Analysis of Complex Networks in Optimization Algorithms

Joshua Triana, Alexander Redondo, and Victor Bucheli,

***Abstract*—The aim of this paper is to present the relationship between complex networks and optimization algorithms. We investigate the genetic algorithms focus on the relationship of the individuals in each iteration and how the network emerge. The results are promising related to the complex networks properties and show that the Differential Evolution match with the small world networks. We model how the complex network connected component property seems to have relationship with fitness quality in the Imperialist competitive algorithm. Furthermore, we discuss how to improve optimization algorithm based on small networks.**

***Keywords*—Complex Networks, Differential Evolution, Evolutionary Programming, Imperialism Competitive Algorithm**

I. Introduction

IT has been discovered how some of these algorithms can be visualized as complex networks [1]. Also, the fact that not all optimization algorithms modeled as networks have complex network properties has been discussed [2] . In this research, the aim is to study if the algorithms that do present these properties have a better performance than the algorithms where this does not happen.

It is proposed to study the modeling of some optimization algorithms as complex networks in order to study their topological properties. With this network analysis it is expected to better understand the dynamics of the algorithms in their different iterations. In previous works [3], [4] Evolutionary algorithms have been modeled as complex networks and some network analysis have been carried out. In this work, the objective is to understand how the values of these network properties could serve to improve the algorithms.

Section 2 explains what optimization algorithms are and which are their different categories. In Section 3 the differential evolutionary algorithms and the two variants that are used in this work are exposed. Section 4 discusses the imperialism competitive algorithms and their most relevant works. Section 5 introduces the concept of complex networks and the topological properties that are studied in this research. Section 6 discusses the relationship between complex networks and evolutionary algorithms. Section 7 presents the research methodology used in the tests. Section 8 shows the tests and the analysis of networks discussed in the methodology. Finally, Section 9 discusses the results found and the open questions that remain to be answered.

The authors would like to thanks Dr Ivan Zelinka for his contribution and the Masters degree program of Universidad del Valle for the support and encouragement for this project

Joshua Triana is with the Computer Science Department, Universidad del Valle, Colombia (joshua.triana@correounivalle.edu.co)

Alexander Redondo is with the Computer Science Department, Universidad del Valle, Colombia (alexander.redondo@correounivalle.edu.co)

Victor Bucheli is with the Computer Science Department, Universidad del Valle, Colombia (victor.bucheli@correounivalle.edu.co)

II. Optimization

In [5] the author introduces the importance of research on optimization. In many jobs regarding engineering or science, we need to improve some processes which, at first, seem to work properly but when we exchange certain parameters, we can find better and more desirable results, therefore, to optimize is to choose these exchanges in the best way possible, this results in several possible solutions, and in order to accomplish this, we must look for the best way to find those solutions. The author indicates the techniques used to optimize the solution of real-life problems. First, there was the *deterministic search technique* known as the traditional method, which despite finding an optimum global, took a long time in convergence, therefore the researchers focused more on a proper approximate solution, which they can converge in a shorter computational time, this is how a new *stochastic optimization technique* is created, which follows the following algorithm:

1) Beginning: To generate and evaluate an initial collection of S candidate solutions
2) Operation: To produce and evaluate a collection of possible candidate solutions S when operating or making random changes to the selected member of S
3) Replacement: To replace one of the members of S with some of the members of S ' and return to step 2, unless some finalization criterion is reached.

In [5] three categories are discussed:

Category 1: New possible solutions have some slight variations regarding previously generated candidates.
Category 2: New possible solutions are generated when aspects of two or more existing candidates are recombined.
Category 3: The solutions of the current candidates from which new candidates are produced in category 1 or 2 are selected from candidates previously generated, through an stochastic and competitive strategy, which favors the best performance candidates .

