◊ .1

Placing Text Labels on Maps and Diagrams

Jon Christensen
Harvard University
Cambridge, Massachusetts

Joe Marks
Digital Equipment Corporation,
Cambridge Research Lab
Cambridge, Massachusetts

Stuart Shieber
Harvard University
Cambridge, Massachusetts

◊ Introduction ◊

Tagging graphical objects with text labels is a fundamental task in the design of many types of informational graphics. This problem is seen in its most essential form in cartography, but it also arises frequently in the production of other informational graphics such as scatterplots. The quality of a labeling is determined essentially by the degree to which labels obscure other labels or features of the underlying graphic. The goal is to choose positions for the labels that do not give rise to label overlaps and that minimize obscuration of features. Construction of a good labeling is thus a combinatorial optimization problem, which has been shown to be NP-hard (Formann and Wagner 1991, Marks and Shieber 1991). In this gem, we describe a method for heuristically solving this optimization problem using simulated annealing (Kirkpatrick et al. 1983, Cerny 1985). Extensive empirical testing has shown that the simulated-annealing approach to the label-placement problem produces higher quality labelings than other previously published algorithms, and is competitive with respect to efficiency (Christensen et al. 1992).

◊ Iterative Local Improvement ◊

As an example, suppose one is given a set of point features (perhaps a set of city locations on a map or point locations on a scatterplot), each of which may be labeled in one of eight positions, as shown in Figure 1. As a hypothetical baseline algorithm, randomly choosing positions for each label, as in Figure 2 is likely to generate a poor labeling. The apparent poor quality of the labeling can be quantified using a metric

ISBN 0-12-XXXXXX-X

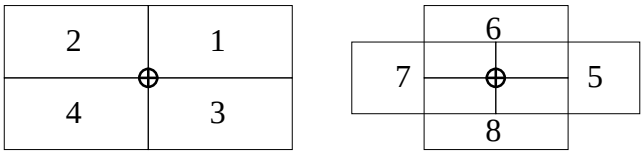

Figure 1. A set of potential label positions and their relative desirability.

that counts the number of *conflicted* labels, i.e., those that obscure point features or other labels. Using this metric, this labeling has a score of 86.

One might attempt to improve the labeling by an iterative local improvement method, adjusting a label position if it leads to a better overall score. Such a local method is prone to ending up in unacceptable local optima. Repeatedly performing the best local improvements (i.e., the local improvements that give the greatest decrease in the number of conflicted labels at each step) on the example of Figure 2 generates the locally optimal labeling in Figure 3, which, though better than the original random labeling, can still be greatly improved upon.

♢ Simulated Annealing ♢

In order to escape from local optima in the search space, we use simulated annealing (Kirkpatrick et al. 1983, Cerny 1985, van Laarhoven and Aarts 1987) to allow occasional label adjustments that make the overall score worse. Of course, such anarchic behavior cannot be tolerated uniformly. Rather, the ability of the algorithm to degrade the solution is controlled by a parameter T, called the *temperature*, that decreases over time according to an *annealing schedule*. At zero temperature, such backwards steps are disallowed completely, so that the algorithm reduces to a local optimization method. At higher temperatures, however, a wider range of the search space can be explored, so that regions surrounding better local optima (and perhaps even the global optimum) may be visited. The following outline describes the essential characteristics of a simulated-annealing algorithm for the label-placement problem:

1. For each point feature, place its label randomly in any of the available potential positions.
2. Initialize the temperature T to T_0.
3. Repeat until the rate of improvement falls below a given threshold:
   (a) Decrease the temperature, T, according to the annealing schedule.
   (b) Pick a label at random and move it to a new position at random.
   (c) Compute ΔE, the change in the score caused by repositioning the label.

(d) If the new labeling is worse, undo the label repositioning with probability $P = 1.0 - e^{-\Delta E/T}$.

