Chapter 9

Street labeling

In day to day life we often have to find our way in cities in which we have never been before. When we travel long distances by car we usually have road maps at our disposal. Here only the main roads are on the map. On arrival in the city we switch to a city map. A city map shows all streets of a particularly city with the corresponding names. To avoid having large map sheets, the scale is chosen so that it is just possible to distinguish different streets and their names. That means that space on city maps for placing the names is limited. It is often not possible to put the full name of the street centered halfway the street, because the names of different streets may not overlap. Therefore, the name of the label must often be abbreviated, broken into pieces or placed partially outside the street. The many possible positionings of a label and the interaction between them make it hard for a cartographer to make a city map in limited time. A computer could be the solution. Also with the advent of vehicle navigation systems the amount of electronically available street data increased rapidly. So street data for automated making of city maps is now in most cases available. There exist a few programs for automated city map labeling. Maplex from ESRI and Label-EZ from MapText are software packages for map labeling and the Dutch company Andes used automated labeling to produce a book with all city maps of the Netherlands. Although there has been progress, high quality automated labeling of city maps is still a difficult task. There is a difference between European style and U.S. style street labeling. In the latter style street names are placed along the single centerlines of the roads. European style has road casings and the street names can be placed within the road casing. This chapter focuses on European style city maps of the Netherlands. But it could easily be adapted for European style labeling of other countries, and with more effort for U.S. style labeling.

In this chapter we use the following approach. First we describe the different label forms allowed for city maps. Then the rules for city map labeling are presented. To assess the quality of a labeling we use a quality function that, given a labeling, outputs a number indicating the quality of the labeling. Our goal is to find the labeling with the highest value of the quality function. First we generate a number of possible labels for a street. These are called candidates. The optimization heuristic simulated annealing is used to select the candidates that together produce a labeling with a value of the quality function that is hopefully close to optimal. We present a few examples of city maps produced by this approach. Our approach differs from existing computer programs in that we use a quality function which is composed of different, almost independent labeling aspects. Due to limited information content of the data used, we could not implement all rules in the quality function. Nevertheless the city maps produced give an impression of the possible power of the approach presented in this chapter.

9.1 Label forms

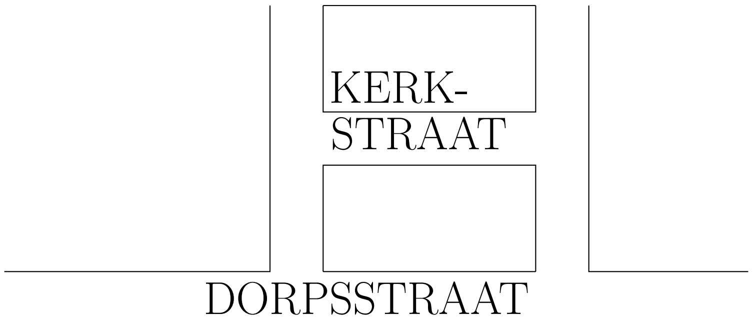

Figure 9.1: The standard placement of a street name ("Dorpsstraat") and the stacked placement ("Kerkstraat").

In contrast to labels for point features there are many possibilities for a label name to be present on a map. A street name can be placed inside the street casing, or it can be stacked in two or three parts with parts above or below the street, see Figure 9.1.

Different pieces of a name can be drawn separately on a map. A label form is a way of writing a label on a map with the label separated in different parts and using abbreviations. The parts of a label form can be placed at some distance from each other in a street. For example John F. Kennedystraat can be written in the following ways (the parts are bracketed out):

- (John F. Kennedystraat)
- (John F.) (Kennedystraat)
- (John F. Kennedy-) (straat)
- (John F.) (Kennedy-) (straat)

- (John F. Kennedystr.)
- (John F.) (Kennedystr.)
- (John F. Kennedy-) (str.)
- (John F.) (Kennedy-) (str.)
- (J. F. Kennedystraat)
- (J. F. Kennedy-) (straat)
- (J. F. Kennedystr.)
- (J. F. Kennedy-) (str.)

A name can be chopped in two at spaces and before general map words as straat, weg, plein (street, road and square in English) and so on. These general map words can also be abbreviated to str, wg, pln. Christian names are often abbreviated in city maps. A single part of a label form usually has more than 2 letters, so "(John)(F.)(Kennedystraat)" is not allowed. Only if the parts of a label form are stacked it is allowed since then the parts are close to each other. Then "(J.F.)(Kennedy)(straat)" is allowed as a label form.

We also add as a label form to each street a number that corresponds to the street. If there is no space on the city map to place a label form without intersections, the number of the street is placed on the map. Adding a small index on the city map then could give the name of the street corresponding to the number.

9.2 Labeling rules

Labeling a city map is fundamentally different from labeling a map with cities, rivers and areas. The streets form a network with a density of crossings that is unlikely to be found on normal maps. We are not aware of any literature on rules for labeling city maps. By studying how cartographers have labeled city maps, we were able to distill a number of rules. Except for the category feature visibility, the categories are the same as in Chapter 7. We did not include the category feature visibility because it is largely implied by the label visibility rules.

9.2.1 Association rules

These rules ensure that a particular label and its street can be associated unambiguously to each other. We start with the definition of the *course* of a street, see Figure 9.2:

- The course of a street follows corners and curves in the street.
- At a crossing the course of a street proceeds in the extension of the street.

- The course of a street ends at a T-junction.
- At a Y-junction the course of a street is not clear. It depend on the street types and the exact angles between the streets.

The main association rule for labeling a city map is the following:

- A label for a street is valid for the whole course of the street.