III. Differential evolution algorithm (DE)

(DE) Differential Evolution [6] is a heuristic method of population-based optimization that works on individuals coded in real numbers. For each individual $\overrightarrow{x}_{i,G}$ in the current generation G, DE generates a new test individual $\overrightarrow{x}'_{i,G}$ by adding the weighted disparity between two randomly selected individuals $\overrightarrow{x}_{r1,G}$ y $\overrightarrow{x}_{r2,G}$ to a third individual selected at random $\overrightarrow{x}_{r3,G}$. The resulting individual $\overrightarrow{x}'_{i,G}$ is crossed with the original individual $\overrightarrow{x}_{i,G}$. The fitness of the resulting individual, called disturbed vector $\overrightarrow{u}_{i,G+1}$, which is then compared to the fitness of $\overrightarrow{x}_{i,G}$. If the fitness of $\overrightarrow{u}_{i,G+1}$ is greater than the fitness of $\overrightarrow{x}_{i,G}$, $\overrightarrow{x}_{i,G}$ is replaced with $\overrightarrow{u}_{i,G+1}$, otherwise $\overrightarrow{x}_{i,G}$ remains in the population as $\overrightarrow{x}_{i,G+1}$. The differential evolution is robust, fast, and effective with the ability to obtain a global optimum. It can be seen as a new way to implement the category 2 of [5].

Some limitations in the performance of the conventional DE algorithm depend to a large extent on the generation strategy of the chosen test individual and on the values of the parameters used. Inadequate choice of strategies and parameters can lead to a premature convergence or stagnation. In [7] a SaDE algorithm was developed in which the strategy of generating test individuals and their parameters can be self-adapted gradually according to their previous experiences of generating promising solutions. the investigations that are being developed in this field, are based on investigating the effect of improving the necessary parameters and selecting the best individuals of base tests [8].

IV. Imperialist competitive algorithm

The imperialist competitive algorithm (ICA) was proposed by Atashpaz in 2007 [9]. In [10] the imperialist competitive algorithm (ICA) is described, which is a meta-heuristic algorithm [11] of optimization motivated by one of the sociopolitical models, the imperialist competition, which is an economic theory known as mercantilism, this inspired the government to extend its power and rules beyond its own borders. For the initial population the term "countries" is used and categorized in colonies and imperialist states, respectively. The basic idea behind ICA is to lead the search process towards the powerful imperialists or the optimal points based on their power, that is, the weakest empires will lose their colonies until there is no colony in them. The objective is to find an ideal world in which there is no difference not only between the colonies, but also between the colonies and imperialism.

The steps to implement ICA can be summarized as follows:

1: Define the optimization problem.
2: Generation of initial empires by choosing some random points in the function.
3: Move the colonies to the imperialist states in different directions (ie assimilation).
4: Random changes occur in the characteristics of some countries (ie the revolution).
5: Exchange of position between a colony and imperialist.
6: Calculate the total cost of all empires.
7: Use imperialist competition and choose the weakest colony of the weakest empire.
8: Eliminate empty empires.
9: Check if the maximum iteration is reached, go to step 3 for a new beginning. If a specified completion criteria is met, stop and return the best solution.

V. Complex Networks

In 2002 [12], it was reviewed the progress in the evolving networks, where they focused on the structural properties of complex networks regarding communications, biology, social sciences and economics. At that time, artificial giant networks of such type were being created, which allowed the study of their topology, evolution and the complex processes that occur in them. These networks have a large set of scale properties. Some of them are scale-free and show a surprising recovery capacity when facing random failures. Despite the large sizes of these networks, the distances between the majorities of their vertices are short, a feature known as the *small world* effect.

It was defined complex networks as systems composed of a large number of highly interconnected dynamic units [13]. To capture the global properties of such systems, they must be modeled as graphs whose nodes represent the dynamic units, and whose edges represent the interactions between them. From this research, some questions which are relevant to the study of the dynamics of complex networks arise, such as: how a large set of dynamic systems that interact through a topology of complex networks can behave collectively. In this reference, the main concepts and results achieved during 2006 regarding the study of the structure and dynamics of complex networks were reviewed.

