# Map Point-Labeling with Rotation in Slider Model Using an Efficient Evolutionary Algorithm

Amir Zafar Asoodeh
Department of Computer Science and Engineering
Science and Research Branch, Islamic Azad University
Tehran, Iran

Farshad Rostamabadi
Department of Computer Science and Engineering
Science and Research Branch, Islamic Azad University
Tehran, Iran

Ali Ahmadi
Electrical & Computer College,
Khajeh Nasir Toosi University of Technology
Shariati St., Seyedkhandan,
Tehran, Iran

## ABSTRACT

Given n point coordinates and their various labels' length, our algorithm places a rotated collision-free label for each point. Using a combination of genetic algorithms and simulated annealing as an evolutionary algorithm, with qualification function consuming just $O(nlogn)$ time, we achieve a fast near-optimal algorithm.

## Keywords

Map-Labeling, Point Feature Label Placement, Rotation, Slider Model, Genetic Algorithms, Simulated Annealing, Sweep-Line Algorithm.

## 1. INTRODUCTION

Map-Labeling is a crucial step in map generation and usage. Automated label placement, in its simple form is to automatically attach labels to special features of maps i.e. points, lines and areas. Point Feature Label Placement (PFLP) is one of the remarkable sub-problems of automated label placement that has received good attention. In a valid label placement, labels should be located adjacent to their feature points (could be attached to or parted with a defined space) and they must be pairwise disjoint. Moreover, map clarity is an optional parameter argued in some documents on account of its emotional and human-based nature in maps. The first approach towards automated map-labeling belongs to Edward Imhof in [Imh75a] who tried to distinguish different steps of labeling and gave a systematic solution for all features of maps. After proving time complexity of map-labeling problem which is NP-complete in [For91a], heuristic approaches to solve the problem arose significantly. The first heuristic-based map-labeling solution was published in 1984 by Noma in [Nom84a] which placed labels according to an abridged algorithm in non-colliding space. Wagner and Wolf in [Wag95a] implemented a heuristic approach with a quality guaranty of 50 percent of the optimal solution and running time $O(nlogn)$ in all situations. As PFLP evolved, innovative map-labeling techniques were proposed in order to fulfill unwonted targets. Zhu and Poon heuristic Map-Labeling solution in [Zhu99a] which placed a non-intersecting pair of circular or rectangular labels for each point on the map, is one of the obvious illustrations of this concept. Evolutionary Algorithms (EA) are widely employed in complicated optimization problems. However, Genetic Algorithms (GA) as a sub-category of EA's became prominent due to their power in optimization and parallel processing. The first GA approach to map-labeling was represented by Djouadi in [Djo94a] which was comprised of procedures to calculate overlap and aesthetic constraints on maps to place labels. There are distinct versions of GA-based solutions that are argued in [Dij00a, Bae10a]. One of the main reasons which make fast automated map-labeling seem less perspicuous than the human-made one is restricted candidate space to place labels in the automated technique. Consequently, map-labeling in slider model, which allocates an approximate continuous candidate space to labels, was introduced by Strijk in [Str02a]. Labels' intersection detection is the main part of the evaluation function of map-labeling heuristic approaches, undoubtedly. In this paper, the collision detection procedure originates from Bentley_ Ottmann algorithm that was proposed in [Ben79a].

Applications for this problem can be found in computer graphics, GIS, Navigation systems, Computer games, flight animation and in other related fields.

The remainder of the article contains a complete description of the problem, including search space and cost function, the explanation of different parts of the algorithm and the results that were experimented to show the efficiency of the algorithm.

## 2. DEFENITION OF OBJECTIVES

The problem is precisely defined as follows: We are given a valid labeling $L$ composed of points $P = \{p_1, p_2, \dots, p_n\}$ and variable length rectangular labels $L = \{l_1, l_2, \dots, l_n\}$ where $l_i$ is attached to $p_i$ on the edges or vertexes (depending on labels' relocation values). For each position of map points in $P$, the problem is to make $l_i$ values in a way that the minimum intersections occur. Additionally, it can be thought of as a combinatorial optimization problem with evident search space and cost function which are discussed afterwards.

