This article can be cited as J. Triana, V. Bucheli and A. Garcia, Traveling Salesman Problem Solving Using Evolutionary Algorithms Guided by Complex Networks, International Journal of Artificial Intelligence, vol. 18, no. 2, pp. 101-112, 2020.

Traveling Salesman Problem Solving Using Evolutionary Algorithms Guided by Complex Networks

Joshua Triana[1], **Victor Bucheli**[2] **and Angel Garcia**[3]

[1]Department of Computer Science
Universidad del Valle
Thirteenth street 100-00, Cali, Colombia
joshua.triana@correounivalle.edu.co

[2]Department of Computer Science
Universidad del Valle
Thirteenth street 100-00, Cali, Colombia
victor.bucheli@correounivalle.edu.co

[3]Department of Computer Science
Universidad del Valle
Thirteenth street 100-00, Cali, Colombia
angel.garcia@correounivalle.edu.co

ABSTRACT

The evolutionary computation is an optimization strategy of the artificial intelligence, it is widely used for solving problems, which traditional algorithms cannot solve. Moreover, complex networks model systems, in which, the relationship structures are represented as graphs. The Traveling Salesman Problem (TSP) is a well known computational problem solved using many strategies, included evolutionary computation. The literature review shows that the population of evolutionary computation strategies can be represented as complex networks, as follows: a graph represents the group of solutions, where the vertices (nodes) represent the individuals within the population of solutions, and the edges (links) represent the crosses among individuals (solutions). Thus, to improve solutions, links among existing nodes of the population are created. This work aims at the development of an algorithm to guide the population dynamics of an evolutionary algorithm that solves TSP. It is based on the hypothesis that one strategy guided by a complex dynamic network yields better results than a traditional algorithm. Furthermore, the results substantiate that the model proposed not only improves convergence compared to the traditional solution, but also enables shorter execution times. Strategies based on small-world networks show enhanced overall performance in the experiments conducted compared to those using other network topologies, and subsequently the traditional solution.

Keywords: Evolutionary algorithms, Network optimization, Artificial Intelligence
Computing Classification System (CCS) Computing methodologies $\sim$ Computational control theory, Mathematics of computing $\sim$ Evolutionary algorithms, Theory of computation $\sim$ Network optimization

1 Introduction

Evolutionary computation has become a reference for artificial intelligence (Lawrence J. Fogel and Walsh, 1966). Different researchers have tried to improve the performance of this strategy, like exploring control mechanisms for the parameters (Eiben, Hinterding and Michalewicz, 1999) of the strategy in order to improve the quality of the solutions and convergence.

Complex networks present a graph model where the vertices and edges have a set of defined properties that generate a complexity and structure of interest (Dorogovtsev and Mendes, 2003). These networks have been useful to understand the behavior of many real systems from the relationship of their individuals and their dynamics. It has been possible to find underlying networks in some evolutionary algorithms that have properties of complex networks (Zelinka, Davendra, Snášel, Jašek, Šenkeřík and Oplatková, 2010). In this investigation underlying networks are built through the interaction between individuals and some experimental conclusions are reached.

Some investigations in progress suggest that a complex network that guides the interaction between individuals in an evolutionary computation strategy could improve the overall quality of the solution (Triana, Victor and Garcia, 2020). It is important to make a comparative analysis of the proposed model in this work with the original model in order to measure the proposed model's impact. The results of this research could help to advance in the idea of including the complex networks dynamics in optimization algorithms to improve its performance.

In Chapter 2 the evolutionary computation and complex networks model are presented, with review of the most relevant applications that worked with both. In Chapter 3 the TSP problem and its optimization problem is described. In Chapter 4 the methodology of the algorithm is explained. In Chapter 5 the experimental results are discussed. Finally, in Chapter 6 we discussed the conclusions of this research and some possible directions for future work.