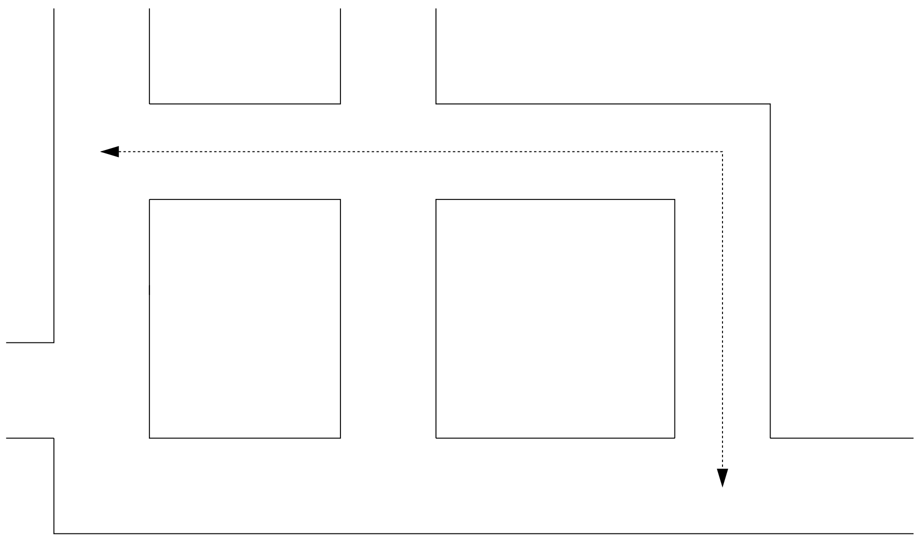

Figure 9.2: The course of a street (dotted) follows corners, follows the extension at crossings, and stops at T-crossings.

If the course of a street has two different labels then the change of the name of the street is indicated by the following situations:

- At a 90 degree corner.
- At the crossing of the course of a street with a major road.
- At the crossing of the course of the street with an equally important side-street.
- At the boundary of a municipality, province or country.
- At the crossing with a river or canal.

A cul-de-sac without a label has the same name as the street at which it starts. This is also true for general street patterns without label which all lead into the same street.

Some other association rules:

- A label should be either inside the road or parallel to it.
- If the letters of the label are not all capitals, then preferably the label should be placed on top of the street rather than below it.
- There should not be too much spacing between different parts of the same label.

9.2.2 Visibility rules

The visibility rules deal with how well the label is visible on the map. It is influenced by other features and labels on the map.

- There should be no overlap of labels.
- A label should not cross the edges (road casing) of other streets.
- If necessary it is allowed that the label sticks out at an end of the street.

9.2.3 Aesthetic rules

The aesthetic rules deal with the shape of the label itself. They are not influenced by the position of the label on the map, nor by other map features.

- Do not place the label upside-down.
- The label should not curve too much.
- Only use abbreviations if necessary.
- Do not repeat the label too often on a short interval.
- Use roughly the same spacing between different parts of the same label.
- The label should not have too many parts.

9.3 Quality function

The elements that define the quality function are in many aspects the same as in Chapter 7 of this thesis. We compute three types of quality of a labeling. The association quality shows how well the streets and their labels are associated to each other. The label visibility quality is a measure for how well the labels are visible, taking into account overlap between labels and intersections with other streets. The aesthetic quality represents how good the labels look in isolation from other labels and map objects. It takes into account curvature, abbreviations, and number of parts. Only the most important rules are expressed in the quality function. The other rules could be incorporated in the quality function to make it more fine grained.

Elementary road pieces are pieces of roads between crossings, boundaries, sharp curves, rivers, or canals. The quality function is a weighted sum of the partial quality functions for association, label visibility, and aesthetic quality. The association quality is the sum of the association quality of the elementary road pieces. The label visibility and aesthetic quality are, respectively, the sum of the label visibility and aesthetic qualities of the individual labels.

Let L be the set of labels of the city map, F the set of all fixed features on the city map (road casing, symbols), and R the set of elementary road pieces,

The general form of our quality function thus is:

$$\begin{aligned}\text{Quality}(L \cup R \cup F) \; = \; & \\ & w_{\text{aesthetics}} \cdot \textstyle\sum_{l \in L} \rho_{\text{aesthetics}}(l) \cdot \text{quality}_{\text{aesthetics}}(l) \\ + \; & w_{\text{label-vis}} \cdot \textstyle\sum_{l \in L} \rho_{\text{label-vis}}(l) \cdot \text{quality}_{\text{label-vis}}(l, F, L) \\ + \; & w_{\text{association}} \cdot \textstyle\sum_{r \in R} \rho_{\text{association}}(r) \cdot \text{quality}_{\text{association}}(r, R, L)\end{aligned}$$

where $w_{\text{aesthetics}}, w_{\text{label-vis}}$, and $w_{\text{association}}$ are factors that weigh the relative importance of the three partial quality functions. The constants $\rho_{\text{aesthetics}}(l)$, $\rho_{\text{label-vis}}(l)$ and $\rho_{\text{association}}(r)$ are the priorities given to each separate feature. We set $\rho_{\text{aesthetics}}(l) = \rho_{\text{label-vis}}(l) = 1/|L|$ with $|L|$ the number of labels, and we set $\rho_{\text{association}}(r) = d(r)/d(R)$ with $d(r)$ the length of elementary road piece r and $d(R) = \sum_{r \in R} d(r)$.

In the next three subsections we give relatively simple examples of quality functions for measuring association, label visibility, and aesthetics.

