Graph-based comparison and analysis of metaheuristics

Maria Clara O. D. Ruas
Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
quietbrelax@ufmg.br

Mateus Bastos Soares
Department of Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
mateusbs22@ufmg.br

João Pedro A.F. Campos
Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
jpafcampos@ufmg.br

Michel Bessani
Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
mbessani@ufmg.br

***Abstract*—The comparison of metaheuristics is a challenge in the optimization community. While searching for a solution, metaheuristics can have different forms of exploring the search space and mapping the objective space to find the best solution. Often, the behaviour of these methods can be quite unclear for direct human interpretation, making the comparison and selection of available methods subjective. Using graph theory and graph-related metrics, we can improve the understanding of how different metaheuristics estimate the optimal solution and have clear metrics for performance comparison. This leads to a more efficient and explainable approach for choosing the most suitable method for each problem and scenario. In this paper, we present the application of a graph-based representation and analysis of metaheuristics' behaviour and discuss the interpretation of complex network metrics applied to compare the different metaheuristic methods, showing aspects relevant to the choice of which method is better for a specific problem. A case study is presented to illustrate the comparison between two metaheuristics. Results highlight how the graph-based comparison explains how the algorithms map the objective space in a visual and trackable manner, while the complex network metrics allow a direct comparison between the behaviour of the two different metaheuristics.**

***Index Terms*—metaheuristics, optimization, graph, search trajectory network, complex networks**

I. Introduction

Optimization problems and models have long become more and more complex, with the number of variables and dimensions increasing exponentially to mimic the multiple possibilities of the real world [1]. Moreover, the optimization methods created to solve these problems have also evolved to accompany the rising complexity, requiring more time, energy, and computer power. Naturally, with the increase of complexity, it becomes harder to visualize, compare, and understand the processes by which the methods find the optimal solution [2]. That hardship results in difficulties in comparing methods, whether in identifying methods' weaknesses and strengths or understanding how the methods explore the search space and map the objective spaces for each optimization problem.

On the other hand, the application of graphs to represent abstract concepts, complex models, and solutions, with the possibility of various interpretations, is widely investigated within the mathematical and engineering fields. Accompanied by the representation, graphs also carry many known metrics that can be applied to different contexts, such as colours for sorting, bipartition for conjuncts in functions and systems [3], or betweenness in social and engineering networks [4].

There are different proposed approaches for better understanding metaheuristic behaviour. Among them, the Search Trajectory Networks [5], which will be further discussed in the following sections, showed great potential by combining complex networks' representation and metrics to model the behaviour of metaheuristics. Analogously, the work in this paper aims to implement and analyse a comparison method of metaheuristics based on graph theory, focusing on ways to make solid statements about the methods and how they behave during the optimal solution estimation. The proposed method enables both individual analyses of metaheuristics and direct comparisons between multiple ones, all evaluated using a consistent set of metrics. In this work, we use the method to compare two metaheuristics for learning the structure of a Bayesian network from data. These two approaches were previously evaluated by Campos et al. [6], who applied statistical tests to demonstrate that their proposed method, MAGABN, outperforms a classical genetic algorithm (GA) across several datasets. Our graph-based analysis findings are compared with the key results found in their previous work.

Given the relevance of metaheuristics comparison, this manuscript aims to further the studies about interpretations and metrics using the search trajectory networks approaches already developed. Given a bigger emphasis on flexibility

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior- Brasil (CAPES)- Finance Code 001, the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

of analysis parameters and behaviour probability insights, this paper brings to light different paths to judge important aspects of a metaheuristic algorithm, including exploitation, exploration, robustness, and other features. Furthermore, the case study is brought to exemplify the method, presenting a deeper understanding of the algorithms' behaviour.

The remainder of this paper is organized as follows: Section II reviews approaches for metaheuristics comparison, including graph-based ones. Section III presents the proposed method, describing how to construct the graph representation and how to extract the complex networks' metrics. In Section IV, the case study is presented, and previous findings from statistical comparisons among the algorithms are stated. Section V describes and discusses the results, and Section VI concludes this manuscript.

.

II. Related Work

As stated in [2], explainable models gain more trust from users and decision makers. However, the challenges in comparing and stating which metaheuristic is better are an active research field. As analysed in [1], metaheuristics are often accompanied by real-world analogies and multiple classifications. However, the translation of these concepts into the actual space exploration is still blurry. As stated in the work, theoretically proving efficiency is not a simple task and usually relies on empirical results.

Works such as [7] highlight the ongoing problem in the comparison of metaheuristics. The lack of a deeper understanding about the behaviour of existing methods creates space for the repeated use of the same algorithms under different interpretations. The referenced paper critiques precisely this issue by questioning how one can confidently claim that a newly proposed algorithm is genuinely novel. This highlights the importance and urgency for the optimization community to develop theoretical foundations and standardized approaches for comparing algorithms.