A geometric annealing schedule is recommended, where the temperature remains constant for a given number of iterations, and then is decreased by a constant factor. We have used the following particular annealing schedule with success. The initial temperature T_0 is set so that $P = 1/3$ when $\Delta E = 1$. This gives a value for T_0 of about 2.5. The temperature is decreased by 10 percent every $50n$ iterations through loop (3), where n is the number of point features. If more than $10n$ successful configuration changes are made at any temperature, the temperature is immediately decreased. This process is repeated for at most 50 temperature stages. However, if we stay at a particular temperature for the full $50n$ steps without accepting a single label repositioning, and if the cost is the lowest seen so far, then the algorithm stops with the current labeling as the final solution. In limited experimentation we found the particular choice of annealing schedule to have a relatively minor affect on the performance of the algorithm. This schedule was chosen primarily to provide reasonable execution times; longer annealing schedules result in moderately improved solutions.

Figure 4 shows the labeling generated by this algorithm for the same point data used in Figure 2. This labeling was generated in under a second on a DEC 3000 Model 400 AXP workstation.

♢ Augmentations and Comparisons ♢

To make the algorithm practical, care must be taken in the computation of ΔE. Cartographic practice requires that provision be made for a priori preferences among label positions and the ability to delete labels in congested areas (Yoeli 1972). In this section, we discuss these issues and conclude with an informal comparison along several dimensions of the simulated annealing approach and other previously proposed approaches to label placement.

Computation of ΔE

Because simulated annealing is a statistical method that relies on a large number of evaluations for its success, the best scoring functions are those for which ΔE can be computed easily. The most straightforward scoring function simply counts the number of pairwise overlaps. In practice, however, it is often preferable to have a single label with three overplots (four labels conflicted) instead of three distinct pairwise overplots (six labels conflicted). Therefore the scoring function we choose counts the number of labels obstructed by at least one other label or graphical object.

Initialization of the algorithm involves the following five steps:

1. Pre-compute all intersections between potential label positions, recording for each label position a list of which points and label positions overlap it.
2. Generate an initial random labeling.
3. For each point, store a counter of the number of pairwise overlaps between its current label position and those of the other points.
4. Calculate an initial score, the number of points with non-zero counters.

Calculation of ΔE at each step can now be done with reasonable efficiency. Given a single label repositioning for a source point p, the change in the score is simply the number of newly conflicted labels minus the number of newly cleared labels. To count the number of newly cleared labels, each point q that potentially conflicts with the old label position is examined, using the table set up in step (1). If the current label position of q is in a conflicting position with the old label position of p (as pre-computed in step (1)), the conflict counter for q (initialized in step (3)) is decremented. If the conflict counter of q reaches zero, the number of cleared labels is incremented. The calculation of newly created conflicts is analogous: The algorithm examines the list of points that potentially conflict with the new position. If a point switches from zero pairwise conflicts to one or more, then the candidate repositioning has created a new label conflict. Lastly the algorithm notes whether the source point p itself has switched from unconflicted to conflicted, or vice-versa, and adjusts the score accordingly.

Placement Preferences

Typically, not all of the potential label positions for a given point are equally desirable. It is standard in cartography applications, for instance, to give highest preference to the upper-right label position, all else being equal (Yoeli 1972). Although this additional consideration makes the labeling problem more difficult, the simulated-annealing algorithm handles preferences easily by adding a penalty to the scoring function for each point feature that reflects the relative undesirability of its label position. Given a ranking of the positions such as that given by the numbers in Figure 1, a penalty of $(r-1)/N$ is associated with the r-th ranked of N positions. In the example, the upper-right position is therefore given a penalty of 0, the upper-left a penalty of 1/8, and so forth. Placement preferences were incorporated in the runs of the simulated-annealing algorithm that produced Figures 4 and 5.

Deletion of Labels

In many applications, especially automated cartography, an alternative to placing a label in a highly congested area is to delete the label, and often its associated point feature. This strategy can be incorporated into the algorithm by modifying the scoring function to be the number of conflicted labels plus the number of deleted labels. This is

equivalent to maximizing the number of labels displayed without conflicts and insures that an optimal map will have no conflicted labels. Optionally, deletion terms may be weighted to discourage deletion of important features, such as capital cities.

