

## Genetic Graph Drawing

A. Rosete<sup>a</sup> & A. Ochoa<sup>b</sup>

<sup>a</sup>*Department of Information Science, CEIS, ISPJAE,  
Marianao 19390, Havana, Cuba*

*EMail: rosete@informatica.ispjae.edu.cu*

<sup>b</sup>*Artificial Intelligence Centre, Institute of Cybernetics,  
Mathematics and Physics, 15 e/ C y D, Havana, Cuba*

*EMail: ochoa@cidet.icmf.inf.cu*

## Abstract

The drawing of graphs is widely recognized as a very important task in diverse fields of research and development. Examples include VLSI Design and plant layout. Also, many CASE methodologies and interactive data-intensive applications should perform graph drawing and should manage a large amount of constraints including semantic ones. On the other hand, when the information that must be visualized is changing constantly, is necessary to develop automatic methods to partially (or totally) redraw the diagrams. We want to obtain drawings that are optimal in the sense of readability. However, is difficult to create a general-purpose graph drawer and is also computationally expensive. Genetic algorithms (GAs) have recently received a growing interest and have been applied in many design problems. In this paper we describe why the GAs-based approach is a good alternative to obtain a general flexible tool for Graph Drawing Problems, dealing also with semantic aesthetics. Free Form Deformation (FFD) has been used in structural design problems in combination with GAs. We also present an approach to automatic graph drawing based on GAs and FFD, for improving of a given layout by exploring its neighborhood. The search space is reduced and the readability of the drawings is improved. Also the Mental Map is almost preserved.

## 1 Introduction

Graphs are something very popular in the science family and the drawing of them is widely recognized as a very important task in diverse fields of research and development. For instance, many CASE methodologies use diagrams in its stages of analysis and design<sup>5,18</sup>. Cooperative interactive data-intensive applications process a lot of information that should be presented in form of charts, diagrams, or drawings. These systems should manage a large amount of constraints including semantic ones. On the other hand, this information is changing constantly. For this reason, is necessary the development of automatic methods to partially (or totally) redraw the diagrams. In this context, our aim is to obtain optimal drawings in the sense of readability, i.e. ease to understand due to the way it is drawn<sup>5,18,30</sup>. Many algorithms for drawing graphs have been proposed so far. Extensive surveys have been conducted in<sup>5,18,30</sup>. However, is difficult to create a general-purpose graph drawer and is also computationally expensive. It is very frequently a large-scale optimization task. The field of genetic algorithms has been growing since the early 1970s, but only recently GAs have yielded real-world applications that demonstrate commercial potential<sup>7</sup>. In this paper we present an approach for automatic graph drawing based on genetic algorithms. The principal justifications for the genetic approach in this context, are:

- We get a more general, robust and unique method to almost all kinds of diagrams.
- We can exploit the exploration capabilities of GAs in dealing with complex and large search spaces.
- We are dealing with structural and semantic drawing rules in the same simple manner. This means, we are more capable to consider the meaning of the representations. We believe that the semantic aspects of this problem have not yet received the necessary attention in the specialized literature<sup>5</sup>.
- We can combine the advantages of both genetic and numerical optimizations by some sort of hybridization technique<sup>2,8,11,19,24</sup>.

In this paper we illustrate the potential of the proposed approach. Also, we analyze its bad sides and propose a solution based on Free Form Deformation, a structural design technique<sup>31</sup>.

## 2 State of the art

As is stated in<sup>2</sup>, the problem of drawing a graph nicely can be regarded as searching for an optimal layout of a given graph according to some measurable aesthetics. In<sup>30</sup> the authors make a classification of algorithms for automatic graph drawing. They classify the algorithms attending to:

- Class of graphs (tree, planar, directed, etc)
- Graphical standard (straight-line, grid, hybrid, etc)
- Constraints (structural, semantic)
- Aesthetics

Some commonly used aesthetics are:

- A1: minimizing total area
- A2: minimizing the amount of bends
- A3: (line-crossing) minimizing the amount of edges crossings
- A4: uniform density of vertices

Some commonly used constraints are:

- C1: to place a set of vertices in the center
- C2: to draw a sub-graph with a given shape
- C3: to place a sequence of vertices along a straight line

A complete classification and a survey of the most important job in this field can be obtained in<sup>5</sup>. This paper outlines some open problems for Automatic Graph Drawing including:

- General Strategy for Straight Line Drawings
- Dynamic Drawing Algorithms

We consider that these objectives depend on a robust approach that separates as most as possible the desire from the method. Many of the approaches consist of algorithms that solve efficiently a kind of problem but are not easily applied for solving a different one. In this paper we intend to illustrate the possibilities of the genetic approach to become a general tool for solving this kind of problem.

### 3 A genetic based approach

Genetic Algorithms (GAs) are modern optimization methods inspired by the principles of the natural selection and genetics. GAs reduce the solution for optimization problem to simulate a natural environment where many individuals (i.e. solutions) fight for staying alive. GAs have three main processes: Selection, Crossover and Mutation.

