This article can be cited as J. Triana, V. Bucheli and O. Solarte, Knapsack Problem Solving Using Evolutionary Algorithms Guided by Complex Networks, International Journal of Artificial Intelligence, vol. 20, no. 1, pp. 136-146, 2022.

Knapsack Problem Solving Using Evolutionary Algorithms Guided by Complex Networks

Joshua Triana[1], **Victor Bucheli**[2] **and Oswaldo Solarte**[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
oswaldo.solarte@correounivalle.edu.co

ABSTRACT

The knapsack problem is a well-known optimization problem solved using evolutionary computation. In this work, we present that the population of the evolutionary computation can be represented as complex networks, as follows: a graph where nodes represent individual solutions and the links represent the crosses among solutions. Furthermore, to improve solutions, new links among nodes of the population are created, thus, the algorithm is guided and the population dynamics of the evolutionary algorithm to solve the knapsack problem. We evaluate that one strategy guided by a complex dynamic network yields better results than a traditional algorithm. The results show that the model proposed improves convergence compared to the traditional solution and has shorter execution times. Strategies based on small-world networks are promising in the experiments compared to those using the traditional evolutionary strategy.

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

1 Introduction

The optimization problems commonly have been solved by evolutionary algorithms, in which the evolutionary computation is related to the population of solutions and crosses between

them (Fogel, O. and Walsh, 1998). However, the evolutionary solutions have well-known problems such as long run time or premature convergence, and others. Thus, to improve the performance of the evolutionary approach, it is explored control mechanisms for the parameters in order to improve the quality of the solutions and convergence (Eiben, Hinterding and Michalewicz, 1999; Precup, Hedrea, Roman, Petriu, Szedlak-Stinean and Bojan-Dragos, 2021; Precup, David, Roman, Szedlak-Stinean and Petriu, 2021).

In this paper, we study the knapsack problem and apply an evolutionary algorithm guided by complex networks. The Knapsack is a combinatorial optimization problem, named as given a set of items, for each one the weight and a value are given, the restrictions are as follows: the total weight is less than a given limit, and the total value is as large as possible. Then, the maximum number of items to include in the collection is searched. The name of the knapsack problem is related to the limitation of a fixed-size knapsack, which must be filled with the most essential stuff.

The knapsack problem is an integer program with one constraint (Belegundu and Chandrupatla, 2011). It is a common problem in areas such as optimization or evolutionary algorithms. The knapsack problem can be described as with a given a set of integers $\{a_1, a_2, ..., a_n\}$, and a given target number, finding a subset that adds up precisely to the target. For instance, the set $\{1, 2, 5, 9, 10\}$ there is a subset that adds up to 10 but not 11.

According to (Triana, Victor and Garcia, 2020a; Triana, Victor and Garcia, 2020b) complex networks can guide the interaction between solutions in evolutionary computation; the complex networks improve the overall quality of the solution. Other works related to this topic are presented in (Sheng, Chen and Wang, 2016). This work presents the implementation of this approach to the knapsack problem. Thus, the complex networks represent a graph model of solutions, where it generates a complex structure of relations between solutions (Zelinka, Davendra, Snášel, Jašek, Šenkerík and Oplatková, 2010). In other contexts, these networks have been helpful to understand many systems from the relationship of their individuals and their dynamics (Zapata, Perozo, Angulo and Contreras, 2020), for instance, internet, citation networks, economy, etc (Dorogovtsev and Mendes, 2003).

This work presents an evolutionary algorithm guided by a complex network. The selection step of the evolutionary algorithm is made by using an underlying network that has properties of complex networks. A comparative analysis of the proposed model and the original model is presented, so a measure of the proposed model's impact can be done. The results could help to advance in the idea of including the complex network dynamics in optimization algorithms to improve its overall performance.

Chapter 2 is presented the relation between complex networks and evolutionary computing. In Chapter 3, the knapsack problem and its optimization problem are described. In Chapter 4, the methodology of the proposed strategy is explained. In Chapter 5, the experimental results are

shown. Finally, in Chapter 6, we discussed the conclusions of this research and some possible directions for future work.