A. Structural Properties

Complex networks are modeled as graphs where their vertices are nodes connected by edges or links. The edges can be directed or not directed. It is established that the length of all the edges will be one.

The structure of the network is described by the *adjacency matrix*, $\hat{B}$, whose elements are ones and zeros. An element of the adjacency matrix of a network with non-directed edges, $b_{\mu,v}$, is 1 if the vertices formula μ and v are connected, and otherwise it will be 0. Therefore, the adjacency matrix of a network with no directed edges it's symmetrical. For networks with directed edges, an element of the adjacency matrix, $b_{\mu,v}$, is equal to 1 if there is an edge of the vertex μ to the vertex v, and otherwise is equal to 0.

In the case of a random network, an adjacency matrix only describes a particular member of the entire random graph statistical set. Therefore, what one observes is only a particular realization of this statistical set and the adjacency matrix of this graph is only a particular member of the corresponding set of matrices.

The statistics of the adjacency matrix of a random network contain complete information about the structure of the network and, in principle, only the adjacency matrix must studied. In general, this is not an easy task, which is why, instead of this, only a very restricted set of structural characteristics is considered.

1) Degree: The simplest and most studied characteristic of a vertex is the degree. The degree, k, of a vertex is the total number of its connections called connectivity. The indegree, k_i, is the number of input edges of a vertex. The outdegree,

k_o is the number of edges that come out of the vertex. Therefore, $k = k_i + k_o$.

The degree is actually the number of nearest neighbors of a vertex, z_1. Total distribution of the degrees of vertices of a complete network, $P(k_i, k_o)$ the joint distribution of indegree and outdegree, $P(k)$ the distribution of the degree, $P_i(k_i)$ the distribution of the indegree, and $P_o(k_o)$ the distribution of the outdegree are the vertex basic statistical characteristics . Thus

$$P(k) = \sum_{k_i} P(k_i, k - k_i) = \sum_{k_o} P(k_o, k - k_o) \quad (1)$$

$$P_i(k_i) = \sum_{k_o} P(k_i, k_o) \quad (2)$$

$$P_o(k_o) = \sum_{k_i} P(k_i, k_o) \quad (3)$$

In short, instead of $P_i(k_i)$ and $P_o(k_o)$ we usually use the notation $P(k_i)$ and $P(k_o)$. If a network does not have any connection with the outside, then the average input and output are the same:

$$\bar{k}_i = \sum_{k_i, k_o} k_i P(k_i, k_o) = \bar{k}_o = \sum_{k_i, k_o} k_o P(k_i, k_o) \quad (4)$$

Although the degree of a vertex is a local quantity, we will see that a distribution of degrees often determines some important global characteristics of random networks. In addition, if the statistical correlations between vertices are absent, $P(k_i, k_o)$ determines, in a total way, the structure of the network [12]

2) Average path length: The shortest path: One can define a geodesic distance between two vertices, μ and v, of a graph with edges of unit length. It is the average path length, $\ell_{\mu v}$, from the vertex μ to the vertex v. If the vertices are directed, $\ell_{\mu v}$ is not necessarily equal to $\ell_{v\mu}$. It is possible to introduce the distribution of the shortest path lengths between pairs of vertices of a network and the average shortest path length $\bar{\ell}$ of a network. the average is applied to all the pairs of vertices between which there is a route, also to all the realizations of a network.

ℓ is called the "diameter" of a network. Determines the effective "linear size" of a network, the average separation of pairs of vertices. For a section of dimension d that contains N vertices, obviously, $\bar{\ell} \sim N^{1/d}$. In a completely connected network, $\bar{\ell} = 1$. One can roughly estimate the $\bar{\ell}$ of a network in which random vertices are connected. If the average number of nearest neighbors of a vertex is z_1, then on z_1^{ℓ} the vertices of the network are at a distance ℓ from the vertex or closer. Thus $N \sim z_1^{\bar{\ell}}$ and then $\bar{\ell} \sim \ln N / \ln z_1$.

