

An Online Human-AI Collaborative Tool for Steering a Genetic Algorithm to Address the Maximum Clique Problem

Brynner Barbosa de Brito, Thiago Sylas Antunes da Costa, Kevin Washington Azevedo da Cruz, Natã Ferreira Lobato, Nelson Cruz Sampaio Neto, and Carlos Gustavo Resque dos Santos(✉)

Universidade Federal do Pará, Belém, Brazil
carlosresque@ufpa.br

Abstract. This paper presents an online Human-AI collaborative tool designed to steer a genetic algorithm in solving the Maximum Clique Problem (MCP). By employing human-in-the-loop principles, the system enables users to dynamically adjust algorithmic parameters in real-time. We developed an interactive tool to investigate possible directions for these challenges, enabling an intense collaboration between human experts and artificial intelligence in addressing an NP-Hard problem. We assess the tool by conducting tests with DIMACS benchmark datasets to validate the effectiveness of the approach and gain insights about this form of Human-AI interaction. Additionally, a heuristic evaluation was carried out with domain experts, who interacted with the system to assess usability. Heuristic evaluation confirmed that the tool meets basic requirements, with 17.29% of observations highlighting areas for improvement, particularly in tool functionality and visualization enhancements. The experimental results indicate that the collaborative tool achieves competitive performance while enhancing user engagement and control over the optimization process. Additionally, the results showed interaction preferences, strengths, and weaknesses.

Keywords: Human-in-the-Loop · Genetic Algorithm · Maximum Clique Problem · Human-AI Collaboration

1 Introduction

The integration of human expertise and artificial intelligence (AI) systems has gained prominence to enhance optimization processes [26]. In particular, the human-in-the-loop (HITL) paradigm has emerged in machine learning and optimization tasks, allowing users to interact with algorithms to refine solutions, improve interpretability, and guide the decision-making process. By incorporating human intuition, experience, and contextual knowledge, AI systems become more adaptable, flexible, and transparent, mitigating the "black box" [1].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
O. Gervasi et al. (Eds.): ICCSA 2025, LNCS 15648, pp. 88–105, 2025.
https://doi.org/10.1007/978-3-031-97000-9_6

In the field of combinatorial optimization, the Maximum Clique Problem (MCP) stands out as a challenge with applications in bioinformatics, social network analysis, artificial intelligence, and cryptography. In a graph $G\ (V, E)$, with vertices set V and edges set E connecting pairs of vertices, a complete subgraph of G is one whose vertices are mutually adjacent. A clique is a maximal complete subgraph [3]. The MCP aims to determine a largest clique in a graph G. It is one of the NP-complete problems.

Optimization approaches often struggle due to the inherent complexity of the MCP and experts must settle for near-optimal solutions. In such scenarios, it is not just about the final solution but also about how an expert can interact with the optimization process, dynamically adjusting parameters, exploring alternative solutions, and refining constraints based on domain knowledge [29].

One of the greatest advantages of Human-AI collaboration is the ability of users to dynamically adjust parameters while the algorithm is running. This approach contrasts with conventional methods, where the parameters are predefined based on prior knowledge or determined through extensive and time-consuming parameter sweeps [2].

Dynamic parameter tuning, guided by human intuition, enables real-time adaptation of algorithms, making the search process more transparent. This approach allows users to adjust critical variables as they observe the results, optimizing the exploration of the solution space and improving the overall performance of the system [13].

In this study, we propose a tool to integrate human interaction into solving the MCP based on genetic algorithms. Our approach emphasizes user engagement throughout the evolutionary process, enabling dynamic parameter adjustments and real-time feedback and visualizations.

This paper is structured as follows: Sect. 2 provides an overview of related works. Section 3 presents the developed tool. Section 4 presents experimental results from pilot tests. Section 5 presents a heuristic evaluation conducted by experts in the field, and finally, Sect. 6 discusses future research directions and concludes with reflections on the broader implications of interactive optimization paradigms.

2 Related Works

Llorà et al. [20] introduced Active Interactive Genetic Algorithms (aiGA), where users rank solutions without altering selection or mutation. The tool tracks user interactions, updates preference models, and presents new solutions. Hettenhausen et al. [16] employed an interactive heatmap to guide particle swarm movement and applied a novel technique in using multiple visualizations, including parallel coordinates and scatter plots, with user interaction through the selection of value ranges to guide the algorithm.

Lutton and Fekete [21] evaluated visual analytics tools like ScatterDice and GeneaQuilts for complex Evolution Algorithm (EA) data sets. Farooq et al. [14] introduced Visualization of Genetic Algorithm on 2-D (VIGA-2D) Graph, which

requires occasional human intervention to propose new solutions. Boukhelifa et al. [4] evaluated and analyzed a framework for Evolutionary Visual Exploration (EVE) created by them named EvoGraphDice, an interactive system interface that reflects the evolutionary process and user input in real time via a scatterplot matrix.

Cruz et al. [9] introduced tools for focusing generations and tracking lineages with options to select individuals, and manipulate visual details like fitness, genetic operators, and offspring information through tree and parallel coordinates, while Walker [33] used treemaps and 2D charts to visualize multi-objective data. Radhika and Velayutham [28] explored visual analysis of Differential Evolution (DE) search, using scatter plots and difference vectors to detect premature convergence and stagnation across six DE variants.

Vallerio et al. [32] presented a multi-objective framework where users select weights for Pareto front optimization, visualizing relationships with parallel coordinates. McPhee et al. [22] analyzed ancestry in evolutionary computation runs through visualizations of parent-child relationships. Medvet and Tušar [23] proposed a heatmap summarizing genotype diversity and gene contributions across different EAs.