For a complete discussion about GAs and Evolutionary Algorithms we suggest to read<sup>1,7,20</sup>. It is well known that a genetic algorithm to solve a particular problem need a representation of problem solutions in terms of chromosomes and genes, a population initialization procedure, a suitable cost function, some genetic operators to evolve the population and a vector of GA parameters. Therefore, GAs-based approach for automatic graph drawing must define a strategy to give meaning to the above components. In our most general case, chromosomes (individuals) are chains like:

$$X_1, Y_1; \dots X_i, Y_i; \dots X_n, Y_n$$

where each  $X_i$  and  $Y_i$  is a bit string and  $n$  is the number of nodes. The tuple  $\langle X_i, Y_i \rangle$  represents the coordinate of a vertex in some system. Our representation is based on integers and it is a difference respect to<sup>2</sup>. This representation may be changed to accommodate specific kinds of graphs. For instance, one coordinate may be not relevant (i.g. for hierarchies<sup>22,25</sup>), or it can represent a relative value<sup>25</sup>. For the later case, we have found very effective to consider a variation of the Free Form Deformation (FFD) technique<sup>31</sup>. Sometimes in the evolution process, we may want to deal with individuals that are close to some specific patterns. This is also a good choice to deal with constraints like C1 and C2. On the other hand, we have experimented mainly with random initialization with the aim of obtaining a general tool. A source of initial individuals can be the results of numerical optimization techniques if we have a hybrid system, as is frequently suggested for GAs<sup>2,11,24</sup>. Besides, we use conventional genetic operators in our experiments.

Automatic graph drawing can be seen as a maximization optimization problem. We use a fitness function like:

$$F = K_1 * F_1 + K_2 * F_2 + \dots K_j * F_j + \dots + K_t * F_t$$

where  $t$  is the number of aesthetic criteria, each  $K_j$  is a coefficient related to the importance of each criteria and each  $F_j$  measures a

| Coefficient | Function | Meaning                                                         |
|-------------|----------|-----------------------------------------------------------------|
| $K_{o-}$    | $F_{o-}$ | Minimizing node overlappings                                    |
| $K_{o+}$    | $F_{o+}$ | Maximizing node overlappings                                    |
| $K_{c-}$    | $F_{c-}$ | Minimizing line-crossings(aesthetic A3)                         |
| $K_{c+}$    | $F_{c+}$ | Maximizing line-crossings                                       |
| $K_{l-}$    | $F_{l-}$ | Minimizing the global edges length<br>(related to aesthetic A1) |
| $K_{l+}$    | $F_{l+}$ | Maximizing the global edges length                              |

Table 1: Aesthetic criteria: coefficients, functions and meanings

constraint or aesthetic satisfaction. This is not the only alternative for combining aesthetic-criteria. Also, could be interesting to consider the problem as a multi-objective optimization problem<sup>25,26</sup>. As many of the aesthetics criteria could be opposed, it's important to express the real objective of the optimization task. This is the moment when our approach is tailored to one specific problem. The coefficients, related functions, and its meaning are in Table 1.

Corresponding with each one of the previous coefficients exists a function that measures the accomplishment of each criteria. So, each function assigns a value computing the rates between the quantity of the criterion in the evaluated graph drawing and the maximum possible value of the criterion for that graph. On the other hand, we estimate the maximum value for the global edges length as a bound value, maybe never achievable. We set this maximum bounded value by multiplying the maximum length of an edge by the number of edges. The maximum length of an edge is the square root of the sum of the squares of both coordinates resolution. We say that this a bounded maximum because in many graph is impossible to draw a graph in a way that all its edges are as long as possible. For a function that evaluates the minimization of a criterion the function value is calculated as saved values. For instance, the value of  $F_{c-}$  is the rate between the quantity of line crossings saved in a graph drawing and the maximum saved value of this criterion. For combining many criteria in the objective function we use a linear combination of all functions. So, specifying the previous formula, the objective function is as follows:

$$F = K_{o-} * F_{o-} + K_{o+} * F_{o+} + K_{c-} * F_{c-} + K_{c+} * F_{c+} + K_{l-} * F_{l-} + K_{l+} * F_{l+}$$

| Identifier | Meaning                                            |
|------------|----------------------------------------------------|
| EX         | Identifier of each experiment                      |
| NR         | Number of runs made for each experiment            |
| MG         | Maximum Number of generations for the GA           |
| NI         | Number of individuals in each generation           |
| PC         | Crossover Probability                              |
| PM         | Individual mutation Probability (not bit mutation) |
| TC         | Type of crossover used in the GA                   |
| TS         | Type of selection scheme for the GA                |
| PO         | Percent of the runs where the optimum is obtained  |
| AB         | Average of the best values obtained in all runs    |
| WB         | Worst of the best values obtained in all runs      |
| BB         | Best of the best values obtained in all runs       |

Table 2: Convention: identifiers and meanings

In the previous formula,  $K_{o-}, K_{o+}, K_{c-}, K_{c+}, K_{l-}, K_{l+} \in [0, 1]$  and also,  $F_{o-}, F_{o+}, F_{c-}, F_{c+}, F_{l-}, F_{l+} \in [0, 1]$ . Of course, is not possible to try to attain contradictory criteria (e.g.  $F_{o-}$  and  $F_{o+}$ ) in the same optimization task. So, the value of the coefficient for opposite criteria will never be positive coincidentally. Besides, for ensuring that  $F \in [0, 1]$ , we add a new constraint to the coefficients:

$$K_{o-} + K_{o+} + K_{c-} + K_{c+} + K_{l-} + K_{l+} = 1$$

This constraint permits to stop the genetic search when a solution with maximum evaluation value (equals to 1) is obtained.