9.3.1 Association quality

For each elementary road piece r we compute the association to its label. This value ranges from 0 (unclear association) to 1 (clear association). First we compute all labels placed at streets that include r in their course. If one of those labels does not belong to the street of r, the association quality of r is 0, otherwise it is 1. Thus the association quality of an elementary road piece is only 1 if it is not possible that it is associated to a label of another street.

It is also possible to make the association more refined by incorporating the distances from r to the labels to which it is possibly associated. If the distance to its own label is much smaller than the distance to the other labels it can also be given an association quality of 1. Not only the distance is important but also the presence of a name change indicator. One could increase the value representing the distance if, on the path from r to a label, the street makes sharp curves or if the path makes turns smaller than 160 degrees at crossings.

9.3.2 Label visibility quality

The label visibility quality is good if the label does not intersect other labels and lies completely inside the road casing. If it intersects other labels, the label visibility quality is decreased. The same is true if the label intersects the road casing of another street. Also a decrease in plain label visibility quality is incurred if the label extends beyond the end of the street. If all label candidates of a street intersect another label or the road casing of another street, it is better to place the number instead of the name of the street. This is achieved by allowing a number to be used to label a street, but with a low visibility value. To quantify these aspects we define the following functions:

- *intersectLabels*(l, L) is the number of intersections of label l with other labels.
- *intersectStreets*(l, F) is 1 if l intersects the road casing of another street, else it is 0.
- *sticksOut*(l, F) is 1 if l sticks out at an end of its street, else it is 0.
- *isNumber*(l) is 1 if l is a number, and 0 otherwise.

The function *mapIntersectLabels* is a discrete function that maps the value of *intersectLabels* to a value in the interval $[0, 1]$. More information on mapping functions can be found in Section 7.5.1. Similarly we use discrete functions *mapIntersectStreets*, *mapSticksOut*, and *mapIsNumber* to convert the values 0 or 1 from *intersectStreets*, *sticksOut*, and *isNumber* to values in the interval $[0, 1]$ to weigh the importance of the different aspects.

The label visibility quality function is:

$$\begin{aligned} \text{quality}_{\text{label-vis}}(l, F, L) = \; & \mathit{mapIntersectLabels} \; (\mathit{intersectLabels}(l, L)) \\ & \cdot \mathit{mapIntersectStreets} \; (\mathit{intersectStreets}(l, F)) \\ & \cdot \mathit{mapSticksOut} \; (\mathit{sticksOut}(l, F)) \\ & \cdot \mathit{mapIsNumber} \; (\mathit{isNumber}(l)) \end{aligned}$$

The reason for multiplying the quality aspects is given in Section 7.5.1.

9.3.3 Aesthetic quality

The aspects that influence the aesthetic quality of a label are among others the curvedness, the use of abbreviations and the number of parts a label is divided into.

To quantify these aspects we define the following functions:

- *curvedness*(l) is the average change in angle between two consecutive letters of the label.
- *abbreviations*(l) is the number of abbreviated words in the label text.
- *parts*(l) is the number of parts the label is divided into.

The function *mapCurvedness* is a fading function that maps the value of *curvedness* to a value in the interval $[0, 1]$, where 0 represents a very curved label and 1 a straight label. Similarly we use discrete functions *mapAbbreviations* and *mapParts* to convert the values of *abbreviations* and *parts* to the interval $[0, 1]$ or a subinterval of it.

The aesthetic quality function is:

$$\begin{aligned}\text{quality}_{\text{label-vis}}(l) \quad = \quad & mapCurvedness \quad (curvedness(l)) \\ & \cdot\, mapAbbreviations \; (abbreviations(l)) \\ & \cdot\, mapParts \quad (parts(l))\end{aligned}$$

9.4 Generation of candidate labels

In principle we would like to have all possible label positions for a street as candidates. Since the number of possible label positions is too large, we restrict ourselves to generating candidates that are different. If two candidate labels can be made equal by shifting one candidate to the left or to the right without having an end of the candidate cross a junction, the two candidates are not really different. For each label form we generate all really different candidates. If the number of generated candidates is large, we reduce this number by only keeping the best candidates. Since for a street with many candidates, only one of the best candidates would be chosen for the labeling, the removal of candidates probably does not have a negative effect on the quality of the labeling.

First we look at the case where the label form consists of one part. The candidates for this label form are all positions of the label where the start of the label is just beyond a junction or where the end of the label just stops before a junction. If a label candidate just stops before a junction and the candidate lies completely in the street, then we can remove this candidate because it can be made equal to another candidate that starts at a junction. We can also remove a candidate if it spans all the elementary road pieces and junctions of another candidate, see Figure 9.3.

If a label form has more than one part, we use a recursive procedure. The first part is placed the same way as a single part label (beginning just after a junction or ending just before a junction of the street), then for one such label position we know that the other parts of the label form must be placed to the right of the first part. The second part can be placed with its begin just after a junction, but with the restriction that the second part is placed to the right of the first part. A possible third part of the label form is placed in the same manner to the right of the first and second part. The recursion ends when we place the last part. In this way we generate a number of candidates.

If a street has many elementary road pieces, the number of candidates generated in this way is very large. In this case we select the best candidates and delete the others. We delete candidates with abbreviated words and candidates with a part sticking out of the end of the street.