A. Strategies for Comparison of Metaheuristics

Due to unique functionalities [8], different test-beds [9], difficulties in replicating the exact behaviour [9], or many other factors, the comparison of different algorithms is indeed a complex task. Many authors developed their works focusing on good practices for a more reliable comparison, such as common shared test-beds, parameter complexity analysis, solution quality, and runtime comparisons [8], [9]. However, for new proposals of methods or new problems, it is a challenge to systematically compare the new algorithm with existing ones. With so many affirmations of improvement, the lack of proper comparison raises doubts about the true superiority in performance [7].

To address that, different empirical standardized comparisons for metaheuristics were developed. In [10], a discussion is presented about adequate experimental protocols. The discussion focuses on the required sample sizes to perform a statistical comparison with desired statistical properties. In [11], factors such as sample size, number of repetitions, and number of instances are debated, showing ideal values to perform statistical hypothesis tests without highlighting minimal differences with sample sizes bigger. Following the work, (Sample size calculations) also utilized statistically principled methods to compare multiple algorithms. It is stated that experimental evaluation remains a central aspect for understanding metaheuristics. Even so, in the work, it is stated how the purely empirical analysis is highly criticized for the disconnect between theory and experimentation and the multiplicity of methodologies, which can cause reliability issues.

Another approach to understand high-dimensionality metaheuristics problems would be the usage of mathematical assets to turn the problem visible. For example, [2] uses dimensionality reduction (via PCA) to visualize and interpret metaheuristics. Even if this approach loses some information about the method during the dimensionality reduction, it brings the possibility to visualize said problems. The capability of visualizing is a great asset when it comes to explaining and understanding an optimization method [12].

In fields like software systems, complex networks are already used to analyse complexity and behaviour. For example, in [13], complex networks help software developers gain insights into code vulnerability, modularity, and structure. In optimization, [14] applies network measures to extract more than just the best solution, enabling better understanding and transfer learning. This approach also supports algorithm comparison by analysing behavioural differences.

A related line of work is the use of Local Optima Networks (LONs) for modeling continuous optimization landscapes. Here, local optima are nodes, and transitions are edges based on search dynamics, offering insights into algorithm behaviour, especially in multimodal problems. Inspired by LONs, Search Trajectory Networks (STNs) were later proposed [15] to further explore algorithm dynamics.

B. Search Trajectory Networks

Search Trajectory Networks (STN) is a method proposed to help analyse and visualize the behaviour of metaheuristics during the optimization process [5]. Since the problem of comparing metaheuristics and understanding their behaviour is still an open problem, STN come as a data-driven graph-based model, making it possible to apply complex networks' metrics onto the behavioural analysis as well as visualize it, even in problems with multiple dimensions. The objective of STN is to model the search behaviour of the algorithms and how they explore the solution space, in contrast to other methods that compare only the objective space or individuals [5]. The STN have the potential to differentiate algorithms not only by the high-level metaphors, but also by their behaviour, giving important insights about it, such as early convergence and the pattern of connections.

The concept of STNs involves selecting representative solutions at each step of the optimization process and defining locations in the search space that group similar solutions [5].

The trajectory is then formed by the sequence of these representative solutions in the order they are discovered. From this trajectory, a graph is built where nodes represent the defined locations and edges capture transitions between them. Once constructed, various complex network metrics such as node and edge counts, community structures, and centrality, can be applied to gain deeper insights into the algorithm's behaviour. The approach also supports alternative sampling and analysis strategies, offering flexibility for future exploration.

The continuous evolution of STN is noticeable, with further developments leading to WEB applications [16], [17] and studies applying it to multi-objective evolutionary algorithms [18]. When compared to Local Optima Networks, which represent a subset of solutions, the STN approach does not limit the nodes, often representing a given state or a solution status of the evolution of the algorithm during the optimization process [12].

III. Methodology

The method has two main components: graph construction from metaheuristic execution data, and analysis of the resulting graphs. During construction, options like granularity, visualization, and whether to represent one or multiple metaheuristics are defined. The analysis then applies relevant metrics to interpret and compare the graphs objectively. Both components are detailed in the following subsections.

A. Graph Construction

To successfully use graph theory to compare different types of metaheuristics, we first need to build an effective way to transform the trajectory of a metaheuristic execution into a graph. Keeping that in mind, we can separate the tasks to achieve the final graph into the data storage and retrieval, the definition of granularity, and the graph construction itself.

As detailed in our case study, we compare the performance of two metaheuristics on a Bayesian network structure learning task. Each algorithm was executed 20 times per dataset. During each run, the fitness (objective function value) of the best individual at each iteration was recorded. These values are used to map the optimization trajectory, with each point uniquely represented in the graph.

For numerical data, a useful strategy to manage precision is dividing the solution space by merging nearby points, in order to reduce redundancy. This allows control over what constitutes a meaningful improvement in the objective function versus what can be treated as equivalent. In the implementation, the solution space can be discretized as needed for the analysis or kept continuous. Once this step is completed, the data is processed and formatted to generate the graph.