2 Complex networks in evolutionary computing

Recent research in artificial intelligence show promising results in modeling signatures in expert systems (Pozna, Precup and Pârvan, 2014), artificial neural networks applied to decision making in healthcare (Albu, Precup and Teban, 2019), machine learning approach to classify pedestrians events (Ahmed, Brickman, Dengg, Niklas Fasth and Norman, 2019) and Tensor product-based model transformation approach to tower crane systems modeling (Hedrea, Precup, Roman and Petriu, 2021).

Complex networks can be represented as graphs where links connect their nodes. The links can be directed or not directed and weighted or unweighted. In an evolutionary algorithm, the solutions are not connected, and the information about crosses is lost. The crosses between solutions are commonly random, and they are not guided or restricted to obtain better results. This work proposes an evolutionary algorithm guided by complex networks to solve the knapsack problem.

The complex networks used in some researches are known as Small world networks (Amaral, Scala, Barthelemy and Stanley, 2000). The network structure is characterized by the shortest average distance between a pair of nodes, proportional to the natural logarithm of the number of nodes. The clustering coefficient is enormous, indicating that the network links tend to be strongly connected.

According to SOMA (Zelinka, 2004) and Differential Evolution (Storn and Price, 1997), evolutionary algorithms may have non-trivial relationships between individuals or solutions. The models use the father/child offspring relationships as networks, where the shortest average path length property with small values and power-law type distributions are highlighted. The author proposes a future direction to use the properties of this type of network as an improvement. The authors applied some updates to the algorithms studied (Zelinka, 2011; Zelinka, Davendra, Roman and Roman, 2011).

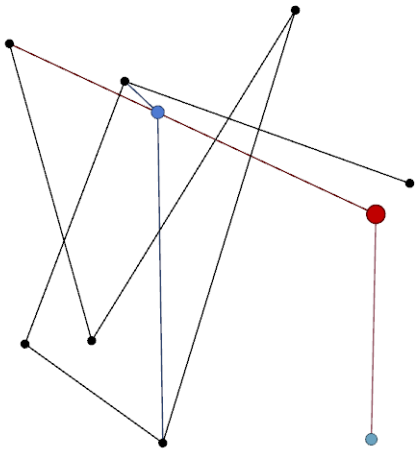

Figure 1: Complex networks in evolutionary computation

The main idea of the previously proposed models is to link each individual in a population of solutions to a node in a network. When an offspring (creation of a new individual) between some existing solutions occurs, a new edge between the nodes related to the existing solutions is created in the network. As shown in Figure 1 the red node has two connections, meaning that these two neighbors were involved in an offspring creation with the red node. The resulting network in these iterations have properties of complex networks, which were the direct results of the researches.

A research where a review on complex networks in evolutionary computation has been made (Sheng et al., 2016). The author indicates how the degree of distribution could have a relationship with premature convergence, the average path length between the nodes to determine the emergence of a complex network, clustering coefficient could also be used to analyze the individual in each iteration and use the property of centrality between the nodes to find the best individual. These studies, although promising, lack implementation, so they are only conceptually presented.

The relationship of complex networks in evolutionary computing in previous research given in this chapter is key to understand the proposed model's inspiration and new contributions in Chapter 4.

3 Optimization problem

The knapsack problem (KP) can be defined as follows: Given a set of n items with profit $\boldsymbol{P_i}$, weight $\boldsymbol{W_i}$, and the capacity value **C**, then the objective is to select a subset of n such that the total profit of the selected items is maximized and the total weight does not exceed **C** (Lazarev, Salnikov and Baranov, 2011). Next equations define the KP problem mathematically. Each $\boldsymbol{X_i}$ represents the number of item i to include in the Knapsack.

$$\sum_{i=1}^{n} \boldsymbol{P_i X_i} \tag{3.1}$$

$$\sum_{i=1}^{n} \boldsymbol{W_i X_i} <= \boldsymbol{C}\ and\ \boldsymbol{X_i}\ \in\ \{0,1\} \tag{3.2}$$

The KP problem has been of interest to the research community for many years. This fact is due to it representing different practical real-life situations (Assi and Haraty, 2019). Knapsack problems appear in the actual life decision-making problems specified by a resource allocation and a cost constraint.