### 2.1 Relocation Techniques

On account of decreasing the overlap between map features and labels, map providers use some replacement methods to position the labels in non-colliding space. We use two relocation techniques described as follows:

#### *2.1.1 Rotation method*

It includes a rotation of rectangular label around the corresponding point with a degree ranging from 0 to 360. Based on the slider model, we utilize this method in a continuous space in order to increase candidate places. Figure 1 depicts the rotation method around a map point.

Note that Θ indicates the degree between the horizontal axis and the rectangle's length which connects map point and label's vertex horizontally in the initialization stage.

#### *2.1.2 Translation Method*

This method incorporates translation with maximum displacement equal to the labels' length. It also employs slider model to relocate labels around the points. Figure 2 illustrates the translation method. Notice that L is the translation displacement value between 0 and each label's length.

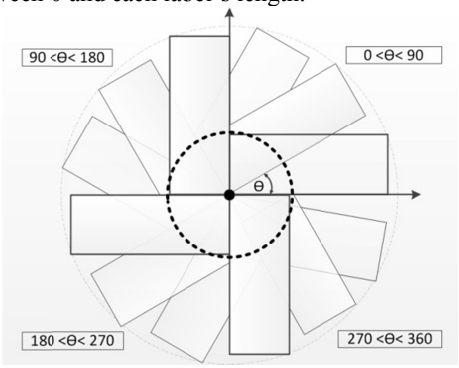


**Figure 1. Demonstration of various label positions under the sliding label scheme with rotation method**

The relocation methods are called in an ordered sequence during algorithm runtime. By virtue of the aforementioned slider-based label replacement

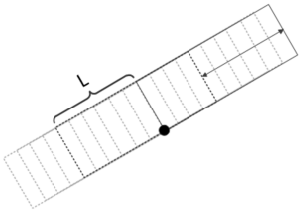


**Figure 2. Presentation of a range of label positions in slider model with translation method**

Techniques, it seems that the candidate space required to attach labels to dense and complex maps is satisfied.

### 2.2 Cost Function

In combinatorial optimization problems, cost function is an estimation of distance between the optimal and the existing solution. Hence, defining a rational and reasonable cost function leads to higher proficiency. In this paper, cost function is comprised of two different values:

1-Overlaps of the candidate label with the other labels.

2-Clarity preference.

The combination of these two items can be used as a suitable cost function in order to clarify the closeness of the solutions to the optimal result. Let $C$ be a set of n label cost, $C = \{C_1, C_2, \dots, C_n\}$ where $C_i$ belongs to $L_i$ from labels. The attributes of label cost are defined as follows:

- $c_i$: number of overlaps of $l_i$ with $\forall l_j \in L, i \neq j$
- $p_i$: a penalty of the current position. In this paper, the inclination of initialized and relocated label of each point is the main parameter of this clarity option. In other words, labels with less rotation from horizontal axis receive lower penalty. If α is the degree of rotation of a relocated label, positions between 0 and 90 ($0 \leq \alpha \leq 90$) are considered as preferred locations and the penalty value of these candidate spaces are set to 0 and all the remaining positions have penalties equal to 1.

$C_i$ is then defined as follows:

$$C_i = a_1 . c_i + a_2 . p_i$$

Where $a_1$ and $a_2$ are constant values. In our experiment 1.0 and 0.1 are used for $a_1$ and $a_2$, respectively. It is important to notice that owing to attached labels to the points in all degrees of rotation and translation variation, the algorithm does not need to calculate point/label collisions. Regarding that the cost function of map-labeling is $\sum_{i=1}^{n} C_i$, F(s) definition as a cost function for a given set of points S is as follows:

$$F(S) = \sum_{i=1}^{n} (a_1 . c_i + a_2 . p_i)$$

By utilizing this function, the algorithm chooses labels with fewer overlaps and straighter positions as lower cost candidate solutions.

## 3. THE ALGORITHM

The algorithm designed to solve the previously described map-labeling problem is an evolutionary algorithm that utilizes a mixture of GA and SA ideas to offer a near-optimal map labeling solution. In this section, the basic principles of suggested genetic algorithm will be stated.

