یازدهمین کنفرانس بین المللی کامپیوتر انجمن کامپیوتر ایران، پژوهشگاه دانشهای بنیادی، پژوهشکده علوم کامپیوتر، ۴ تا ۶ بهمن ۱۳۸۴، تهران، ایران.
11th International CSI Computer Conference (CSICC'2006), School of Computer Science, IPM, Jan. 24-26, 2006, Tehran, Iran.

Incremental Labeling in Closed-2PM Model

Farshad Rostamabadi Mohammad Ghodsi
Computer Engineering Department,
Sharif University of Technology, Tehran, Iran

IPM School of Computer Science,
Institute for Studies in Fundamental Sciences, Tehran, Iran
farshad@mehr.sharif.edu ghodsi@sharif.edu

ABSTRACT

We consider an incremental optimal label placement in a closed-2PM model where labels are disjoint axis-parallel square-shaped of maximum length each attached to its corresponding point on one of its horizontal edges. Our goal is to efficiently generate a new optimal labeling for all points after each point insertion. Inserting each point may require several labels to flip or all labels to shrink. We present an algorithm that generates each new optimal labeling in $O(\lg n + k)$ time where k is the number of required label flips, if there is no need to shrink the label lengths, or in $O(n)$ time when we have to shrink labels. The algorithm uses $O(n)$ space in both cases.

Keywords

Computational Geometry, Map Labeling, Online Labeling.

1. INTRODUCTION

Automated label placement is an important problem in map generation and geographical information systems (GIS). This is to attach one or more labels (regularly in text) to each feature of the map, which may be a point, a line, a curve, or a region. The point feature label placement has received considerable attention. There are two basic requirements of any labeling: labels should be pairwise disjoint, and each label should have a common point with its feature [9, 19, 6]. Other variations of the problem let the features to receive more than one labels [2, 10, 14], or use specific shapes for labels [1, 8, 17, 16]. There are also two labeling models: fixed model, where some fixed positions given as possible label positions [9], and slider model, where the labels can be placed at any position while touching the feature[15, 7]. Optimal labeling of a set of points is generally an NP-Complete problem, but with some restrictions, it can be solved in polynomial time like the problem stated in [13] or the elastic labeling introduced in [4, 5].

We consider labeling of a set of points in the *closed-2PM model*, a variation of the well known 2P model [1], where labels are disjoint equal-length axis-parallel squares each attached exclusively to its corresponding point on the middle of one of its horizontal edges ('M' in 2PM comes from this property). In a closed-2PM labeling, two labels with intersecting edges are not disjoint. A closed-2PM labeling with the maximum label length is referred here as an *optimal labeling*. We will show that the time required to generate an optimal labeling in closed-2PM model is $\Omega(n \lg n)$ in algebraic computational tree model.

In this paper, we study the problem of *incremental labeling* where the goal is to insert a series of points, one at a time, in an initial optimal labeling, such that an optimal labeling is computed efficiently after each point insertion. A naive strategy to achieve this goal is to generate a new optimal labeling from scratch whenever a new point is inserted. This can be done by deciding for existence of a fixed-length labeling for all the given points by a transformation to an instance of 2SAT problem. Since there are at most $O(n)$ possible values for all label lengths [18, 3], an optimal labeling can be found with a binary search in $O(n \lg n)$ time, noting that any instance of 2SAT problem can be solved in $O(n)$ time [1]. So, inserting n points in an empty point set and generating an optimal labeling after each insertion needs $O(n^2 \lg n)$ time with this strategy.

The outline of our algorithm, presented in Sect. 3, is as follows. Given a set of n points $\mathcal{P}$, we compute some data structures in $O(n \lg n)$ time to build an initial optimal labeling in $O(n)$ time with a sweep line algorithm, introduced in Sect. 2. For each new point, we update our data structures and then decide if an optimal labeling of the same length, including the new point, exists. This decision, in addition to generating an optimal labeling, takes $O(\lg n + k)$ time where k is the number of changes that should be applied to previously optimal labeling. Otherwise, we use our updated data structures to generate a new optimal labeling in $O(n)$ time again with the sweep line algorithm. By this approach, inserting n points to an empty point set and generating an optimal labeling after each point insertion, requires $O(n^2)$ time in the worst case, but we expect a much better performance on the average.