Figure 5 shows the result of allowing point deletion for the set of point features used in Figure 2.

Comparison with Other Algorithms

A wide variety of approaches — greedy heuristics (Doerschler and Freeman 1992, Jones 1989), physical models (Hirsch 1982), and reductions to integer programming (Zoraster 1990), among others — have been pursued to solve the label-placement problem. (A more complete bibliography is provided by Christensen et al. (1992).) Empirical testing of a wide variety of algorithms, including essentially all practical published algorithms, has shown that the simulated-annealing method finds better solutions at all label densities (Christensen et al. 1992, 1993). The simulated-annealing algorithm is also competitive with its alternatives in terms of efficiency, falling roughly in the mid-range of running times. In practice, the algorithm requires only a few seconds on a modern workstation for page-sized maps at typical cartographic labeling densities. The simulated-annealing algorithm may be the easiest algorithm to implement beyond random placement or iterative local improvement. Finally, the simulated-annealing approach has the advantage of generality. Although the discussion here has used labeling of point features to exemplify the algorithm, simulated annealing can be used to solve the labeling problem for any features — points, lines, areas — for which alternative label positions can be generated and quality of labeling can be quantified.

♢ Acknowledgements ♢

The research reported in this article was supported in part by a contract with U S WEST Advanced Technologies, by grant IRI-9157996 from the National Science Foundation, and by a matching grant from Digital Equipment Corporation.

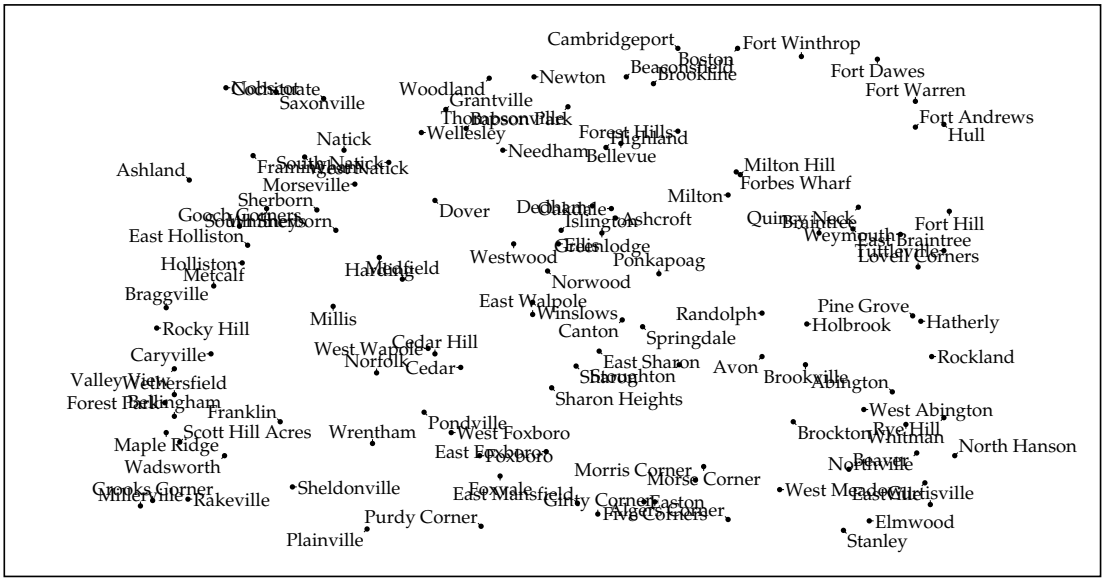

Figure 2. Initial random labeling of a set of 120 points in Massachusetts (86 labels conflicted).

