

ELICIT – Evolutionary Computation visualization

António Cruz
CISUC
Department of Informatics
Engineering
University of Coimbra
antonioac@student.dei.uc.pt

Filipe Assunção
CISUC
Department of Informatics
Engineering
University of Coimbra
fga@student.dei.uc.pt

Penousal Machado
CISUC
Department of Informatics
Engineering
University of Coimbra
machado@dei.uc.pt

António Leitão
CISUC
Department of Informatics
Engineering
University of Coimbra
apleitao@dei.uc.pt

ABSTRACT

ELICIT is a generic tool that enables the visual exploration of evolutionary computation algorithms. It is characterized by the use of simple visual elements to represent information and by the adoption of interactive techniques which allow the navigation between different granularity levels, i.e., it allows the visualization of data from single runs as well as the display of aggregated data, resulting from multiple evolutionary runs. Visualization of lineages is supported, assisting the user in understanding how a given solution was reached. It also provides several visualization modes to inspect the genetic heritage and offspring of individuals and populations. The application was built with the purpose of being capable to deal with different types of representation, allowing the visualization of both genotypes and phenotypes. This paper describes the main visualization modes offered by the tool, presenting examples of its application to tree and graph-based representations.

CCS Concepts

•Human-centered computing → Visualization; Scientific visualization; •Computing methodologies → Genetic algorithms; Genetic programming;

Keywords

Evolutionary Computation, Information Visualization, User Interaction

1. INTRODUCTION

Although Evolutionary Computation (EC) approaches have been successfully applied to many different fields, analysing the evolutionary process and its outcomes is typically a complicated task due to the sheer magnitude of the data.

The use of information visualization techniques may assist the researcher in the detection of meaningful patterns by presenting information through simple and intuitive visual elements. Additionally, the adoption of interactive techniques, that allow the navigation between different levels of detail, may also be valuable to gain insight.

Over the years, several visualization tools for the analysis of EC approaches have been developed in order to provide new points of view over the data and facilitate its analysis [3, 4, 5]. More often than not, these tools tend to focus on the analysis of the evolution of fitness values. Moreover, they tend to be limited in terms of interactivity.

This paper describes ELICIT – EvoLutIonary Computation vISualizaTion – an interactive tool designed to visualize evolutionary processes, which gives particular emphasis to the visualization of the solution formation process. The application is an interactive visualization platform, which allows the navigation between multiple levels of detail, layers of information, and visualization modes by means of simple interactions.

Currently, ELICIT is an offline application, in the sense that it receives as input one or multiple datasets, each of them corresponding to a completed evolutionary run. These datasets supply the necessary information to build the visualization. In detail, for each generation, a dataset includes information on each individual, namely: a representation of its genotype and phenotype, its fitness value, indexes pointing to its parents and which genetic operators were applied to produce it. By gathering this information from all the generations, it becomes possible to recreate the entire evolutionary process and to present it visually and interactively. The representation and aggregation of these features using simple and distinctive elements promotes the apprehension of these large volumes of data. The size and complexity of the data was addressed not only visually but also through interaction, allowing the user to zoom in/out on the data, changing the level of abstraction and detail.

ELICIT was designed to be a generic visualization platform, in the sense that it can be applied to different evolutionary models and representations. It currently supports tree and graph-based representations. For the purpose of illustrating our description, we will resort to datasets concerning the use of tree-based Genetic Programming (GP) for a symbolic regression task [6], and graph-based GP for the evolution of 2D shapes [8].

The paper is structured as follows: Section 2 presents a brief survey of related work in this area; Section 3 provides a general overview of ELICIT, focusing on the nature of the input data, the main visualization modes, levels of abstraction and data aggregation options; Section 4 describes the general view, which provides visualizations regarding the entire evolutionary runs, while Section 5 presents the individual view, which focuses on the visualization of specific individuals; finally, Section 6 presents conclusions and future work.

2. RELATED WORK

Evolutionary algorithms have been used to find solutions to a variety of problems in different fields of study, however, due to their complexity, it is necessary to adjust their parameters in order to guarantee their efficiency as well as effective results for each particular problem. The candidate solutions generated by the algorithm depend not only on the parameters but also on the problem itself and as there is no exact formula to determine these values, it is often necessary to analyse the results of one or several runs of the algorithm and then fine-tune the parameters accordingly. The size and complexity of the data can grow exponentially with the increase in individuals per population and number of generations, which is why visualization tools have been developed over the years in order to provide new ways of viewing and facilitating its analysis. These tools were made with different objectives, sometimes targeting specific problems and, therefore, they vary in functionality, complexity and usability.

Li and Zhang [7] created a tool to help teach how a genetic algorithm works, using a simplified user interface and focusing on the traveling salesman problem (TSP). The program allows the specification of experimentation parameters, such as number of generations and the probability of mutations and crossover. It also allows the users to visualize what occurs during each step of the process and to interact with it. However, the representation is limited to text and lacks graphical elements, which could make it easier for the user to differentiate the individuals and to identify the relationships between them.