2. CLOSED-2PM LABEL PLACEMENT PROBLEM

Given a set of n points $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, we define a *valid labeling* of $\mathcal{P}$ as a placement of equal-length axis-parallel square-shaped labels $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ such that ℓ_i is attached to p_i on the middle of its horizontal edges and with no pairs of intersecting labels[1]. The *length* of $\mathcal{L}$, denoted by $\sigma(\mathcal{L})$, is the (same) length of all labels in $\mathcal{L}$. A valid labeling with the maximum value of $\sigma(\mathcal{L})$ is referred to as an optimal labeling. Given $\mathcal{P}$, the problem of finding an optimal labeling is referred to as the *closed-2PM label placement problem*.

The time needed to decide for existence of a labeling of a given length for $\mathcal{P}$ is $\Omega(n \lg n)$, since we can reduce the ϵ-closeness problem to this problem as follows. Given a real number ϵ and n real numbers $\phi_1, \phi_2, \ldots, \phi_n$, two numbers ϕ_i and ϕ_j satisfying $|\phi_i - \phi_j| < \epsilon$ exist if and only if no valid labeling of length ϵ for a set of n points $\mathcal{P} = \{(\phi_i, 0) | 1 \leq i \leq n\}$ exists.

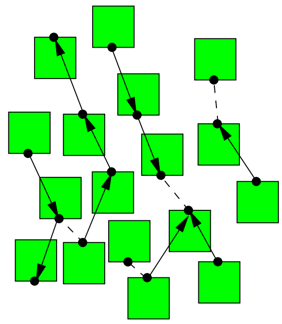

Figure 1: A given labeled map and the conflict graph: $\mathcal{B}$ edges (dashed), directed edges (solid).

2.1. Preliminaries

We define $\tau_i^{\uparrow}(\gamma)$ (resp. $\tau_i^{\downarrow}(\gamma)$) as the label of length γ attached to p_i on the middle of its top (resp. bottom) edge. Besides, $\tau_i^{\uparrow}$ (resp. $\tau_i^{\downarrow}$) is an abbreviation of $\tau_i^{\uparrow}(\sigma(\mathcal{L}))$ (resp. $\tau_i^{\downarrow}(\sigma(\mathcal{L}))$).

Assuming x_i and y_i are the coordinates of p_i, we define the distance of two points p_i and p_j as $\Delta(p_i, p_j) = \max(|x_i - x_j|, |y_i - y_j|)$. Moreover, $\eta(p_i)$ is denoted as the minimum distance between p_i and all points below p_i. If there is no point below p_i then $\eta(p_i)$ is $+\infty$. Obviously, the value of $\eta(p_i)$ is also the maximum length of ℓ_i when $\tau_i^{\uparrow}$ labels p_i. It is easy to verify that, if a valid labeling of length $\eta(p_i)$ for all points below p_i exists, then ℓ_i may intersect at most four other labels, where their corresponding points are not farther than $2\eta(p_i)$ from p_i.

We introduce a weighted and directed *adjacency graph* $\mathcal{G}$ to look for possible label intersections when assigning a label to a given point. Precisely, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $v_i \in \mathcal{V}$ corresponds to $p_i \in \mathcal{P}$ $(1 \leq i \leq n)$ and a directed edge $(v_i, v_j) \in \mathcal{E}$ exists if we have, (a) p_i lies above p_j (i.e., $y_i > y_j$), and (b) $\Delta(p_i, p_j) \leq 2\eta(p_i)$. Given a labeling for all points below p_i, $\tau_i^{\uparrow}$ (or equally $\tau_i^{\downarrow}$) may intersect at most four other labels, so we only store four edges starting at v_i in $\mathcal{E}$ that corresponds to four nearest neighbors of p_i. Clearly, $\mathcal{G}$ has $O(n)$ vertices and edges and can be constructed in $O(n \lg n)$ time.