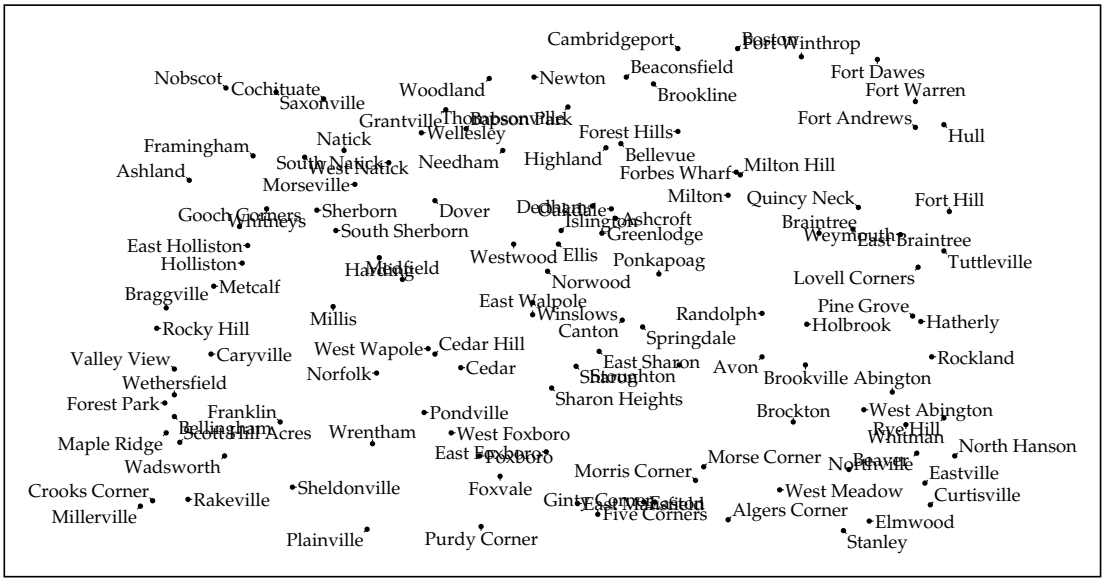

Figure 3. Random labeling improved by iterative local improvement (42 labels conflicted).

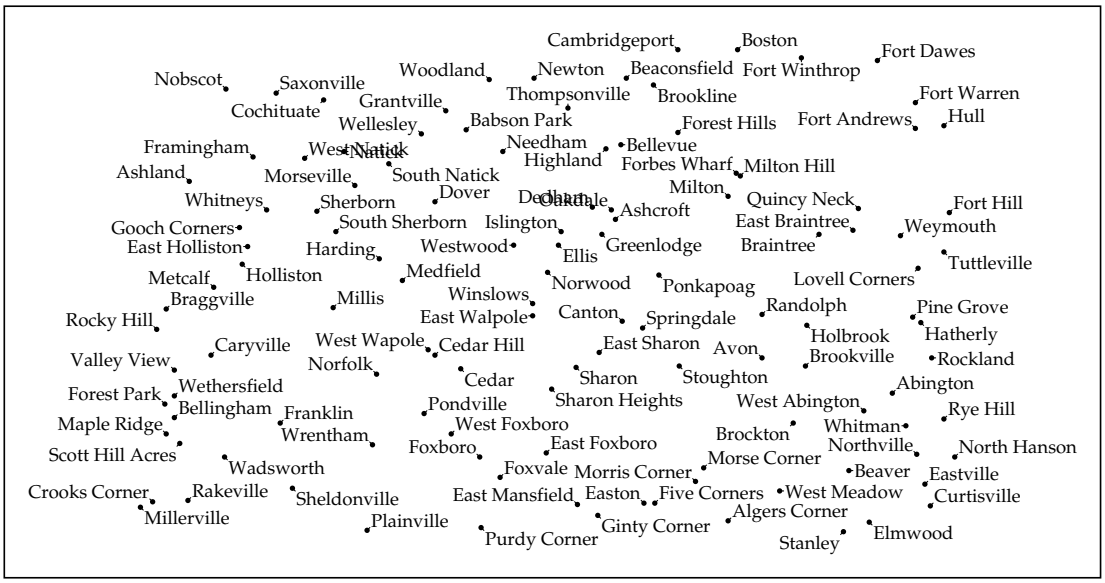

Figure 4. Labeling of the same set of points generated by the simulated-annealing algorithm (4 labels conflicted).