It is also possible to introduce the maximum length of shortest path over all the pairs of vertices between which a path exists. This feature determines the maximum length of a network[12].

3) The Clustering coefficient: For the description of the connections in the environment closest to a vertex, the so-called *grouping coefficient* is introduced. For a network with non-directed edges, the number of all the possible connections of the nearest neighbors of a vertice μ ($z_1^{(\mu)}$ nearest neighbors) equal to $z_1^{(\mu)}$ ($z_1^{(\mu)}$ - 1)/2. Leave only $y^{(\mu)}$ of these which are present. The clustering coefficient of this vertex, $C^{(\mu)} \equiv y^{(\mu)}/z_1^{(\mu)}(z_1^{(\mu)} - 1)/2]$, is the fraction of the connections existing between the nearest neighbors of the vertex. Averaging $C^{(\mu)}$ over all the vertices of a network produces the network's grouping coefficient C. The grouping coefficient is the probability that two nearest neighbors of a vertex are also neighbors to each other. The network's grouping coefficient reflects the "distinction" of the neighborhood closest to the average of a network vertex, that is, the measure to which the nearest neighbors of a vertex are the nearest neighbors also to each other[12].

4) Connected Components: In randomized graphs, the sizes of the connected components [14] are given by a random variable, which, in turn, depends on the specific model. The model $G(n, p)$ has three regions with apparently different behavior:

- Subcritical $np < 1$: All components are simple and very small, the largest component has a size of $|C_1| = O(\log n)$
- Critical $np = 1$: $|C_1| = O(n^{\frac{2}{3}})$
- Supercritical $np > 1$: $|C_1| \approx yn$ where $y = y(np)$ is the positive solution to the equation: $e^{-pny} = 1 - y$

Where C_1 and C_2 are the largest and the second largest components, respectively. All of the other components have their magnitudes of size: $O(\log n)$.

VI. Relationship between evolutionary algorithms and complex networks.

Evolutionary computation can effectively address complex problems that the traditional optimization algorithm can not solve [5], but evolutionary computation also has some deficiencies [4], for example, the way of describing traditional evolutionary algorithms is unclear. Since it only pays attention to the results, but ignores the intermediate processes that can affect the result. Therefore, through the analysis regarding the optimization process, the interactive process can be described through the implicit process of the structure of the complex networks and some opinion of improvement is proposed.

Is it possible to visualize and simulate an evolutionary process with a complex network? If complex networks are hidden behind the dynamics of evolutionary algorithms, then it is possible to analyze and control them.

The traditional approach generally ignores the intermediate procedures of the evolutionary algorithms, which have a great impact on the results. If the people chosen to reproduce are close relatives, it is detrimental to the development of the species. Complex networks can correct the disfigurement of evolutionary algorithms. In the process of optimization of the algorithm, there is a type of relationship between individuals. This relationship was proven to be a complex network topology. The optimization process of the algorithm described by the complex network topology structure could be more intuitive and easier to understand and, besides, it could benefit population growth. The relationship between each individual in the optimization process of the algorithm could be shown dynamically and the population's evolutionary process could be described using the dynamic characteristics of complex networks.

In the evolution process, the relationship between individuals is a changing process, and the topology structure model can be established according to the relationship between individuals. The topology structure of the algorithm is analyzed by the statistical characteristic of the complex network, and the characteristics of each algorithm is verified by the structure diagram through the study of the network topology, in order to analyze the advantages and disadvantages of the algorithm. Then, according to the results of the graph analysis, the best specific opinions of the algorithm are proposed.
In this article, we propose a visualization of evolutionary algorithms as complex networks based on [1] and the idea of using these properties to improve the algorithm is proposed. In this order of ideas, the following question arises: How to control the dynamics of evolutionary algorithms using the characteristics of complex networks?

