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 $P(k) \sim k^{-\gamma}$.

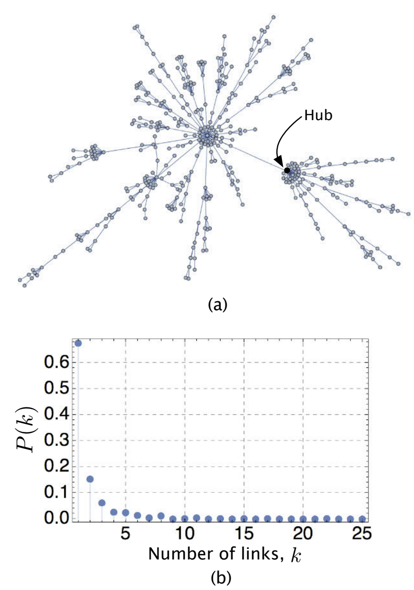


Fig. 1. (a) Example of a scale-free complex network with 400 nodes. (b) Node degree probability density function of a network similar to that represented in (a).

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.

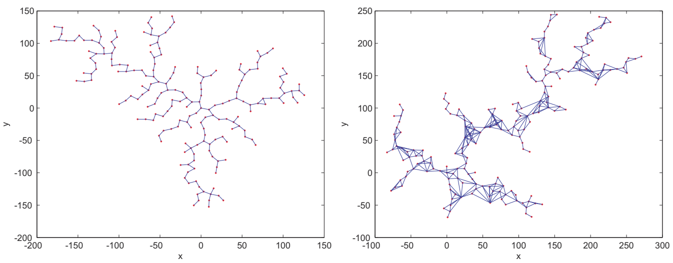
II. Growing geographically-embedded complex networks over random-distributed nodes

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:

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

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.

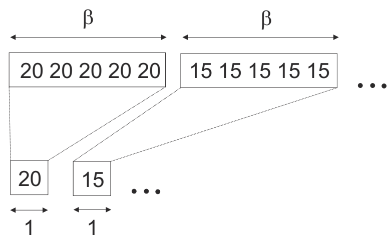
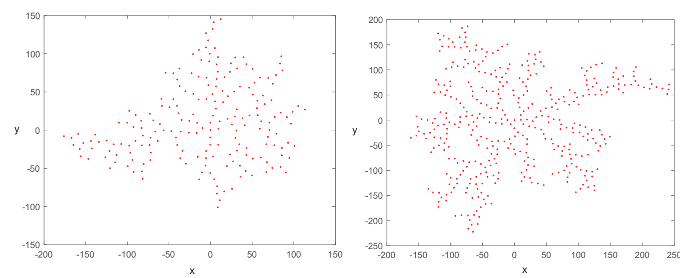


Fig. 3. Compressed encoding example ($\beta = 5$), useful in the evolution of complex networks.

- **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 $x_i' = x_i^1 + F(x_i^2 - x_i^3)$ 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 $10 \leq R_l^i \leq 100$. 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.

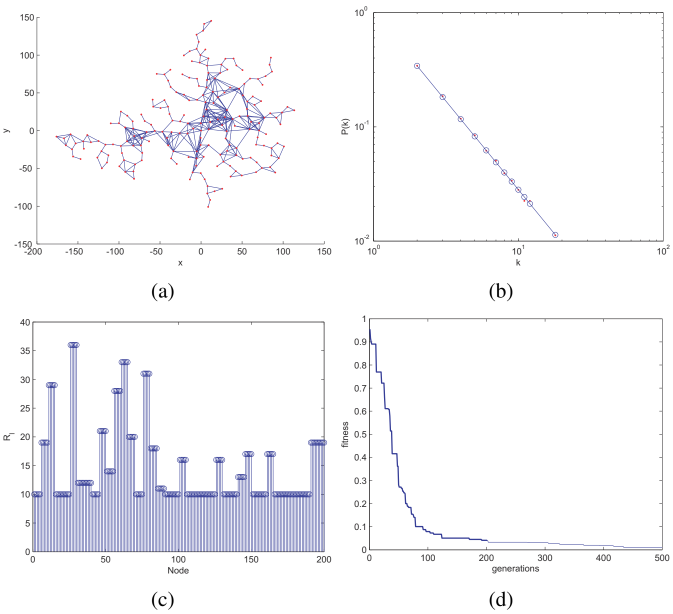


Fig. 5. Example of geographically-embedded complex network with $N = 200$ nodes, evolved with the CRO-SL algorithm; (a) Resulting spatial network obtained; (b) Node degree distribution for the network represented in (a): blue circles stand for the power law distribution $k^{-1.55}$, and red points for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