When constructing the graph, the first step is the definition of all the unique points that will be the vertices. Each unique point present in any run after the prior definitions will be saved as a vertex in the graph. After the selection of all valid points, looking at the runs, it saves information about the special points, such as initial and final points of each run, points that are common between more than one run, and the best optimal point estimated on that batch of runs. These special points are selected for better visualization, with different colours and shapes, and for robustness analysis of the metaheuristic.

Finally, following the track of each run, all the directed connections between the unique visited vertices are built. After all the runs have been processed, we end up with a graph that represents all the paths from each initial point to the final point, and the respective connections. This graph can be visualized and further analysed with complex networks' metrics, which helps to understand how each of the different behaviours from each metaheuristic is reflected in the objective space.

B. Quantitative Graph Analysis

After the graph construction, we can start to analyse it and interpret the network metrics, relating them to metaheuristics' behaviours. Each metric can be used to raise insights about efficiency and behaviour in a specific manner. The main observation features are related to the explored region in the solution space, the robustness and consistency of the optimization process, the ability to escape from local optima, the capacity of the method to reach the estimated global optimal solution, and the capability to achieve better results for the objective function.

When comparing methods and implementations, it is essential to use consistent granularity. The first metric, *number of vertices*, reflects how many unique points the algorithm visited, being an indicator of exploration capacity, depending on the level of discretization.

The *subgraph simple path* counts distinct paths sharing vertices and leading to the estimated global optimum, revealing the algorithm's robustness in finding multiple routes to the best solution. *Betweenness* identifies key connection points in the graph, offering insights into how the algorithm balances exploration and exploitation based on the centrality of these nodes.

The *probability of improvement* measures the likelihood of achieving better solutions at each step, shedding light on each algorithm's potential to escape local optima. The *number of common points* shows how often an algorithm revisits or shares locations in the search space, either within its own runs (indicating convergence patterns) or across different algorithms (revealing similarity or divergence in behaviour).

The *passage through local optima* captures an algorithm's ability to escape suboptimal solutions by identifying runs that share points but proceed to better outcomes. Lastly, the *number of self-loops* indicates how often the algorithm stalls without improving the objective function, correlating with wasted computational effort or minor changes based on the defined granularity. All metrics are summarized in Table I.

IV. Case study

In the case study here proposed, we analyse an application of metaheuristics for Bayesian Networks Structural Learning, a work thoroughly developed in [6]. The scripts developed and used for the case study can be found on the repository "Graph Based Analysis" for reproducibility, as well as the datasets

TABLE I
DESCRIPTION OF MAIN METRICS COLLECTED

Feature	Description
Number of vertices	Unique vertices visited by each algorithm
Sub-graph Simple Path	Different paths able to reach best optimal point estimated
Betweenness	Relationships between relevant vertices
Improvement probability per iteration	Histogram indicating the probability of improvement
Number of Common Points	Number of points shared between runs or algorithms
Passage through local optimal points	Number paths that improve an final point of another path
Number of Self-loops	Number of times the algorithm didn't improve given the periodicity of data collection

used. For the case study, a PC with an AMD Ryzen 5-5500U processor and 12GB of RAM was used. The metaheuristics execution data was previously collected, and the STN graph construction process required less than one minute of computational time. Further in this section, a summary of the problem will be presented, alongside the metaheuristics that will be compared in this paper and their respective performances analysed in previous works.

A. Problem Description

Bayesian Networks (BN) are probabilistic graphical models that represent how multiple random variables relate to each other in a compact manner. These networks can be learned from data, building connections among the variables and finding patterns that may not be explicit. The concept is useful, as the knowledge constructed can be auxiliary in critical decision-making scenarios, such as the treatment of diseases [19], maintenance and monitoring of complex systems [20], and forecasting load consumption in energy systems [21].

Building the BN requires mainly two steps: first, learning the structure, and then learning the model parameters [6]. The structure embeds the conditional dependence among random variables, and the parameters are the conditional probabilities of the learned relations. With the increasing growth in the collection of data around the world, there is a vivid necessity to be able to extract important information from all the data gathered. However, constructing the right BN structure representing the dependencies between variables is not trivial. With the increase of variables, the construction of the network becomes more and more challenging, requiring more computational power, and possibly still not getting optimal networks.

Considering the BN's structure learning as a combinatorial NP-hard problem, many algorithms and approaches to solve such a problem have been proposed. Therefore, metaheuristics such as evolutionary algorithms are a popular method to try to search for a relatively good network structure. To achieve that goal, however, it is important to choose methods that have good stability and robustness, the ability to escape local optima, and that overall show a great balance between resources used and solution quality.