2 Evolutionary Computation and Complex Networks

2.1 Evolutionary Computation

In (Bäck, 1996) Evolutionary techniques began, when the Genetic Algorithms (GA) invented by J. Hollan, appeared in 1975. They were a family of search and optimization methods inspired by biological evolution. Since its invention, GAs have inspired many related methods and have led to the most advanced field of evolutionary computing, with scientific and commercial applications (Holland, 1992). In 1976 Schwefel (Schwefel, 1976) and in 1973 Rechenberg (Rechenberg, 1973), Invented the evolution strategy (ES) which is an optimization technique based on ideas concerning evolution. The evolution strategies use natural representations which depend on the problem, and mainly mutation and selection as search operators. Operators in common with evolutionary algorithms, are applied in a loop. An iteration of the cycle is called generation. The sequence of generations continues until a termination criterion is met. It belongs to the general class of evolutionary computation or methodologies of artificial evolution. In 1998 Lawrence J. Fogel (Fogel, 1998) used evolutionary programming for the first time which is one of the four main paradigms of evolutionary algorithms and is similar to genetic programming, but the structure of the program to be optimized is fixed, while its numerical parameters can evolve. Fogel used finite-state machines as predictors and evolved them. Currently, evolutionary programming is a broad evolutionary computational dialect without a fixed structure, in contrast to some of the other dialects. All these designs were favored by the appearance of more powerful and easily programmable computers, thanks to the aforementioned, interesting problems were addressed for the first time and the evolutionary calculation began to compete and became a serious alternative to other optimization methods.

The Evolutionary Algorithm (EA) (Vikhar, 2016) is a subset of evolutionary computation and is inspired by biological evolution. The algorithm produces a pseudo random population of chromosomes (individuals usually represented as vector numbers) and then evaluated by their performance to solve a given optimization problem. They are selected according to that performance and then matched in pairs in order to reproduce and generate a new individual. The new individual may have some random mutation in the chromosome and finally replaced to the main population following some rules. The process repeats until a stop criteria is met.

2.2 Complex Networks

In 2002, Dorogovtsev reviewed the evolution in networks(Dorogovtsev and Mendes, 2002) and focused on the structural properties of complex networks in communications, biology, social sciences and economics. At that time, such artificial giant networks were being created, which allowed the study of their topology and evolution. These networks have a large set of scale properties. Some of them are scale free and show surprising resilience in the face of random failures. Despite the large sizes of these networks, the distances between the majorities of their vertices are short, a characteristic known as the effect of *small world*.

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 complex networks used in this work are known as Small world networks(Amaral, Scala, Barthelemy and Stanley, 2000). This networks are those that have the shortest average distance between a pair of vertices proportional to the natural logarithm of the number of nodes and the clustering coefficient is large, which indicates that the vertices of the network tend to be strongly connected.

2.3 Complex networks in evolutionary computing

There are evolutionary computation strategies that allow you to visualize relationships between your individuals over time through complex networks, among which it is possible to highlight: SOMA(Zelinka, 2004) and Differential Evolution(Storn and Price, 1997).

(Zelinka et al., 2010) uses the SOMA algorithm, developed by the author, and the differential evolution to model complex networks. This investigation models the father/child offspring relationships as networks, among which the shortest average path length with small values and power law type distributions stand out. The author proposes as a future direction to use the properties of this type of networks as an improvement. The author applied some modifications to the algorithms already studied (Zelinka, Davendra, Roman and Roman, 2011; Zelinka, 2011; Zelinka, n.d.).

A review on complex networks in evolutionary computation has been made (Sheng, Chen and Wang, 2016). In this work the author explains how the degree of distribution could be studied to analyze the premature convergence, the average path length between the nodes to determine the existence of complex networks, the clustering coefficient could be used to analyze the individual progress in each iteration and the centrality between the nodes as an alternative to find the best individuals. These studies seem promising although they lack implementation and analysis of results, that is to say, they are only conceptually presented.

3 Optimization problem

3.1 The Traveling Salesman Problem

The Traveling Salesman Problem tries to solve the problem where given a list of cities and the distances between each pair of them, determine the shortest route possible to visit each city exactly once and finally returns to the city of departure (Laporte, 1992). A set of n cities enumerated 0, 1, ..., n-1 to be visited with the distance between each pair of cities i and j is given by c_{ij}. The algorithm will decide variables y_{ij} for each (i,j) such that:

$$y_{ij} = \begin{cases} 1 \, if \; city \; j \; is \; visited \; immediately \; after \; city \; i \\ 0 \; otherwise \end{cases} \tag{3.1}$$

The objective of TSP is to minimize $\sum_i \sum_j c_{ij} y_{ij}$. This is a classic NP-Hard problem well known and used in computer science in general and for this reason was selected to test the proposal. In optimization problems, there has been different models to path problems (Purcaru, Precup, Iercan, Fedorovici, David and Dragan, 2013), (Alvarez, Johanyák and Kovács, 2018), (Precup and David, 2019) and particularly for the Traveling Salesman Problem (Osaba, Del Ser, Sadollah, Bilbao and Camacho, 2018). It has been studied some complex networks properties in TSP such as Community Detection (Liu, Feng, Hu and Jia, 2014) and PageRank Distance (Jiang, Liu and Wang, 2016). The model proposed in this papers aims to contribute as an alternative to solve this problem.