Dinis et al. [2] proposed Graph EA, a tool that was also created for educational purposes but that contrasts with Li and Zhang’s approach by having an emphasis on 3D representations. It possesses some of the same functions, although with more elaborate representations, such as monitoring the fitness values and specifying various parameters. Its graphical representations include multiple features such as 2D plots of fitness values throughout the generations, diagrams depicting the traversal of cities in TSP problems or 3D representations of individuals as segmented colored blocks that correspond to genes. Overall, it appears as a more complete tool than the previous one, although some of the 3D representations might seem unnecessary as they do not always take advantage of the extra dimension while increasing the complexity in navigating and analysing the

data.

One of the more complete programs in the field is HeuristicLab (dev.heuristiclab.com), created by created by Wagner and Affenzeller [9], a flexible environment for heuristic and evolutionary algorithms, which was created in 2002 and whose development is still ongoing. It allows the parameterization of a variety of genetic algorithms as well as the visualization of both the individual’s genotypes and the evolution of fitness.

There are also those which have tried to aim for more creative approaches, such as Egger and Karren [3]. They developed a tool capable of identifying and listing the ten fittest individuals of the entire run. Their tool can also represent the best individuals in each generation as a “barcode” which shows which genes pass from generation to generation. However, its main differentiating feature is the representation of the evolution process which affects each individual, referred to as an “evolution map”. This visualization is generated by drawing geometrical shapes to represent the various operators which led to the creation of a selected individual. These shapes reflect properties of the individuals through shape and size, but their position is randomized leading to what the authors describe as “pictures you might encounter in a museum of modern art”, that can be visually interesting but difficult to interpret.

3. OVERVIEW OF ELICIT

This section provides a general description of ELICIT. It is important to notice that ELICIT is an offline visualization tool, i.e., it allows the user to visualize and explore evolutionary process that, typically, have already been concluded. As such, ELICIT does not include a specific evolutionary engine, instead it is fed by previously gathered data.

The data is passed to ELICIT through several external data-files, which should adhere to a specific file format. For the purpose of the paper it is sufficient to say that each data-file corresponds to a single evolutionary run and contains information on all the individuals generated during the evolutionary process of that specific run, such as its genotypes, associated fitness values, indexes pointing to their parents and information on which genetic operators were used to generate them. ELICIT handles and aggregates data from multiple evolutionary runs, organized in separate files, provided that they have the same number of generations and individuals per population.

To display the phenotypes associated with each genotype the user may provide them as separate image files, stored in specific folders and following a pre-defined naming scheme, or supply a genotype/phenotype method that performs the interpretation of the phenotypes and their mapping.

ELICIT offers two main levels of detail:

General View – Provides an overview of the data regarding the entire run;

Individual View – Provides in-depth information regarding a specific individual;

The user may switch between these modes by clicking on the corresponding tab (see Figure 1). The functionalities associated with these two levels of detail will be described in the next two sections.

It is important to keep in mind that ELICIT is an interactive tool and that the responsiveness of the tool, the

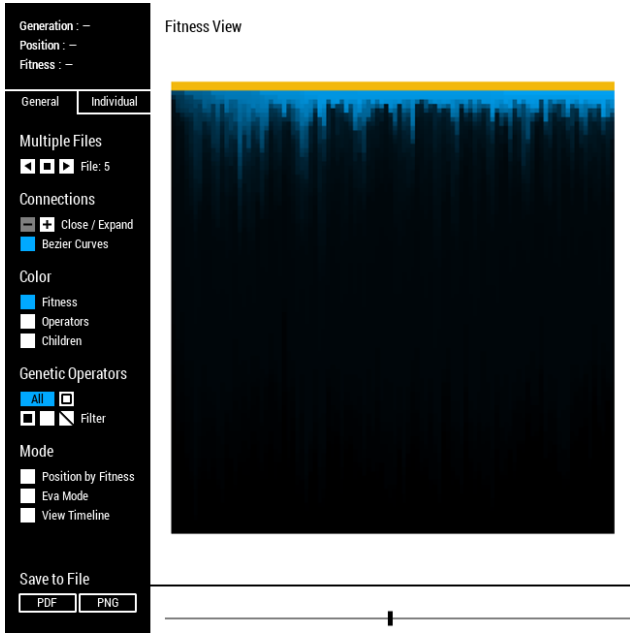


Figure 1: Interface of ELICIT while on the general view visualization mode.

feedback given to the user and the opportunities for rapidly exploring the data are its greatest assets. As such, it is difficult to convey the overall feeling and user experience of the users through a paper with static images.

