Real-time Placement of Labels on a Geographical Map

Enterprise
GIRO Inc.

Coordinator
Odile Marcotte
CRM and UQÀM

Reference
An article entitled "Fast Point-Feature Label Placement Algorithm for Real Time Screen Maps" (by M. Yamamoto, G. Camara, and L. A. N. Lorena) is available on the problems page and can be downloaded.

Abstract

Labels are displayed on a geographical map in order to identify map components (i.e., streets, clients, routes, cities, boroughs, etc.). Almost any map component may have a label. The problem consists of placing labels in such a way that each label is close to the corresponding component, while minimizing the overlap between this label and the other ones. The placement must be computed very fast since it is updated every time the user moves on the map or zooms on a specific map area. A computation taking more than half a second is considered slow. We are interested in a generalization of this problem that includes priorities for labels, preferences for label positions, and (maybe) relative costs for overlaps.

A list of allowable positions is available for each label. Each label has a weight reflecting its priority (relative to those of the other labels). A label with a high priority must have "smaller overlaps" than labels with low priorities. Each position has a weight reflecting its "quality". The cost of the overlap between two positions may be absolute or relative; in the latter case, it depends upon the overlap percentage (with respect to the smallest label). Then one can build a graph whose vertices are the allowable positions, and in which two vertices are joined by an edge if and only if the corresponding positions overlap or pertain to the same label.

In this graph, the weight of a vertex equals the sum (or product) of the label priority and the position weight. The cost of edge uv equals a large constant (denoted M) if the positions corresponding to the vertices u et v pertain to the same label. If not, the cost of edge uv equals the product of a negative constant (denoted $-C$), the maximum weight of u and v, and the sum of the weights of u and v (if the overlaps are considered to be absolute), or to the same product multiplied by the overlap percentage (if the overlaps are considered to be relative). For example, if $-C$ equals -0.02, the weights of u and v are respectively 10 and 8, and the overlap percentage equals 50%,

then the cost of the edge uv is equal to $-0.02 \cdot 0.5 \cdot 10 \cdot (10+8) = -1.80$. If the weights of u and v are both equal to 9, the cost of the edge uv will be equal to -1.62.

One is thus looking, in the graph defined as above, for a stable set of maximum weight, where the weight of a stable set includes the weights of the vertices and those of the edges. The stable set obtained is actually a set of labels placed in such a way that the total overlap is acceptable. One may also enforce the constraint that all labels have positions; this constraint can have a large impact on the solution strategy.

For each label, one must at least compute its "preferred" position. Computing other positions, however, is costly, given that the algorithm should not consume much time. One must avoid computing the alternative positions, as far as possible. We wish to implement the following strategy, consisting of five steps:

1. generate the preferred positions of all labels,
2. if there is one preferred position whose alternative positions have not been computed and that overlaps with another position, then compute its alternative positions,
3. find a large stable set of maximum weight by executing the greedy algorithm MIN (which consists of selecting the vertex of largest weight and smallest overlap, and of discarding the vertices overlapping the selected vertex or pertaining to the same label),
4. assign the remaining labels to the least costly positions, and
5. run through the set of labels five times and change the current label position for a position of the highest possible quality.

We wish to explore other solution strategies, which will avoid computing the alternative positions (as far as possible). The implementation of the above strategy risks generating alternative positions for most of the labels.