If all candidates have parts that stick out at an end of the street, we know that even the abbreviated name of the street is too long to be placed inside the street. Generating candidates with stacked labels is in those cases the next step. If a label form has two parts, the first part can be placed above the street and the second part inside the street, or the first part inside the street and the second part below the street. If a label form has three parts, the first is placed above the street, the second inside the street, and the third part below the street. The

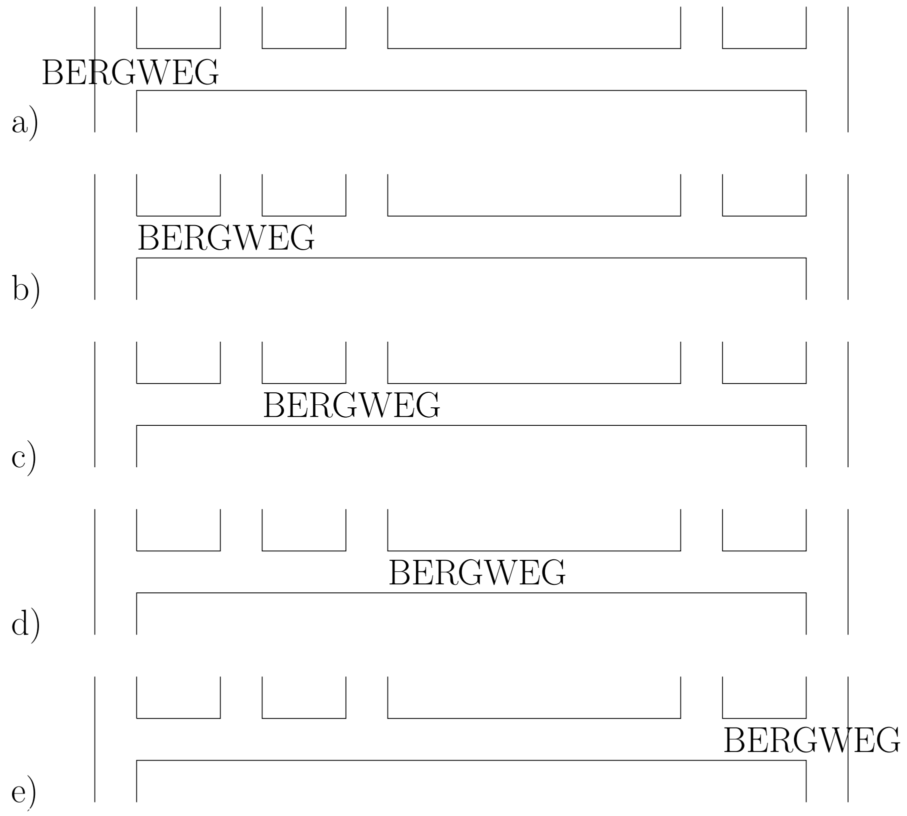

Figure 9.3: The candidates for this street are a, b, c, d, and e. Candidate a ends at a junction and sticks out. Candidates b, c, d, and e begin at a junction. Candidate c is removed because it spans the same and even more elementary road pieces and junctions than candidate d.

first part can be placed just after a junction, the other parts are then placed below the first part. We delete candidates that have a part above or below a street that intersects another street.

To avoid labels with letters upside-down we only keep candidates for which the average angle of the letters of the label with respect to the x-axis is between -90 and 90 degrees. But we keep all candidates if there would be too few candidates left after this operation, for example in case of a very curved street or a square.

9.5 Optimization

We optimize the value of the quality function by using simulated annealing, see Kirkpatrick, Gelatt, and Vecchi [67]. This algorithm derives its inspiration from the thermodynamic process by which solids are heated and cooled gradually (annealed) to a crystalline state with minimum energy. Steps in the search space that make the solution worse are allowed with a certain probability. This makes it possible to escape from local optima in the search space.

Initially each street is randomly assigned one of its label candidates. In an iterative fashion, a street is chosen randomly and its candidate is replaced randomly by one of its other candidates. The difference of the value of the quality function as a result of this candidate change is computed. This computation is done locally by investigating the changes in the neighborhood of the old and new candidate. If the quality increases, the change is accepted. Else, it is only accepted with a certain probability depending on how much worse the quality would become. During the algorithm the probability of accepting changes that worsen the quality becomes smaller and smaller. So after a number of iterations the quality stays the same and the final labeling is output.

9.6 Test results

The data for the street network are part of the Dutch "Nationaal Wegenbestand (NWB)"maintained by a branch of the Dutch Ministry of Traffic. It contains all streets and roads of the Netherlands, and when possible the names associated to them. We obtained the data set for the town "Zeist", which has 461 named streets. The data format of the data set is the GDF-standard. GDF stands for Geographic Data Files and is a European standard that is used to describe and transfer road networks and road related data. The data does not contain the width of streets nor the importance of the streets, so we could not take these aspects into account for our test. Also we did not abbreviate Christian names.

In Figures 9.4 and 9.5 a labeling produced by our method is shown. It took 10 minutes to compute the whole labeling. The road casing is at some points not correct. It was computed using the center line of the streets. Improving the algorithm for computing the road casing could solve this problem. The way we generate the candidates enables the optimization algorithm to place many

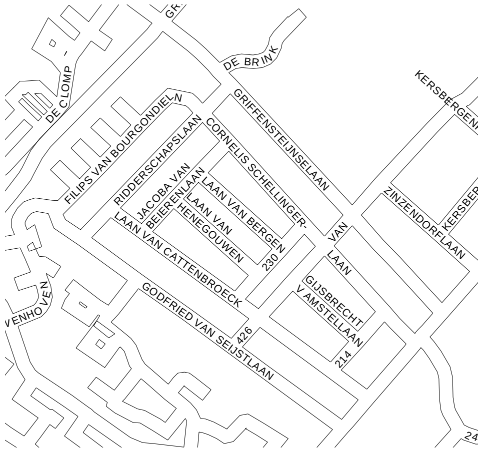

Figure 9.4: Part of the street labeling of Zeist.

street labels. Letting label parts start just after or letting them end just before a junction creates space at junctions to prevent that labels intersect each other. Adding more rules to the quality function would further improve the labeling. As a post-processing step the label parts could be centered with respect to the elementary road pieces they span.