As it can be observed in Figure 1, the screen is split into two main areas: on the left, the control buttons and check boxes that allow the user to access and change between different options and visualization modes; on the right, the visualization area. The interaction with the visualization can be done mostly through the use of the mouse, by dragging the panel, selecting components or zooming in and out using the mouse wheel. By choosing the save to file option, the current visualization will be fully exported into PNG or PDF format, including any parts that might be off-screen, while retaining any of the selected options.

4. GENERAL VIEW

When entering the general view, by default, the visualization will depict an overview of the EC process. Each individual is represented as a square belonging to a specific column, which represents a population. Columns are chronologically lined according to their generation from left to right. Within each column, individuals are ranked by fitness from top to bottom (fittest first). Therefore, the top left square represents the fittest individual in the first population and the bottom right square represents the least fit of the last population. A color gradient is used to indicate fitness, ranging from a light blue to black (Figure 1).

The user may choose to focus on a specific evolutionary run or to visually aggregate and visualize the data from all runs. This is accomplished by using the arrow symbols on the top left (see Figure 1) to browse through runs, or the square symbol to aggregate the results. The possibility to aggregate results is available for all visualizations created in the general view. Figure 2 depicts the visualization of the

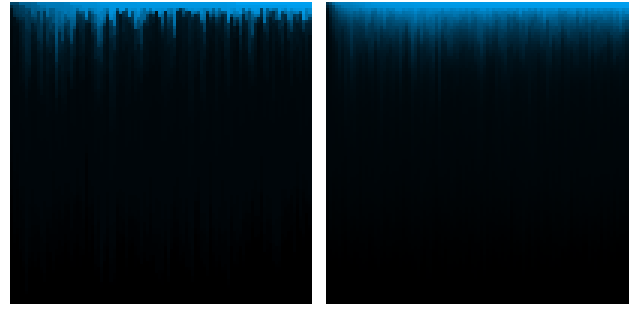


Figure 2: Visualizing the distribution of fitness of a single evolutionary run (left) and of multiple runs using visual aggregation of the results (right).

evolution of fitness through generations with and without aggregation of results. In order to calculate a color for each individual, all runs and generations are considered, meaning that the fittest individual will correspond to the brightest blue and the less fit to pure black. When viewing a specific run, individuals are simply represented using their color while on the aggregate view each cell's color will average the color of the individuals that it merges.

By default individuals at each generation are ranked by fitness, however, we also give the user the possibility of depicting the individuals using fitness proportionate y coordinates. As it can be observed in Figure 3, this results in a visualization that is comparable to the conventional charts depicting the evolution of fitness through time. There are, however, two important differences: (i) instead of depicting the best and average fitness values for each population, ELICIT draws all individuals of the population, arguably allowing the user to get a better grasp of the spread of fitness values; (ii) when aggregating results from multiple runs, it will also draw all individuals of all runs. The squares corresponding to individuals will however be drawn using transparency so that as they overlap they become darker, thus allowing the user to grasp the number of overlapping individuals and the variance of fitness through multiple runs.

Although it would be easy to mathematically calculate the fitness values typically presented in charts (such as average fitness and maximum fitness), we prefer to aggregate the data visually, presenting the maximum amount of information to the user. Since we are dealing with a large amount of information and taking into account that users are typically not familiarized with these visualization schemes, it may require some practice to successfully interpret and analyze the outcome of our tool. However, our conviction is that these visualizations are valuable complements and alternatives to typical charts, offering different opportunities for analysis and insights.

In the visualizations shown so far in Figures 1 and 2, color was used to represent fitness. In addition to this option, ELICIT also allows the user to use color and shape to represent the number of offspring produced by each individual as well as the genetic operators applied.

Figure 4 depicts the number of offspring of the individuals corresponding to each grid position, averaged over multiple runs. Similarly to fitness, a brighter blue indicates a higher number of descendants. Although color is used to represent the number of offspring, individuals are still ranked verti-

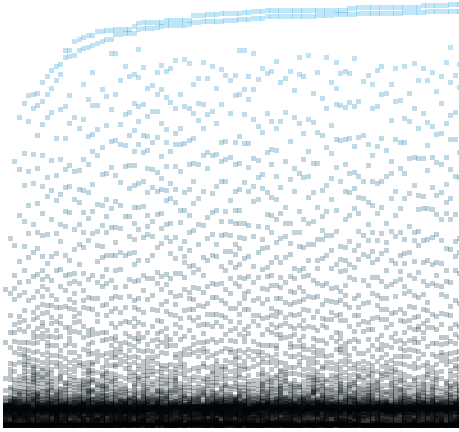


Figure 3: Visualizing the distribution of fitness using fitness proportionate y coordinates.

