PRACTICAL EXPERIENCE WITH A MAP LABEL PLACEMENT PROGRAM

Steven Zoraster
Stephen Bayer
ZYCOR, Inc.
220 Foremost
Austin, Texas 78745

ABSTRACT

Mathematical optimization algorithms have previously been shown to provide a solution to some map label placement problems. This article discusses computational and cartographic difficulties encountered and solved while implementing such techniques in commercial map label placement software. The solution of the optimization problem, the automation of label overplot detection, and the representation of deleted labels in the manner best suited to assist human intervention are discussed.

Despite remaining questions about the best way to implement this type of computer program, our work has shown that mathematical optimization techniques are an excellent method for performing map label placement. This conclusion is based on experience resolving labeling problems on maps compiled from typical data encountered in oil and gas exploration activities.

INTRODUCTION

Most algorithms designed for the placement of map labels around point symbols are characterized by sophisticated data structures and complex, rule-following logic (Hirsch 1982; Ahn and Freeman 1983; Ahn 1984; Basoglu 1984; Pfefferkorn, et al. 1985; Langran and Poiker 1986; and Mower 1986). In fact, these algorithms are attempts to solve a specific type of combinatorial optimization problem by heuristic or artifical intelligence procedures. Because artificial intelligence is not used to solve combinatorial optimization problems in non-cartographic applications, we have investigated alternate methods for solving the map label placement problem. Our work has produced a computer program which is suitable for production mapping. Our program solves the underlying optimization problem directly.

The optimization model for label placement was originally exploited by Cromley (1985), who developed an overplot resolution algorithm for point symbol labels based on a linear programming relaxation procedure and interactive detection of overplots. Its use was extended by Cromley (1986) and by Zoraster (1986), who have both used integer programming techniques along with automated detection of label overplots.

This paper provides a review of the optimization model for label placement, compares this type of algorithm to other placement algorithms, and discusses various implementation problems.

THE OVERPLOT RESOLUTION ALGORITHM

Simple label overplotting problems can be resolved without sophisticated algorithms. An advantage of using an optimization algorithm for this purpose is that it can resolve extremely complex problems efficiently. The only requirement is that there be a finite number of label placement options for each label. A prioritized set of options is shown in Figure 1 by numbers placed around a map symbol (X). Given such a placement priority for each label, the best label placement for a large number of labels can be found by use of integer programming.

3	1	2
5	X	4
8	6	7

Figure 1. Possible Label Position Options

Linear Programming and Integer Programming

Linear programming is a mathematical optimization technique used to minimize (or maximize) the value of a linear objective function subject to linear constraints on the values that can be assumed by problem variables (Chvatal 1983). Given an N element vector **c**, an M by N matrix **A**, and an M element vector **b**, the standard linear programming problem is to choose the N elements of the vector **x** to maximize

$$\mathbf{c}^T\mathbf{x},$$

subject to the constraints

$$\mathbf{A}\ \mathbf{x} \leq \mathbf{b}.$$

Integer programming problems are linear programming problems with the added constraint that the elements of **x** must be integer. 0-1 integer programming problems require that the elements of **x** be either 0 or 1. Our overplot resolution program uses 0-1 integer programming to resolve overplot problems.

Many linear programming algorithm implementations are available in the form of software subroutine libraries. Only a few of these implementations handle integer or 0-1 variables efficiently. This difficulty is discussed in more detail later in this paper.

The Overplot Problem Formulated As a 0-1 Integer Programming Problem

Assume that each of the K labels on a map can be placed in one of N_k possible positions, one of which corresponds to deletion from the map. In the mathematical formulation of the overplot resolution problem, each position for label k will correspond to a single variable $X_{i,k}$ ($i = 1,2,\ldots,N_k$) which can take on the values of 0 or 1. Only one variable corresponding to each label is allowed to be 1. This restriction is enforced by the following constraints:

$$\sum_i x_{i,k} = 1 \qquad k=1,2,...,K. \qquad (1)$$

If an overplot is possible between the i-th label position for label k, and the j-th label position for label m, then the following constraint needs to be enforced:

$$x_{i,k} + x_{j,m} \leq 1. \qquad (2)$$

Each position for each label will have a penalty $W_{i,k}$ associated with its use. Normally the optimal position will have a penalty of 0 and other positions will have positive penalties proportional to the difficulty caused by attempting to associate a well label in that position with its correct symbol. The penalty associated with deletion of a label from the map will be the largest penalty. Our goal is to attempt to minimize the total penalty represented by

$$\sum_i \sum_k w_{i,k} \quad x_{i,k}. \qquad (3)$$

subject to the constraint sets (1) and (2).