The KP is known to be a Not Polynomial (NP) problem. It cannot be solved in polynomial time by a deterministic algorithm. Several approaches have been used: Greedy, Dynamic programming, Evolutionary computation and Greedy, Dynamic programming and Evolutionary computation.

The idea of the greedy approach is to start with an empty knapsack and go through the items in this decreasing order of efficiencies adding every item under consideration into the Knapsack if the capacity constraint is not violated. However, the Greedy approach provides a simple way that does not provide an optimal solution.

On the other hand, Dynamic programming approaches have been used extensively to solve the KP problem. (Martello and Toth, 1984) proposed a general approach combining dynamic programming and Deep first search to address the KP problem. Several studies using dynamic programming to deal with The KP have been proposed (Chebil and Khemakhem, 2015; Figueira, Paquete, Simões and Vanderpooten, 2013).

Evolutionary algorithms have also been widely applied for a range of stochastic optimization problems, such as the knapsack problem. In (Xie, Harper, Assimi, Neumann and Neumann, 2019) is proposed an Evolutionary approach to solve the KP problems using Chance-Constrained techniques. The authors argue that applying a chance-constrained to the Knapsack problem provides a clear benefit compared to its single-objective formulation. In (Shah, 2019) proposes an evolutionary algorithm capable of delivering optimum solutions within a reasonable amount of computational duration.

Although the above proposals have been shown promising results to deal with KP problem, these results can be improved or addressed by other approaches. Complex networks can be integrated with evolutionary algorithms to deal with the KP problem. Moreover, Complex networks have not been explored yet to deal with this problem.

In this chapter, the KP problem and the optimization problem are explained in order to clarify its boundaries and limitations. This required background is useful for understanding the problem the proposed model in chapter 4 aims to improve.

4 Methodology

The knapsack problem is an integer program with one constraint (Belegundu and Chandrupatla, 2011). It is relatively simple since all coefficients and objectives are non-negative. It can be summarized as determining the highest ratio of objective coefficient/constraint coefficient per variable. The knapsack problem can be represented as a 0-1 problem representing the data, variables, constraints and function to be maximized as follows:

Data:

- n number of objects;
- a_j the weight of object j
- c_j the utility of object j
- b knapsack capacity

Variables

$$x_j = \left\{ \begin{array}{c} \text{1 if the object is put in the knapsack} \\ \text{0 otherwise} \end{array} \right\}$$

Constraints: not exceed this capacity

$$\sum_{j=1}^{n} a_j x_j <= b \tag{4.1}$$

Maximized this utility

$$Z = \sum_{j=1}^{n} c_j x_j \tag{4.2}$$

There is a set of items that have a utility, and there exists a limitation on the number of items can carry according to Equation 4.1. The problem consists of selecting a subset of items to maximize the sum of the utilities and never exceed the carrying capacity of the carrier.

The knapsack problem has been studied for a lot of time, with firths works dating 1897. The decision problem form of the knapsack problem is NP-complete and there is no known algorithm both correct and fast (polynomial-time) in all cases. There is also a fully polynomial-time approximation scheme, which uses the pseudo-polynomial time algorithm (Shachnai and Tamir, 2007; Lai, 2006).

In this work, we propose an evolutionary algorithm guided by complex networks generated from the relationships between their individuals or solutions. The relationship detected in this strategy is the father-son relationship, established when a solution serves to create or improve another solution. This interaction happens in the reproduction process where, through one selection technique, two solutions generate a new solution by recombining their genes. With this

idea in mind, it is possible to create networks based on the relationship of offspring of solutions.

In the complex networks analysis, the relationships including social entities such as friendships among individuals or communication in a group. The complex networks analysis discovers cohesive groups or communities, or identifies similar individuals.

According to the lines below, the solutions of one evolutionary algorithm are similar to the social entities, so we proposed a method to map the interactions in an evolutionary algorithm to the complex network. Thus, the patterns of communities and paths in the graph are used to guide the process of crossing. This change in the original evolutionary algorithm improves the phase of selection and crossing, which is commonly random.

In this work, the crossing of solutions is represented as an adjacency matrix, which is also known as a connectivity matrix. Furthermore, an entry a_{ij} of the adjacency matrix is the number of undirected edges from solution v_i to solution v_j. The pseudo-code 1 implements the process to produce the adjacency matrix to the evolutionary algorithm. For a graph on vertices, the adjacency matrix has two dimensions. For an undirected graph, the adjacency matrix is symmetric, and it must have zeros on the diagonal.