B. Applied metaheuristics

One of the many proposed approaches for BN structure learning involves using metaheuristics. In this case study, it will be used as base data for our graph-based method, the best solutions gathered along the optimization process of the Genetic Algorithm (GA) and Multi-Agent Genetic Algorithm for Bayesian Network Structural Learning (MAGABN). The classical Genetic Algorithm mimics the concepts of genetic evolution, treating the candidate solutions as individuals that suffer mutations and crossover, improving aspects of the solution until convergence. A common occurrence in GA is the early convergence, which can result in missing relationships or the creation of ghost connections in the BN, where the structure states a relationship between variables that does not correspond to the real world. That challenge shows the importance of using algorithms to build a good structure for the BN.

Proposing a balance between premature convergence and exploration of different areas in the search space, in [6], it is proposed to usage of a known metaheuristic method, MAGA. Originally, the MAGA method was proposed for global numerical optimization [22]. Similarly to the classical GA, the method uses a multi-agent system. In the method, there is a cooperative improvement between neighbouring agents via crossover, combining the knowledge of each solution they represent. The MAGABN resulted in a superior performance that will be discussed next.

C. Statistical Comparison Findings

The BN structure learning solution previously described used the BIC score as the objective function. It measures the fitness of an agent, where the agent represents a binary vector that can be decoded into a BN structure. Additionally, in [6], the performance of both algorithms was statistically compared using two topological metrics that evaluate how closely the learned Bayesian network matches the ground truth structure. The two metrics are the Structural Learning Factor (SLF) and the Topological Learning Factor (TLF).

The former, SLF, quantifies the proportion of correctly identified arcs in the learned structure compared to the true network, emphasizing structural accuracy. The latter, TLF, captures how well the learned topology preserves the broader connectivity patterns of the ground truth, including indirect dependencies and overall graph shape. Together, these metrics complement the BIC score by offering a structural and relational perspective on the quality of the learned networks.

As key findings, it is shown that MAGABN tends to outperform the classical GA, presenting higher values of SLF for the benchmark problems. The superiority of the method prevails as the difference in SLF values shows statistical significance, with average SLF higher than GA and other methods, while also having the lowest standard deviation. These statistical results indicate a more stable and consistent performance, a higher

adaptability to diverse problem instances, scalability to larger networks, and overall better structure learning capability. The aim of the case study presented in this paper is to confirm these conclusions, checking if they can be reached by using the presented method on the two metaheuristics, and comparing the ways each of them searches for solutions. By that, it puts into light not only the final result, but also how the methods reached the final result.

V. Results

In this section, the key results from the methodology are shown. The BIC score was used as the primary representative solution, however, the structure and variables are also viable representative solutions, as long as the locations are properly assigned. With the behavioural graphs constructed, insights about relations between the data collected and the behavioural characteristics of each metaheuristic can be found. This section was separated into different aspects of analysis, regarding the granularity, probability of improvement, robustness, search space exploration, redundancies, early convergence, and capabilities to surpass local optima.

The granularity, or how the search space is discretized for the analysis, results in changes in the graph representation for the same metaheuristic data. In Figure 1, the effect of different granularity values are illustrated for the runs of the algorithm MAGABN for the benchmark problem Child. We can observe that the higher the granularity, the fewer nodes will be displayed, resulting in a combination of different solutions due to the small differences in objective function value, which also results in more common nodes between executions.

The rate at which nodes are combined as the granularity is increased indicates the ability of the algorithm to improve the quality of solutions. If the rate is too high, the algorithm does not show significant objective function improvements given the same number of steps. When comparing two or more metaheuristics using the same values for granularity, this rate indicates that the metaheuristic with a higher grouping rate consumes more energy, as it needs more iterations to find relevant improvements to the solution. In Figure 2, the rates of nodes grouping of the MAGABN and GA for the same benchmark problem are displayed together, showing the MAGABN with a lower grouping rate growth, reflecting that less energy is necessary to achieve better solutions, concluding that MAGABN overall presents a higher diversity in the nodes visited, meaning significant improvements in the objective function.

In Figure 3, a complementary cumulative histogram for objective function improvements at all iterations of the MAGABN and GA executions is presented. It allows an evaluation of the different magnitudes of improvement probabilities, reflecting the behaviour of each algorithm in generating higher objective function improvements and escaping local optima. We can notice in Figure 3, that both curves decrease with a similar shape. Since the MAGABN uses some mechanisms of the GA, similarities like this are expected. However, it can be seen how GA's curve decreases faster.

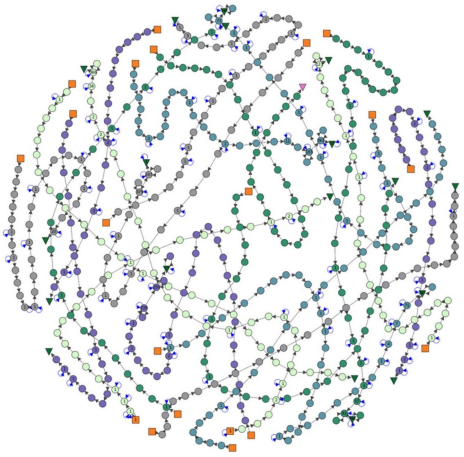
On the other hand, MAGABN presents overall higher values of probability and magnitudes of improvements. That can be interpreted as MAGABN having higher chances of improvement per iteration, reflecting a greater capability of escaping local optima. It is also related to the robustness of each metaheuristic. The higher the probability at the tails, the higher the probability of generating greater improvements and escaping local optima at different executions. We can notice that the MAGABN probabilities are higher for all magnitudes of improvement than the GA ones.