As GAs are stochastic methods we did some runs in order to compute average values. Following, we describe the characteristics of the experiments. In all cases, we use simple GA, non steady-state<sup>7</sup>. By convention, we identify the parameters of the GA and the results in each experiment using the identifiers in Table 2. The software package used to implement the GA is GAL, an object-oriented package designed by the authors<sup>23</sup>. The main characteristic of GAL is its large library of genetic algorithm objects and its ease of customization. GAL was augmented with a graph analyzer module which implement the problem-dependent classes.

| EX | Functions involved       | Coefficients                                 |
|----|--------------------------|----------------------------------------------|
| A1 | $F_{o-}, F_{c-}$         | $K_{o-} = 0.25, K_{c-} = 0.75$               |
| A2 | $F_{o-}, F_{c-}, F_{l-}$ | $K_{o-} = 0.15, K_{c-} = 0.75, K_{l-} = 0.1$ |
| A3 | $F_{l-}$                 | $K_{l-} = 1.0$                               |
| A4 | $F_{l+}$                 | $K_{l+} = 1.0$                               |

Table 3: Experiments with a simple planar graph

## 4 Drawing a Simple Planar Graph

The graph used in the following experiments is an extension of the graph presented by Nishizeki and Chiba<sup>21</sup> as an example of planar graph that haven't convex drawing. So, the algorithm presented in<sup>21</sup> can't draw it.

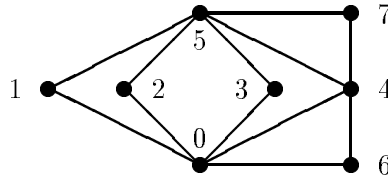


Figure 1: A simple planar graph

As observed in the Figure 1, this graph has 8 nodes and 12 edges. The nodes 6 and 7 aren't in the original graph, neither the edges adjacent to these nodes. In the following experiments, graph nodes can be placed in a grid with 8 coordinates in  $X$ -axis and 4 coordinates in the  $Y$ -axis. Consequently, for coding one node we need 5 bits and the length of the chromosome is 40 bits. Next, we summarize the experiments we did for illustrating the robustness of the proposed approach. The experiments, functions involved and coefficients are in Table 3, and the GA parameters are in Table 4. Unif corresponds to the uniform crossover originally proposed by Syswerda. LS corresponds to the linear scaling, and Cmult is the expected number of copies of the best individual in the mating pool, as is defined by Goldberg<sup>7</sup>.

| EX | NR | MG | NI | PC  | PM   | TC   | TS              |
|----|----|----|----|-----|------|------|-----------------|
| A1 | 30 | 60 | 60 | 0.8 | 0.16 | Unif | LS, Cmult = 1.4 |
| A2 | 10 | 30 | 60 | 0.8 | 0.16 | Unif | LS, Cmult = 1.4 |
| A3 | 20 | 30 | 60 | 0.8 | 0.16 | Unif | LS, Cmult = 1.4 |
| A4 | 10 | 30 | 60 | 0.8 | 0.16 | Unif | LS, Cmult = 1.4 |

Table 4: GA parameters for drawing a simple planar graph

| Experiment | PO | AB     | WB    | BB    |
|------------|----|--------|-------|-------|
| A1         | 93 | 0.9992 | 0.988 | 1.0   |
| A2         | -  | 0.9564 | 0.946 | 0.967 |
| A3         | 0  | 0.85   | 0.767 | 0.872 |
| A4         | -  | 0.7821 | 0.641 | 0.849 |

Table 5: Experimental results for a simple planar graph

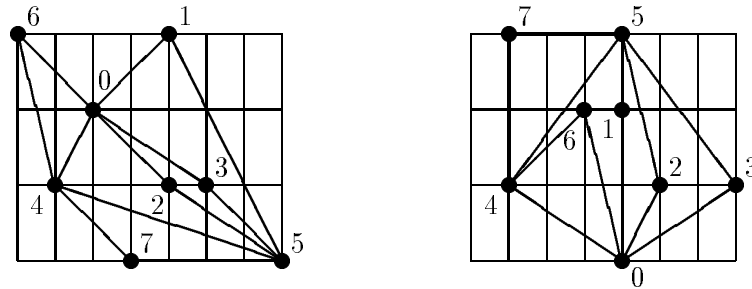


Figure 2: One of the layouts obtained for experiment A1(left).  
One of the layouts obtained for experiment A2(right).

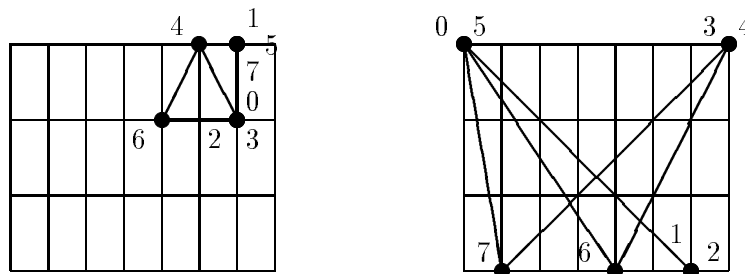


Figure 3: The best of the layouts for experiment A3(left).  
One of the best layouts for experiment A4(right).