3.2 Optimization problem for TSP

TSP is a combinatorial optimization problem usually named Integer Linear Programming (ILP) (Papadimitriou and Steiglitz, 1982), because some or all of the variables are restricted to be integers. ILP is usually expressed as:

$$\begin{aligned}&\text{Maximize } \mathrm{c}^{t}x\\&\text{subject to: } \mathrm{Ax} <= \mathrm{b,x}>=0 \text{ and } \mathrm{x} \in Z^{n}\end{aligned} \tag{3.2}$$

Where c,b are vectors of integers and A is a matrix of integers. It has been proved that using a reduction from minimum vertex cover to integer programming will proof the NP-hardness (Kannan and Monma, 1978).

This optimization problem would be solved using the evolutionary algorithm (EA), where the aim is to minimize the distances the travel using initial pseudo random paths (population), and then following the EA model in the process of selection, crossover and mutation for new paths.

4 Methodology using complex networks

In the traditional evolutionary algorithm, networks could be generated from the relationships between their individuals. The detectable relationship in this strategy is the father-son relationship, established when an individual serves to create or improve another individual. This interaction occurs in the reproduction process where, through some selection technique, two individuals generate a new individual by recombining their genes. With this modeling it is possible to create networks based on the relationship of offspring of individuals.

The generation of networks was successful, but its topological properties do not coincide with that expected in complex networks. This result and the apparent competitive advantage of the algorithm, where underlying complex networks emerge, are the inspiration for the model in (Triana et al., 2020). It is proposed to take a network using the small world model proposed by Watts and Strogatz (Watts and Strogatz, 1998) to guide the strategy of traditional evolutionary computation.

Initially, a population of individuals are created to initiate the evolutionary algorithm and a small world network with the same number of nodes as the size of the population of individuals. A random association of an individual from the population is made with a node in the network, to proceed to start the iteration of the algorithm, where each individual from the population is taken and a neighbor of its associated node is selected to reproduce. In this way the reproduction of individuals through the network of the small world and the relationship between the nodes are guided. If the new offset has a better performance than the selected father, a direct replacement of the son by the father is made to maintain the size of the population. This last step is necessary so that the network continues to have the same structure and does not lose the properties of the small world.