9.7 Conclusions

We have presented a method for the automated labeling of city maps. This method was devised for city maps of the Netherlands, but it can easily be adapted for labeling cities in other countries. We studied the rules for labeling the streets of a city. These rules were divided into three categories: association, label visibility, and aesthetics. We presented a quality function that describes the quality of a city map labeling. It is composed of quality functions for the three categories. Using an optimization method we sought an optimum value for this quality function. Examples of optimal labelings for a town in the Netherlands were given. We think that by incorporating more rules in the quality function and by generating more candidate labels the labelings could be made even better. Our approach is flexible and is a step forward in automatically producing high-quality labelings of city maps.

Figure 9.5: Part of the street labeling of Zeist.

Bibliography

[1] Pankaj K. Agarwal and Jeff Erickson. Geometric range searching and its relatives. Technical Report CS 1997-11, Department of Computer Science, Duke University, 1997.

[2] Pankaj K. Agarwal, Marc van Kreveld, and Subhash Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11:209–218, 1998.

[3] John Ahn and Herbert Freeman. A program for automatic name placement. *Cartographica*, 21(2–3):101–109, 1984.

[4] David H. Alexander and Carl S. Hantman. Automating linear text placement within dense feature networks. In *Proc. Auto-Carto 12*, pages 311–320. ACSM/ASPRS, Bethesda, 1995.

[5] Georges Alinhac. *Cartographie Théorique et Technique*, chapter IV. Institut Géographique National, Paris, 1962.

[6] Hiromi Aonuma, Hiroshi Imai, and Yahiko Kambayashi. A visual system of placing characters appropriately in multimedia map databases. In *Proceedings of the IFIP TC 2/WG 2.6 Working Conference on Visual Database Systems*, pages 525–546. North-Holland, 1989.

[7] Mathieu Barrault. An automated system for name placement which complies with cartographic quality criteria: The hydrographic network. In *Proceedings of the Conference on Spatial Information Theory (COSIT'97)*, volume 1329 of *Lecture Notes in Computer Science*, pages 499–500, Pittsburgh, PA, 1997. Springer-Verlag.

[8] Mathieu Barrault. *Le placement cartographique des écritures: résolution d'un problème à forte combinatoire et présentant un grand nombre de contraintes variées.* PhD thesis, Marne-la-Vallée University, November 1998.

[9] Mathieu Barrault and François Lecordix. An automated system for linear feature name placement which complies with cartographic quality criteria. In *Proc. Auto-Carto 12*, pages 321–330, Charlotte, NC, March 1995. ACSM/ASPRS, Bethesda.

[10] Martin Beale and John A. Tomlin. Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables. In J. Lawrence, editor, *Proceedings of the Fifth International Conference on Operations Research*, pages 447–454, London, 1970. Tavistock Publications.

[11] D. Beus and D. Crockett. Automated production of 1:24,000 scale quadrangle maps. In *Proceedings of the ASPRS/ACSM Annual Convention and Exposition 1*, pages 94–99, 1994.

[12] Wilhelm Bonacker. *Die Namenstellung in geographischen Karten*. Studien zur Kartographie 1 a. Haller, Berlin, 1957.

[13] A. Raymond Boyle. Computer aided map compilation. Technical report, Department of Electrical Engineering, University of Saskatchewan, Canada, 1973.

[14] A. Raymond Boyle. Report on symbol and name manipulation and placement. Technical report, Department of Electrical Engineering, University of Saskatchewan, Canada, 1974.

[15] Alberto Caprara, Matteo Fischetti, and Adam Letchford. On the separation of maximally violated mod-k cuts. *Mathematical Programming*, 87:37–56, 2000.

[16] Jon Christensen, Joe Marks, and Stuart Shieber. Algorithms for cartographic label placement. In *Proceedings of the American Congress on Surveying and Mapping 1*, pages 75–89, 1993.

[17] Jon Christensen, Joe Marks, and Stuart Shieber. Placing text labels on maps and diagrams. In Paul Heckbert, editor, *Graphics Gems IV*, pages 497–504. Academic Press, Boston, MA, 1994.

[18] Jon Christensen, Joe Marks, and Stuart Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.

[19] Anthony C. Cook and Christopher B. Jones. A Prolog interface to a cartographic database for name placement. In *Proceedings 4th International Symposium on Spatial Data Handling*, pages 701–710, 1990.

[20] Anthony C. Cook and Christopher B. Jones. A Prolog rule-based system for cartographic name placement. *Computer Graphics Forum*, 9(2):109–126, 1990.

[21] Thomas Cormen, Charles Leiserson, and Ronald Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 1990.

[22] Robert G. Cromley. An LP relaxation procedure for annotating point features using interactive graphics. In *Proc. Auto-Carto 7*, pages 127–132, 1985.

[23] Harlan P. Crowder, Ellis L. Johnson, and Manfred W. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834, 1983.

[24] George B. Dantzig and Mukund N. Thapa. *Linear Programming, 1: Introduction*. Springer-Verlag, Berlin, 1997.

[25] Ludwig Danzer and Branko Grünbaum. Intersection properties of boxes in $\mathbb{R}^d$. *Combinatorica*, 2(3):237–246, 1982.

[26] Amitava Datta, Hans-Peter Lenhof, Christian Schwarz, and Michiel H. M. Smid. Static and dynamic algorithms for k-point clustering problems. *J. Algorithms*, 19:474–503, 1995.