Daneshpajouh and Zakaria [10] introduced a clustering-based visualization tool for Genetic Algorithm (GA) search refinement interactively in a 3D environment, while Mexicano-Santoyo et al. [25] provided a Differential Evolution (DE) analysis tool with interactive views through 2D and 3D graphics where users can manipulate data views by adjusting generations, individuals, and algorithm parameters like mutation factor and crossing probability. Sathyajit and Velayutham [30] investigated the visual analysis of Genetic Algorithms (GAs) applied to the simple 0–1 Knapsack problem[1] using heat maps.

Dolson and Ofria [12] developed a Virtual Reality (VR) framework to explore evolutionary history through 3D fitness landscapes and population lineages. Meiguins et al. [24] presented tools for exploring evolutionary processes in AutoClustering, providing various visualizations like scatterplots, heatmaps, and parallel coordinates. Chae et al. [7] explored the evolutionary process of neural networks through lineage views and fitness hyperparameter adjustments.

Vidaurre et al. [8] developed a system for visualizing and tuning genetic algorithms to generate software unit tests. Ivanov et al. [18] allowed to guide the algorithm by manipulating tiles, which represent evolving genomes on a hexagonal grid. Walter et al. [34] introduced a tool for visualizing algorithm search on discrete dual- and multi-objective knapsack problems, and Hakanen et al. [15] employed coordinated multiple views for optimization through user preferences via parallel coordinates and other views.

Although there are various tools and methods of interacting with optimization algorithms, in general, these tools do not allow online interaction, or do not provide visualizations that update at runtime, or are alternatives for very

[1] The 0/1 Knapsack Problem aims to maximize total value by selecting items within a weight and value limit without exceeding the knapsack's capacity, where each item can be either included (1) or excluded (0).

specific applications. The tool proposed in this article is innovative by applying these characteristics to a problem that can be used in a series of applications and can be adapted to other similar classical problems without much effort.

3 Proposed Tool

The proposed tool was developed primarily to support human-in-the-loop interactions. It was built using JavaScript to enable the creation of visualizations and interactive interfaces that update at runtime. The user also has control over the execution time by adjusting hyperparameters that make the execution slower or faster. Table 1 describes the hyperparameters that the user can adjust at runtime.

Table 1. GA hyperparameters that can be adjusted by the user during execution.

Variable Name	Description
$populationSize$	Size of the population
$survivalRate$	Rate of individuals that survive from the current to the next generation
$mutSelectionRate$	Mutation rate applied to selected individuals
$mutationRate$	Probability that a new individual undergoes mutation
$partialReset$	Performs a reset on $1 - survivalRate$ to the next generation
$1sProb$	Probability of generating a 1 in the bitstring of a randomly generated individual
$hasMaxAge$	When set to $true$, resets the fitness of individuals that survived more than $maxAge$ generations
$maxAge$	Maximum number of generations an individual can survive when $hasMaxAge = true$
$hasImprovement$	When set to $true$, performs local improvement on new individuals
$preventEqual$	Prevents the creation of duplicate individuals already present in the population
$calcUpperBound$	Computes a simple heuristic to estimate the chromatic number

This section is divided into two parts. The first explains the implementation details of the Genetic Algorithm. The second part explains the interfaces and visualizations that the user has to analyze and manipulate the GA.

3.1 GA Implementation

The Genetic Algorithm (GA) was selected due to its ease of understanding and widespread adoption, which reduces the tool's learning curve and enhances user familiarity. GA was also considered suitable for an initial implementation because of its large number of hyperparameters, providing users with extensive flexibility for algorithm tuning.

The algorithm design was inspired by the implementation of Huang [17] due to its simplicity and reasonable efficiency. Certain aspects were modified to prioritize speed over efficiency, as user interactions are required to occur in real time.

Each individual is represented by a binary mask, where each bit indicates the presence (1) or absence (0) of the vertices in a graph G, thus defining a subgraph. Since the binary mask can represent any subgraph, it is necessary to verify whether it forms a clique by checking if all selected vertices are pairwise connected. If the subgraph is a valid clique, its size is determined by counting the number of 1 s in the mask. Each new individual is generated in three steps. Initially, a zero-filled array is created; then, 1 s are randomly inserted with $1sProb$ probability; if $hasImprovement = true$ the extraction and improvement operations are applied (these operations are described next). This procedure is used both for initializing the population and during the partial reset operation [31], which can be triggered by the user.

The fitness[2] is calculated by summing the number of 1 s in the binary mask. If $hasImprovement = false$, individuals may represent a subgraph that is not a valid clique. In this case, the fitness is the negative number of missing edges required to obtain a valid clique. In the crossover[3], two parents are randomly selected from the population, and a single-point crossover is applied at a randomly chosen position. If $hasImprovement = true$, the extraction and improvement operations are applied.

The mutation[4] is applied only into offspring generated via crossover, with a probability $mutSelectionRate$. The operation consists of performing a bit swap on each bit of the binary mask with a probability defined by the $mutationRate$ parameter. In other words, the $mutSelectionRate$ controls how many offspring undergo mutation, and the $mutationRate$ controls the amount of bit swaps in a mutation.

The $survivalRate$ variable controls the percentage of the current population that is retained. Only the top-performing individuals from the previous population are preserved. The remaining individuals who will form the new population are generated through crossover followed by mutation. The extraction operation randomly removes vertices from the clique until it represents a valid clique. The improvement operation randomly adds vertices until the subgraph becomes a maximal clique. Together, these two operations ensure that the resulting individual represents a valid (after extraction) and maximal (after improvement) clique.

[2] In genetic algorithms, *fitness* quantifies how well a candidate solution satisfies the problem's objective.

[3] In genetic algorithms, *crossover* combines parts of two parent solutions to generate new offspring, promoting genetic diversity.

[4] In genetic algorithms, *mutation* introduces random alterations to a solution to maintain diversity and explore new regions of the search space.

The partial reset eliminates $1 - survivalRate$ individuals in the new population and replaces them by randomly generated individuals, following the same procedure used for initializing the original population.