Our algorithm assigns labels to all points, one at a time, and generates an optimal labeling after each label assignment. Label candidates of each new point, may intersect previous labels. To make room for the new label, we can flip a series of labels, shrink all labels to a smaller length or both. Maintaining the 2PM property, all points must remain at the middle of one of horizontal edges of their labels after doing each shrink or flip operation. A flipped version of ℓ_i is denoted by $f(\ell_i)$ and ℓ_i resized to length α is denoted by $r(\ell_i, \alpha)$.

We present another special weighed and directed graph called *conflict graph* to represent all possible flip and resize operations of a given labeling [11, 12]. For clarity, we define the conflict graph precisely and briefly in the following.

For a given $\mathcal{L}$, the conflict graph $\mathcal{H} = (\mathcal{W}, \mathcal{F} \cup \mathcal{B})$ is a weighted and directed graph where each $w_i \in \mathcal{W}$ corresponds to $p_i \in \mathcal{P}$ $(1 \leq i \leq n)$. There is a directed edge (w_i, w_j) if $f(\ell_i)$ intersects ℓ_j. Moreover, $(w_i, w_j) \in \mathcal{B}$ if $f(\ell_i)$ also intersects $f(\ell_j)$ and $(w_i, w_j) \in \mathcal{F}$ if $f(\ell_i)$ is disjoint from $f(\ell_j)$. A directed path of edges in $\mathcal{F}$ represents a series of label flips and an edge in $\mathcal{B}$ represents when no more flipping is reasonable. Since all edges in $\mathcal{B}$ are reflective (i.e., if $(w_i, w_j) \in \mathcal{B}$ then $(w_j, w_i) \in \mathcal{B}$), we show and treat edges in $\mathcal{B}$ as undirected edges. An example of a conflict graph is shown in Fig. 1.

The weight of $(w_i, w_j) \in \mathcal{F} \cup \mathcal{B}$, which is the optimal label length resolving the intersection of $f(\ell_i)$ and ℓ_j is denoted by $t_e(w_i, w_j)$ without any further label flips. If no flipping is allowed, the intersection of $f(\ell_i)$ and ℓ_j can be resolved by shrinking both labels to a suitable length. $t_e(w_i, w_j)$ is the maximum value of ρ where $r(f(\ell_i), \rho)$ does not intersect $r(\ell_j, \rho)$.

The vertex weight of w_i in $\mathcal{H}$, denoted by $t_v(w_i)$, is the length of an optimal labeling, if ℓ_i is forced to flip. To define $t_v(w_i)$ precisely, we consider these three cases:

1. If w_i has no outgoing edge, then $t_v(w_i)$ is $\sigma(\mathcal{L})$ since $f(\ell_i)$ has no intersection with other labels.

[1] Recall that in closed-2PM model, two labels with touching edges are intersecting.

2. If w_i has no outgoing edge in $\mathcal{F}$, then flipping ℓ_i causes no more label flips. Hence $t_v(w_i)$ is the minimum edge weight among all edges in $\mathcal{B}$ attached to w_i.
3. Otherwise, we need to consider any label intersection corresponding to an outgoing edges of w_i. We define a function $h(w_i, w_j)$ representing the optimal label length resolving the intersection of $f(\ell_i)$ and ℓ_j. If $(w_i, w_j) \in \mathcal{B}$, $h(w_i, w_j)$ is $t_e(w_i, w_j)$ which is the maximum shrink length to make $f(\ell_i)$ and ℓ_j disjoint. But if $(w_i, w_j) \in \mathcal{F}$, we can either shrink all labels to $t_e(w_i, w_j)$ or flip ℓ_j and generate a labeling of length $t_v(w_j)$. Hence, $h(w_i, w_j)$ for $\mathcal{F}$ edges is $\max(t_e(w_i, w_j), t_v(w_j))$. Finally, $t_v(w_i)$ is the minimum value of h function over all outgoing edges of w_i.

2.2. Properties of 2PM Label Placement

Let $\mathcal{L}$ be an optimal labeling of $\mathcal{P}$ with length $\gamma = \sigma(\mathcal{L})$ and $p_{n+1} = (x_{n+1}, y_{n+1})$ be an unlabeled point that lies above all points in $\mathcal{P}$. Also, let $\mathcal{L}^+$ be an optimal labeling of $\mathcal{P}^+ = \mathcal{P} \cup \{p_{n+1}\}$ with length γ^+. The following lemmas, which are concluded directly from the vertex weight definition of $\mathcal{H}$, form the main ideas of our algorithm.