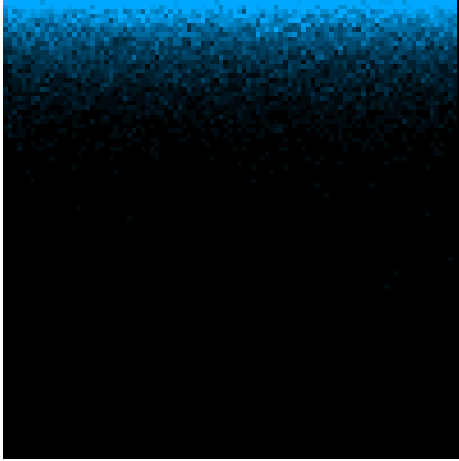


Figure 4: Visualizing the number of children of an average of multiple files.

cally accordingly to their fitness. As it can be observed, and as would be expected, the fittest individuals tend to produce most of the offspring.

Figure 5 depicts the genetic operators used to create each individual through symbols. A filled blue square indicates that a mutation occurred, a filled black square represents a copy operation, a diagonal slash represents crossover, and an empty square with a black outline indicates that crossover and mutation occurred simultaneously. The user can also filter the visualization for any of these options, depicting only the individuals that were created using the selected operator, turning the visualization into a simple binary color scheme.

The visualization of genetic lineages takes advantage of the interactive capabilities of ELICIT. By selecting an individual using the left mouse button, the user is able to see the genetic lineage associated with it (ancestors and descendants). As Figure 6 shows, this option is available both when visualizing individual runs or aggregating results from several runs. Selecting an individual in the aggregate view

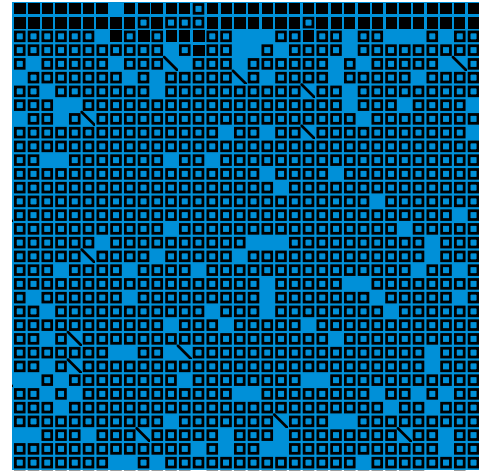


Figure 5: Visualizing a close-up of the genetic operators of a single file, which are represented with symbols.

will show the lineages of every individual in that position. That is accomplished using transparencies to overlap individuals in the same cell. The selected individual is depicted in orange and individuals that do not belong to its lineage are greyed out.

By clicking on the top orange row (see Figure 1) the user can select an entire column, i.e., an entire population, and depict the lineage associated with all individuals of that population. Figure 7 presents an example where the last population was selected, thus allowing us to observe all the individuals whose genetic code contributed, somehow, to the selected population.

In addition to the visualization of the lineages of individuals and populations, we are also interested to find and visualize the closest common ancestor of a set of individuals, namely of a given population. As the name indicates, the closest common ancestor is the individual that is an ancestor to all individuals in a given population, while minimizing the distance in generations. Algorithm 1 details how the closest common ancestor is found¹. In this procedure, *selected* corresponds to the index of the selected population, *explore* to the index of the first population to explore (which should be initially set to *selected* - 1) and *ccas* to a list of the closest common ancestors found. In ELICIT this option is called *eva* mode and can be accessed through a button on the left panel of the interface.

As it can be observed in Figure 8, the distance between the population and its closest common ancestor varies significantly from run to run. The analysis of the causes and impacts of such a variation is however outside the scope of the present paper.

Although the visualizations presented so far have the advantage of being compact, they fail to depict explicitly the lineage connections among individuals. To do so the user may use the expand button on the left panel (see Figure 1). This creates a space between the populations' columns and draws Bezier lines among progenitors and descendants. All the visualizations described so far can also be depicted in this mode. The interaction with the visualization, namely

¹If ties exist, more than one individual is returned.

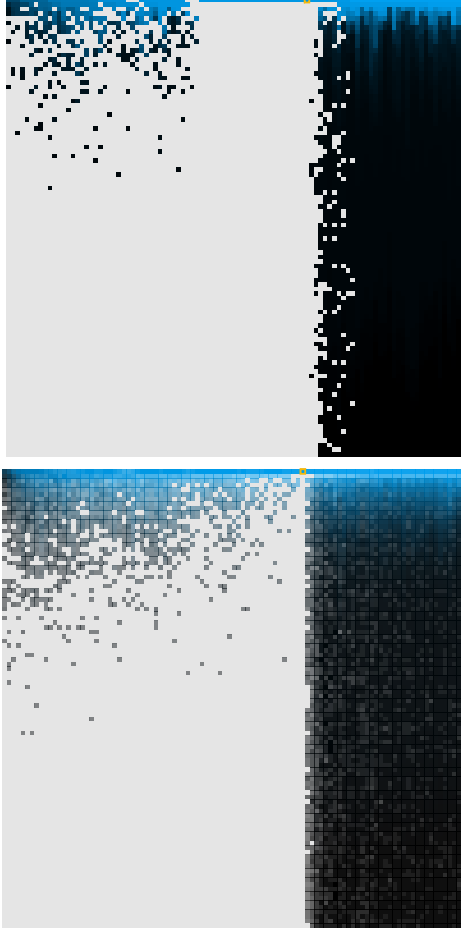


Figure 6: Genetic lineage of an individual for a single evolutionary run (top) and for an aggregation of multiple runs (bottom).