The experimental results are in Table 5. Note that in experiments A2 and A4 the optimum value is unknown so we don't know when

| EX | Functions involved | Coefficients                 |
|----|--------------------|------------------------------|
| B1 | $F_{o-}, F_{c-}$   | $K_{o-} = 0.3, K_{c-} = 0.7$ |
| B2 | $F_{c+}$           | $K_{c+} = 1.0$               |
| B3 | $F_{o-}, F_{c+}$   | $K_{o-} = 0.3, K_{c+} = 0.7$ |

Table 6: Experiments with a simple non-planar graph

it's found. In the experiment A3, the optimum value corresponds to the overlapping of all nodes in one identical position. Unfortunately, it is never obtained. Results can be improved with more generations, but we think that the speed is also very important in this kind of problems<sup>8,15,28</sup>. Figures 2 and 3 show examples of the layouts obtained in the experiments. The achievement of each objective is remarkable.

## 5 Drawing a Simple Non-Planar Graph

The graph used in the following experiments is one of the simplest non-planar graphs<sup>21</sup>, i.e. is impossible to draw without line crossings. So, we can modify the way the maximum number of saved line-crossings is calculated. Instead of doing so, we maintain the same previous criteria for evaluating  $F_{c-}$ . Consequently,  $F_{c-}$  never gets a value equal to 1. The graph is known as K5 and is shown in Figure 4. This graph has 5 nodes and 10 edges. In the following experiments, the nodes of the graph can be placed in a grid with 4 coordinates in  $X$ -axis and 4 coordinates in the  $Y$ -axis. Consequently, for coding one node we need 4 bits, and the length of the chromosome is 20 bits. In Table 6, we summarize the experiment we did for showing the robustness of the proposed approach, and the coefficient we used. GAs parameters are in Table 7.

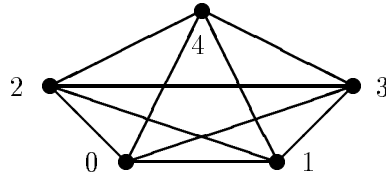


Figure 4: K5, one of the littlest non-planar graphs

| EX | NR | MG | NI | PC  | PM   | TC   | TS                   |
|----|----|----|----|-----|------|------|----------------------|
| B1 | 10 | 40 | 50 | 0.8 | 0.16 | Unif | LS, with Cmult = 1.4 |
| B2 | 10 | 20 | 30 | 0.8 | 0.16 | Unif | LS, with Cmult = 1.4 |
| B3 | 10 | 20 | 30 | 0.8 | 0.16 | Unif | LS, with Cmult = 1.4 |

Table 7: GA parameters for drawing a simple non-planar graph

| Experiment | PO  | AB     | WB    | BB    |
|------------|-----|--------|-------|-------|
| B1         | 100 | 0.984  | 0.984 | 0.984 |
| B2         | 50  | 0.966  | 0.93  | 1.0   |
| B3         | -   | 0.8649 | 0.84  | 0.876 |

Table 8: Experimental results for a simple non-planar graph

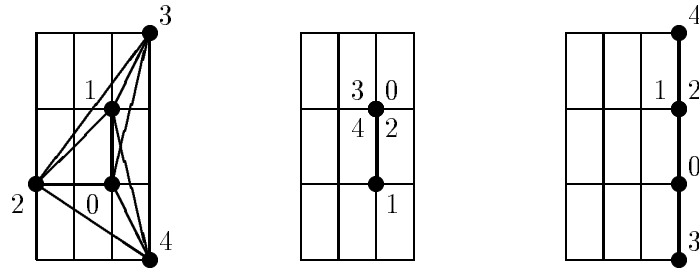


Figure 5: Three good layouts obtained for graph K5 by genetic search: B1(left), B2(center) and B3(right).

The experimental results are in Table 8. It must remarked that in experiments B3 the optimum value is unknown so we don't know when it's found. In the experiment B1, the optimum value corresponds to the solution having just one line crossing. In this case the value of  $F_{c-}$  is 0.984. As this solution is obtained in all the runs we consider  $PO = 100\%$ , unless  $F_{c-}$  is never equal to 1.0. Figure 5 shows examples of the obtained layouts for the experiments.

## 6 Exploring the Neighborhood of a Layout

Although the algorithm performance in very good we notice that when the size of graph grows the time of the algorithm to converge increases. This problem is related to the evaluation of the objective function. When more edges exist, increasing number of comparison between edges is necessary to determine the number of line crossing

in the drawing. We are investigating many techniques to estimate this value instead of calculate it<sup>27</sup>. Consequently, runtimes will be reduced. On the other hand, when the complexity of the graph increases, is necessary to embed it in a bigger space. For this reason, the problem is harder to solve due to the increase of chromosomes size. We noticed that in few generations the algorithm achieve a great advance in the quality of the drawing but converge to not optimal solution near the optimal one. In this point we develop a method<sup>25</sup> inspired in FFD<sup>31</sup> (Free Form Deformation). In all these cases, the solutions are not necessary to be modified so far to obtain the optimum. This situation was the inspiration of the following experiment. It deals with the exploration of the neighborhood of a layout, with the hope of finding a better one. What we code is not the absolute coordinate of the nodes. Instead we code a relative magnitude, i.e. an offset respect to the position of the node in the layout we want to explore around. So, now the chromosomes are as follows:

$$DX_1, DY_1; \dots DX_i, DY_i; \dots DX_n, DY_n$$

In order to obtain the absolute coordinate of the drawings we have the coordinates of the nodes in the original (explored around) graph:

$$IX_1, IY_1; \dots IX_i, IY_i; \dots IX_n, IY_n$$

Now, the coordinates of each node can be calculated as:

$$X_i = IX_i + DX_i; Y_i = IY_i + DY_i; i = 1, 2, \dots, n$$

This solution is inspired in FFD, a structural-design technique used for deformations studies of objects. This technique use a discrete description of the objects by means of control points that can be moved in the space for obtaining deformations of the original object<sup>31</sup>. We consider this method as a good way to face the problem of Mental Map preservation<sup>6,17</sup>, also for doing dynamic drawings<sup>3,15,28</sup>. This is a typical problem in many CASE tools with evolving diagrams that are frequently modified but never in a disruptive way. In these cases, is convenient to maintain the new drawing as close to the original as can be possible. In our approach, this objective can be achieved by restricting the feasible value for  $DX_i$  and  $DY_i$ . In order to define the explored neighborhood we define a magnitude called *DELTA*. *DELTA* is the maximum possible variation of each node in both coordinates respect to the original layout. It's closely related to the size

| EX | Functions and $DELTA$               | Coefficients                                      |
|----|-------------------------------------|---------------------------------------------------|
| C1 | $F_{o-}, F_{c-}, DELTA = 2$         | $K_{o-} = 0.25, K_{c-} = 0.75$                    |
| C2 | $F_{l-0}, DELTA = 1$                | $K_{l-} = 1.0$                                    |
| C3 | $F_{o-}, F_{c-}, F_{l-}, DELTA = 1$ | $K_{o-} = 0.15, K_{c-} = 0.75,$<br>$K_{l-} = 0.1$ |
| C4 | $F_{c-}, F_{l-}, DELTA = 2$         | $K_{c-} = 0.9, K_{l-} = 0.1$                      |
| C5 | $F_{o-}, F_{c-}$                    | $K_{o-} = 0.1, K_{c-} = 0.9$                      |

Table 9: Experiments by exploring the neighborhood of a layout

| EX   | NR | MG    | NI | PC    | PM    | TC   | TS              |
|------|----|-------|----|-------|-------|------|-----------------|
| C1-4 | 20 | 10    | 20 | 0.8   | 0.16  | Unif | LS, CMult = 1.4 |
| C5   | 20 | 20/10 | 30 | 0.9/0 | 0.1/1 | Unif | LS, Cmult = 1.4 |

Table 10: GA parameters for exploring the neighborhood of a layout

of the search space. That means,  $DX_i, DY_i \in [-DELTA, +DELTA]$  Conceptually,  $DELTA$  can be different for each node, also for each coordinate, but in all the following experiments we use the same value at all. We consider that this exploration can be view as a codification-based exploration

Another technique for exploring the neighborhood, is dynamic adjustment of the GAs parameters. In this sense, we consider the GA run as two (multiple, in general) runs. Each run has different parameter values, but maintains the population obtained until this moment. We suppose that in the last runs, is better to raise the influence of the mutation in the GAs search. The intention is to explore the neighborhood of the best layouts we have obtained. Besides, as all the layouts are very similar due to the convergence, crossover is less important. Consequently, the crossover probability will be decreased. We consider this approach as a parameters-based exploration. See an example in the experiment C5. More complex and complete discussion about parameters and operators adjustment can be found in<sup>9,10,12,13</sup>.

The graph used as reference in the following experiment is one of the layouts obtained for the experiment A1. It is shown in Figure 6 (left). We did some experiments exploring the neighborhood of the layout presented before. The experiments and coefficients are in Table 9. The parameters of the GA in the experiments are in Ta-

| Experiment | PO  | AB     | WB    | BB    |
|------------|-----|--------|-------|-------|
| C1         | 100 | 1.0    | 1.0   | 1.0   |
| C2         | 0   | 0.724  | 0.688 | 0.757 |
| C3         | -   | 0.9654 | 0.953 | 0.971 |
| C4         | -   | 0.959  | 0.937 | 0.972 |
| C5         | 50  | 0.99   | 0.959 | 1.0   |

Table 11: Experimental results for a simple planar graph

ble 10. In the case of C5, the couple of values in columns MG, PC and PM means the values for the related parameters before and after the change is done.

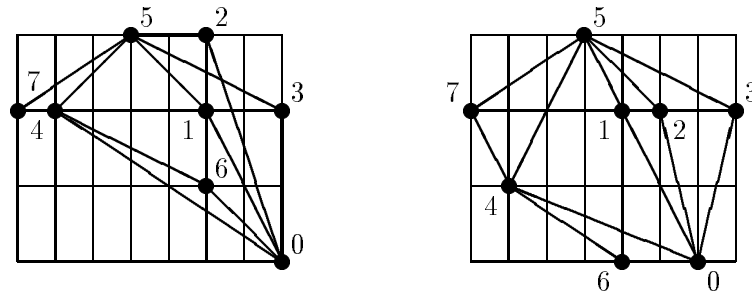


Figure 6: The layout used for the experiment as reference for the initial nodes positions, and consequently, it's the layout which its neighborhood will be explored(left). One of the obtained layouts for experiment C1(right).