Lemma 1 *There exists an optimal labeling for $\mathcal{P}^+$ in which $\tau^{\downarrow}_{n+1}(\gamma^+)$ is attached to p_{n+1}.*

Lemma 2 *If all labels intersecting $\tau^{\downarrow}_{n+1}(\gamma)$ have vertices of weight greater than or equal to γ, then there exists an optimal labeling of length γ where $\tau^{\downarrow}_{n+1}(\gamma)$ is attached to p_{n+1}.*

Obviously, if $\tau^{\downarrow}_{n+1}(\gamma)$ intersects ℓ_i where $t_v(w_i) < \gamma$, then $\gamma^+ < \gamma$. To resolve this intersection, we can either shrink all labels to a length $\Delta(p_{n+1}, p_i)$ or flip ℓ_i to generate a labeling of length $t_v(w_i)$. So, the length of the resulting is $h'(p_{n+1}, p_i) = \max(\Delta(p_{n+1}, p_i), t_v(w_i))$. Therefore,

Lemma 3 *If $\tau^{\downarrow}_{n+1}(\gamma)$ intersects a label with vertex of weight less than γ, then γ^+ is the minimum value of h' function over all such intersecting labels.*

The above lemmas give a *bottom-edge labeling scheme* for labeling a set of points from bottom to top. Lemma 1 and 2 give the clue to attach each new label on its bottom edge to its corresponding points and, Lemma 3 reveals the required conditions for flipping down a series of labels. The key property of the above scheme is that no label will be flipped twice, since by flipping up a label (i.e., the second flip of a label), a series of label flips is generated that may introduce some intersections with other labels, that can only be resolved with further label shrinks. So:

Lemma 4 *In the bottom-edge labeling scheme, no label flips more than once.*

Doing a series of label flips to make room for the label of the newly inserted point, invalidates some previously calculated vertex weights in the conflict graph. Lemma 5 and Lemma 6 show that these invalidated vertex weights are not required to be re-calculated.

Lemma 5 *In the bottom-edge labeling scheme, the vertex weights of flipped labels are not required to be updated.*

Proof. According to Lemma 4, a flipped label will not be flipped any further. Hence its vertex weight will never be used during the calculation of h' function in Lemma 3. □

Lemma 6 *Given an optimal labeling $\mathcal{L}$, if there is a directed downward path from w_i to w_j with $\mathcal{F}$ edges, then after flipping ℓ_j, $t_v(w_i)$ requires no update.*

Proof. Considering $\mathcal{L}$, if flipping ℓ_i does not force ℓ_j to be flipped to generate an optimal labeling, then flipping ℓ_j has no effect on $t_v(w_i)$. Otherwise, consider a path π of edges in $\mathcal{F}$ from w_i to w_j. The last edge of π, say (w_k, w_j), will be removed from $\mathcal{F}$ after flipping ℓ_j. Hence, the value of $t_v(w_k)$ may increase since a constraint in calculation of $t_v(w_k)$ is removed. The vertex weights of all vertices on π from w_k to w_i may also increase, if $t_v(w_k)$ is increased. Denote the new weight of $t_v(w_i)$ by $t'_v(w_i)$. If $t_v(w_i) = t'_v(w_i)$ then the flipping of ℓ_j has no effect on $t_v(w_i)$. Otherwise, it is easy to see that $t_v(w_i) = t_v(w_j)$. So, we already have an optimal labeling of length at most $t_v(w_j)$ after flipping ℓ_j. This way, updating $t_v(w_i)$ to a value greater than the current label length is not necessary. □

2.3. The Closed-2PM Label Placement Algorithm

The basic idea of the bottom-edge labeling scheme is to stop a horizontal sweep line at y-coordinate of each point in the ascending order and label that point with the current optimal label length (Lemma 1 and Lemma 2). This may cause other intersecting labels to flip down which may occur at most once per label (Lemma 4), or all labels to

