Automatic drawing of octalinear maps for public transportation: using Genetic Algorithms

Author
Marcelo de Lima Galvão[1]
Department of Computer Science - Universidade de Brasília

Advisers
Prof. Dr. Marcus Vinicius Lamar
Department of Computer Science - Universidade de Brasília
Prof. Dr. Pastor Willy Gonzales Taco
Transport Post-Graduation Program - Universidade de Brasília

Keywords: *automated cartography, genetic algorithm, graph drawing, subway map, public transportation*.

1 Introduction

This article presents results from using genetic algorithms to draw octalinear graphs. This type of diagram, commonly used in subways around the world, was first conceived by Henry Beck in 1933. Figure 1 shows its first publication. The idea of Beck's Map was to restrain the maps layout in order to communicate information in the most efficient way. Its main feature consists in restraining orientation of the line segments in four orientations (vertical, horizontal and both diagonals). The map became a world-class design and nowadays it is the main pattern to represent collective ways of transportation [1994].

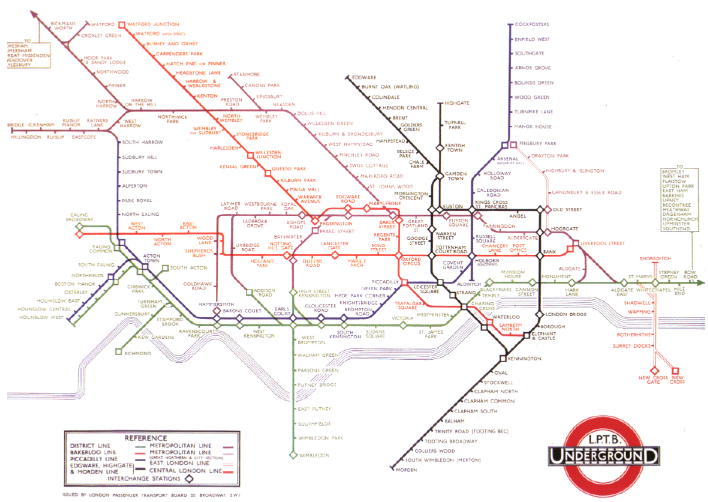


Figure 1: Tube Map - First official publication of an octalinear map, 1933 [2006]

Even though it can be applied to public transportation, there are no commercial software that can automatically generate maps of this type. They would have to be hand made by professional designers. According to Nöllenburg [2005], the problem to draw octalinear maps is NP-hard, which

[1]**Contact:** galvaobrasil@gmail.com **address:** SHIS QI 28 conjunto 02 casa 27, Brasília-DF Brasil 71670220 **Phone Fax:** 55 61 33676855

explain the difficult to find this kind of software in the market.

To solve the problem of how to make the drawing automatic, it was developed an algorithm capable of schematize graphs in an octalinear manner. The problem was partially solved by using an optimized search engine known as genetic algorithms. This solution can be considered unique, the use of genetic algorithms for schematizing octalinear maps has not been yet identified in previous studies. To test this algorithm, it was used Brazils capital (Brasília, D.F) public service transportation data. The results obtained were proven adequate once they managed to be obtained in short periods of time.

2 Developing

Genetic algorithms are inspired by natural selection. This means that problems are solved through an evolution process which returns an adequate solution. Basically, what a genetic algorithm does is create an initial population of possible solutions to the problem so they can be used to breed a new generation of solutions. This process is motivated by the expectancy that a new population will have better solutions that its ancestors [1999].

An important step in developing a genetic algorithm is choosing how to encode each individuals chromosome. Individuals from a given population are the fundamental units of a genetic algorithm. They are represented by chromosomes and identify possible solutions to a given problem. At this present study, an octalinear line is considered an individual. Due to the geometric nature of the problem, the most intuitive way of encoding a chromosome is using the cartesian coordinated system of the stopping points of the line. So to facilitate other operation involved in a genetic algorithm, instead of using Cartesian coordinate system, it was used a sort of relative polar coordinate system.

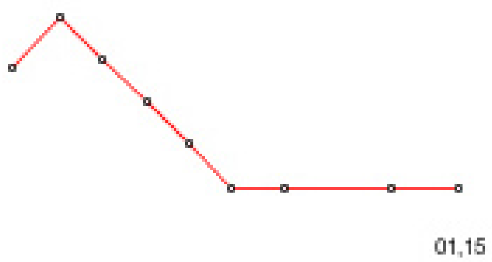


Figure 2: Octalinear line.