Experimental results are in Table 11. Note that in experiments C3 and C4 the optimum value is unknown so we don't know when it's found. In the experiment C2 is important to remark that the optimum value is impossible to reach from the initial layout using the value of DELTA assigned. In experiment C2, C3 and C4 is remarkable the minimization of the global length of the edges respect to experiment C1, which doesn't consider this aesthetic. Figure 6 (right) shows one example of the result of experiment C1. This layout is considered as reference for the minimization of the global length of the vertices in the examples of the Figures 7, 8 and 9. Note in Figure 8 and in Table 11 that the results of experiment C3 and C4 are very similar. This result could be caused by the little space to search. Besides, the best result for accomplish the three criteria needs to put

the nodes as close as possible but taking line-crossings and nodes-overlappings into account. Is easy to note that with  $DELTA = 1$  it's possible to almost obtain the best results. See in Figure 8, in both cases, is reached a better layout that reduces the global length (respect to the layout obtained for C1 in Figure 6, right) and also maintains the optimality respect to the crossings and overlappings criteria.

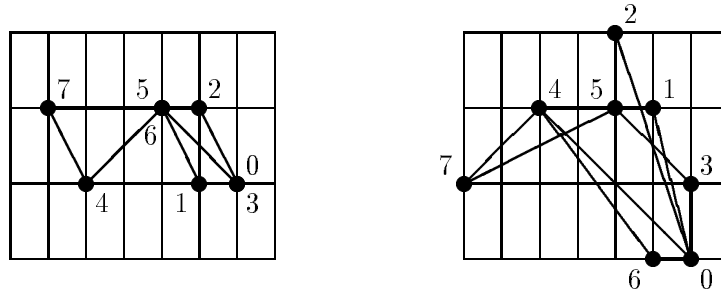


Figure 7: Two of the best obtained layout for experiment C2: the best of all the runs(left) wich reduces the global length to 67.77% (respect to the layout obtained in C1, Figure 6, right) and the worst of all runs(right), wich reduces the same magnitude to 85.55%. Note that no other aesthetic is taken into account.

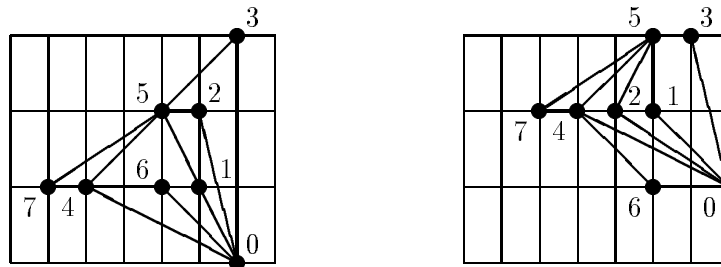


Figure 8: Two of the best results for experiments C3 and C4: one of the results for C3 (left), wich reduces the global length to 79.59% (respect to the layout of Figure 6, right) and one of the results of C4(right), wich reduces the same magnitude to 78.82%

Concerning the last experiment (C5) is important to note that the results are not better than the obtained results for A1. Specially, is very worst respect to PO. However, in experiment C5

a considerable amount of layouts evaluations is saved. Only 900 solutions are explored. On the other hand, experiment A1 explores 3600 solutions. Consequently, the results of C5 are quickly obtained and not bad results at all. This method could be used for obtain fast results, something important (e.g. in CASE tools) to permit efficient user work<sup>8,15,28</sup>. At last, we think that exploring the neighborhood (using codification-based exploration or parameters-based exploration) is a solution for preserving the Mental Map. This is an important aspect for graphical interface applications that change continuously its contents, i.g. CASE tools during the development of Information Systems. It's convenient to maintain the representation as close as possible to the previous one when a modification (e.g. inserting nodes, deleting edges) was done. Our approach would permit to set the freedom of the search and consequently the maximum modification we allow.

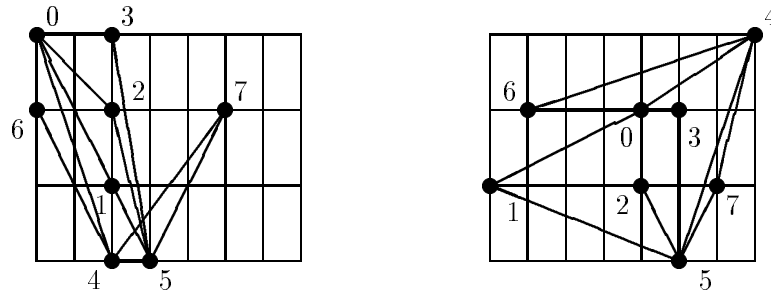


Figure 9: Two of the best layout obtained for experiment C5: the best(right) and the worst(left) among all runs.