Figure 4 illustrates an example of a graph generated for the two metaheuristics at once in the Asia benchmark problem. We can notice, by the number of nodes represented as red circles, that the two metaheuristics have shared points, and they can often pass through the same points. However, the MAGABN trajectory has a path towards the best optima, it also presents the behaviour of passing through the final points of other runs. In contrast, GA presented more self-loops, indicating a higher chance of being trapped in local optima than MAGABN, which has a higher probability of escaping.

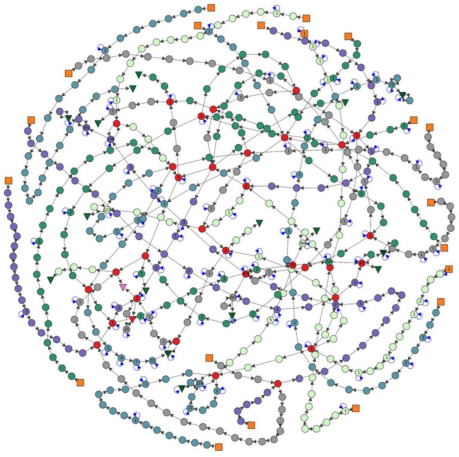
The Table II summarizes quantitative metrics for the MAGABN and the GA for the Child benchmark problem under different granularity values. Starting from the average path length, the lower its value, the more interconnected the graph is. In this context, the interconnections indicate whether the algorithm has a consistent performance on different executions, since the nodes have shortest paths to each other. MAGABN shows consistently lower values than GA for the two granularity levels, meaning that the networks constructed by the MAGA runs are more tightly connected. The number of common points, similarly, creates a certain debate about robustness and exploration. While a higher number of common points can mean consistency in the search, it also indicates a lack of diversity in the exploration, which is prejudicial to the optimization process. Therefore, this metric can be used as supplementary information to the average path length.

In the Table II, GA presents higher values of path length and also higher values of common points. The combination of those two means that while the runs pass through the same points more often, the network is still not as interconnected as MAGA. In that way, one conclusion might be that the common points shared by GA are not significant, as they show a lower the consistency in patterns of exploration when compared to MAGA, that present a network interconnected by key common points, and a good diversity in local improvements after the findings of common points, presenting relevant exploitation.

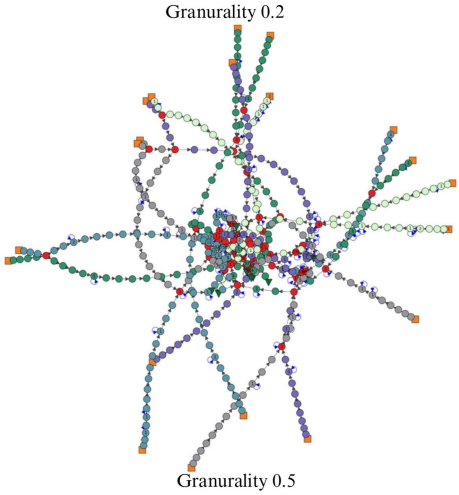
Proceeding to the next metrics, the self-loops indicated the number of iterations the algorithm stays in the same node. Logically, the higher the number of self-loops, the higher the power consumption without significant improvements. In optimization, self-loops often appear when the algorithm is trapped in local optima. GA presents higher numbers of self-loops, both in absolute value and in comparison with the global number of nodes, showing itself to be trapped with a higher



Null Granurality



Granurality 0.2



Granurality 0.5

Fig. 1. Visualizations of MAGABN Search Trajectory obtained from 20 executions and generated with different granularity values. The ○ symbols are the nodes representing each objective function value found by an execution, each colour reflects one execution of the metaheuristics for the same optimization problem, and the red ○ indicates common nodes between executions. Each orange □ represents the objective function value associated with each run, and the green ▽ is the final node associated with a run.

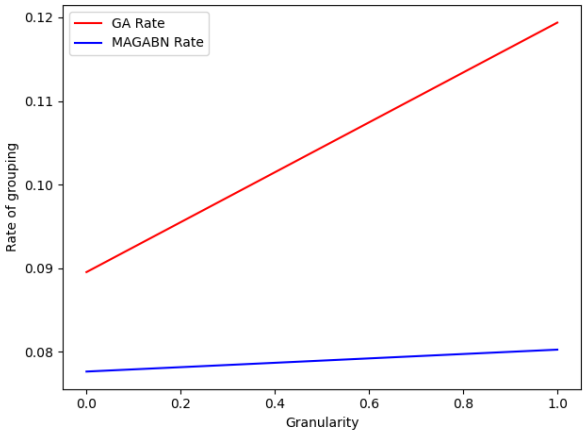


Fig. 2. Visualisation of the granularity rate comparison between MAGABN and GA, showcasing the rhythm each algorithm agglomerate its nodes based on granularity.