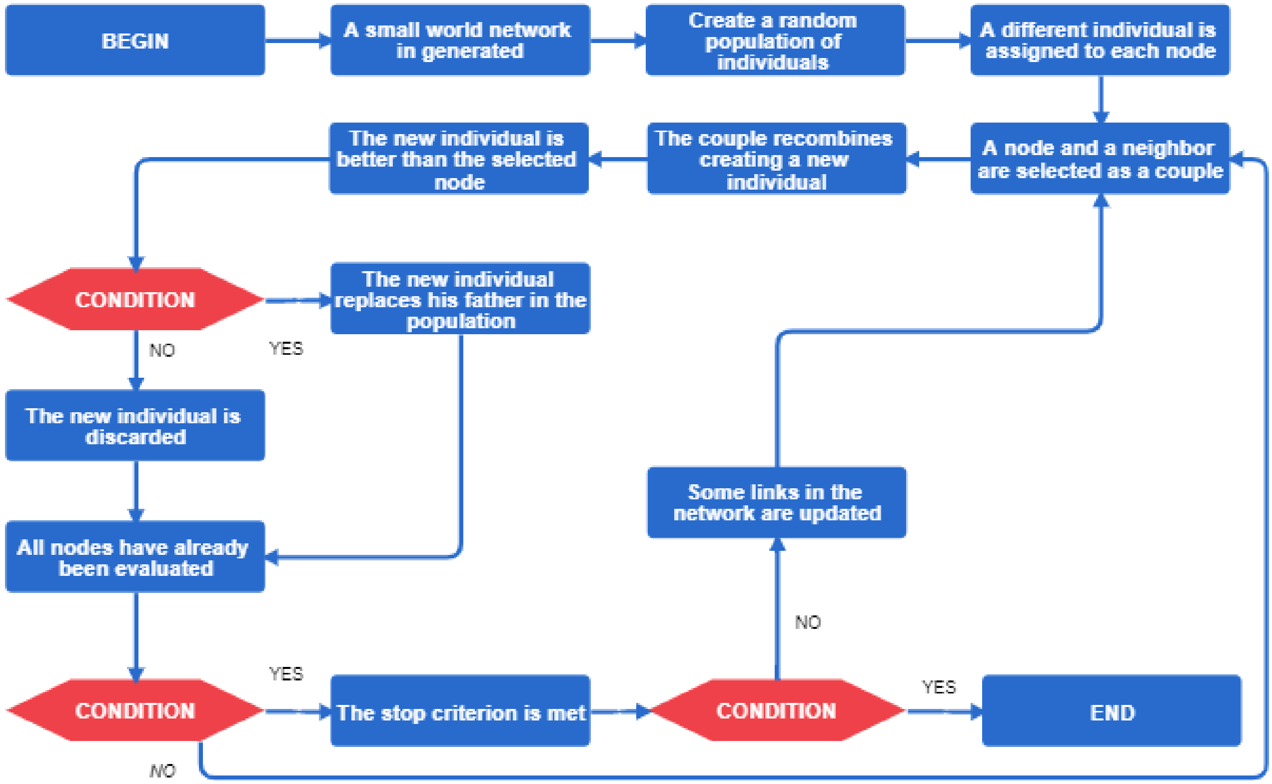

Figure 1: Flow diagram of the proposed model of this paper

This mechanism has a problem because keeping the same network for all generations could present a premature convergence. The solution is to follow the idea of Watts (Watts and Strogatz, 1998) and do a rewiring process. Each link of each node X' of the network that connects to another node X'' with a certain probability (between 10 and 30 %), breaks the link and joins another node X'''. In this way, at each iteration the network changes slightly and adds dynamism to it.

As shown in Figure 2 node can only select another neighbor node as a candidate to reproduce. The decision that the node must make is which of its neighbors to reproduce itself with. To solve this problem, possible variants of neighbor selection are considered.

- Standard model: This variant is called Small World, and it consists in the random selection of any neighbor. It is expected to represent with its name the intention that what is important is the guidance of these networks to individuals, rather than which neighbor it selects.

- Best neighbor model: We call this alternative SW Best, and it consists in the selection of the neighbor through the roulette method. The intention is to favor the selection of the neighbor with better performance (fitness)
- Higher degree model: This variant is called SW High Degree, and it consists in the selection of the neighbor with the highest grade. The intention is to select the most relevant neighbor in the network.

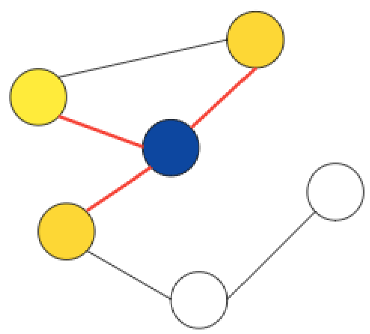

Figure 2: Selection of neighbours (yellow) for an individual (blue)

In (Triana et al., 2020) was proposed this model and some experiments where made using test optimization algorithms for optimization. In this investigation we test the proposed model using a real life NP-Problem and analyze the model's scalability over fitness, time and iterations.

5 Results

The experiments were made using these technologies:

- Python 3.7, as a programming language
- Spyder 3.3.3, Framework for scientific computing in Python
- Networkx 2.4, For modeling complex networks in Python
- Mathematica 12, For the analysis and visualization of networks and graphics in general

The implementation and test of the model can be found in this repository:
`https://github.com/jodatm/complex_networks_in_EA`

The relevant parameters for the experiments are:

- Population size: 100
- Mutation rate: 0,01

- Stop criterion: 50 Iterations
- Number of cities: 25

The population size and mutation rate are selected according to previous investigations of evolutionary computation approaches to solve TSP (Wei, Chen, Hu and Zhang, 2019). The stop criterion was selected due to manual test made before this experiments where later iterations did not show changes, for computational cost we decided to limit them. The number of cities were selected to test the model with a standard size problem and would be explored with more sizes in the scalability tests.