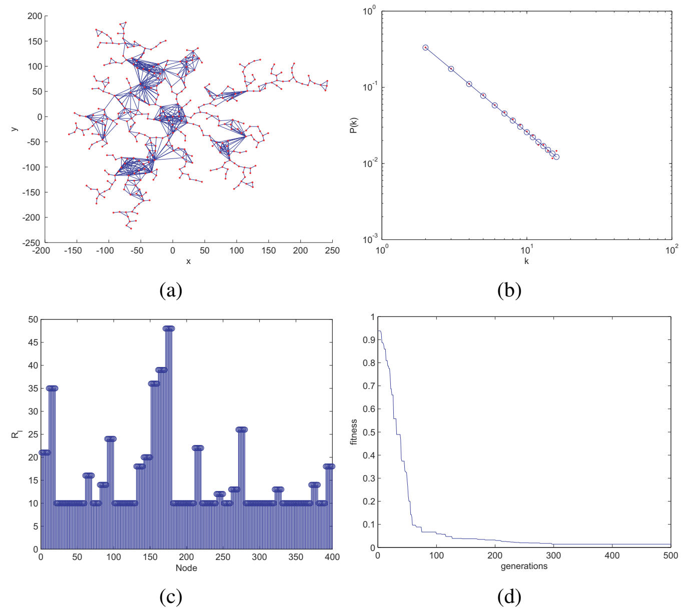


Fig. 6. Example of geographically-embedded complex network with $N = 400$ nodes, evolved with the CRO-SL algorithm; (a) Resulting geographically-embedded network ; (b) Node degree distribution for the network represented in (a): blue circles stand for the power law distribution $k^{-1.59}$, and red points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

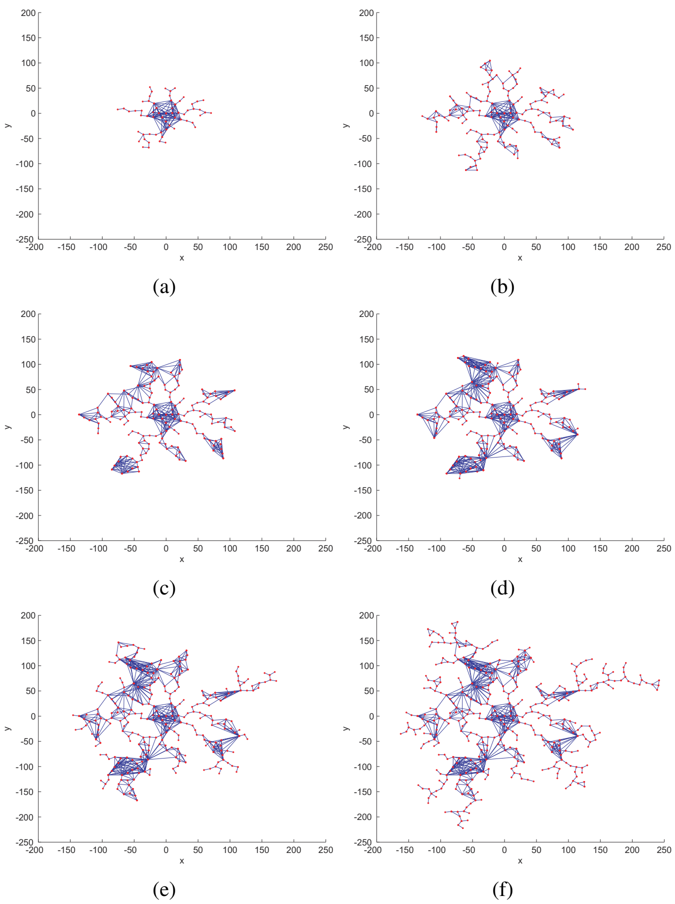


Fig. 7. Network evolution process in 6 steps for the $N = 400$ case (best solution obtained with the CRO-SL algorithm).

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.