Each individual has an *age* counter that is incremented every generation it survives. If $hasMaxAge = true$ and the individual has $age > maxAge$, its fitness becomes 0. This parameter allows the user to control when the population elite should be refreshed.

3.2 Interfaces and Visualizations

Figure 1 shows a screenshot of the interfaces available for user interaction with the GA. These interfaces are web pages that can be freely arranged by the user, with each operating independently. This allows users to customize their own dashboard layout.

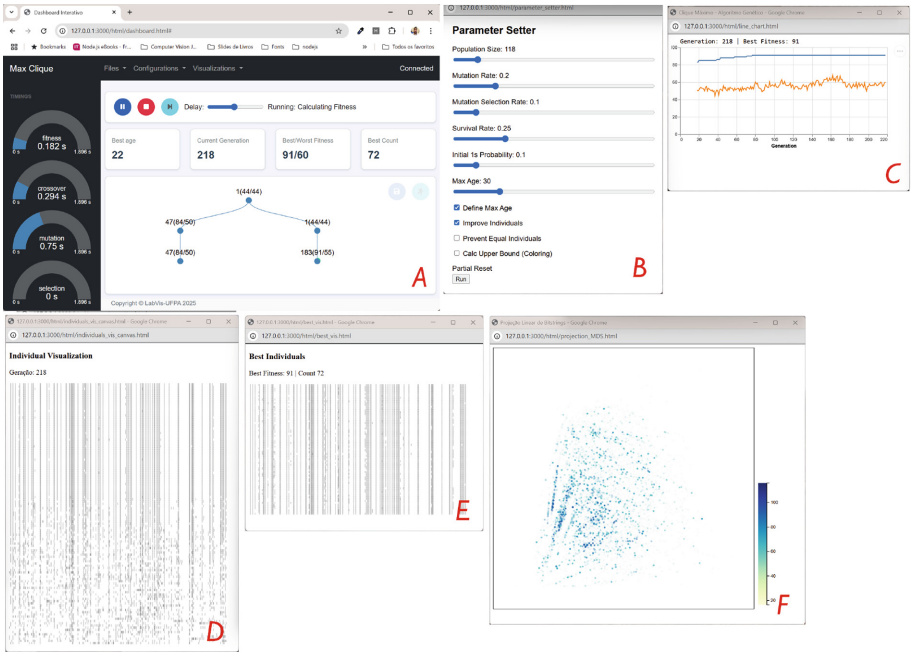

Fig. 1. Screenshot of the tool displaying all currently available interfaces.

The main dashboard provides a real-time overview of the algorithm's execution, displaying key metrics and interactive controls. In the menu bar, the user can upload and manage datasets and saves. The execution bar (below the menu bar) shows the current execution status, allowing the user to pause, restart, stop, process the next generation, and adjust the runtime speed. Performance

gauges (on the left side) monitor the execution time of each core operation in the algorithm, enabling performance analysis throughout the evolutionary process.

The informational cards (bellow execution bar) summarize the algorithm's execution. It shows the current generation; the best and worst fitness of the current population; the best count card shows the number of different solutions encountered with the same best fitness; and the best age card provides the age of the best individual in the current population. During the GA execution, the user can save the current GA state (all hyperparameters and population) to be able to recover to this saved point. The dendrogram visualization documents the saves' history, where each node represents a specific state, similar to a breadcrumb idea. Users can click on any node to restore that specific execution state and run the algorithm from this point with distinct parameters.

The parameter configuration interface allows users to dynamically adjust key genetic algorithm hyperparameters, directly influencing its convergence and performance. The interface includes controls for tuning $populationSize$, $mult\text{-}SelectionRate$, $mutationRate$, $survivalRate$, $1sProb$, and the $maxAge$ of individuals, as shown in Table 1. Below these controls, additional options allow toggling specific settings $maxAge$, $hasImprovement$, $preventEqual$, and $calcUpperBound$. A $partialReset$ button triggers the partial reset into the next generation. All changes are applied immediately without interrupting the algorithm's execution, allowing for continuous and dynamic tuning throughout the evolutionary process.

The line chart visualization tracks the evolution of $fitness$ across generations, enabling real-time monitoring of the genetic algorithm's performance. At the top, key information is displayed, including the current generation, the best fitness found, and the best upper bound found.

The individual visualization provides a grid visual representation of individuals in the population, where each line shows the individual binary mask (1 in black) and each column represents a vertice of the graph. The density and pattern of these markers indicate the population's diversity degree and convergence trends. Whenever a new individual with higher fitness is found, it is recorded in a separate visualization using the same format (Best Individuals window). The number of lines increases as different individuals with the same fitness are discovered.

In scatterplot visualization, each individual is represented as a point obtained by projecting its bits array into $\mathbb{R}^2$. The Multidimensional Scaling (MDS)[5] is used due to its proven effectiveness [11]. MDS is applied only to the initial generation to avoid computational overhead and ensure real-time updates. Being the initial population random, it provides good spatial dispersion. In subsequent generations, new points are projected using K Nearest Neighbor (KNN), ensuring plot stability and that identical individuals across different generations are

[5] Multidimensional Scaling (MDS) is a class of dimensionality reduction techniques that use proximities among any kind of objects to project high-dimensional data into a lower-dimensional space while preserving the pairwise distances between points as closely as possible.

plotted in the same position. K=5 is used because it empirically yielded good results, but this value can be configured and better evaluated in future works.

The purpose of this visualization is to illustrate the GA's trajectory through the search space over time. Hence, each generation's points are drawn on the canvas, with a fade-out effect applied to the previous points. This effect emphasizes recent generations without completely removing the previous ones, offering a continuous view of the population's trajectory on the search space.