In this paper, it is generated the adjacency matrix accordingly to the small world model, this is known as Watts and Strogatz model (Watts and Strogatz, 1998) and the new form to related solutions in the cross-process is guided by the Watts and Strogatz model. The underlying complex networks improved the result and the competitive advantage of the algorithm, and it is the same inspiration for the model (Triana et al., 2020a).

Algorithm 1 Pseudocode of proposed method during evolutionary selection stage

```
Population ← pseudo randomly generated population of size n
Small world network ← Each node is associated to a unique agent of population
while stopCriterionNotReached do
    for i ← 1 to n do
        agentA ← i agent of population
        for i ← 1 to k do
            neighborsA ← set of all agentA's neighbors
            if length(neighborsA) > 0 then:
                agentB ← choose one agent from neighborsA by a given criterion
                Reproduction Stage with agentA and agentB to generated a new agentC
                Mutation Stage with agentC
                if (agentC.fitness) > (agentA.fitness) then:
                    Replace agentA with agentC in population
                end if
            end if
        end for
    end for
end while
```

The process of the proposed algorithm 1 is as follows: initially, a population of solutions is created to initiate the evolutionary algorithm, and a small-world network with the same number of nodes as the size of the population is created too. A random association of an individual from the population of solutions is made with a node in the network to start the algorithm iteration, where each solution from the population is taken, and a neighbor of its associated node is selected to reproduce. Furthermore, the reproduction of solutions is through the network of the small world, and the relationship between the nodes is 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. For a flow diagram of the model, see (Triana et al., 2020b).

Furthermore, to improve the solutions and not fall into premature convergence, the rewiring process was integrated into the algorithm in the same form as Watts and Strogatz (Watts and Strogatz, 1998). Thus, each link of each node X_0 to another node X_{00} with a certain probability (between 10 and 30%), breaks the link and joins to another node X_{000}. In sum, for each iteration, the network changes slightly and adds dynamism to the evolution of the solutions. As explained above, each node can select a candidate to reproduce between neighbor nodes. The node into the neighbor is selected according to:

- Small World: random selection of any node of the neighbor.
- SW Best: selection of the neighbor through the roulette method, it is to favor the selection of the neighbor with better performance (fitness)

- SW High Degree: selection of the neighbor with the highest grade

In (Triana et al., 2020a; Triana et al., 2020b) was proposed a model where an underlying small world complex network is applied to an evolutionary computation strategy to solve test optimization problems, where each individual in the population is associated with a node and it's limited to reproduce with its networks neighbors. In this investigation, we test the proposed model using a real-life NP-Problem, the Knapsack Problem, and analyze the model's scalability over fitness, time, and iterations.

The proposed model explained will be tested in the next chapter to validate the central hypothesis and make a discussion of obtained results.

5 Results

In computer science, the Knapsack problem is an optimization challenge. In this problem, an analogous situation arises when filling a backpack, which supports maximum weight and can be put all or part of a set of objects within it, each with a specific weight and value. The objects placed in the backpack should maximize the total value without exceeding the maximum weight (R.M., 1972).
The relevant parameters for the experiments are:

- Population size: 100
- Mutation rate: 0,1
- Stop criterion: 200 Iterations
- Knapsack capacity: 30 objects

In each experiment, 200 iterations were made. The results are the averages of 500 experiments in each iteration. Given the poor results of the No SW strategy is omitted from the figures.