### 3.1. GA

In this paper, we adapt a GA to solve map-labeling problem. The algorithm consists of different parts that are described as follows:

#### *3.1.1 Chromosome Structure*

In map-labeling problem, there are many variables which have the capability to be placed inside chromosomes. However, in order to prevent extra complexity, we endeavor to design chromosomes as plainly as possible. Figure 3 illustrates the chromosome design of GA algorithm. In this structure, each pair of $R_i$ and $L_i$ relates with a label.


**Figure 3. Presentation of chromosome structure**

Note that $R_i$ belongs to the degree of rotation of each label and $L_i$ is described as follows:

$$L_i = \frac{TranslationValue_i}{Length_i}$$

Where $Length_i$ is the label's length that could vary for each point. By this structure, utilizing additional techniques like masking to avoid chromosome disruption after operation is not required. The chromosome size equals to $2.(point\ num) + 1$ where the addition of one unit pertains to the chromosome fitness value.

#### *3.1.2 Genetic Operators*

In this paper, we use two different GA operators which are called after fitness calculation of each generation.

**Crossover**: It generates two offspring from two parents by swapping the information beyond random points. Owing to the large size of chromosomes in dense maps, we use a two-point crossover operator to function more drastically.

**Mutation**: It evolves chromosomes by transmuting some bits identified in a random manner. Provided that the selected bit belongs to the degrees of rotation, the offspring mutated bit is calculated as

$$R^*_3 = 360 - R_3.$$

And if the selected location contains information of a transformation displacement, the mutated bit is computed as

$$L^*_1 = 1 - L_1.$$

The amount of altered bits in a chromosome by mutation operation is defined as the number of genes divided by 10. In other words, each 10 genes in a single chromosome include one mutation. After calling GA operators, the offspring and parents are saved in a chromosome pool to launch the selection procedure.

#### *3.1.3 Selection Methods*

We use four different selection methods to compare their proficiency in this special GA solution. The methods are:

- Rank selection
- Roulette wheel selection
- Elitist Roulette wheel selection
- Elitist Rank selection

In which elitist methods perform selection procedure by taking advantage of the former generation directly. Note that running time and selection accuracy are two criteria to measure the competence of the selection methods that will be discussed later.

#### *3.1.4 Fitness Method*

We have benefited from Bentley-Ottman algorithm as the basis of GA evaluation function due to its distinct advantages and have adjusted it to the specification of the problem. If $n$ represents the number of lines and $K$ is the number of overlaps in the map, the time complexity of Bentley-Ottman algorithm equals $O((n+k).logn)$. The labels in a map are rectangles with expected diverse sizes clearly. If Bentley-Ottman runs on a map with rectangular labels, the vertices of labels are added to the intersection points undesirably owing to the fact that they should not be considered overlaps between the labels. To make an adjustment, the following definition is assumed.

**Definition 1** reduced rectangle is described as a rectangle which contains sides decreased by ε and it is denoted as $R_\varepsilon$.

**Lemma 1** Overlap detection by Bentley-Ottman algorithm in a map which contains $R_\varepsilon$ labels has lower time complexity than that one in a map with rectangular labels.

**Proof** The number of intersections between lines in the plane affects running time in Bentley-Ottman algorithm undoubtedly. If $k$ is the number of overlaps between lines, $p$ is the quantity of points in the map and $i$ is the number of intersections between labels, rectangular label map has the following equation:

$$k = 4.p + i$$

On the other hand, number of overlaps between lines in the map which contains $R_\varepsilon$ labels is defined as:

$$k = i$$

Thus, time complexity with $R_\varepsilon$ labels is calculated as $O((n+i).logn)$ resulting in lower running time.

### 3.2 SA Function

In this paper, we take advantage of the SA idea to develop dynamic operator rates. Temperature as a significant parameter of SA is interpreted as average amount of fitness in a GA generation. In other words, if a generation is more fitted than the preceding one, the temperature will be reduced and vice versa.