[27] Mark de Berg, Mark van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, second edition, 2000.

[28] Borden D. Dent. *Cartography*, chapter 14. Wm. C. Brown Publishers, 1996.

[29] Y. Djouadi. Cartage: A cartographic layout system based on genetic algorithms. In *Proc. EGIS'94*, pages 48–56, 1994.

[30] Srinivas Doddi, Madhav V. Marathe, Andy Mirzaian, Bernard M.E. Moret, and Binhai Zhu. Map labeling and its generalizations. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms (SODA'97)*, pages 148–157, New Orleans, LA, 4–7 January 1997.

[31] Srinivas Doddi, Madhav V. Marathe, and Bernard M.E. Moret. Point labeling with specified positions. In *Proc. 16th Annu. ACM Sympos. Comput. Geom. (SoCG'00)*, pages 182–190, Hongkong, 12–14 June 2000.

[32] Jeffrey S. Doerschler and Herbert Freeman. An expert system for dense-map name placement. In *Proc. Auto-Carto 9*, pages 215–224, 1989.

[33] Jeffrey S. Doerschler and Herbert Freeman. A rule-based system for dense-map name placement. *Communications of the ACM*, 35:68–79, 1992.

[34] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized curve or its caricature. *Canadian Cartographer*, 10(2):112–122, 1973.

[35] L.R. Ebinger and A.M. Goulette. Automated name placement in a non-interactive environment. In *Proc. Auto-Carto 9*, pages 205–214, 1989.

[36] Shawn Edmondson, Jon Christensen, Joe Marks, and Stuart Shieber. A general cartographic labeling algorithm. *Cartographica*, 33(4):13–23, 1997.

[37] Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric graphs. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, Washington, DC, 7–9 January 2001. to appear.

[38] ESRI. MAPLEX—a fully automated cartographic name-placement software. `http://www.esri.com/software/maplex/`, 1998.

[39] Evermap. Evername—an advanced cartographic label-placement software for MapInfo Professional. `http://www.evermap.com/evername.htm`, 1999.

[40] M. Feigenbaum. Method and apparatus for automatically generating symbol images against a background image without collision utilizing distance-dependent attractive and repulsive forces in a computer simulation, 1994. Assigned to Hammond Inc., Maplewood, New Jersey. U.S. Patent filed 11/5/93, received 10/11/94.

[41] Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Annu. ACM Sympos. Comput. Geom. (SoCG'91)*, pages 281–288, 1991.

[42] Robert J. Fowler, Michael S. Paterson, and Steven L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12(3):133–137, 1981.

[43] Greg N. Frederickson and Donald B. Johnson. Generalized selection and ranking: Sorted matrices. *SIAM Journal on Computing*, 13(1):14–30, 1984.

[44] Herbert Freeman. An expert system for the automatic placement of names on a geographic map. *Information Sciences*, 45:367–378, 1988.

[45] Herbert Freeman. Computer name placement. In D.J. Maguire, M.F. Goodchild, and D.W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, pages 445–456. Longman, London, 1991.

[46] C. Friden, Alain Hertz, and Dominique de Werra. A technique for finding stable sets in large graphs with tabu search. *Computing*, 42:35–44, 1989.

[47] Alain Hertz, Eric Taillard, and Dominique de Werra. Tabu search. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 121–136. John Wiley & Sons Ltd, Chichester, 1997.

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

[49] Daniel P. Huttenlocher, Gregory Klanderman, and William J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.

[50] Hiroshi Imai and Takao Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *Journal on Algorithms*, 4:310–323, 1983.

[51] Hiroshi Imai and Takao Asano. Efficient algorithms for geometric graph search problems. *SIAM Journal on Computing*, 15(2):478–494, 1986.

[52] Eduard Imhof. Die Anordnung der Namen in der Karte. In *International Yearbook of Cartography*, pages 93–129, Bonn Bad Godesberg, 1962. Kirschbaum.

[53] Eduard Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[54] Edward H. Isaaks and R. Mohan Srivastava. *An Introduction to Applied Geostatistics*. Oxford University Press, New York, 1989.

[55] Claudia Iturriaga. *Map Labeling Problems*. PhD thesis, University of Waterloo, 1999.

[56] Claudia Iturriaga and Anna Lubiw. Elastic labels: The two-axis case. In *Proceedings of the Symposium on Graph Drawing (GD'97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 181–192, Rome, Italy, 18–20 September 1997. Springer-Verlag.

[57] Claudia Iturriaga and Anna Lubiw. NP-hardness of some map labeling problems. Technical Report CS-97-18, University of Waterloo, Canada, 1997.

[58] Claudia Iturriaga and Anna Lubiw. Elastic labels around the perimeter of a map. In *Proceedings of the 8th International Workshop on Algorithms and Data Structures (WADS'99)*, volume 1663 of *Lecture Notes in Computer Science*, pages 306–317, Vancouver, B.C., Canada, 12–14 August 1999. Springer-Verlag.

[59] Michiel Jansen and Marc van Kreveld. Evaluating the consistency of cartographic generalization. In T.K. Poiker and N. Chrisman, editors, *Proc. 8th Int. Symp. on Spatial Data Handling*, pages 668–678, 1998.

[60] Peter Jolly. Naming places, placing names. *Mapping Awareness*, 10:28–30, 1996.

[61] Christopher Jones. Cartographic name placement with Prolog. *IEEE Computer Graphics & Applications*, 9(5):36–47, 1989.