Cromossomo
(15.0, 36.0)
(25.0, 45.0)
(21.55, 315)
(21.55, 315)
(21.55, 315)
(21.55, 315)
(19.13, 0)
(37.64, 0)
(24.48, 0)

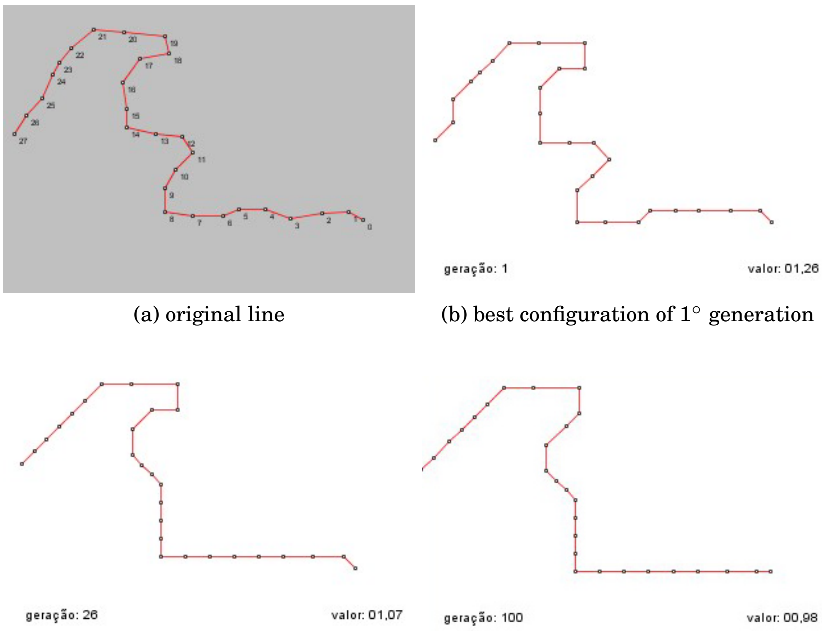
Figure 3: Line chromosome of figure 2.

Relative polar coordinate system represents one line with n points, selecting as origin point p_0 one of the extremes of the line. p_0 is represented by its Cartesian coordinate, i.e., its values (x, y). The only point adjacent to p_0, p_1, it is represented by the polar coordinate relative to p_0, i.e., a value (r, Θ) where r represents the distance between p_0 and p_1, and Θ the angle formed by the line that unites the points p_0 and p_1 and the axis x. The other point adjacent to p_1, p_2, is represented by relative polar coordinate to p_1, i.e., a value (r, Θ). This representation made by a relative polar coordinate goes to the point p_n forming a line chromosome. Figure 2 illustrates and example of a line with a chromosome value.

After encoding the chromosome, the next step is to define the other basic operations of a genetic algorithm. Crossover and mutation will guarantee diversity for each generation. Evaluation operation consists in evaluating adaptation or rejection of an individual as a possible solution to a problem. Chromosome evaluation is made through fitness function, which interprets it and returns a value that will represent chromosome quality.

The ideal solution, according to the rules of the octalinear map, strives for perfect balance between the minimum bends of the line and a coherent positioning of the points in a way that the line will not have much distortion. Therefore evaluation operation tends to appreciate the level in which those aesthetic characteristics are respected. Fitness function was developed to obtain an index that, the lower the value, the better the line will be in regards to this balance..