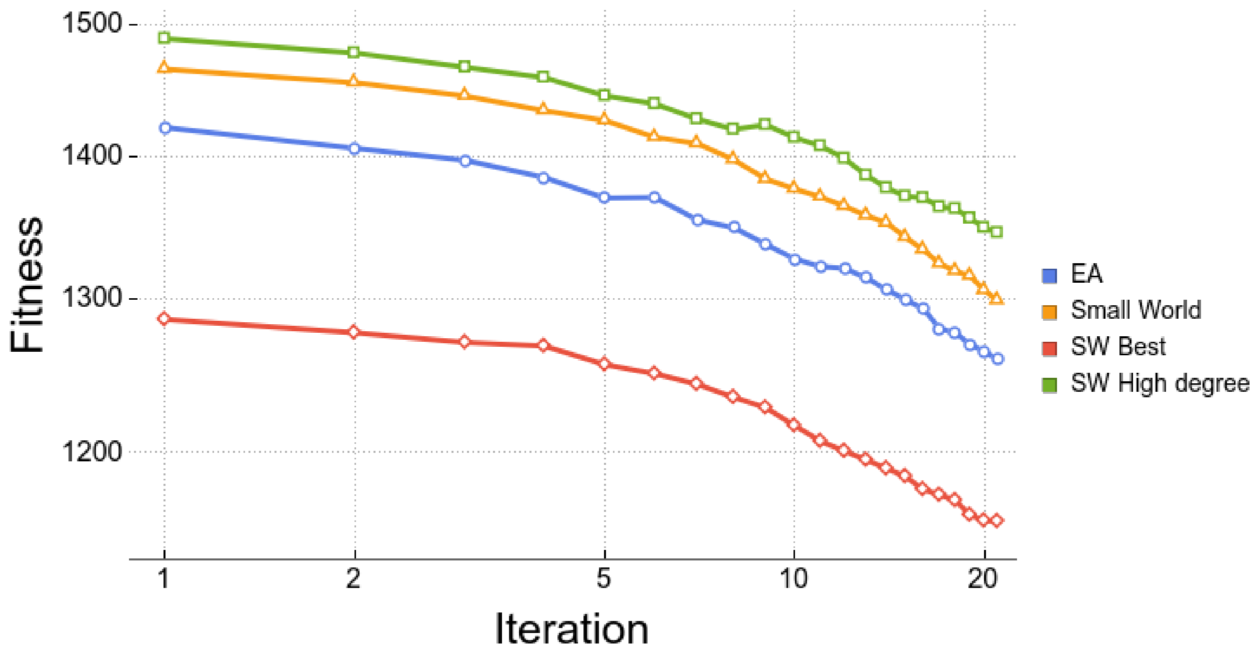

Figure 3: Average performance in TSP problem

The results in Figure 3 show a dominance of the proposed model using SW Best to the traditional algorithm when compared the fitness values over time. The Standard and Higher degree model show worse performance than the traditional algorithm in all iterations. In each experiment 50 iterations were made and the results are the averages of 500 experiments in each iteration.

The tests in the proposed algorithms indicate that the model with neighbor selection with better fitness (SW Best) could be the best alternative, it has better performance than the traditional algorithm in all iterations. This results are expected since the use of the optimization target (fitness) in the selection process may be better than other apparently none relevant evolutionary parameters, such as high degree. Degree and other network topology properties may be important to understand the dynamics of the network and used in other versions of the algorithm.

The convergence of the proposed and traditional algorithm is expected after some iterations due to the nature of the traditional algorithm in the exploration/ exploitation (Crepinsek, Liu and Mernik, 2013). Although the results are promising regardless to fitness and overall performance, the proposed algorithm in this work don't solve the possible and latent local or premature convergence of the traditional evolutionary algorithm.