[62] Konstantinos G. Kakoulis and Ioannis G. Tollis. On the edge label placement problem. In *Proceedings of the Symposium on Graph Drawing (GD'96)*, volume 1190 of *Lecture Notes in Computer Science*, pages 241–256. Springer-Verlag, 1997.

[63] Konstantinos G. Kakoulis and Ioannis G. Tollis. On the multiple label placement problem. In *Proc. 10th Canadian Conf. Computational Geometry (CCCG'98)*, pages 66–67, 1998.

[64] Konstantinos G. Kakoulis and Ionnis G. Tollis. A unified approach to labeling graphical features. In *Proc. 14th Annu. ACM Sympos. Comput. Geom. (SoCG'98)*, pages 347–356, June 1998.

[65] T. Kato and H. Imai. The NP-completeness of the character placement problem of 2 or 3 degrees of freedom. In *Record of Joint Conference of Electrical and Electronic Engineers in Kyushu*, page 1138, 1988. In Japanese.

[66] Sung Kwon Kim, Chan-Su Shin, and Tae-Cheon Yang. Labeling a rectilinear map with sliding labels. Technical Report HKUST-TCSC-1999-06, Hongkong University of Science and Technology, July 1999.

[67] Scott Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[68] Gunnar W. Klau and Petra Mutzel. Optimal labelling of point features in the slider model. In *Proc. Sixth Annual International Computing and Combinatorics Conference (COCOON'00)*, Lecture Notes in Computer Science, Sydney, 26–28 July 2000. Springer-Verlag. to appear.

[69] Lars Knipping. Beschriftung von Linienzügen. Master's thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, November 1998.

[70] Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM J. Discr. Math.*, 5(3):422–427, 1992.

[71] Joshua C. Kramer. Line feature label placement for ALPS5.0. unpublished manuscript, available at `http://paul.rutgers.edu/~jckramer/academics/Report/`, 1997.

[72] Ludek Kučera, Kurt Mehlhorn, Bettina Preis, and Erik Schwarzenecker. Exact algorithms for a geometric packing problem. In *Proc. 10th Sympos. Theoret. Aspects Comput. Sci.*, volume 665 of *Lecture Notes in Computer Science*, pages 317–322. Springer-Verlag, 1993.

[73] Gail E. Langran and Thomas K. Poiker. Integration of name selection and name placement. In *Proc. Auto-Carto 8*, pages 50–64, 1986.

[74] David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[75] William A. Mackaness. A constraint based approach to human computer interaction in automated cartography. In *Proc. of the 17th Int. Cartographic Conference*, pages 1423–1432, 1995.

[76] MapText. Label-EZ for automated cartographic text placement—a high-performance productivity tool for the mapping industry. `http://www.maptext.com`, 1999.

[77] Joe Marks and Stuart Shieber. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard CS, 1991.

[78] S. Masuda, S. Kimura, T. Kashiwabara, and T. Fujisawa. On the Manhattan wiring problem. Technical Report CAS 83-20, Institute of Electronics and Communication Engineers of Japan, 1983. in Japanese.

[79] Kurt Mehlhorn and Stefan Näher. *LEDA: A Platform of Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge, England, 1999.

[80] William Mills. Practical considerations in name placement: A defence of Pinhas Yoeli. *Cartographica*, 31(4):58–62, 1994.

[81] Institut Géographique National. *Les Écritures sur les Cartes Topographiques*. Paris, 1934.

[82] George L. Nemhauser and Les E. Trotter. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974.

[83] George L. Nemhauser and Les E. Trotter. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[84] Gabriele Neyer and Frank Wagner. Labeling downtown. Technical Report TR-324, Department of Computer Science, ETH Zurich, May 1999.

[85] Manfred W. Padberg. On the Facial Structure of the Set Packing Polyhedra. *Mathematical Programming*, 5:199–216, 1973.

[86] Manfred W. Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[87] C. Pfefferkorn, D. Burr, D. Harrison, B. Heckman, C. Oresky, and J. Rothermel. ACES: A cartographic expert system. In *Proc. Auto-Carto 7*, pages 399–407, 1985.

[88] Itzhak Pinto and Herbert Freeman. The feedback approach to cartographic areal text placement. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, pages 341–350. Springer-Verlag, New York, 1996.

[89] Chung Keung Poon, Binhai Zhu, and Francis Chin. A polynomial time solution for labeling a rectilinear map. *Information Processing Letters*, 65(4):201–207, 1998.

[90] Mike Preuß. Solving map labeling problems by means of evolution strategies. Master's thesis, Fachbereich Informatik, Universität Dortmund, February 1998.

[91] Lilian Pun-Cheng and Geoffrey Y.K. Shea. Automatic bilingual name placement of 1:1000 map sheets of hong kong. In *Proceedings of the 19th International Cartographic Conference (ICA'99)*, pages 925–930, 1999.

[92] Zhongping Qin, Alexander Wolff, Yinfeng Xu, and Binhai Zhu. New algorithms for two-label point labeling. In *Proc. 8th Annu. Europ. Symp. on Algorithms (ESA'00)*, volume 1879 of *Lecture Notes in Computer Science*, pages 368–379, Saarbrücken, 5–8 September 2000. Springer-Verlag.

[93] Günther Raidl. An evolutionary approach to point-feature label placement. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, page 807. Morgan Kaufmann, July 1999.

[94] Chong S. Rim and Kazuo Nakajima. Complexity results for rectangle intersection and overlap graphs. Technical Report SRC TR 88-64, Electrical Engineering Department and Systems Research Center, University of Maryland, 1988.

[95] Arthur H. Robinson, Joel L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling, and Stephen C. Guptill. *Elements of Cartography*, chapter 22. John Wiley & Sons, Inc., 1995.