VII. Methodology

The Figure 1 explains the architecture used by the program for network analysis. The process begins by parameterizing the optimization algorithm, indicating the size of the population and the optimization function to be used, among others. Afterwards, the optimization algorithm is executed and the network files are generated (in this case with Pajek format) to be used in the Mathematica software. In Mathematica these files are loaded and then the networks are generated with the libraries of this software. Finally, the measurement of the topological properties of the networks is generated and the graphs are generated and presented in this work.

A. Evolutionary and random algorithm

In this algorithm each new individual of the population x' is created from two individuals x_1 and x_2. The selection of these parents is created by roulette-type methods where the best individuals are encouraged but without completely ruling the worst out. The creation of a link for each parent towards his new son for the network model is proposed.
The random proposal consists of the same evolutionary algorithm but with a totally random selection, that is, two randomly selected individuals regardless of their performance

B. Differential evolution

It is proposed that for each individual of the population $x_{i,G}$, a new individual $x'_{i,G}$ be generated based on the difference of weights of two individuals $x_{r1,G}$ and $x_{r2,G}$ added to another individual $x_{r3,G}$, all randomly selected. This new individual $x'_{i,G}$ is crossed with the original individual

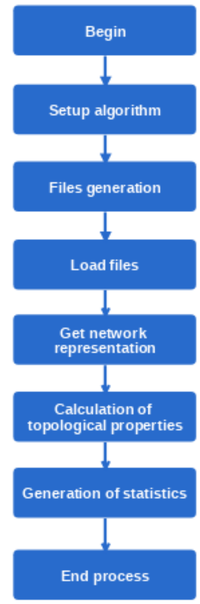

Fig. 1. Software Architecture

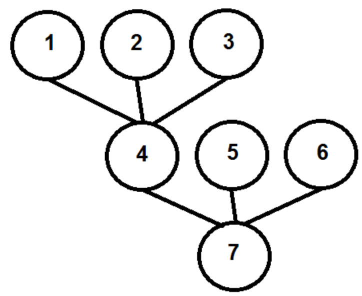

Fig. 2. Example of differential evolution networks modeling

$x_{i,G}$. If this new vector called $u_{i,G+1}$ is better than the original individual, it is replaced by the original individual $x_{i,G}$, otherwise the new solution is discarded.

In this implementation the creation of new chromosomes (nodes) is omitted throughout conventional manner. The values of the predecessor are updated by the values of the new individual, keeping constant the size of the population and creating edges from the individuals who intervened in this new update towards the new individual.
Figure 2 Shows an example of network modeling in the differential evolution algorithm. In this example, individuals 1,2 and 3 are used in the recombination process of individual 4 in order to create a new individual. The links are created because in the recombination process, the creation of an individual, better than the 4, happens. This new individual improves the process and replaces the individual 4 and they keep their previous edges, as in the example where the individual 4 served to improve the process of the individual 7.

C. Imperialist competitive algorithm

In the network model, it is important to determine what the nodes and edges represent. In this model, it is proposed

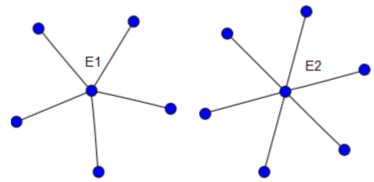

Fig. 3. Example of the imperialist competitive algorithm

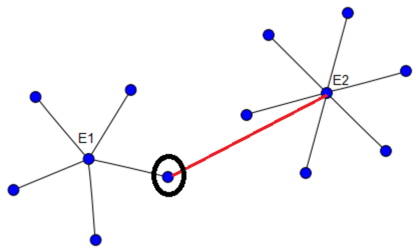

Fig. 4. Example of a new edge in the imperialist competitive algorithm

that each country (node) will have an edge (relationship) with the empire that dominates it. In each iteration of the algorithm, new relationships are added, either by the weaker countries that are dominated by the strongest empires or by the probability that a new country will take control of the empire. These new relationships add dynamism to the network and allow topological properties to converge to a complex network.