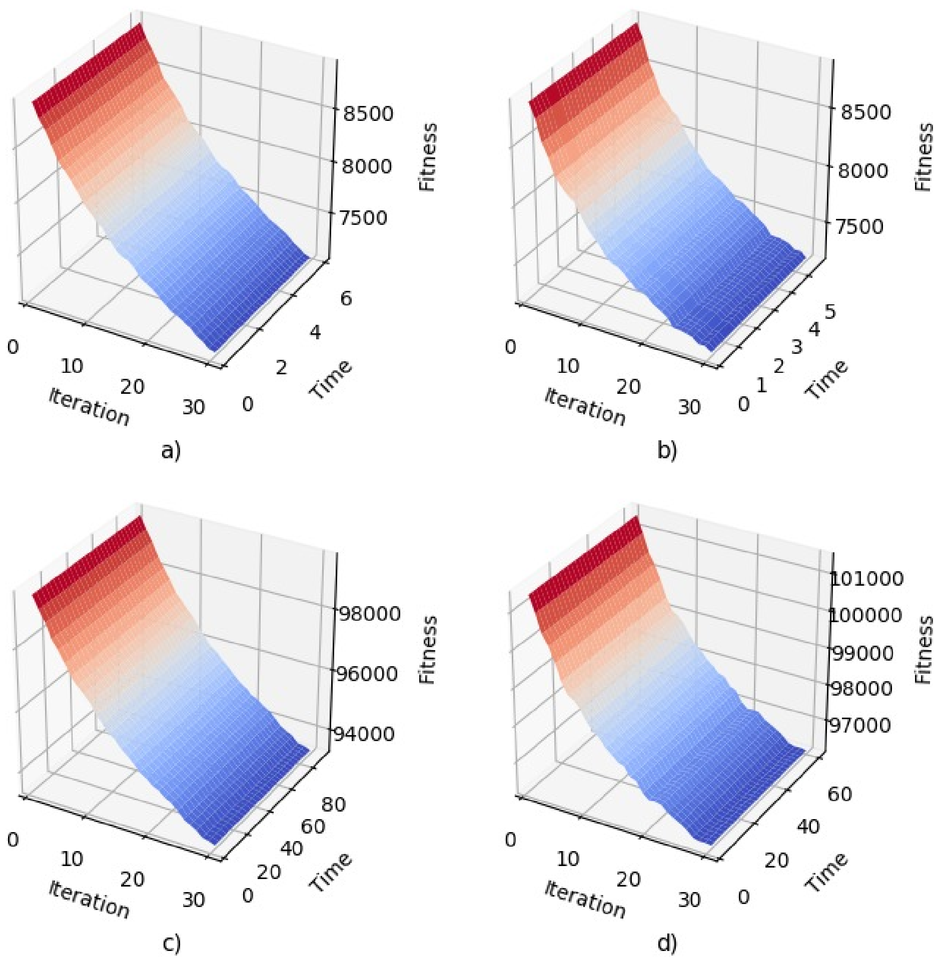

Figure 4: Iterations, time and average fitness of 200 experiments in TSP with 100 and 1000 cities

Finally, the impact of the computational cost of the model in counterpart with the original strategy is analyzed. Figure 3.a shows the traditional evolutionary strategy applied to the TSP problem with 100 cities and Figure 3.b shows the SW Best model (the best neighbour according to results) application to the same problem. It can be seen how the computational cost of the medical proposal in seconds of execution is less than the original strategy. The original strategy has aggregate computational costs of sorting and selecting a node with many nodes, while the proposed model by limiting this process only with its neighbors reduces the entire process over time. We also wanted to see how the behavior exchanges when the order of magnitude rises and analyzed it as the iterations go by. In Figure 3.c the experiment with 1000 cities is repeated using the original strategy and in Figure 3.d using the SW Best model. The same pattern of advantage of the model in terms of computation time and linearity over iterations, time and fitness.

6 Discussion

The proposed algorithm obtained better results compared to the traditional algorithm, same as previous investigation. The idea of a complex network guiding an evolution computation algorithm seems reasonable according to the results.

In all the experiments carried out in this work, the alternative of selecting the best-performing neighbors (fitness) obtained the best results. This selection promotes competition by means of the evaluation factor (the fitness to be improved) among the individuals of the population, which could explain this behavior.

The analysis between time and fitness performance in the comparison between proposed model and the traditional approach is key to understand the scalability of this work. The results are promising and enhance the impact of the model in larger problems. The thinkable additional cost of including complex networks dynamics in the evolutionary algorithm is unsubstantiated according to the results, mainly because a local matching in the selection process cost less than a global and elite selection, due to computational cost in population size and ordering.

Acknowledgment

This work was supported by the computer science department of Universidad del Valle

References

Alvarez, R., Johanyák, Z. and Kovács, T. 2018. Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation, *International Journal of Artificial Intelligence* **16**: 20–40.

Amaral, L., Scala, A., Barthelemy, M. and Stanley, E. 2000. Classes of small-world networks, *Proceedings of the National Academy of Sciences* **97**(21): 11149–11152.

Bäck, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, first printing edn, Oxford University Press.

Crepinsek, M., Liu, S.-H. and Mernik, M. 2013. Exploration and exploitation in evolutionary algorithms: A survey, *ACM Computing Surveys* **45**: Article 35.