Algorithm 1 Closest Common Ancestor

```

procedure EVA(exploration, selected)
  ccas  $\leftarrow$  NewEmptyList()
  for ind  $\in$  populationexploration do
    if all populationselected individuals are descendants
    of ind then
      ccas.add(ind)
    end if
  end for
  if |ccas| > 0 then
    return ccas
  else if exploration > 0 then
    return EVA(exploration - 1, selected)
  else
    return NULL
  end if
end procedure

```

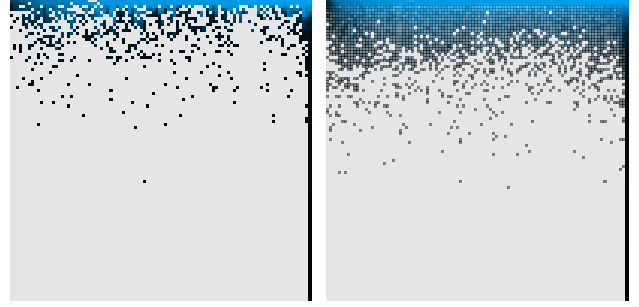


Figure 7: Genetic lineage of the individuals of the last population for a single run (left) and for an aggregation of multiple runs (right).

the selection of individuals, zooming in/out and panning left and right are performed as previously. Additionally, a scroll bar on the bottom can be used to center the visualization on any specific population, facilitating navigation when using this expanded visualization mode.

Figure 9, similarly to Figure 6, depicts the genetic lineage of an individual using the expanded view mode. Likewise, Figures 10 and 11 depict, respectively, the genetic lineage and the closest common ancestor of a population and are analogous to Figures 7 and 8. By explicitly rendering the connections among individuals, additional information is given to the user.

5. INDIVIDUAL VIEW

As previously mentioned, ELICIT offers a second level of detail, an individual view which focuses on the visualization of the details and genetic lineage of a single individual. To access this mode the user begins by selecting an individual in the general view mode, and then picking the individual view tab on the left panel of the visualization window (see Figure 1).

The way genotypes and phenotypes are visualized depends on their structure and nature. Currently, ELICIT deals with tree-based and graph-based genotypes. Concerning the visualization of the phenotypes, the user has three options: (i) using one of the predefined phenotype visualization modules; (ii) supplying a new phenotype visualization module; (iii) supplying images corresponding to the rendering of the phenotype in a chosen format of his/her choice. In the present version, ELICIT only supplies a specific phenotype visualization module for symbolic regression tasks. It is important to note that, despite the fact that currently only images are being addressed as phenotypes, ELICIT can be easily generalised to deal with other formats.

We use two example problems illustrate the properties of the individual view: a classical symbolic regression problem, as described in Anonymous [6]; and an evolutionary art task, dealing with the evolution of 2D shapes, proposed by Machado et al [8]. In the first case the genotypes are tree-based structures mapped to real valued functions, for which we use ELICIT’s built in visualization module for symbolic regression. In the second case the genotypes are extended graphs. We use ELICIT’s graph visualization module for depicting the genotypes and user supplied images to display the phenotypes. These images are supplied to ELICIT following a predefined naming scheme and directory structure.

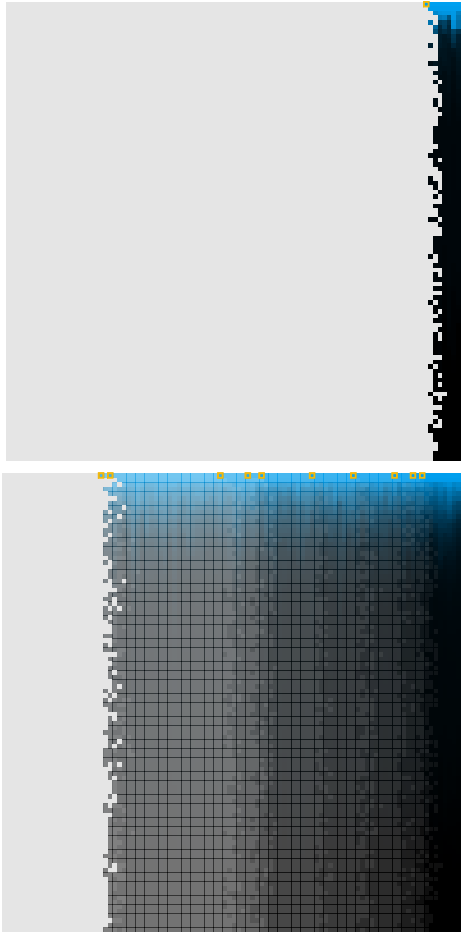


Figure 8: Visualization of the closest common ancestor of the last population for a single evolutionary run (top) and for an aggregation of multiple runs (bottom). The closest common ancestors are depicted by an orange square.