## 4. RESULTS

In order to establish more efficacious results, we have implemented two other solutions and have made a comparison with the previously introduced algorithm. Figure 4 shows the running time of GA-SA with sweep-line and $R_\varepsilon$ labels (suggested algorithm), GA with sweep-line and $R_\varepsilon$ labels which is constructed with constant rates and simple GA which deploys the basic line-by-line overlap detection algorithm to count intersections. Notice that the running time discrepancy between GA's with static and dynamic rates becomes prominent with 220 points or more which exist on a map. Moreover, with 100 points or fewer, the GA with static rates consumes less time due to computational overheads of GA with dynamic rates. However, the more points are added, the bigger difference appears between running time of considered GA's. A simple GA has the time complexity of 46.61% on average higher than the GA with sweep-line and 43.58% higher than GA-SA with sweep-line which reveals the swiftness of suggested map-labeling algorithm. Figure 5 depicts the efficiency of the suggested selection methods by illustrating the overlaps after 10 seconds past running time. As the figure shows, elitist roulette wheel selection method has always the least amount of overlaps in comparison with other algorithms.
Moreover, elitist rank selection method ranked second finally in spite of its lower ability to reduce overlaps compared with roulette wheel selection in lower density Maps.

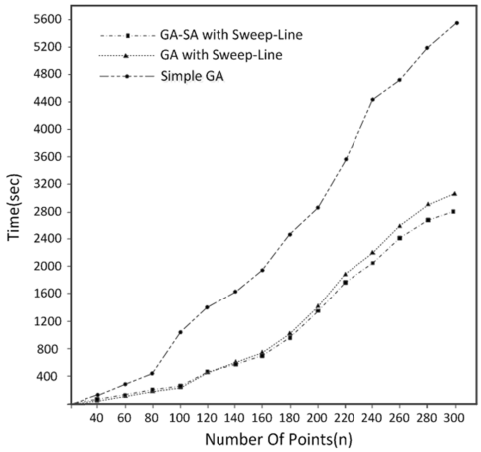


**Figure 4. Running time comparison of three different GA algorithms**

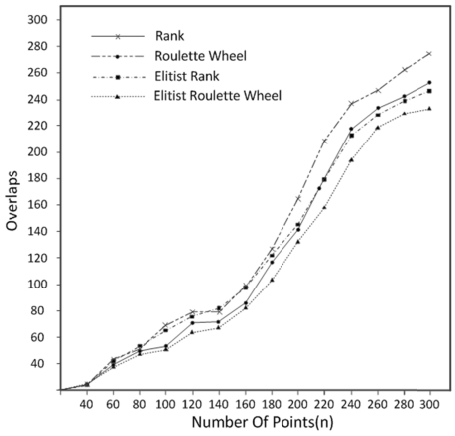


**Figure 5. Performance Comparison of different selection methods**

## 5. CONCLUSIONS

In this paper, we introduced an algorithm to label maps with new rotation techniques in slider model. Using a mixture of GA and SA ideas, the algorithm can perform swift and efficient labeling. By adapting the well-known segment intersection detection algorithm to the map point-labeling problem, we achieve a fast near-optimal solution.

## 6. REFERENCES


[Bae10a] Bae W., Alkobaisi S. and Narayanappa S. and Y. Bae K., Convex Onion Peeling Genetic Algorithm: An Efficient Solution to Map Labeling of Point-Feature, SAC, pp. 892, 2010.

[Ben79a] Bentley J. L. and Ottmann, T. A., Algorithms for reporting and counting geometric intersections, IEEE Transactions on computers, 1979.

[Dij00a] Dijk S. V., Scalability and efficiency of genetic algorithms for geometrical applications, Springer-Verlag, pp. 683-692, 2000.

[Djo94a] Djouadi M., Cartage: A cartographic layout system based on genetic algorithms, EGIS, 1994.

[For91a] Forman M., A packing problem with applications to lettering of maps, ACM, 1991.

[Imh75a] Imhof E., Positioning names on maps, Cartography and Geographic Inf. Science, 1975.

[Nom84a] Noma E., Heuristic Method For Label Placement In Scatterplots, psychometrica, 1984.

[Str02a] Strijk T. and Kreveld M. V., Practical Extensions of Point Labeling in the Slider Model, Springer, pp. 683-692, 2002.

[Wag95a] Wagner F. and Wolff A., Map labeling heuristics: Provably good and practically useful, SoCG'95, pp.109-118, 1995.

[Zhu99a] Zhu B. and Poon C. K., Efficient approximation algorithms for multi-label map labeling, Springer-Verlag, pp. 143-152, 1999.