In Figure 3 An example with the visualization of two empires is presented, there, each country has a link with the empire that dominates it but there are no links between countries. The Figure 4 shows a dynamism of the network, where a country of the empire E1 is dominated by the empire E2 and thus this aforementioned connects the two empires.

VIII. Network tests and analysis

In previous works, it is argued that differential evolution can be visualized as complex networks. The measurements of the Clustering coefficient and the average path length were made in order to study their topological properties in a total of 80 experiments, where the values of each iteration are averaged using the sphere optimization function.

A. *Tests in differential evolution*

In the Figure 5, it is possible to see the clustering coefficient increasing as the iterations happen and the network connects better. In the same way, it is possible to see in the Figure 6 how the average path length decreases as the network connects. These obtained values coincide with the expected values in the small world model and, therefore, they do not refute the hypothesis proposed in previous works of similarity to a complex network.

In the same way, its behavior is studied comparing its performance against the traditional evolutionary strategy and with random walks, in the same 80 iterations. Figure 7 clearly shows how it has a dominant performance in all iterations, although in half it achieves the performance of the evolutionary strategy.

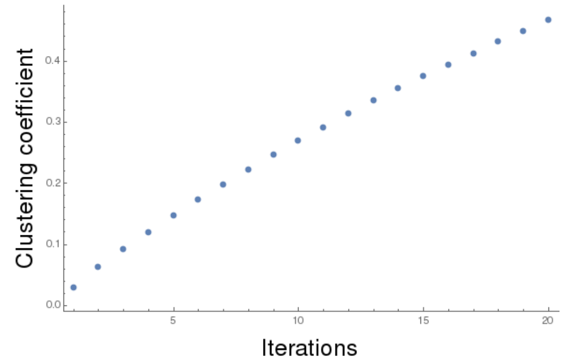

Fig. 5. Clustering coefficient in several experiments of differential evolution

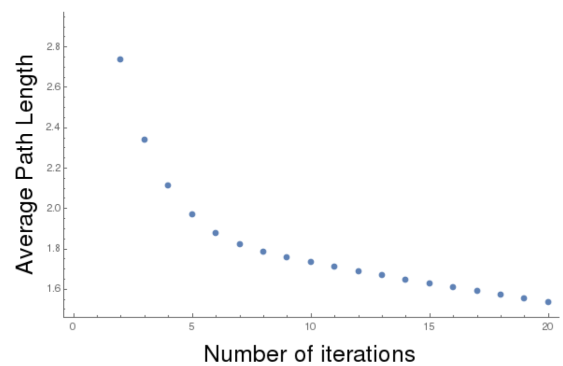

Fig. 6. Average path length in several experiments of differential evolution

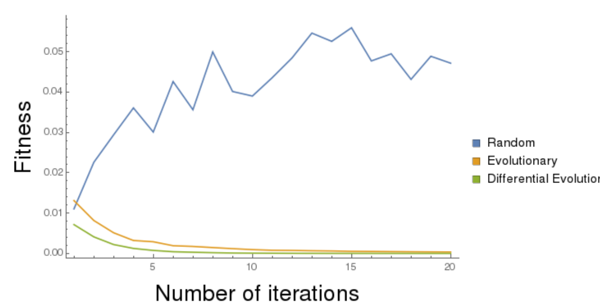

Fig. 7. Differential Evolution Performance with the proposed model

B. *Tests in Imperialist competitive algorithm*

The topological properties of the network obtained in competitive imperialism are analyzed and the conclusion is that it has certain similarities with the scale-free network model.