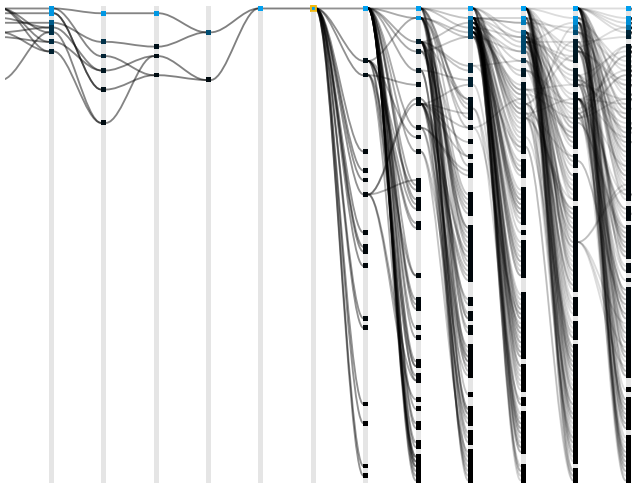


Figure 9: Visualization of the genetic lineage of an individual using the expanded view mode. The selected individual is highlighted in orange.

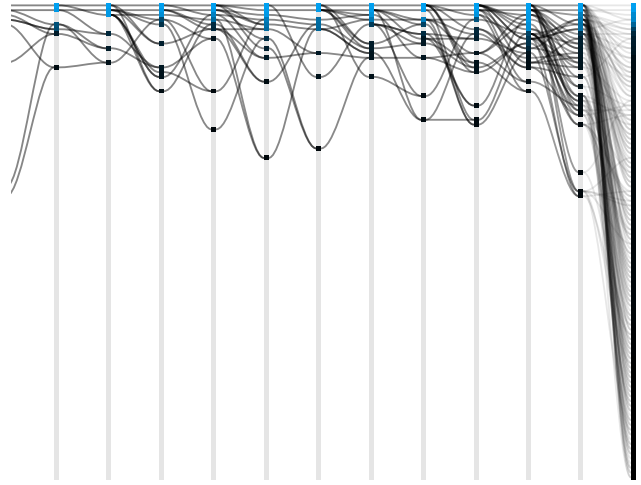


Figure 10: Visualization of the genetic lineage of a population using the expanded view mode.

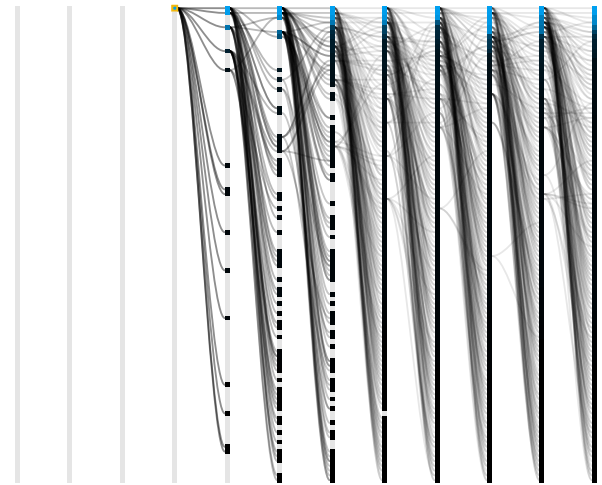


Figure 11: Visualization of the closest common ancestor of a population using the expanded view mode.

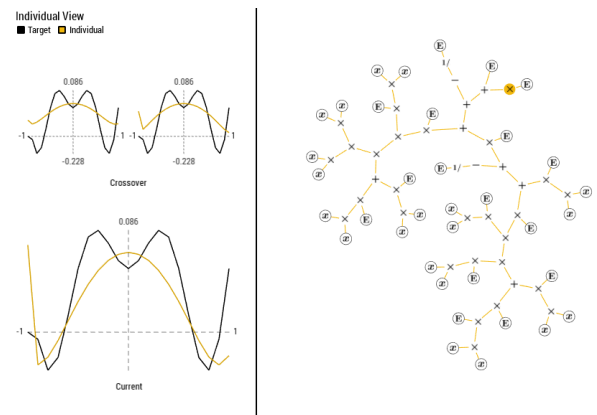


Figure 12: Visualization of a tree-based individual evolved for addressing a symbolic regression problem.

Figure 12 depicts an example of the visualization of a tree-based individual evolved in the course of a symbolic regression task. The screen is split into three areas: the interface panel, on the left; the visualization of the phenotypes, in the middle; and the visualization of the genotype, on the right.