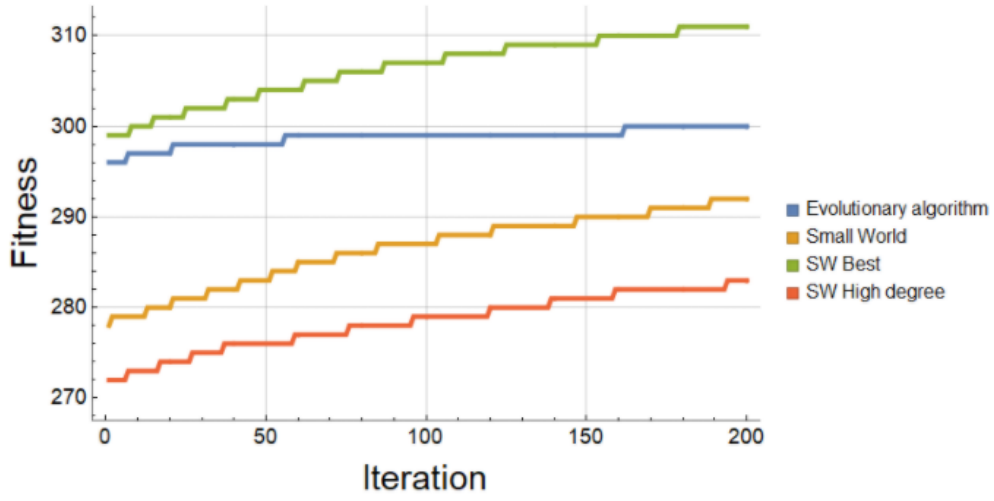

Figure 2: Average performance in Knapsack problem

The results were similar to those obtained in previous investigations (Triana et al., 2020b). The worst variant was SW High Degree, followed by the base model Small world, then the traditional evolutionary, and finally the SW Best variant as the best. The tests in the proposed algorithms indicate that the model with neighbor selection with better fitness (SW Best) could be an excellent alternative to the original strategy.