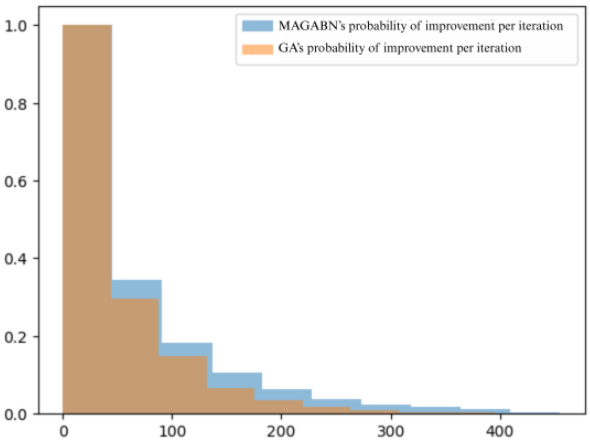


Fig. 3. Probability of improvement comparison between MAGA and GA, Child variation

frequency. That conclusion resonates with the analysis made with the histogram, where there is a lower probability of grand improvements than MAGABN. Still observing the capacity of each method to escape local optima, the next metric of the table shows the ability of self-improvement between runs of the algorithms. By being able to reach a final node of a run and surpass it, the algorithm shows great perseverance and exploration abilities. The higher this metric, the less likely early convergence occurrence, as it can consistently improve previous solutions found. The values of MAGABN and GA are similar, however, MAGABN shows a greater increase rate, showing a greater exploration ability. Additionally, the value of this metric on the MAGABN and GA comparison in the last column is a result of MAGABN being able to surpass the local optima of GA, outperforming the runs of the classic method.

The last three metrics of the table are related to the betweenness, highlighting the relationships that important points of a path hold with each other. The combined analysis of the mean,

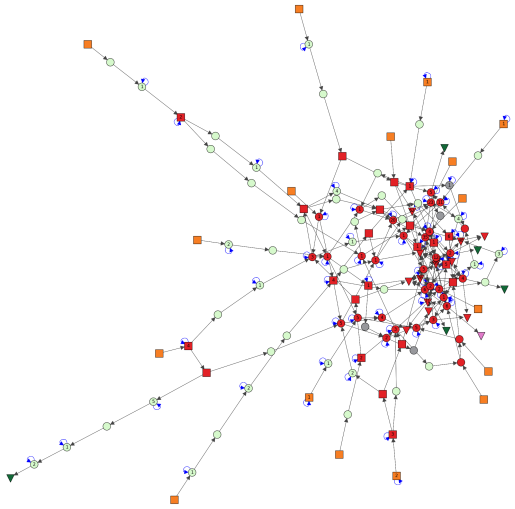


Fig. 4. Visualisation in a single graph of MAGABN and GA Search Trajectory obtained from 20 executions each and generated with granularity 0.5, both in the Asia variation. The comparison showcases the behaviour of both methods.

TABLE II
COMPARISON OF METRICS FOR THE GRAPH CONSTRUCTED FOR THE TWO METAHEURISTICS USING THE CHILD BENCHMARK PROBLEM WITH DIFFERENT VALUES OF GRANULARITY.

Metric	Granularity = 0.2		Granularity = 0.5		Granularity = 0.2
	MAGABN	GA	GA	MAGABN	MAGABN and GA
Number of nodes	511	712	627	470	1132
Average Path Length	14.2	21.0	15.0	13.1	16.3
Common Points	34	63	115	59	165
Self-loops	106	360	323	99	-
Passage through local optimal points	0	1	3	5	4
Mean Betweenness	671.2	2784.8	2799.3	1152.4	4015.9
Median Betweenness	510.0	2220.0	1922.0	822.0	2823.0
Standard Deviation of Betweenness	647.6	2566.6	2904.4	1226.1	4396.9

median, and standard deviation statistics is important as they give insights into the exploration and exploitation capabilities. The betweenness is the number of shortest paths that pass through a node. The lower the mean betweenness, the higher the exploration rate, as the algorithm tends not to pass through the same points and shows a higher diversity. Analogously, the higher the mean, the higher the exploitation rate, due to the presence of more important points found in different runs that pass through the same node and then proceed to a better solution, taking into consideration that the algorithms did not find the same final point within the runs.