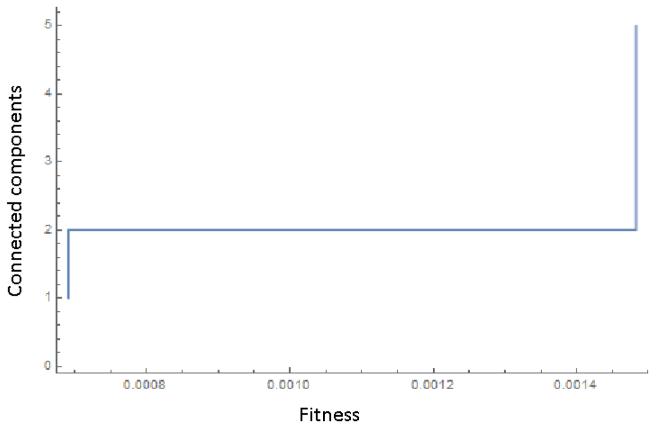

Fig. 8. Connected components vs Fitness in Imperialist competitive algorithm

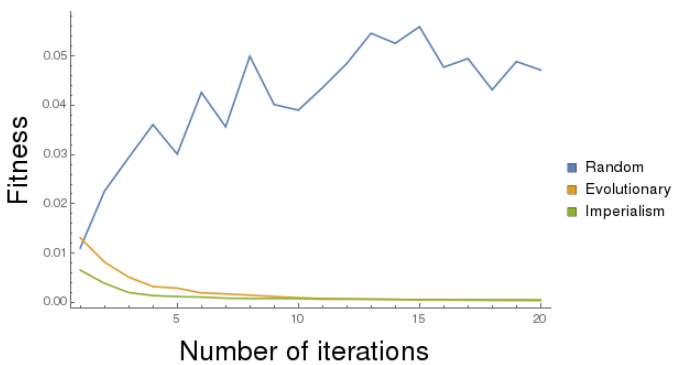

Fig. 9. Performance of Imperialist competitive algorithm with the proposed model

However, the most important measurement of the distribution as a power law is not fully met and this hypothesis is discarded. Other topological properties continue to be analyzed and the measurement shown in Figure 8 is found, which shows how the fitness improves as the number of components decreases. This relationship is important because it could encourage the reduction of this topological property in order to improve the performance of the strategy.

Their behavior is studied comparing their performance against the traditional evolutionary strategy and with random walks in the same 80 iterations. In Figure 9 It is clearly seen how it has a dominant performance in all the iterations, although, in the last iterations, it reaches the performance of the evolutionary strategy.

Finally, a comparison was made concerning the performance (fitness) of the 4 strategies proposed in the different iterations as it can be seen in Figure 10. In this experiment, 80 executions of the four algorithms were performed and averaged for each iteration.

It can be seen how optimization strategies perform much better than a random path. It can be highlighted how the strategy of differential evolution, which presents properties of complex networks, behaves better and has a dominance with regards to the strategies of traditional evolutionary computation and the competitive imperialism in a process of 200

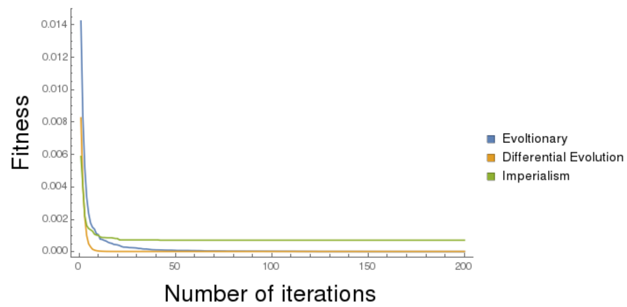

Fig. 10. Performance of optimization strategies over time

iterations as shown in Figure 10

IX. Conclusion

The results obtained from the experiments allow us to reach the same conclusions of other researchers of the field [2] that indicates that the algorithm of differential evolution can be visualized as a complex network structure, but we can also say that it has small world properties.
The results of the experiments carried out answered the open question of How to control the dynamics of evolutionary algorithms using the characteristics of complex networks. It was identified that the performance or dynamics of imperialism could be improved if the decrease in the number of components is encouraged. Moreover, it allows us to know that the strategies that have complex network properties seem to solve the problem in fewer iterations
Other questions that were partially answered with this work:

- How to generalize the presence of complex networks in evolutionary computation?
  It is necessary to work on the basis of these properties and study what conditions do those that do have networks (how many generations, size of the population, etc.) and with this, the aim is to generate a set of common patterns.
- What relationships exist between individuals?
  In the experiments of imperialism it was possible to identify that its performance can be improved depending on the number of components.
  In the algorithm of differential evolution it is necessary to maintain the individuals so that it can be visualized as a complex network.
- What complex network techniques can be implemented?
  So far with the results obtained we know that the small world network can be implemented; more experiments with the algorithm of imperialism are needed in order to know if it can be a free-scale network.
- If complex networks are hidden in the evolutionary algorithms, is it possible to control the flow of the algorithm with the representation?

This open questions are proposed in the relation between complex networks and optimization algorithms based on the results

- How to implement a control mechanism based on these complex networks topologies?

- How to guide the population, selection, reproduction and mutation?
- Improve the quality of solutions and premature convergence?
- What is the impact of the additional cost in each iteration and how it's scaling?

It is necessary to advance in these aspects of control. Still unresolved in optimization research.

A comparative analysis of the computational costs of the 4 selected strategies was also carried out. The random walk and the evolutionary strategy take around 0.02 seconds, the strategy of imperialism competitive algorithm takes 0.12 seconds and the strategy of differential evolution takes approximately 0.27 seconds. The additional computation time is proportional to the quality of the solutions according to the results shown, noting the cost of generating the network model that is used in the differential evolution. This factor is important for the selection of the model and to consider the additional computational cost against the quality of the solution.

REFERENCES

[1] I. Zelinka, D. Davendra, V. Snášel, R. Jašek, R. Šenkeík, and Z. Oplatková, "Preliminary investigation on relations between complex networks and evolutionary algorithms dynamics," *2010 International Conference on Computer Information Systems and Industrial Management Applications, CISIM 2010*, pp. 148–153, 2010.

[2] I. Zelinka, D. Davendra, J. Lampinen, R. Senkerik, and M. Pluhacek, "Evolutionary algorithms dynamics and its hidden complex network structures," *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 3246–3251, 2014.

[3] M. Kirley and R. Stewart, "Multiobjective evolutionary algorithms on complex networks," in *Evolutionary Multi-Criterion Optimization*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 81–95.

[4] L. Sheng, F. Chen, and H. Wang, "A Review on complex network dynamics in evolutionary algorithm," *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016.

[5] G. C. Onwubolu and B. Babu, *New optimization techniques in engineering*. Springer, 2013, vol. 141.

[6] K. V. Price, "An introduction to differential evolution," in *New Ideas in Optimization*, D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli, and K. V. Price, Eds. Maidenhead, UK, England: McGraw-Hill Ltd., UK, 1999, ch. An Introduction to Differential Evolution, pp. 79–108. [Online]. Available: http://dl.acm.org/citation.cfm?id=329055.329069

[7] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, April 2009.

[8] L. Jiang and H. Qiang, "Differential evolution algorithm with base vector group," in *Proceedings of the 32nd Chinese Control Conference*, July 2013, pp. 8006–8009.

[9] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *Evolutionary computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 4661–4667.

[10] B. Xing and W.-J. Gao, "Imperialist competitive algorithm," in *Innovative computational intelligence: A rough guide to 134 clever algorithms*. Springer, 2014, pp. 203–209.

[11] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[12] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," *Advances in Physics*, vol. 51, no. 4, pp. 1079–1187, 2002. [Online]. Available: https://doi.org/10.1080/00018730110112519

[13] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4, pp. 175 – 308, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037015730500462X

[14] Y. Shiloach and S. Even, "An on-line edge-deletion problem," *J. ACM*, vol. 28, no. 1, pp. 1–4, Jan. 1981. [Online]. Available: http://doi.acm.org/10.1145/322234.322235