Dorogovtsev, S. and Mendes, J. F. 2002. Evolution of networks, *Advances in Physics* **51**(4): 1079–1187.

Dorogovtsev, S. and Mendes, J. F. 2003. Evolution of networks: From biological nets to the internet and www, *Physics Today* **57**.

Eiben, A., Hinterding, R. and Michalewicz, Z. 1999. Parameter control in evolutionary algorithms, *IEEE Transactions on Evolutionary Computation* **3**(2): 124–141.

Fogel, D. B. 1998. Unearthing a fossil from the history of evolutionary computation, *Fundam. Inf.* **35**(1-4): 1–16.

Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, USA.

Jiang, Z., Liu, J. and Wang, S. 2016. Traveling salesman problems with pagerank distance on complex networks reveal community structure, *Physica A: Statistical Mechanics and its Applications* **463**: 293 – 302.

Kannan, R. and Monma, C. L. 1978. On the computational complexity of integer programming problems, *in* R. Henn, B. Korte and W. Oettli (eds), *Optimization and Operations Research*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 161–172.

Laporte, G. 1992. The traveling salesman problem: An overview of exact and approximate algorithms, *European Journal of Operational Research* **59**(2): 231 – 247.

Lawrence J. Fogel, A. J. O. and Walsh, M. J. 1966. *Artificial intelligence through simulated evolution*, Wiley, Chichester, WS, UK.

Liu, S., Feng, C., Hu, M.-S. and Jia, Z.-J. 2014. Community detection method of complex network based on aco pheromone of tsp, *in* D.-S. Huang, K.-H. Jo and L. Wang (eds), *Intelligent Computing Methodologies*, Springer International Publishing, Cham, pp. 763–770.

Osaba, E., Del Ser, J., Sadollah, A., Bilbao, N. and Camacho, D. 2018. A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem, *Applied Soft Computing* **71**.

Papadimitriou, C. and Steiglitz, K. 1982. *Combinatorial Optimization: Algorithms and Complexity*, Vol. 32.

Precup, R.-E. and David, R.-C. 2019. Nature-inspired optimization algorithms for fuzzy controlled servo systems, *Butterworth-Heinemann* .

Purcaru, C., Precup, R.-E., Iercan, D., Fedorovici, L., David, R. and Dragan, F. 2013. Optimal robot path planning using gravitational search algorithm, *International Journal of Artificial Intelligence* **10**: 1–20.

Rechenberg, I. 1973. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Problemata, 15, Frommann-Holzboog.

Schwefel, H. 1976. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: Mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategie*, Interdisciplinary Systems Research, Birkhäuser Basel.

Sheng, L., Chen, F. and Wang, H. 2016. A Review on complex network dynamics in evolutionary algorithm, *2016 IEEE Trustcom/BigDataSE/ISPA* pp. 2221–2226.

Storn, R. and Price, K. 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization* p. 341–359.

Triana, J., Victor, B. and Garcia, A. 2020. *Sistema de control para computación evolutiva basado en redes complejas*, Universidad Simon Bolivar.

Vikhar, P. A. 2016. Evolutionary algorithms: A critical review and its future prospects, *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 261–265.

Watts, D. and Strogatz, S. 1998. Collective dynamics of ‘small-world’ networks, *Nature 393* p. 440–44.

Wei, F., Chen, W., Hu, X. and Zhang, J. 2019. An empirical study on evolutionary algorithms for traveling salesman problem, *2019 9th International Conference on Information Science and Technology (ICIST)*, pp. 273–280.

Zelinka, I. 2004. *SOMA — Self-Organizing Migrating Algorithm*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 167–217.

Zelinka, I. 2011. Investigation on relationship between complex networks and evolutionary algorithms dynamics, *AIP Conference Proceedings* **1389**: 1011–1014.

Zelinka, I. n.d.. IWCFTA2012 Keynote Speech III On Close Relations of Evolutionary Dynamics , Chaos and Complexity, *2012 Fifth International Workshop on Chaos-fractals Theories and Applications* p. 10624.

Zelinka, I., Davendra, D., Roman, S. and Roman, J. 2011. Do Evolutionary Algorithm Dynamics Create Complex Network Structures ?, *Complex Systems* **20**(2): 127–140.

Zelinka, I., Davendra, D., Snášel, V., Jašek, R., Šenkeřík, R. and Oplatková, Z. 2010. 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.