4 Internal Validation Tests

This test aims to evaluate the efficiency of the tool developed to address the Maximum Clique Problem (MCP). Additionally, it seeks to identify possible bugs, making the heuristic evaluation with specialists less susceptible to such errors.

4.1 Test Setup

The tests were conducted by the tool's developers, who interacted with the application and recorded relevant impressions regarding its efficiency and functionality. Any bugs encountered in this phase were fixed.

The test was conducted in a dedicated room, equipped with air conditioning and an office chair. The tool was executed on a notebook with an Intel Core i7-10750H Comet Lake processor (6 cores, 2.6 GHz frequency), 64 GB of DDR4 RAM (2933MHz), an NVIDIA GeForce RTX 2070 GPU (8GB GDDR6), and PCIe SSD storage. The test setup included two screens for participant interaction: the notebook's built-in monitor, a 15.6" Full HD (1920×1080p) LED display with a 144 Hz refresh rate, and an external monitor with 27" UHD 4K (3840×2160) IPS display, 5ms response time, and a 60 Hz refresh rate.

The test utilized the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) benchmark [19], which includes datasets with the best known solutions for MCP[6]. The evaluators tested the tool with each of the available datasets. During execution, the evaluators have to adjust the hyperparameter in the GA and have support of real-time visualizations to perceive and analyze the impact of these changes. The evaluators could take note of suggestions about their interaction. The following metrics were recorded (system logs) during execution: (1) best clique found, (2) number of users interactions, and (3) total execution time.

The task consisted of finding the best known clique. For some datasets, the result was found quickly, but for others it was not found at all. There was no time limit to complete the task, but the evaluators were already tired after about 25 min, as showed in the results.

[6] A useful link of datasets: https://iridia.ulb.ac.be/~ fmascia/maximum_clique/.

4.2 Results

Table 2 presents results from three Evaluators (Ev. 1, Ev. 2 and Ev. 3). The values in the Ev. column have the best clique found, the number of interactions (in parentheses), and the time taken in minutes [in brackets]. Bold clique values indicate that the evaluator found the best-known clique ($\omega(G)$).

The evaluators reached the best-known clique in 23 out of 37 datasets (62.16%), with an average execution time of 6.57 min and an average of 45.21 user interactions. This data suggests that, for some datasets, the GA—as designed—could solve the problem quickly by simply selecting appropriate hyperparameters from the start.

Thus, to conduct a more in-depth analysis, we filtered datasets that: (1) did not exceed 15 min of execution time, and (2) did not discover a new best clique after the 5-minute. This filtering resulted in a subset of 8 more challenging datasets. Under these conditions, the average execution time increased to 14.4 min and an average of 105.9 interactions.

Figure 2 presents user interactions across these 8 datasets for each evaluator. Each timeline represents the total execution time, where the black dot indicates when a new best clique was found. Colored dots represent user interactions throughout the run. A random jitter was applied to the y-axis for easier visualization.

Overall, a typical fitness curve behavior is observed—many better cliques are found early in the process, even with minimal user interaction. From approximately the 2-minute mark, the GA struggles to find the best clique with the same ease. After this stage, the visualization reveals user interactions until they find a new best. It also indicates that the most frequently adjusted parameters were: *populationSize*, *survivalRate*, *mutSelectionRate*, and *mutationRate*.

In a follow-up meeting after the tests, the evaluators discussed the strengths and weaknesses of the tool, as well as potential improvements. Strengths included the real-time visualizations, parameter controls for *populationSize*, *survivalRate*, *mutSelectionRate*, *mutationRate*, and *preventEqual*, and the ability to navigate the GA save history. The visualizations enabled real-time analysis of how hyperparameter adjustments influenced the GA's behavior. In particular, when manipulating the five mentioned parameters, evaluators identified an effective strategy: maintaining population diversity by reducing elitism (especially by lowering *survivalRate* and increasing *mutationRate*), while preserving a reasonable portion of the elite to ensure crossover operations produce solutions near high-performing individuals.

Weaknesses included the *partialReset* feature. Although its intended purpose was to increase population diversity when stuck in a local optimum, evaluators observed that when the population had already reached a favorable balance between diversity and elitism—and the fitness had significantly diverged from the initial random state—the new individuals generated by partial reset closely resembled the non-elite individuals and were replaced with ease, having no impact on the algorithm search process.

Table 2. Results of the internal evaluation. The first four columns are some dataset metadata, and the three last columns are evaluators results with $n/i/t$ mask. Where n is the best clique found (bold if benchmark found), i is the number of interactions, and t is the runtime in minutes.