[96] Yukio Sadahiro. Size of map labels used in GIS and loss of literal information. *Cartographica*, 32:29–41, 1995.

[97] Tycho Strijk and Marc van Kreveld. Labeling a rectilinear map more efficiently. *Information Processing Letters*, 69(1):25–30, 1999.

[98] Tycho Strijk and Marc van Kreveld. Practical extensions of point labeling in the slider model. In *Proc. 7th ACM Symposium on Advances in Geographic Information Systems*, pages 47–52, 5–6 November 1999.

[99] Tycho Strijk, Bram Verweij, and Karen Aardal. Algorithms for maximum independent set applied to map labelling. Technical Report UU-CS-2000-22, Department of Computer Science, Utrecht University, 2000.

[100] Tycho Strijk and Alexander Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 2000. to appear.

[101] Steven van Dijk, Dirk Thierens, and Marc de Berg. On the design of genetic algorithms for geographical applications. In W. Banzhaf, J. Daida, A.E. Eiben, M.H. Garzon, V. Honavar, M. Jakiela, and R.E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, pages 188–195. Morgan Kaufmann, July 1999.

[102] Steven van Dijk, Dirk Thierens, and Mark de Berg. Robust genetic algorithms for high quality map labeling. Technical Report UU-CS-1998-41, Department of Computer Science, Utrecht University, 1998.

[103] Steven van Dijk, Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Towards an evaluation of quality for label placement methods. In *Proceedings of the 19th International Cartographic Conference (ICA'99)*, pages 905–913, Ottawa, 14–21 August 1999. Int. Cartographic Association.

[104] Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21–47, 1999.

[105] Jan W. van Roessel. An algorithm for locating candidate labeling boxes within a polygon. *The American Cartographer*, 16(3):201–209, 1989.

[106] Oleg Verner, Roger Wainwright, and Dale Schoenefeld. Placing text labels on maps and diagrams using genetic algorithms with masking. *INFORMS Journal on Computing*, 9(3):266–275, 1997.

[107] Bram M. Verweij. *Selected Applications of Integer Programming: A Computational Study*. PhD thesis, Department of Computer Science, Utrecht University, Utrecht, 2000.

[108] Frank Wagner and Alexander Wolff. Map labeling heuristics: Provably good and practically useful. In *Proc. 11th Annu. ACM Sympos. Comput. Geom. (SoCG'95)*, pages 109–118, 5–7 June 1995.

[109] Frank Wagner and Alexander Wolff. A practical map labeling algorithm. *Computational Geometry: Theory and Applications*, 7:387–404, 1997.

[110] Frank Wagner and Alexander Wolff. A combinatorial framework for map labeling. In Sue H. Whitesides, editor, *Proceedings of the Symposium on Graph Drawing (GD'98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 316–331. Springer-Verlag, 13–15 August 1998.

[111] Frank Wagner, Alexander Wolff, Vikas Kapoor, and Tycho Strijk. Three rules suffice for good label placement. *Algorithmica Special Issue on GIS*, 2000. to appear.

[112] W.T. Wilkie. Computerized cartographic name processing. Master's thesis, Department of Electrical Engineering, University of Saskatchewan, Canada, 1973.

[113] Alexander Wolff, Lars Knipping, Marc van Kreveld, Tycho Strijk, and Pankaj K. Agarwal. A simple and efficient algorithm for high-quality line labeling. In Peter M. Atkinson and David J. Martin, editors, *Innovations in GIS VII: GeoComputation*, chapter 11, pages 147–159. Taylor & Francis, 2000.

[114] Alexander Wolff and Tycho Strijk. A map labeling bibliography. `http://www.math-inf.uni-greifswald.de/map-labeling/bibliography/`, 1996.

[115] Alexander Wolff, Michael Thon, and Yinfeng Xu. A better lower bound for two-circle point labeling. In D.T. Lee, editor, *Proc. 11th Annual International Symposium on Algorithms and Computation (ISAAC'00)*, Lecture Notes in Computer Science, Taipei, 18–20 December 2000. Institute of Information Science, Academia Sinica, Springer-Verlag. to appear.

[116] Laurence A. Wolsey. *Integer Programming*. John Wiley and Sons, New York, 1998.

[117] Chyan Victor Wu and Barbara Pfeil Buttenfield. Reconsidering rules for point-feature name placement. *Cartographica*, 28(1):10–27, 1991.

[118] Pinhas Yoeli. The logic of automated map lettering. *The Cartographic Journal*, 9:99–108, 1972.

[119] Binhai Zhu and Chung Keung Poon. Efficient approximation algorithms for multi-label map labeling. In A. Aggarwal and C. Pandu Rangan, editors, *Proc. 10th Annual International Symposium on Algorithms and Computation (ISAAC'99)*, volume 1741 of *Lecture Notes in Computer Science*, pages 143–152, Chennai, India, 16–18 December 1999. Springer-Verlag.

[120] Steven Zoraster. Integer programming applied to the map label placement problem. *Cartographica*, 23(3):16–27, 1986.

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

[122] Steven Zoraster. Expert systems and the map label placement problem. *Cartographica*, 28(1):1–9, 1991.

[123] Steven Zoraster. Practical results using simulated annealing for point feature label placement. *Cartography and GIS*, 24(4):228–238, 1997.

[124] Peter J. Zwaneveld, Leo G. Kroon, and Stan P.M. van Hoesel. Routing trains through a railway station based on a node packing model. Technical Report RM/97/030, Maastricht University FdEWB, Maastricht, 1997. To appear in European Journal on Operational Research.