3 Results



(a) original line

(b) best configuration of 1° generation

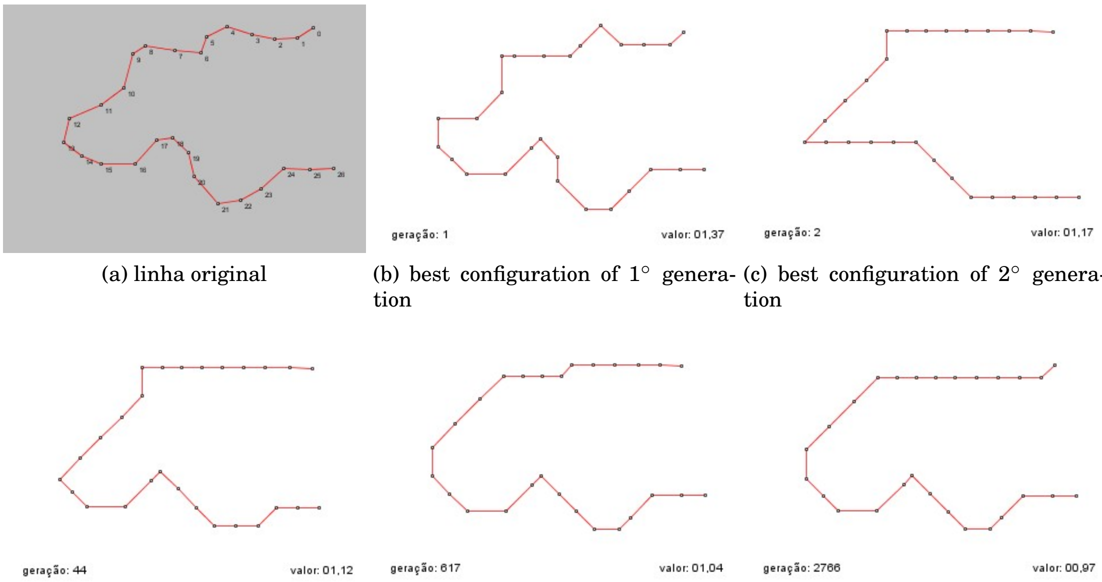
(c) best configuration of 26° generation

(d) final result. 100° generation

Figure 4: Evolution of schematization 1. Results achieved in 0.23 seconds.

The genetic algorithm for line schematization was tested in a software developed exclusively for this. The software allows the user to freely create lines and submit them to schematization. The experiment allows individuals from each generation to be visualized and thus to evaluate algorithm behavior.

In order to present evolution in the schematization, a group of lines differently shaped was created and submitted to the schematizing process. For each line, every time the best solution of a given generation has been altered from the previous generation the configuration image is saved, archiving the progress of the octalinear way. This process keeps going until there is no alteration after passing 1000 generations. Figures 4 e 5 illustrates results from this experiment.



(a) linha original (b) best configuration of 1° generation (c) best configuration of 2° generation

(d) best configuration of 44° generation (e) best configuration of 617° generation (f) final result. 777° generation

Figure 5: Evolution of schematization 2. Results achieved in 0.41 seconds.

This experiment intended to observe a continued improvement caused by the algorithm iteration. The generated lines are always visually pleasant. Also the execution time (less than a second always) is within the tolerable. [2]

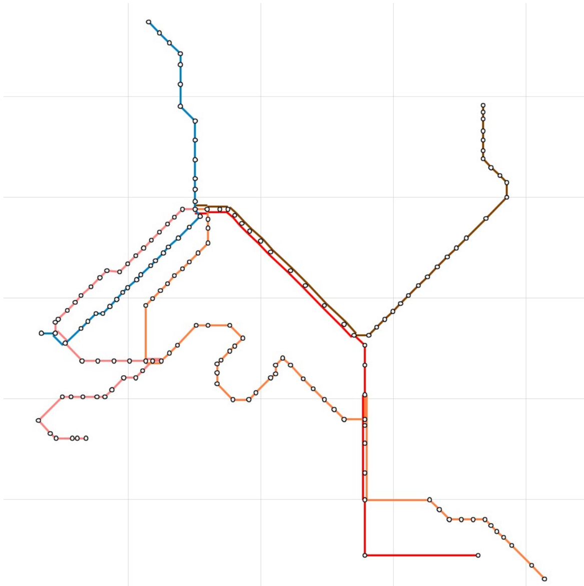
Figure 6 demonstrates results of applicability of the algorithm. Real data was used from Brasilia public transportation to produce those informative maps. Results obtained have good quality and were almost instantly. Related works - such as Avelar [2002] and Nöllenberg [2007] have good visual quality, however, processing can take many minutes.