Dataset	Nodes	Links	$\omega(G)$	Ev. 1 $(n/i/t)$	Ev. 2 $(n/i/t)$	Ev. 3 $(n/i/t)$
C125.9	125	6963	34	**34**/2/0	**34**/5/0	**34**/24/7
C250.9	250	27984	44	**44**/31/6	**44**/31/1	43/63/5
C500.9	500	112332	57	54/102/27	53/153/10	53/237/11
C1000.9	1000	450079	68	63/47/22	64/100/10	63/292/18
C2000.9	2000	1799532	80	66/41/20	70/96/13	66/102/3
DSJC1000_5	1000	499652	15	14/42/25	14/62/3	14/163/14
DSJC500_5	500	125248	13	**13**/16/41	12/34/3	**13**/132/8
C2000.5	2000	999836	16	**16**/8/3	**16**/41/8	14/58/14
C4000.5	4000	4000268	18	16/42/25	15/69/10	15/112/7
MANN_a27	378	70551	126	**126**/17/6	**126**/30/2	**126**/46/3
MANN_a45	1035	533115	345	342/49/17	342/123/15	342/142/3
MANN_a81	3321	5506380	1100	1096/48/20	1096/61/2	1096/99/7
brock200_2	200	9876	12	**12**/9/1	**12**/9/0	**12**/12/2
brock200_4	200	13089	17	**17**/7/1	**17**/12/1	**17**/35/3
brock400_2	400	59786	29	24/78/18	25/164/13	24/258/12
brock400_4	400	59765	33	25/40/13	**33**/68/3	24/191/23
brock800_2	800	208166	24	19/31/11	20/87/11	20/149/8
brock800_4	800	207643	26	20/41/20	20/84/10	20/120/8
gen200_p0.9_44	200	17910	44	**44**/27/4	**44**/28/1	40/55/7
gen200_p0.9_55	200	17910	55	**55**/3/1	**55**/6/0	**55**/6/1
gen400_p0.9_55	400	71820	55	50/16/6	50/67/8	50/98/4
gen400_p0.9_65	400	71820	65	**65**/6/2	**65**/8/1	**65**/8/1
gen400_p0.9_75	400	71820	75	**75**/0/0	**75**/1/1	**75**/2/1
hamming10-4	1024	434176	40	**40**/12/8	**40**/60/11	38/111/8
hamming8-4	256	20864	16	**16**/0/0	**16**/1/1	**16**/1/1
keller4	171	9435	11	**11**/0/0	**11**/0/0	**11**/0/1
keller5	776	225990	27	**27**/7/1	**27**/8/1	**27**/11/1
keller6	3361	4619898	59	52/11/19	52/59/25	55/66/3
p_hat300-1	300	10933	8	**8**/2/0	**8**/2/0	**8**/2/1
p_hat300-2	300	21928	25	**25**/0/0	**25**/1/0	**25**/1/1
p_hat300-3	300	33390	36	**36**/0/0	**36**/1/0	**36**/30/4
p_hat700-1	700	60999	11	**11**/8/3	**11**/9/1	**11**/45/4
p_hat700-2	700	121728	44	**44**/4/1	**44**/5/1	**44**/5/1
p_hat700-3	700	183010	62	**62**/0/0	**62**/1/1	61/47/4
p_hat1500-1	1500	284923	12	11/48/12	10/95/11	10/205/15
p_hat1500-2	1500	568960	65	**65**/3/1	**65**/13/4	**65**/39/5
p_hat1500-3	1500	847244	94	92/27/11	91/38/6	92/105/11

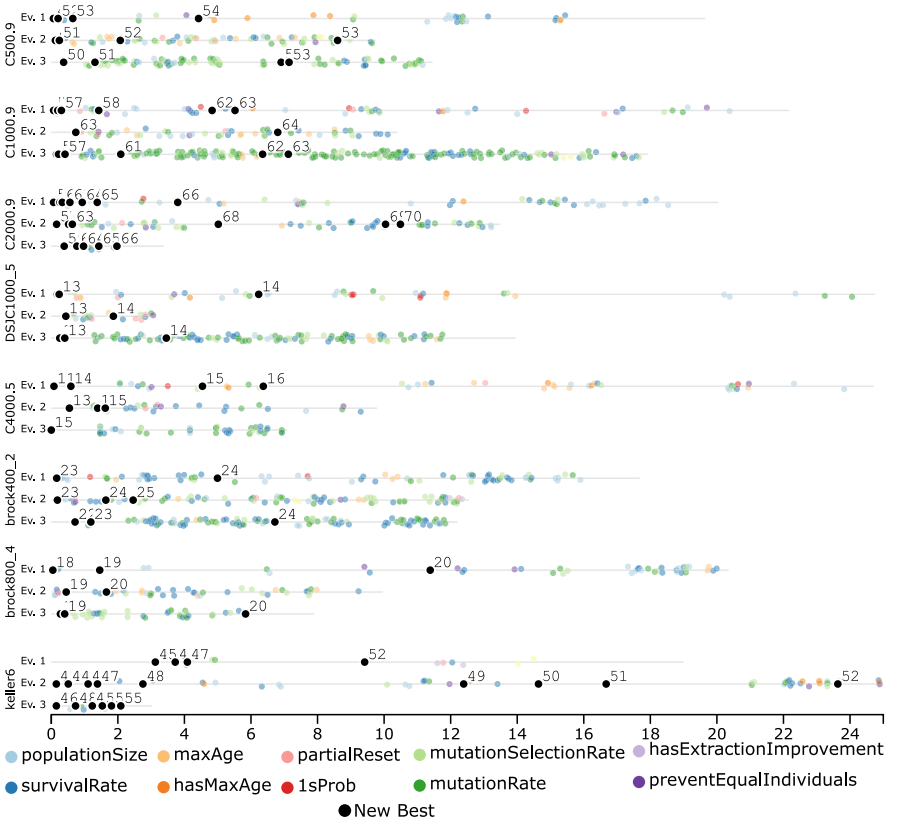

Fig. 2. Visualization of internal evaluation logs for selected challenging datasets. Each row represents the timeline of an evaluator for a specific dataset. Black dots indicate the moment a new best solution was found, while colored dots represent user interactions, as indicated in the legend. Over black dots is the number represents the best clique size. The x-axis is in minutes.

Proposed and documented improvements for the next development cycle include: making the best solution immortal (thus, it can be replaced if $hasMaxAge = true$) or reviving a subset of the best individuals ever found; plotting a fitness color gradient to Individual Visualization; displaying metadata of the running dataset; and providing a one-click option to open all visualizations and controls at once.

5 Heuristic Evaluation

The heuristic evaluation aims to analyze the usability and effectiveness of the tool based on the experience of experts in artificial intelligence and graph theory. The process was based on heuristic guidelines consolidated in the literature,

allowing for a critical assessment of the interface quality and user experience. This evaluation used the same environment as the Internal Validation Tests.

5.1 Test Setup

Three experts were invited: professors who teach artificial intelligence and/or graph theory courses and have published research in the field.

First, the experiment facilitator briefly introduced the tool, explaining all available windows and options. A simple problem was solved as an example of how to use the tool.