ALGORITHM IMPLEMENTATION

Three major concerns in the implementation of any map design software are program speed, the quality of the output, and the ease with which the results can be edited by a cartographer. For this label overplot program, speed is affected primarily by the algorithm used to detect overplots and the algorithm used to solve the optimization problem. The quality of the output is primarily affected by how close the integer programming algorithm can approach an optimal solution to the integer programming problem. The handling of labels deleted from the input map is the most important factor affecting final map editing. Each of these concerns is addressed in the following paragraphs.

Overplot Detection

To detect overploting efficiently, label placement data must be partitioned or sorted. Cromley (1986) has suggested a referencing label positions to a Thiessen diagram based on the map symbols, but we have found that less sophisticated label locating schemes are cost effective for our application. We have used a simple sweep algorithm in which the labels, map symbols, and line segments are first sorted according to their X coordinates and then subject to a hierarchy of tests to detect actual overplots between items whose projections on the map X-axis overlap.

We have implemented a sequence of tests which allow most potential conflicts to be eliminated from consideration quickly. Intelligent programming strategies then make each required interference check simple. For example, when checking for the intersection between an arbitarily oriented label and a point symbol, the point symbol's position is rotated and normalized with respect to the orientation and center of the label, so that the interference check requires measuring only the magnitudes of each of the coordinates of the point symbol in the new coordinate system.

Practical Integer Programming Algorithms

Integer programming problems are generally difficult to solve. For large problems it is often impossible to obtain a feasible solution that is close to the optimal solution using general purpose software. Special techniques based on the structure of the constraints and on the structure of the objective function, along with careful problem formulation are required to solve such large problems (Johnson et al. 1985). A map label placement problem with hundreds or thousands of labels and hundreds of constraints creates a large integer programming problem. Most of the computer time required by our program is spent solving the integer programming problem.

Fortunately, the structure of the overplot resolution problem can be exploited to obtain efficient solutions. The optimization algorithm we use has been developed specifically to solve this map labeling problem. Our solution technique uses Lagrangian relaxation of the interference constraints which are included in the objective function subject to multiplication by a vector of Lagrangian weights. Using matrix notation, the problem we solve is

$$\text{min:} \quad \mathbf{w}^T\mathbf{x} - \mathbf{d}^T[\ \mathbf{1} - \mathbf{A}\mathbf{x}\ ]$$

$$\text{subject to:} \quad \mathbf{P}\mathbf{x} = \mathbf{1}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{1}$$

along with the 0-1 restriction on the elements of x and the restriction $\mathbf{d} \geq 0$. The equality constraints are simply the set (1) written in matrix format, and the inequality constraints are the pairwise conflict constraints (2), again written in matrix format. The elements of the vector **d** are the Lagrangian weights and are approximations to the dual variables of a linear programming relaxation of the original integer programming problem.

The Lagrangian weights are adjusted in an iterative manner according to the "subgradient optimization" method (Held, et al. 1974). When using this algorithm, the penalties for label variables corresponding to unsatisfied label interference constraints are increased at each iteration, while the penalties for label variables corresponding to constraints that are not binding are decreased. The amount of change is reduced periodically in a manner that guarantees algorithm convergence. On most maps this algorithm produces what appears to be very close to an optimal label placment. We have encountered no maps for which this algorithm does not produce at least a reasonable answer. Of course a map which is very densely labeled may require many label deletions to produce a solution.

Many other algorithms exist to solve integer programming problems. We have not seriously investigated the wide range of solution techniques available. Research to determine the best algorithm for this particular problem would be very useful.

Deleted Labels

The optimization algorithm performs the majority of work required to resolve label overplotting, but many maps will benefit from interactive editing to complete the label placement process. One type of data required during an editing session is information about exactly which labels have been deleted in order to obtain a solution. Two obvious graphical methods for recording this information are available. Deleted labels may be written to a distinct graphic layer of the output picture file or to a digital copy of the original input from which all other labels have been removed. Writing the labels to a distinct layer on the output picture is most useful if the program user has access to a color graphic device. Since many users of our software work on monochrome terminals, we have chosen to write the deleted labels, along with their associated symbols, to a second picture.

Better solutions to the problem of handling deleted labels might involve the automatic creation of insert maps at a larger map scale or the display of all deleted labels in a table outside the map border. The labels in the table could be referenced to their true positions on the map by a special class of symbols used to replace those symbols corresponding to deleted labels. Implementation of either of these options would require much work.

ADVANTAGES OF OPTIMIZATION APPROACH TO LABEL PLACEMENT

Most previous algorithms designed to solve the point label placement problem have not made use of mathematical optimization algorithms, but there is nothing in the reported problem formulations which precludes the use of this type of solution. Combinatorial algorithms described in the literature of mathematical optimization make as much use of mathematical techniques as is practical; these include algorithms designed for business planning, transporation analysis, personnel assignment, the allocation of military resources, and many other applications. In each of these domains heuristic techniques are used only when mathematical techniques fail because mathematical techniques produce better solutions. We believe that computer programs based primarily on optimization algorithms will out perform algorithms based on heuristics or artificial intelligence in this application as well.