## 7 Robustness of the proposed method

In many papers the graph drawing problems are solved using deterministic algorithms that found optimal or near optimal solutions for fixed aesthetics and kinds of diagrams<sup>5,18,21,30</sup>. However, these methods are only appropriate for a very restrictive class of graph and esthetics. The genetic-based method is not another algorithm. It can be considered as another approach for the problem. Deterministic algorithms consist of precise procedures for solving a given wish. In the genetic-based approach only the wish is needed. This wish is what we have to express as the GA objective function. GAs then find a good solution for this wish function. As is shown above, we successfully explored the proposed method in planar and not planar graphs. Also

we analyzed the good performance of the approach in hierarchical<sup>22,25</sup> and not hierarchical graphs. Besides, we considered many aesthetics (i.g. line crossing, edge lengths, nodes overlapping). We searched both maximization and minimization of the mentioned aesthetics. We consider our approach robust because it can be easily tailored to a specific graph-drawing optimization task. It performs successfully and reaches satisfying results. We consider our work near to<sup>4,16</sup> where Simulated Annealing (SA) technique is applied. The main difference in concerning with the difference between the optimization methods used, but both can be accepted as possible alternative to the open problems presented in<sup>5</sup>. In the case of GAs the possibilities of becoming parallel can be considered as a serious advantage<sup>2,25,26</sup>, but also a lot of research has been done for becoming SA parallel<sup>14</sup>. In<sup>2</sup> very good result are obtained in more complex graphs by using GAs. We also think that GAs speed has to be taken into account. SA is very slow as is admitted by SA practitioners<sup>8,14</sup>. Another technique, having important general-applicability characteristics is the Spring Model and its extensions<sup>5,29</sup>, but we consider that GAs improve this method in the sense of robustness. The proposed approach become a good alternative for dealing with semantic aesthetics. Many techniques, including but not restricted to Neural Networks can be used to model semantic aesthetics in the drawing. A semantic aesthetic can be related to a kind of diagram or a user specific preference. In our approach is just necessary the evaluation of the semantic aesthetic satisfaction. Then, GAs can be used in a very similar way to explore the space and look for a good solution<sup>25,27</sup>. We consider this as an important advantages of the proposed approach.

## 8 Conclusions

This paper presents a genetic-based approach for automatic graph drawing, which emphasizes aspects of human visual cognition like readability and aesthetics. This contribution is the first step towards the implementation of programs for drawing graphs supporting commercial methodologies. We make experiments with simple graphs and the results allow us to think that GA technology is a very strong candidate to solve this kind of problems. We illustrate how a relative simple schema can yield excellent results in contrast to others cumbersome heuristic methods. The robustness of the proposed approach is

appreciated in many experiment with different criteria and also with combination of them. The difficulties of the approach respect to the growing size of the space explored is solved using a method inspired in Free Form Deformation. This solution is very useful to improve the quality of a given drawing, and is a good alternative to keep the Mental Map of a Diagram, as is wished in many graphical user-interfaces. Besides, the application of this method do not reduce the robustness of the genetic-based approach. We tested our approach in many readability optimization problems and different kinds of graphs with good results. The applicability of the method can be important for other fields such as VLSI Design and plant layout.

## References

- [1] Back, T., *Evolutionary Algorithms : Comparison of approaches, Computing with Biological Metaphors* ed. R. Paton, Chapman&Hall, London, 1994.
- [2] Branke, J., Bucher, F. & Schmeck, H., Using Genetic Algorithms for Drawing Undirected Graphs, *University Karlsruhe (TH), AIFB, Bericht 347*, 1996.
- [3] Cohen, R.F., Di Battista, G., Tamassia, R., Tollis, I.G. & Bertolazzi, P., A Framework for Dynamic Graph Drawing, *Proc. of ACM Symposium on Computational Geometry*, ACM Press, pp. 261-270, 1992.
- [4] Davidson, R. & Harel, D., Drawing Graph Nicely Using Simulated Annealing, *ACM Trans. Graphic*, **15**(4), pp. 301-331, 1996.
- [5] Di Battista, G., Eades, P., Tamassia, R. & Tollis, I.G., Algorithms for automatic graph drawing: An annotated bibliography, *Computational Geometry*, **4**, pp. 235-282, 1994.
- [6] Eades, P., Lai, W., Misue, K. & Sugiyama, K., Preserving the Mental Map of a Diagram, *Proc. of COMPUGRAPHIC'91*, Portugal, pp. 34-43, 1991.
- [7] Goldberg, D.E., *Genetic Algorithms in Search Optimization and Machine Learning* Addison-Wesley, Reading:MA, 1989.