The experts then used the tool, interacting with it and adjusting the GA parameters to attempt to find the maximum clique. After completing the tests, the experts participated in an interview in which the questions were structured based on heuristic evaluation proposed by [27] and on the guidelines proposed by [2]. The participants were allowed to interact freely with the tool if they needed to recall something during the interview.

The interviews were recorded and transcribed for later qualitative analysis. A thematic analysis was conducted to identify patterns in responses and create a mental map of the main positive and negative aspects and areas for improvement.

The specialists were informed about the goal of assessing the usability, efficiency, and effectiveness of the proposed tool through a heuristic evaluation based on predefined guidelines. The interviews were recorded in audio format for posterior transcription. Based on the guidelines, the following questions were asked:

- 1. How was your overall experience interacting with the tool?
- 2. Did you have any difficulty understanding the interface or commands?
- 3. Was the interaction flow intuitive, or were there moments of confusion?
- 4. Were the on-screen information clear and well-organized?
- 5. Was the visual feedback sufficient to understand what was happening?
- 6. Was it easy to learn how to use the tool?
- 7. Does the system communicate what it is capable of doing?
- 8. Does the system demonstrate that it can perform its intended tasks?
- 9. Does the system provide a good response time for executed actions?
- 10. Does the system display contextually relevant information?
- 11. Does the system help the user efficiently reject unnecessary commands?
- 12. Does the system record recent interactions?
- 13. Does the system prevent information overload?
- 14. Does the system inform users about the consequences of their actions?
- 15. Does the system provide global controls?
- 16. Does the system present changes in AI behavior?
- 17. Have you used any similar tools before? How does this one stand out?
- 18. What could be improved in the tool?
- 19. Could the tool benefit a broader audience?
- 20. Do you think this approach increases reliability in AI?

5.2 Thematic Analysis

After transcribing the interviews, two coders conducted the thematic analysis [6], following its six-phase approach. The coders didn't participate in the tool development or evaluation process. They began by thoroughly familiarizing themselves with the data, carefully reviewing all transcriptions to ensure consistency. During this initial phase, they shared their general impressions, identifying preliminary patterns and key aspects that would guide the subsequent coding and theme development. Next, an independent coding of all content was performed. Individually, each coder also generated themes by grouping similar codes. The independently generated themes were validated collaboratively, with both coders comparing differences and jointly deciding on the theme definition and naming.

After validating the themes, descriptions were extracted, and analyses were conducted to generate results that revealed key insights from the participants regarding the tool. To facilitate this thematic analysis, the QualCoder software [5] was utilized, allowing for the organization and examination of textual data. QualCoder enabled the identification of patterns and relationships within the dataset, enhancing the depth and clarity of the findings.

5.3 Results

Figure 3 shows the mind map that graphically organizes the main categories of feedback and suggestions for improvement. The structure of the map shows how the experts perceive the tool, highlighting areas for improvement and positive points of the system.

The results of the heuristic evaluation demonstrated that the expected objectives were achieved, highlighting that a significant portion of the assessments consisted of improvement suggestions. Based on the thematic analysis conducted using QualCoder software, approximately 17.29% of the observations were related to recommendations for enhancing the tool. These suggestions encompassed general system improvements and refinements in the presentation of visualizations. The percentages in parentheses represent the number of observations from participants relative to the total number of observations. An observation can be related to one or more codes.

- **Tool Improvement** (10.6%): General recommendations to improve the user experience and the functionality of the system. For instance, the Participant 2 (P2) stated, "Inclusion of new variation options and crossover techniques, as well as adaptive features to suggest hyperparameters and optimize the user experience [...]".
- **Improve Visualizations** (6.69%): Suggestions to optimize the presentation of data and the usability of graphical representations. For instance, Participant 3 (P3) suggests that "For the graph of the best and worst individual's fitness, show a curve representing the average fitness of the population [...]".

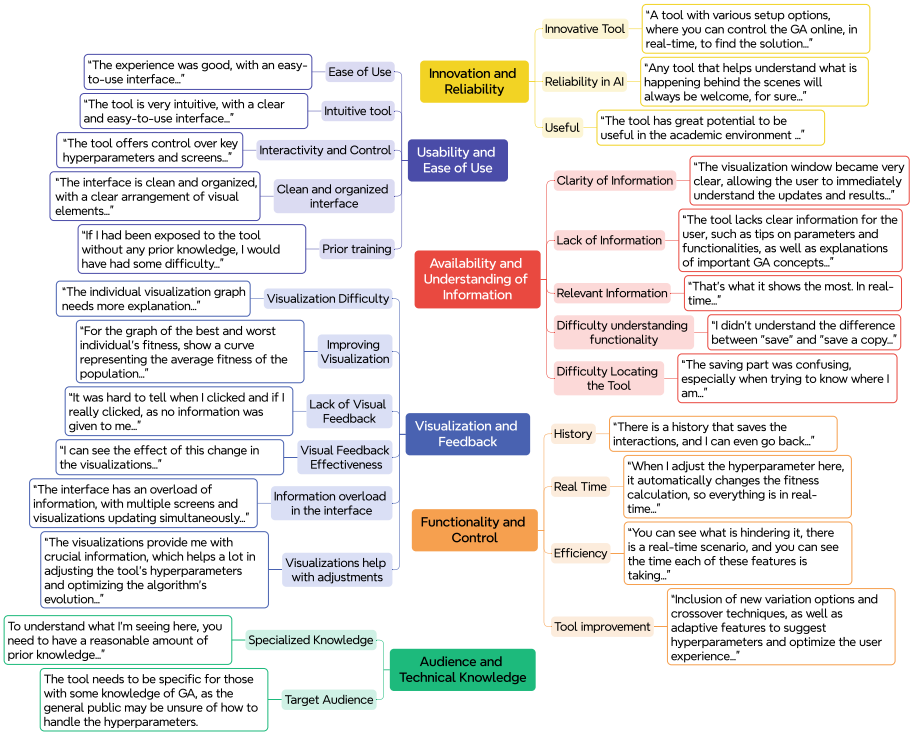

Fig. 3. Results of the thematic analysis. The figure provides a visual summary of the identified themes along with their respective codes