An efficient optimization method should be able to balance these rates, and by that, should present a medium value of mean and a high value of standard deviation. Also, the method that presents a median value closer to the mean will show a great equilibrium of local and global improvements, as the method will present higher values of betweenness on key nodes and proceed with the refinement. Analysing the table, it can be seen that both methods present a great diversity of betweenness, obtaining a balanced ratio of mean and standard deviation. These metrics, alongside the median, show how both algorithms tend to value exploration. However, MAGABN also shows a median closer to the mean, as can be seen on the metric for a granularity = 0.5, where MAGABN median is 71.33% of its mean value, while the GA median is 68.66% of its mean. Deeper analysis also shows how the points visited by multiple paths concentrate in the middle of the path, confirming the statements of efficient global search succeeded by local improvements and refinements.

Finally, the plot of the simple paths of the MAGABN algorithm can be found in Figure 5, with the starting point being the initial node of each run and the final point being the best optima estimated by the algorithm. The simple path in graph theory is the path between two vertices that does not pass through a node more than once. A way to interpret it is as all the alternative paths that could reach the best optima estimated, given all the starting points. Having multiple paths towards the best optima estimated reflects the robustness of the MAGABN. Since having multiple paths that lead to the best estimated solution increases the chances of returning a good solution. That analysis also needs to consider other metrics to be valuable, such as the average path length and betweenness, to guarantee that the algorithm did not early converge, presents good knowledge retention while also exploring the search space, and does not only visit the same points in all runs.

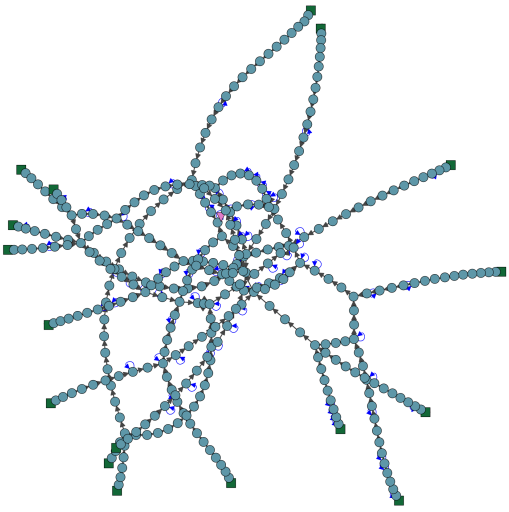


Fig. 5. Visualisation of MAGABN Simple Path, obtained from the Search Trajectory generated for 20 executions of MAGABN Child variation with granularity 0.2, showcasing the interconnected paths between the initial points, represented by □ and the global optima estimated by the method, represented by ▽.

With these results, our method indicates that MAGABN showed greater robustness and balance of exploration and exploitation. While still presenting some similar behaviours

to GA, it is also shown how some modifications can cause significant changes in the optimization procedure patterns. The results stated in this paper corroborate the results found by statistical hypothesis tests performed in [6], which concluded that MAGA is more consistent, stable, and returns better BN structures. The concordance between the empirical analysis and the graph-based analysis using complex networks' metrics highlights the usefulness of our approach to investigate the behaviour of metaheuristics, opening possibilities for deeper understandings of metaheuristic methods and more palpable and viewable comparisons.

VI. Conclusions

Given the relevance related to the understanding of how metaheuristics reach the final solution and its importance in the comparison and selection of methods for each application, the development and improvement of different analysis methods is a great advance. Using the vast studies of complex networks applied to metaheuristics, the behaviour of optimization methods can become more interpretable, helping in decision making about which algorithm should be used, searching for improvements within the same methods, and recognizing patterns. The methodology developed in this article presents a versatile and relatively simple independent tool that allows various visions and assimilation about the behaviour of each metaheuristic, whether on the search space or solution space. In this way, not only is the final solution considered the main aspect of a good method, but the paths taken during optimization are also equally important.

The next steps following this article will focus on generalizing the methodology to a wider range of metaheuristics and exploring metrics like sensitivity of granularity and betweenness more deeply. Future research will also aim to better interpret complex network metrics in relation to how metaheuristics behave during optimization. While using optimization variables to generate graphs is feasible, it was not the main focus here. Thus, future case studies should explore this approach and how it affects the insights into each method's behaviour. Another promising direction is applying this tool to multi-objective problems, enhancing decision-making support with clearer method explanations and new analytical tools.

References

[1] A. Gogna and A. Tayal, "Metaheuristics: review and application," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 503–526, 2013.

[2] M. Fyvie, J. A. W. McCall, L. A. Christie, A. E. I. Brownlee, and M. Singh, "Towards explainable metaheuristics: Feature extraction from trajectory mining," *Expert Systems*, vol. 42, no. 1, p. e13494, 2025.

[3] M. Bessani, L. d. S. Batista, and F. Campelo, "Characterization of distribution systems topological flexibility using bipartite multigraphs," in *Congresso Brasileiro de Automática-CBA*, vol. 3, 2022.

[4] M. Bessani, J. A. Massignan, R. Z. Fanucchi, M. H. Camillo, J. B. London, A. C. Delbem, and C. D. Maciel, "Probabilistic assessment of power distribution systems resilience under extreme weather," *IEEE Systems Journal*, vol. 13, no. 2, pp. 1747–1756, 2018.