- [8] Harel, D. & Sardas, M., Randomized Graph Drawing with Heavy - Duty Preprocessing, *Journal of Visual Languages and Computing*, **6**, pp. 233-253, 1995.
- [9] Herrera, F. & Lozano, M., Adaption of Genetic algorithms parameters based on Fuzzy Logic Controllers, *Genetic Algorithms and Soft Computing* ed. F. Herrera & J.L. Verdegay, Physica-Verlag, Spain, 1996.
- [10] Herrera, F., Lozano, M. & Verdegay, J.L., Dynamic and Heuristic Fuzzy Connective-Based Crossover Operators for Controlling the Diversity and Convergence of Real-coded Genetic Algorithms, *International Journal of Intelligent Systems*, **II**, pp. 1013-1040, 1996.
- [11] Julstrom, B.A., Seeding the Population : Improved Performance in a Genetic Algorithm for Rectilinear Steiner Problem, *Proc. of the 1994 ACM Symposium on Applied Computing* eds. E. Deaton, D. Oppenheim, J. Urban & H. Berghel, ACM Press, New York, pp. 222-226, 1994.
- [12] Julstrom, B.A., An Inquiry into the Behavior of Adaptive Operator Probabilities in Steady-State Genetic Algorithms, *Proc. of the Second Nordic Workshop on Genetic Algorithms and their applications, 2NWGA* ed. J.T.Alander, Vaasa, Finland, pp. 15-26, 1996.
- [13] Julstrom, B.A., Adaptive Operator Probabilities in a Genetic Algorithms that applies three operators, *Proc. of the 1997 ACM Symposium on Applied Computing, SAC'97* eds. B. Bryant, J. Carroll, D. Oppenheim, J. Hightower & K.M. George, ACM Press, New York, pp. 233-238, 1997.
- [14] Kurbel, K., Schneider, B. & Singh, K., VLSI Standard Cell Placement by Parallel Hybrid Simulated Annealing and Genetic Algorithm, *Proc. of the International Conf. On Artificial Neural Networks and Genetic Algorithms, Ales* eds. D.W. Pearson, N.C. Steele, R.F. Albrecht, Springer-Verlag Wien New York, pp. 491-494, 1995.
- [15] Lueders, P. & Ernst, R., The Dynamic Screen- Beyond the Limits of Traditional Graphic User Interfaces, *Proc. of 13th IFIP*

*World Computer Congress 1994, Vol. 1*, Hamburg, Germany, 1994.

- [16] Lueders, P., Stille, S. & Ernst, R., An Approach to Automatic Display Layout using Combinatorial Optimization Algorithms, *Software Practice and Experience*, **25(11)**, pp. 1183-1202, 1995.
- [17] Misue, K., Eades, P., Lai, W. & Sugiyama, K., Layout adjustment and the Mental Map, *Journal of Visual Languages and Computing*, **6**, pp. 183-210, 1995.
- [18] Misue, K., Nitta, F., Sugiyama, K., Koshiba, T. & Inder, R., Techniques for DUI Platforms : Developing Graph Drawing Applications on D-Abductor, *Inst. For Social Information Science, Fujitsu, ISIS-RR-96-7E*, Japan, 1996.
- [19] Miyata, Y., Furuhashi, T. & Uchicawa, Y., A proposal of a New Local Improvement Mechanism for Genetic Algorithm and a Creative Thinking Support, *Proc. of International Conf. On Evolutionary Computation, ICEC'95*, University of Western Australia, Perth, Australia, 1995.
- [20] Muehlenbein, H. & Schlierkamp-Voosen, D., Predictive Models for the breeder genetic algorithm I. Continuous parameter optimization, *Evolutionary Computation*, **1**, pp. 25-49, 1993.
- [21] Nishizeki, T. & Chiba, N. *Planar Graphs: Theory and Algorithms* Elsevier Science Publishers B.V., North Holland, Amsterdam, 1988.
- [22] Ochoa, A. & Rosete, A., Automatic graph drawing by genetic search, *Proc. of 11th ISPE/IEE/IFAC International Conference on CAD, CAM, Robotics & Factories of the Future, CAR&FOF'95* Pereira, Colombia, pp. 982-987, 1995.
- [23] Ochoa, A., Rosete, A. & Ponce, E., GAL: an object-oriented library for genetic search and optimization, *Institute of Cybernetic, Math. and Physics, Havana, T.R. 95-04*, Cuba, 1995.
- [24] Powell, D. J., Skolnick, M. M. & Tong, S.S., Interdigitation: A Hybrid Technique for Engineering Design Optimization Employing Genetic Algorithms, Experts Systems, and Numerical Optimization, *Handbook of Genetic Algorithms* ed. L. Davis, Van Nostrand-Reinhold, New York, pp. 313-331, 1991.

- [25] Rosete, A. & Ochoa, A., Trazado automatico de grafos utilizando algoritmos geneticos, *CREPIAI, ISPJAE, Havana, Tesis de Maestria*, Cuba, 1995.
- [26] Rowe, J., Vinsens, K. & Marvin, N., Parallel Genetic Algorithms for Multi-objective functions, *Proc. of the Second Nordic Workshop on Genetic Algorithms and their applications, 2NWGA* ed. J.T. Alander, Vaasa, Finland, pp. 30-40, 1996.
- [27] Soto, M.R. & Ochoa, A., Evaluacion Parcial en algoritmos geneticos, *CREPIAI, ISPJAE, Havana, Tesis de Maestria*, Cuba, July 1997.
- [28] Stille, S., Minocha, S. Ernst, R., A2DL- An Adaptive Automatic Display Layout Systems, *Proc. of the IEEE 3rd Annual Symposium on Human Interaction with Complex Systems , HICS'96*, IEEE Computer Society Press, Dayton, Ohio, 1996.
- [29] Sugiyama, K. & Misue, K., Graph Drawing by Magnetic Spring Model, *Journal of Visual Languages and Computing*, **6**, pp. 217-231, 1995.
- [30] Tammassia, R., Di Battista, G. & Batini, C., Automatic graph drawing and readability of diagrams, *IEEE Trans. Syst., Man, Cybern.*, **18**, pp. 61-79, 1988.
- [31] Watabe, H. & Okino, N., A study on Genetic Shape Design, *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, ed. S. Forrest, Morgan Kaufmann, San Mateo, pp. 445-450, 1993.