4 Conclusion

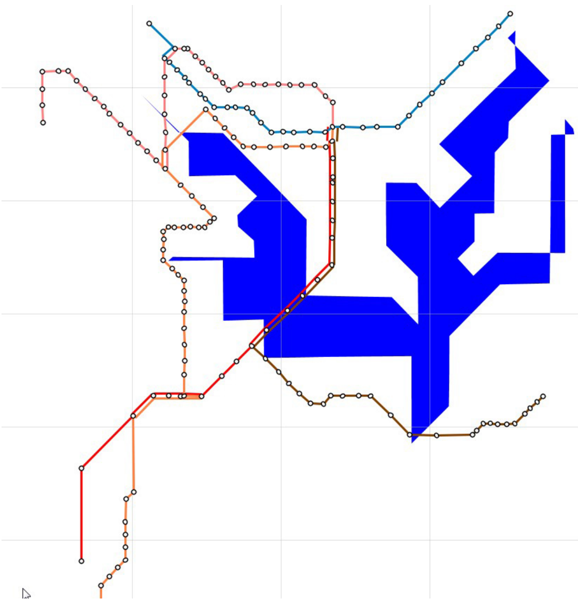
The octalinear map (subway map) is an informative tool for a public transportation system. Nowadays, their production is handmade and it is rapidly outdated. Consequently it becomes a costly task especially for bus lines that are more complex and have a fast modification dynamic. As a solution to the problem of automatic drawing those diagrams, it was developed a technique that uses graph algorithms and genetic algorithms. Data from Brasilia public transportation system was adopted to apply the technique. Results obtained have good quality and were almost instantly. Genetic algorithm was proved to be an adequate solution.

[2] All methods used in this experiment where implemented in Java, and executed in a Intel Core Duo, with 2GHz clock and 3GB of RAM.

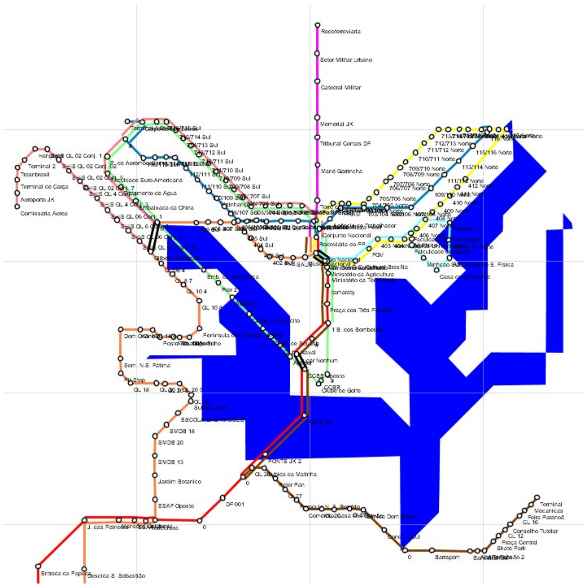
(a) Bus lines of Brasília. Non-schematic map.



(b) Yellow lines of figure 6(a). Processing time 0.8 seconds.



(c) Same set of lines of figure de 6(b), but if different orientation. Processing time 0.8 seconds.



(d) All lines schematic, including labels. Processing time 1.1 seconds.

Figure 6: Pratical results of the genetic algorithm.

References

Silvana Avelar. *Schematic Maps On Demand: Design, Modeling and Visualization*. PhD thesis, Institut of Technology Zurich, 2002.

Giuseppe Di Battista et al. *Graph Drawing: algorithms for the visualization of graphs*. Allan Apt, New Jersey, Estados Unidos, 1999.

Ken Garland. *Mr Beck´s Underground Map*. Capital Transporp Publishing, 1994.

Seok hee Hong, Damian Merrick, and Hugo A. D. Do Nascimento. The metro map layout problem. In *In Proc. Graph Drawing 2004*, pages 482–491. Springer-Verlag, 2005.

Seok-Hee Hong, Damian Merricka, and Hugo do Nascimento. Automatic visualisation of metro maps. *Journal of Visual Languages and Computing*, 2006.

Mitchell Melanie. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, England, 1999.

Martin Nöllenburg. Automated drawings of metro maps. Technical report, Fakultät für Informatik, Universität Karlsruhe, 2005.

Martin Nöllenburg. A mixed-integer program for drawing high-quality metro maps. Master's thesis, Fakultät für Informatik, Universität Karlsruhe, 2007.

Klemesn Polatschek. Die schönheit de untergrundes. Frankfurter Allgemeine Sonntagszeitung, julho 2006.

Alexander Wolff. *Drawing subway maps: A survey*. Universität Karlsruhe, 2007. Suported by German Reasearch Foundation.