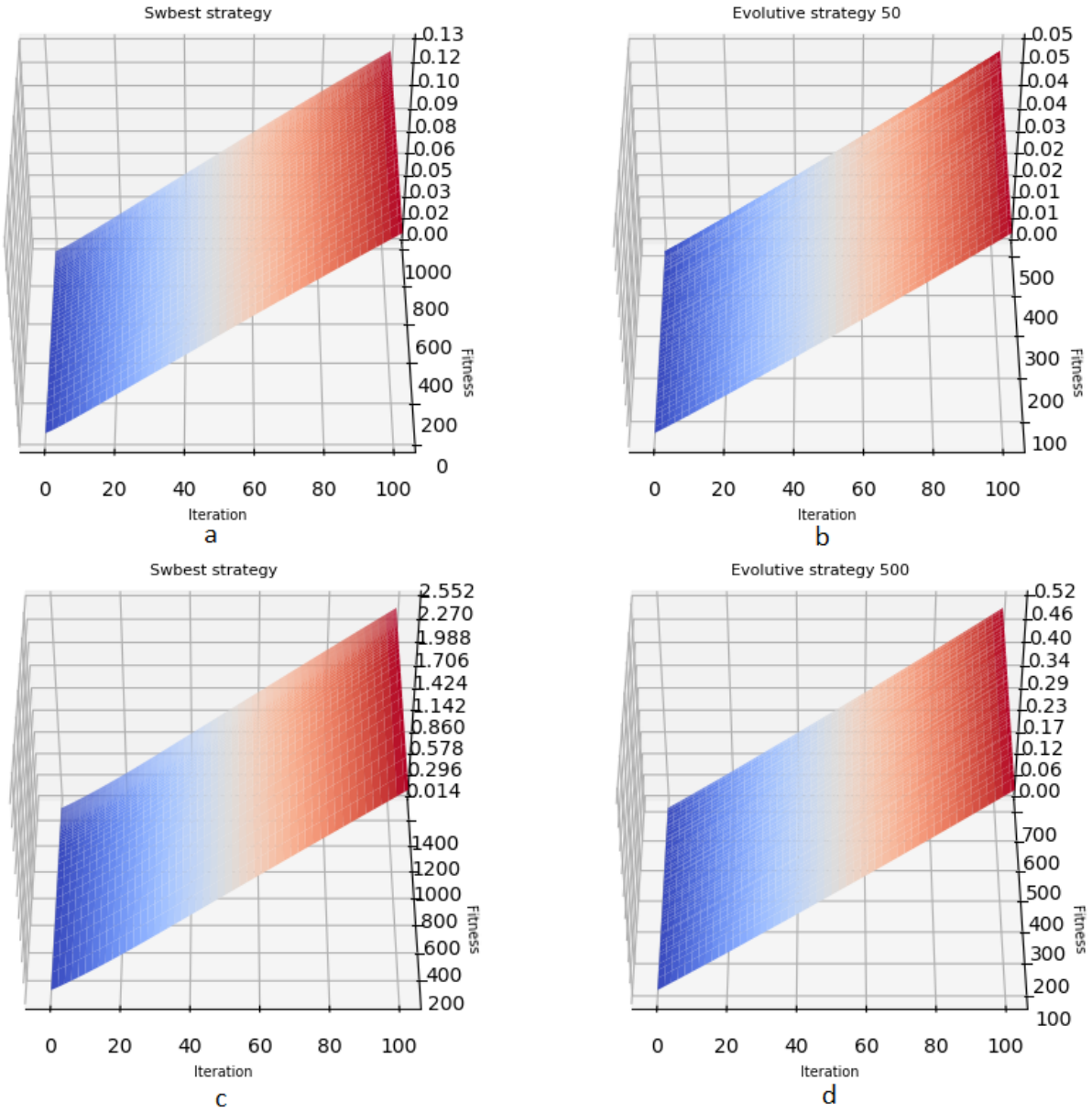

Figure 3: Iterations, time and average fitness of 200 experiments in Knapsack with 50 and 500 population size

Finally, the impact of the computational cost of the model in counterpart with the original strategy is analyzed. Figure 3.a shows the average of iteration,fitness and execution time of SWBest model over 50 experiments in the Knapsack Problem, and Figure 3.b shows the same for the traditional evolutionary strategy. It can be seen how the fitness in better in the proposed model, but the computational cost in seconds of execution is higher than the original strategy. The proposed model has aggregate computational costs over the entire process. We also wanted to see how the behavior exchanges when the number of experiments rises. Figure 3.c shows the average of iteration, fitness and execution time of SWBest model over 500 experiments in the Knapsack Problem, and Figure 3.b shows the same for the traditional evolutionary strategy. The same pattern of advantage of the model in terms of computation time and linearity over iterations, time and fitness.

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

6 Discussion

It is possible to obtain networks with small world characteristics through the dynamics of the relations between the individuals of some evolutionary algorithms. These relations of father-son descent modeled as graphs resemble the Watts-Strogatz model.

The alternative selection of matching neighbors with a greater degree tends to be the worst model, often worse than the original strategy. The network's topological properties are not taken into account to guide the control mechanism, which is why it is understood that it does not provide information relevant to the system's evolution.

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 using the evaluation factor (the fitness to be improved) among the individuals of the population, which could explain this behavior.

The obtained results are similar to previous research (Triana et al., 2020a; Triana et al., 2020b), where the proposed model has better results than the traditional strategy. In this research, a widely known optimization problem has been tested compared to previous research using test optimization functions(Triana et al., 2020a). Compared to a simple model with few optimization parameters, such as the number of cities in TSP (Triana et al., 2020b), the results obtained in this research reinforce the potential applications of the proposed model.

The Knapsack Problem has been solved with some alternatives using evolutionary computation. These solutions (Zitzler and Thiele, 1999; Zhang and Li, 2007) may be better in performance and execution time than the proposed model in this work. The proposed model is an excellent alternative to the traditional evolutionary strategy but may not be the best alternative overall possible models.

Future works involve complexity analysis of the complex networks and the optimization problem with the proposed model. This analysis may be essential to understand the consequences of applying the adjacent complex network in an optimization problem and why it improves fitness and time performance.

Acknowledgment

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

References

Ahmed, M. U., Brickman, S., Dengg, A., Niklas Fasth, M. M. and Norman, J. 2019. A machine learning approach to classify pedestrians' events based on imu and gps, *International Journal of Artificial Intelligence* .

Albu, A., Precup, R.-E. and Teban, T.-A. 2019. Results and challenges of artificial neural networks used for decision-making and control in medical applications, *Facta Universitatis, Series: Mechanical Engineering* **17**(3): 285–308.

Amaral, L., Scala, A., Barthelemy, M. and Stanley, H. 2000. Classes of small-world networks, *Proceedings of the National Academy of Sciences of the United States of America* **97**: 11149–52.

Assi, M. and Haraty, R. A. 2019. A survey of the knapsack problem., *International Arab Conference on Information Technology, Werdanye, Lebanon. IEEE* .