As it can be observed, the visualization depicts the phenotypes of the parent(s) and of the individual, indicating the genetic operation, or operations, that were applied. The target function is rendered in black while the functions resulting from the interpretation of the phenotypes are depicted in orange.

To visualize the tree-structures, ELICIT creates a physics-based simulation. In simple terms, each node is a particle, edges are springs and nodes repel each other. Although this option may be unconventional, it allows us to depict large tree structures in a compact fashion.

The user can interact with the visualization of the genotypes by using the mouse to drag it and to zoom in and out. The reset position button allows the user to reposition the genotype to its default position and zoom level.

One of our main goals when developing the individual view was to allow the user to inspect and visualize the solution construction process. In other words, to analyse and understand how evolution created a given solution. To this end we focused on developing an interactive method to visualize the lineage of an individual.

When an individual is selected in the general view, its entire lineage is also, indirectly, selected (see Figure 9). This lineage is used to create a timeline which is built by recursively selecting the fittest progenitor, as described in Algorithm 2. In what concerns the addition of the descendants of the individual to the timeline, we begin by selecting the fittest individual of the last generation of the lineage (notice that this is not necessarily the last population of the run, since the lineage may end sooner). Then, we recursively build the timeline from that descendant to the selected individual by applying the same algorithm. By following this procedure we ensure that the timeline encompasses the full range of the individual’s lineage.

Algorithm 2 Recursive creation of the timeline

```

procedure ANCESTOR_TIMELINE(ind, timeline)
  timeline.add(ind)
  l ← progenitors(ind)
  if l ≠ NULL then
    f ← fittest(l)
    return ancestor_timeline(f, timeline)
  else
    return timeline
  end if
end procedure

```

The bottom scroll bar in this visualization is used to navigate forward and backwards through the current timeline, advancing to the fittest ancestor or descendant in the previous or next generation. It is also possible to navigate the timeline using the left and right buttons on the keyboard or on the interface panel. The panel also contains a play button that advances the timeline forward automatically, with brief intervals between frames, producing an animation.

These interaction possibilities allow the user to quickly visualize the changes that have occurred in the phenotype and in the genes between two directly related individuals. To

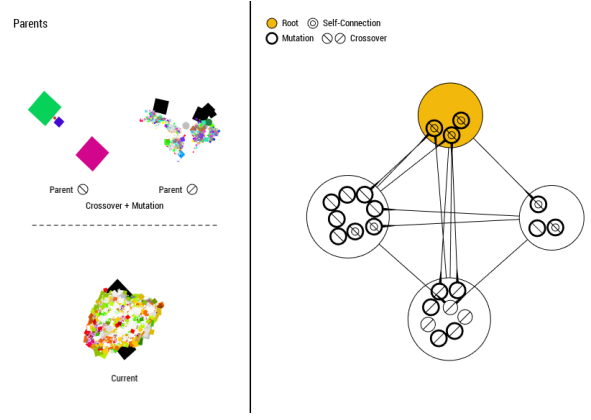


Figure 13: Visualization of a graph-based individual evolved for the creation of 2D shapes.

enhance the visualization of the changes when browsing the timeline, the modified genes and added nodes are depicted in orange. Additionally, when drawing a genotype, the initial positions of the unmodified nodes are inherited from the genotype previously on screen. Then, the physics simulation engine quickly adapts to the structural changes provoked by the removal or addition of nodes. The inheritance of positions and the quick adaptation to changes have the effect of highlighting structural changes in the genotype, guiding the eye of the user to affected portions of the genome.

As a second example of the individual view we focused on the visualization of graph-based genotypes. ELICIT is prepared to handle the visualization of standard graphs, directed graphs, and hypergraphs (directed or otherwise). In a hypergraph, we have nodes and hypernodes, with hypernodes being a set of nodes and corresponding edges.

In the case presented in this paper, each hypergraph represents a Context Free Design Grammar (CFDG) [1]. Each node is a production rule indicating an expansion rule for a non-terminal symbol of the grammar. CFDGs are non-deterministic and, as such, different expansion rules may be applied to the same non-terminal symbol. As such, the set of all production rules, i.e., nodes, associated with a given non-terminal are encapsulated in an hypernode. The edges between nodes correspond to “calls” in the production rules to non-terminal or terminal symbols. Additionally, each edge is annotated with a label indicating the parameters that may be associated with a given call. The representation scheme, genetic operators used to manipulate the representation, and fitness functions used to guide evolution are fully described in the original work by Machado et al. [8].

The graphs are drawn with simple elements, where circles represent the nodes and lines with small arrows at their beginning represent their relationships and the direction of the connection between nodes. The call from one node to itself is represented through a small circle inside the circle’s node. As previously, this is also a force directed layout which pulls connected nodes closer together through springs, but does not allow nodes to overlap. Additionally, the physics engine also forces the nodes to stay within the limits of the corresponding hypernode. The root node is depicted in orange.

