] N. Sorenson and P. Pasquier, "Towards a generic framework for automated video game level creation," in *Applications of Evolutionary Computation - EvoApplications 2010*, ser. Lecture Notes in Computer Science, C. Di Chio *et al.*, Eds., vol. 6024. Berlin Heidelberg: Springer Verlag, 2010, pp. 131–140.

[14] C. Nutt, "The undying allure of the metroidvania," Gamasutra, 2015, https://www.gamasutra.com/view/news/236410/The_undying_allure_of_the_Metroidvania.php [accessed 29-Jun-18].

[15] J. H. Ward Jr., "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.