Belegundu, A. and Chandrupatla, T. 2011. Dynamic programming. in optimization concepts and applications in engineering, *Cambridge: Cambridge University Press* pp. 385–399.

Chebil, K. and Khemakhem, M. 2015. A dynamic programming algorithm for the knapsack problem with setup., *Computers and Operations Research* pp. 40–50.

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

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

Figueira, J. R., Paquete, L., Simões, M. and Vanderpooten, D. 2013. Algorithmic improvements on dynamic programming for the bi-objective, *Computational Optimization and Applications* pp. 97–111.

Fogel, L. J., O., A. J. and Walsh, M. J. 1998. Artificial intelligence through simulated evolution, *Wiley-IEEE Press* pp. 227 – 296.

Hedrea, E.-L., Precup, R.-E., Roman, R.-C. and Petriu, E. M. 2021. Tensor product-based model transformation approach to tower crane systems modeling, *Asian Journal of Control* **23**(3): 1313–1323.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.2494*

Lai, K. J. 2006. The knapsack problem and fully polynomial time approximation schemes (fptas), *18.434: Seminar in Theoretical Computer Science* .

Lazarev, A., Salnikov, A. and Baranov, A. 2011. Graphical algorithm for the knapsack problems, *in* V. Malyshkin (ed.), *Parallel Computing Technologies*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 459–466.

Martello, S. and Toth, P. 1984. Mixture of dynamic programming and branch-and-bound for the subset-sum problem., *Management Science* pp. 765–771.

Pozna, C., Precup, R. and Pârvan, V. 2014. Applications of signatures to expert systems modelling, Vol. 11, Acta Polytechnica Hungaria.

Precup, R.-E., David, R., Roman, R.-C., Szedlak-Stinean, A.-I. and Petriu, E. 2021. Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using slime mould algorithm, *International Journal of Systems Science* pp. 1–16.

Precup, R.-E., Hedrea, E.-L., Roman, R.-C., Petriu, E. M., Szedlak-Stinean, A.-I. and Bojan-Dragos, C.-A. 2021. Experiment-based approach to teach optimization techniques, *IEEE Transactions on Education* **64**(2): 88–94.

R.M., K. 1972. Reducibility among combinatorial problems, *Complexity of Computer Computations. The IBM Research Symposia Series* pp. 85–103.

Shachnai, H. and Tamir, T. 2007. Polynomial time approximation schemes - a survey, *In Handbook of Approximation Algorithms and Metaheuristics (Ed. Teofilo F. Gonzalez), Chapman Hall/CRC Computer and Information Science Series* .

Shah, S. 2019. Genetic algorithm for a class of knapsack problems, *ArXiv* p. 10–12.

Sheng, L., Chen, F. and Wang, H. 2016. A review on complex network dynamics in evolutionary algorithm, *IEEE Trustcom/BigDataSE/ISPA* p. 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. 2020a. Sistema de control para computación evolutiva basado en redes complejas, *Investigación e Innovación en Ingenierías, Barranquilla, Colombia* .

Triana, J., Victor, B. and Garcia, A. 2020b. Traveling salesman problem solving using evolutionary algorithms guided by complex networks, *International Journal of Artificial Intelligence* pp. 101–112.

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

Xie, Y., Harper, O., Assimi, H., Neumann, A. and Neumann, F. 2019. Evolutionary algorithms for the chance-constrained knapsack problem, *ArXiv* p. 338–346.

Zapata, H., Perozo, N., Angulo, W. and Contreras, J. 2020. A hybrid swarm algorithm for collective construction of 3d structures, *International Journal of Artificial Intelligence* **18**: 1–18.

Zelinka, I. 2004. Soma — self-organizing migrating algorithm, *Springer Berlin Heidelberg* p. 167–217.

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

Zelinka, I., Davendra, D., Roman, S. and Roman, J. 2011. Do evolutionary algorithm dynamics create complex network structures?, *Complex Systems* p. 127–140.

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

Zhang, Q. and Li, H. 2007. Moea/d: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* pp. 712–731.

Zitzler, E. and Thiele, L. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, *IEEE Transactions on Evolutionary Computation* pp. 257–271.