The feedback from the experts is the following:

Usability and Ease of Use: the codes about Clean and Organized Interface (3.15%) are illustrated by the P2 statement: "The interface is clean and organized, with a clear arrangement of visual elements [...]" and the Intuitive Tool (1.57%) codes are illustrated by the P1 statement: "The tool is very intuitive, with a clear and easy-to-use interface [...]". Indicate that the system already has a solid foundation in terms of usability, allowing the user to rearrange screens according to their needs. About Ease of Use (9.45%) the P2 stated that "the experience was good, with an easy-to-use interface [...]". The system has a learning curve that could be optimized. Although it is intuitive in some aspects, it could benefit from initial training to make it easier to handle the buttons and visualizations. About Interactivity and Control (5.51%) the P2 stated that "the tool offers control over key hyperparameters and screens [...]" and that there is the possibility of expanding functionalities to allow for greater customization.

Visualization and Feedback: in the code Information Overload on the Interface (3.15%) some users reported that the amount of information displayed simultaneously can create difficulties in understanding, especially for those without

previous experience. P3 stated that "the interface has an overload of information, with multiple screens and visualizations updating simultaneously [...]". In the code Visualizations Help with Adjustments (5.12%), the visualizations used make it easier to understand and make adjustments to the tool. Suggestions include creating new visualizations, a curve of the best average fitness of the population, with the standard deviation behind it, in a very smooth way, so as not to pollute so much. The P2 stated that "the visualizations provide me with crucial information, which helps a lot in adjusting the tool's hyperparameters and optimizing the algorithm's evolution [...]".

Availability and Understanding of Information: Clarity of Information and Relevant Information (5.12%) codes reveals that the organization and terminology could be refined. Experts highlighted the importance of displaying data in a clear and accessible way within the system. P1 stated that "the visualization window became very clear, allowing the user to immediately understand the updates and results [...]" and P3 stated that "that's what it shows the most. In real-time [...]".

Functionality and Control: Real Time (5.12%) codes reveals that the experts emphasized the importance of updating data in real time. The system's response to changes in hyperparameters was considered efficient and useful for monitoring the process. P1 stated that "when I adjust the hyperparameter here, it updates everything in real-time".

Some participants felt they needed training to use the tool better, mentioning aspects such as Prior training (2.76%) and Specialized knowledge (2.76%). In addition, certain functionalities, such as dimensionality reduction and clustering, were perceived as complex, suggesting the need for additional explanatory materials. The P3 stated, "If I had been exposed to the tool without any prior knowledge, I would have had some difficulty [...]" and "to understand what I'm seeing here, you need to have a reasonable amount of prior knowledge".

The Lack of Information in some areas of the interface was also pointed out (7.09%), making interpretation of the results more difficult, especially for users unfamiliar with Genetic Algorithms (GA). The P1 stated that "The tool lacks clear information for the user, such as tips on parameters and functionalities, as well as explanations of important GA concepts [...]".

Among the challenges pointed out were Difficulty Understanding Functionality (4.72%) and Visualization Difficulty (4.33%). P1 stated, "I didn't understand the difference between save and save a copy [...]", and P2 stated that "The individual visualization graph needs more explanation [...]". Suggestions include adding tooltips to explain elements and adjusting the shapes and colors of elements in the visualizations for better understanding.

The Reliability in Artificial Intelligence (AI) code (2.76%) reflects that users consider the technology employed to be reliable, since they point out that a tool that helps to understand what is going on inside the algorithm is always welcome. P3 stated that "any tool that helps understand what is happening behind the scenes will always be welcome, for sure [...]". The Innovative Tool codes (2.76%) indicate that the system was recognized as innovative due to its visual approach

and real-time integration with the optimization technique's hyperparameters. For instance, P2 stated that the proposed tool is a "[...] tool with various setup options, where you can control the GA online, in real-time [...]".

Therefore, the heuristic evaluation showed that the tool meets the fundamental requirements but has significant room for improvement. Suggestions for improvement focused on optimizing the system, and the presentation of visualizations can positively impact the user experience. In addition, the experts' feedback provides important direction for ongoing adjustments and refinements, ensuring that the solution effectively meets users' expectations and needs.

6 Conclusion and Future Works

We introduced a Human-AI collaborative system for addressing the Maximum Clique Problem (MCP) through a steerable genetic algorithm. The system allows users to guide the evolutionary process in real time, putting the human inside the optimization process and actively changing the algorithm behavior.

Internal testing on Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) benchmark datasets demonstrated that the proposed method identifies acceptable solutions while encouraging the user to collaborate with the algorithm. The results reveal that user manipulations over population size, survival rate and mutation rates are more effective to address this problem compared with available interactions.

Furthermore, a heuristic evaluation conducted with domain experts analyzed the user experience, interaction quality, and perceived control within the system. The results indicated that the tool has a solid foundation in usability and interactivity, with strengths in interface clarity and real-time feedback. However, experts highlighted areas for improvement, such as information overload, the need for clearer explanations, and better onboarding resources. Enhancing visual clarity and expanding guidance materials were key suggestions to improve accessibility and user understanding, particularly for those without prior knowledge.