There are at least two other significant reasons to favor the use of mathematical optimization algorithms to place map labels. First, mathematical optimization algorithms are easily implemented in procedural languages. Although there has been much talk about object oriented languages such as LISP or PROLOGUE for cartographic applications, most mapping systems are still coded in languages such as FORTRAN or C. Because cartographic problems often are mathematical in structure, procedural languages will continue to be used for the implementation of many cartographic programs. The program we have developed for map label placement is coded in FORTRAN 77.

Second, this type of algorithm has been proven using our computer program on many complex maps created by the oil and gas industry. To our knowledge, none of the other techniques previously presented in the cartographic literature as solutions to this problem have actually been used in production mapping. Figures 2 and 3 show label placements before and after overplot resolution for a small map with 40 point symbol labels. Similar quality results have been obtained on much larger data sets. For example, problems with over 2100 point symbol labels and over 1600 label conflicts have been executed on a VAX 750 in less than 1 CPU hour.

CONCLUSIONS

The optimization model for map label placement provides a practical solution to an important problem in automated cartography. Its usefulness has been proven in the compilation of maps with thousands of point feature labels and hundreds of label conflicts.

There are several unresolved technical problems in the implementation of this type of algorithm. Research to compare different ways of solving the optimization problem would be very useful. Also, better methods for displaying information about labels deleted from a map and methods to assist the interactive editing of the map created by this type of algorithm need to be investigated.

REFERENCES

Ahn, J. 1984. Automatic Map Name Placement System. Image Processing Laboratory Technical Report 063; Electrical, Computer, and Systems Engineer Department, Rensselaer Polytechnic Institute, Troy, New York.

Ahn, J. and Freeman, H. 1983. A Program for Automatic Name Placement. Proceedings, AUTO-CARTO VI, Vol. 2, pages 444-453.

Basoglu, U. 1984. A New Approach to Automated Name Placement Systems. Ph.D. Dissertation, Department of Geography, University of Wisconsin-Madison, available through University Microfilms International.

Chvatal, V. 1983. Linear Programming. W. H. Freeman and Company.

Cromley, R.C., 1985. An LP Relaxation Procedure for Annotating Point Features Using Interactive Graphics. Proceedings, AUTO- CARTO VII, pages 127-132.

Cromley, R.C. 1986. "A Spatial Allocation Analysis of the Point Annotation Problem," Proceedings, Second International Symposium on Spatial Data Handling, pages 38-49.

Held, M., Wolfe, P., and Crowder, H. 1974. Validation of Subgradient Optimization. Mathematical Programming, Vol 6, pages 62-88.

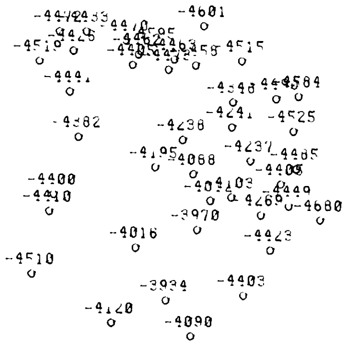

Figure 2. Point Symbol Labels With Overplotting

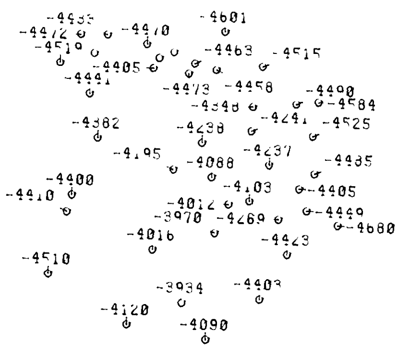

Figure 3. Point Symbol Labels With Overplots Resolved

Hirsch, S.A. 1982. An Algorithm for Automatic Name Placement Around Point Data. The American Cartographer, Vol. 9, No. 1, pages 5-17.

Johnson, E.L., Kostreva, M., and Suhl, V. 1985. Solving 0-1 Integer Programming Problems Arising from Large Scale Planning Models. Operations Research, Vol. 33, No. 4, pages 803-819.

Langran, M.S. and Poiker, T.K., 1986. "Integration of Name Selection and Name Placement," Proceedings, Second International Symposium on Spatial Data Handling, pages 50-64.

Mower, J.E., 1986. "Name Placement of Point Features Through Constraint Propagation," Proceedings, Second International Symposium on Spatial Data Handling, pages 65-73.

Pfefferkorn, C., Burr, D., Harrison, D., Heckman, B., Oresky, C., and Rothermel, J., 1985. ACES: A Cartographic Expert System. Proceedings, AUTO-CARTO VII, pages 399-407.

Zoraster, S. 1986. "Integer Programming Applied to the Map Label Placement Problem," Cartographica, Vol 22, No 3, pages 16-27.