The visualization also provides information regarding the origin of each node, as follows: a thick outline indicates that

the node was affected by a mutation; a slash indicates that the node results from crossover. The direction of the slash (“\” or “/”) indicates from which of the parents the node was inherited.

The hypernodes and nodes of the graph can be selected with the left mouse button in order to display additional information about each one, such as a list of the nodes they are connected with, and the production rule(s) in text format. Double clicking on an hypernode allows the user to expand/contract it, showing or hiding the associated nodes.

As with the symbolic regression example, the phenotype of the individual and of its parents is also depicted. In this case, ELICIT fetches the images that were rendered during the evolutionary runs from external files. Although it would be easy to provide a module to render the phenotypes in real-time, we chose this option to promote the responsiveness of the interface since rendering can be time-consuming.

The construction and visualization of the timeline is not affected by the details of the representation and, therefore, is similar to the previous example.

6. CONCLUSIONS AND FUTURE WORK

This paper presents ELICIT, an evolutionary Computation Visualization tool that enhances the visual exploration of the behavior of evolutionary algorithms. Such tools have been previously proposed as a means to help researchers have a better grasp on what happens during the evolutionary process.

ELICIT proposes a higher interactivity than previous approaches as well as a greater emphasis on the visualization of the solutions formation process, by allowing the navigation between multiple levels of details, different layers of information and various visualization models. This is accomplished using simple means of interaction.

In summary, ELICIT is currently an offline tool that receives input from one or multiple EC runs through external datasets supplying the information necessary for visualization, thus making it possible to recreate and present entire evolutionary processes visually and interactively. It supports the visualization of individual runs or the aggregation of data from multiple runs, as well as behavior assessment using two levels of detail, a general view of the evolution process and an individual view, where details on specific individuals can be better explored. Additionally, it supports viewing and analysing the genetic lineage of individuals interactively on multiple ways as well as displaying genetic and phenotypic information.

Presently, ELICIT and most of its features support generic application to multiple problems and datasets, assuming the format rules are met. Visualization of genotypes and phenotypes are an exception as it depends on their structure and nature. ELICIT provides modules for visualizing tree-based and graph-based genotypes but also allows the users to include new modules for other representations. Regarding phenotypes, ELICIT includes a module to map and visualize symbolic regression tasks. It is up to the user to provide alternative mapping modules for other purposes. Alternatively, users can provide images of rendered phenotypes, which ELICIT loads, thus allowing a general application with no specific module requirements.

Along this paper we have shown how to interact with ELICIT, which views are available and what features they include. We have covered how users can successfully take

the most benefit from the application when analysing evolutionary processes by illustrating two examples. The first regards a symbolic regression task and the second one regards a graph-based approach for the creation of 2D shapes. We have also shown that ELICIT can be easily extended for other applications through the coupling of new modules.

Future work includes adding support for more genotype representations as well as for mapping alternative phenotypes. Moreover, we believe that including new visualization models covering other metrics such as diversity measures can become a relevant feature for further analysis of algorithm behavior. Furthermore, we are currently using ELICIT on different problems and want to make it available to a larger user base in order to test it on other scenarios and to increase its support.

Acknowledgments

The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

7. REFERENCES

- [1] C. Coyne. Context Free Design Grammar. <http://www.chriscoyne.com/cfdg/>, last accessed in February 2014.
- [2] R. Dinis, A. Simões, and J. Bernardino. Graphea: a 3d educational tool for genetic algorithms. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pages 1293–1300. ACM, 2013.
- [3] T. Egger and A. Kerren. Visualization tool for evolutionary algorithms. Interactive Poster, ACM SoftVis 03, San Diego, CA, USA, 2003.
- [4] E. Hart and P. Ross. Gavel-a new tool for genetic algorithm visualization. *Evolutionary Computation*, *IEEE Transactions on*, 5(4):335–348, 2001.
- [5] A. Kerren and T. Egger. Eavis: A visualization tool for evolutionary algorithms. In *Visual Languages and Human-Centric Computing, 2005 IEEE Symposium on*, pages 299–301. IEEE, 2005.
- [6] A. Leitão, J. C. Neves, and P. Machado. A self-adaptive mate choice model for symbolic regression. In *IEEE Congress on Evolutionary Computation*, pages 8–15, 2013.
- [7] J. Li and Z. Zhang. Notice of retraction a learning tool of genetic algorithm. In *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, volume 1, pages 443–446. IEEE, 2010.
- [8] P. Machado, J. Correia, and F. Assunção. Graph-Based Evolutionary Art. In A. Gandomi, A. H. Alavi, and C. Ryan, editors, *Handbook of Genetic Programming Applications*. Springer, Berlin, 2015.
- [9] S. Wagner and M. Affenzeller. *Heuristicslab: A generic and extensible optimization environment*. Springer, 2005.