This tool represents our first investigation about this type of Human-AI interaction, opening up several insights for future directions. As future works, we will investigate new visualizations and interactions; make visualizations interactive, such as creating filters in the search space visualized in the projected scatter plot; investigate the use of different types of meta-heuristics; verify the feasibility of human-machine interaction with the best local searches developed specifically for MCP; compare this approach with adaptive meta-heuristics and include suggestions of interactions for the user based on the these algorithms strategies; and investigate the possibility of other styles of collaboration, such as many IA algorithms or many humans on the same objective.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Amershi, S., et al.: Guidelines for human-AI interaction. In: CHI Conference on Human Factors in Computing Systems. ACM (2019)
3. Balas, E., Yu, C.S.: Finding a maximum clique in an arbitrary graph. SIAM J. Comput. **15**(4), 1054–1068 (1986)
4. Boukhelifa, N., et al.: Evolutionary visual exploration: evaluation of an IEC framework for guided visual search. Evol. Comput. **25**(1), 55–86 (2017)
5. Brailas, A., Tragou, E., Papachristopoulos, K.: Introduction to qualitative data analysis and coding with QualCoder. Am. J. Qual. Res. **7**(3), 19–31 (2023)
6. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**(2), 77–101 (2006)
7. Chae, J., et al.: Visualization system for evolutionary neural networks for deep learning. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 4498–4502. IEEE (2019)
8. Cota Vidaurre, A., et al.: TestEvoViz: visualizing genetically-based test coverage evolution. Empirical Softw. Eng. **27**(7) (2022)
9. Cruz, A., et al.: ELICIT: evolutionary computation visualization. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 949–956. GECCO '15, ACM (2015)
10. Daneshpajouh, H., Zakaria, N.: A clustering-based visual analysis tool for genetic algorithm. In: Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 233–240. SCITEPRESS - Science and Technology Publications (2017)
11. De Lorenzo, A., et al.: An analysis of dimensionality reduction techniques for visualizing evolution. In: Genetic and Evolutionary Computation Conference Companion, pp. 1864–1872. GECCO '19, Association for Computing Machinery, New York, NY, USA (2019)
12. Dolson, E., Ofria, C.: Visualizing the tape of life: exploring evolutionary history with virtual reality. In: Genetic and Evolutionary Computation Conference Companion, pp. 1553–1559. ACM (2018)
13. Farhood, H., Saberi, M., Najafi, M.: Human-in-the-Loop Optimization for Artificial Intelligence Algorithms, pp. 92–102. Springer International Publishing (2022)
14. Farooq, H., et al.: An interactive visualization of genetic algorithm on 2-D graph. In: IEEE 10th International Conference on Cognitive Informatics and Cognitive Computing (ICCI-CC'11), pp. 144–151. IEEE (2011)
15. Hakanen, J., et al.: Interactivized: visual interaction for better decisions with interactive multiobjective optimization. IEEE Access **10**, 33661–33678 (2022)
16. Hettenhausen, J., et al.: Interactive multi-objective particle swarm optimisation using decision space interaction. In: 2013 IEEE Congress on Evolutionary Computation, pp. 3411–3418. IEEE (2013)
17. Huang, B.: Finding Maximum Clique With a Genetic Algorithm. Master's thesis, Penn. State Harrisburg (2002)
18. Ivanov, A., Willett, W., Jacob, C.: EvoIsland: interactive evolution via an island-inspired spatial user interface framework. In: Genetic and Evolutionary Computation Conference, pp. 1200–1208. ACM (2022)
19. Johnson, D.S., Trick, M.A.: Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, October 11-13, 1993, vol. 26. American Mathematical Soc. (1996)

20. Llorà, X., et al.: Analyzing active interactive genetic algorithms using visual analytics. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. GECCO06, ACM (2006)
21. Lutton, E., Fekete, J.D.: Visual Analytics and Experimental Analysis of Evolutionary Algorithms. Tech. rep, INRIA (2011)
22. McPhee, N.F., et al.: Visualizing genetic programming ancestries. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion, pp. 1419–1426. GECCO '16, ACM (2016)
23. Medvet, E., et al.: Unveiling evolutionary algorithm representation with DU maps. Genet. Program Evolvable Mach. **19**(3), 351–389 (2018)
24. Meiguins, A., et al.: Visual analysis scenarios for understanding evolutionary computational techniques' behavior. Information **10**(3), 88 (2019)
25. Mexicano-Santoyo, A., et al.: Visual Analysis of Differential Evolution Algorithms, pp. 512–521. Springer International Publishing (2018)
26. Mosqueira-Rey, E., et al.: Human-in-the-loop machine learning: a state of the art. Artif. Intell. Rev. **56**(4), 3005–3054 (2022)
27. Nielsen, J., Molich, R.: Heuristic evaluation of user interfaces. In: SIGCHI Conference on Human Factors in Computing Systems Empowering People, pp. 249–256. ACM Press (1990)
28. Radhika, P.R., Velayutham, C.S.: Visualization – A Potential Alternative for Analyzing Differential Evolution Search, pp. 31–41. Springer International Publishing (2015)
29. Ramos, G., et al.: Interactive machine teaching: a human-centered approach to building machine-learned models. Hum. Comput. Interact. **35**(5–6), 413–451 (2020)
30. Sathyajit, B.P., Velayutham, C.S.: Visual Analysis of Genetic Algorithms While Solving 0-1 Knapsack Problem, pp. 68–78. Springer International Publishing (2018)
31. Tendresse, I.l., Gottlieb, J., Kao, O.: The effects of partial restarts in evolutionary search. In: International Conference on Artificial Evolution (Evolution Artificielle), pp. 117–127. Springer (2001)
32. Vallerio, M., et al.: An interactive decision-support system for multi-objective optimization of nonlinear dynamic processes with uncertainty. Expert Syst. Appl. **42**(21), 7710–7731 (2015)
33. Walker, D.J.: Visualising multi-objective populations with treemaps. In: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 963–970. GECCO '15, ACM (2015)
34. Walter, M.J., Walker, D.J., Craven, M.J.: An Explainable visualisation of the evolutionary search process. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1794–1802. GECCO '22, ACM (2022)