[5] G. Ochoa, K. M. Malan, and C. Blum, "Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics," *Applied Soft Computing*, vol. 109, p. 107492, 2021.

[6] J. P. Campos, I. G. Machado, and M. Bessani, "Multi-agent genetic algorithm for bayesian networks structural learning," *Knowledge-Based Systems*, vol. 310, p. 113025, 2025.

[7] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stützle, "A comparison of the performance of different metaheuristics on the timetabling problem," in *Practice and Theory of Automated Timetabling IV* (E. Burke and P. De Causmaecker, eds.), (Berlin, Heidelberg), pp. 329–351, Springer Berlin Heidelberg, 2003.

[8] J. Silberholz and B. Golden, *Comparison of Metaheuristics*, pp. 625–640. Boston, MA: Springer US, 2010.

[9] J. Silberholz, B. Golden, S. Gupta, and X. Wang, *Computational Comparison of Metaheuristics*, pp. 581–604. Cham: Springer International Publishing, 2019.

[10] F. Campelo and F. Takahashi, "Sample size estimation for power and accuracy in the experimental comparison of algorithms," *Journal of Heuristics*, vol. 25, pp. 305–338, 2019.

[11] F. Campelo and E. F. Wanner, "Sample size calculations for the experimental comparison of multiple algorithms on multiple problem instances," *Journal of Heuristics*, vol. 26, no. 6, pp. 851–883, 2020.

[12] K. Malan and G. Ochoa, "Landscape analysis of optimization problems and algorithms," in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, GECCO '23 Companion, (New York, NY, USA), p. 1416–1432, Association for Computing Machinery, 2023.

[13] L. Šubelj and M. Bajec, "Software systems through complex networks science: review, analysis and applications," in *Proceedings of the First International Workshop on Software Mining*, SoftwareMining '12, (New York, NY, USA), p. 9–16, Association for Computing Machinery, 2012.

[14] R. Santana, R. Armañanzas, C. Bielza, and P. Larrañaga, "Network measures for information extraction in evolutionary algorithms," *Int. J. Comput. Intell. Syst.*, vol. 6, no. 6, p. 1163, 2013.

[15] G. Ochoa, K. M. Malan, and C. Blum, "Search trajectory networks of population-based algorithms in continuous spaces," in *Applications of Evolutionary Computation* (P. A. Castillo, J. L. Jiménez Laredo, and F. Fernández de Vega, eds.), (Cham), pp. 70–85, Springer International Publishing, 2020.

[16] C. Chacón Sartori, C. Blum, and G. Ochoa, "Stnweb: A new visualization tool for analyzing optimization algorithms," *Software Impacts*, vol. 17, p. 100558, 2023.

[17] C. Chacon-Sartori, C. Blum, and G. Ochoa, "Search trajectory networks meet the web: A web application for the visual comparison of optimization algorithms," in *Proceedings of the 2023 12th International Conference on Software and Computer Applications*, ICSCA '23, (New York, NY, USA), p. 89–96, Association for Computing Machinery, 2023.

[18] Y. Lavinas, C. Aranha, and G. Ochoa, "Search trajectories networks of multiobjective evolutionary algorithms," in *Applications of Evolutionary Computation* (J. L. Jiménez Laredo, J. I. Hidalgo, and K. O. Babaagba, eds.), (Cham), pp. 223–238, Springer International Publishing, 2022.

[19] D. Kong, R. Chen, Y. Chen, L. Zhao, R. Huang, L. Luo, F. Lai, Z. Yang, S. Wang, J. Zhang, *et al.*, "Bayesian network analysis of factors influencing type 2 diabetes, coronary heart disease, and their comorbidities," *BMC Public Health*, vol. 24, no. 1, p. 1267, 2024.

[20] J. Yu, Y. Song, D. Tang, and J. Dai, "A digital twin approach based on nonparametric bayesian network for complex system health monitoring," *Journal of Manufacturing Systems*, vol. 58, pp. 293–304, 2021.

[21] M. Bessani, J. A. Massignan, T. M. Santos, J. B. London Jr, and C. D. Maciel, "Multiple households very short-term load forecasting using bayesian networks," *Electric Power Systems Research*, vol. 189, p. 106733, 2020.

[22] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for combinatorial optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 1, pp. 229–240, 2009.

[23] G. T. L. da F. Costa, F. A. Rodrigues and P. R. V. Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167–242, 2007.

[24] S. Ólafsson, "Chapter 21 metaheuristics," in *Simulation* (S. G. Henderson and B. L. Nelson, eds.), vol. 13 of *Handbooks in Operations Research and Management Science*, pp. 633–654, Elsevier, 2006.

[25] L. Velasco, H. Guerrero, and A. Hospitaler, "A literature review and critical analysis of metaheuristics recently developed," *Archives of Computational Methods in Engineering*, vol. 31, pp. 125–146, 2024.