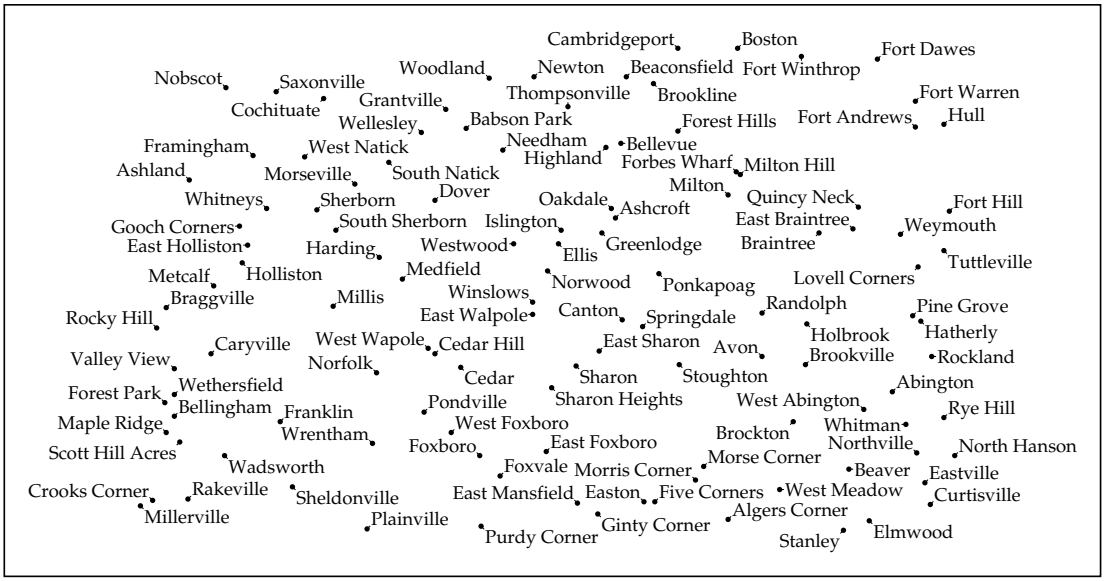

Figure 5. A labeling of the same set of points with point deletion (2 points deleted).

Bibliography

(Cerny 1985) V. Cerny. A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory Applications*, 45:41–51, 1985.

(Christensen et al. 1992) Jon Christensen, Joe Marks, and Stuart Shieber. Labeling point features on maps and diagrams. Technical Report TR-25-92, Harvard University, November 1992.

(Christensen et al. 1993) Jon Christensen, Joe Marks, and Stuart Shieber. Algorithms for cartographic label placement. In *Proceedings of the American Congress on Surveying and Mapping '93, Vol. 1*, pages 75–89, New Orleans, Louisiana, February 15–18 1993.

(Doerschler and Freeman 1992) Jeffrey S. Doerschler and Herbert Freeman. A rule-based system for dense-map name placement. *Communications of the Association of Computing Machinery*, 35(1):68–79, January 1992.

(Formann and Wagner 1991) Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, pages 281–288, North Conway, New Hampshire, July 1991. ACM.

(Hirsch 1982) Stephen A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

(Jones 1989) Christopher Jones. Cartographic name placement with Prolog. *IEEE Computer Graphics and Applications*, 9(5):36–47, September 1989.

(Kirkpatrick et al. 1983) S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

(Marks and Shieber 1991) J. Marks and S. Shieber. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard University, March 1991.

ISBN 0-12-XXXXXX-X

(van Laarhoven and Aarts 1987) P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. D. Reidel, Dordrecht, Holland, 1987.

(Yoeli 1972) P. Yoeli. The logic of automated map lettering. *The Cartographic Journal*, 9(2):99–108, December 1972.

(Zoraster 1990) Steven Zoraster. The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 38(5):752–759, September-October 1990.