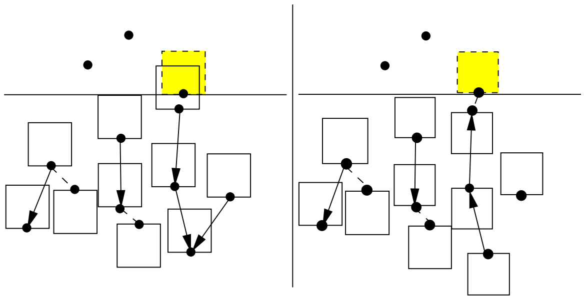

Figure 2: (a) Inital placement of the new label. (b) Labeling after resolving the intersection.

shrink (Lemma 3) without the need to have other vertex weights updated (Lemma 5 and Lemma 6).

The inputs to the closed-2PM label placement algorithm are a sequence of n points sorted according to their y-coordinates in ascending order, and the adjacency graph $\mathcal{G}$ of $\mathcal{P}$. Without loss of generality, we assume $y_1 \leq y_2 \leq \ldots \leq y_n$. Initially, an optimal labeling of the first three points (i.e., p_1, p_2 and p_3) is calculated. Assuming that $\mathcal{L}_{i-1} = \{\ell_1, \ell_2, \ldots, \ell_{i-1}\}$ is an optimal labeling for $\mathcal{P}_{i-1} = \{p_1, p_2, \ldots, p_{i-1}\}$, a sweep line stops at each y_i $(i > 3)$, and generates an optimal labeling $\mathcal{L}_i$ for $\mathcal{P}_i$ as follows:

1. **Let** $\gamma_i = \min\{\{\gamma_{i-1}\} \cup \{h'(w_i, w_j) | \tau_i^{\downarrow}(\gamma_{i-1}) \cap \ell_j \neq \emptyset\}\}$ (Lemma 1, 3).
2. **If** $\gamma_i = \gamma_{i-1}$ **then** p_i can be labeled by $\tau_i^{\downarrow}(\gamma_{i-1})$ (Lemma 2).
3. **Else** generate a labeling of length γ_i [12].
4. Add p_i to $\mathcal{H}_{i-1}$ and build $\mathcal{H}_i$.

In the first step, building the set of labels intersecting $\tau_i^{\downarrow}(\gamma_{i-1})$ can be done in $O(1)$ time by visiting the edges in $\mathcal{G}$ attached to v_i. Other steps are done in $O(1)$ amortized time since no label flips more than once (Lemma 4). Hence, the algorithm needs $O(n)$ time to generate an optimal labeling. Fig. 2 shows the algorithm in action.

The following theorem states the main result of this section.

Theorem 1 *For a given sequence of points sorted by their y-coordinates and the adjacency graph $\mathcal{G}$ of $\mathcal{P}$, an optimal labeling $\mathcal{L}$ can be found in $O(n)$ time.*

3. INCREMENTAL ALGORITHM

Given $\mathcal{P}$, we build the adjacency graph $\mathcal{G}$ of $\mathcal{P}$ in $O(n \lg n)$ and build an optimal labeling of $\mathcal{P}$ in $O(n)$ with the closed-2PM label placement algorithm.

Suppose a new point p_{n+1} is inserted in P. This point should be optimally labeled and the previously optimal labeling $\mathcal{L}$ should also be updated to $\mathcal{L}^+$. It is easy to see that a new vertex can be inserted in both $\mathcal{G}$ and $\mathcal{H}$ graphs and updated graphs can be constructed in $O(\lg n)$ time after each insertion.

We now show that in $O(\lg n + k)$ time we can verify the existence of a series of k label flips that makes room for the new label of the same length. This can be done by checking each label candidate (i.e., $\tau_{n+1}^{\uparrow}$ and $\tau_{n+1}^{\downarrow}$) of p_{n+1}. To check a candidate, say $\tau_{n+1}^{\uparrow}$, we should find and flip all labels intersecting with $\tau_{n+1}^{\uparrow}$. These label flips may cause intersections with other labels and force them to flip and this may cause domino effect. If all these flips are possible, then we have found a series of label flips that makes room for $\tau_{n+1}^{\uparrow}$. This can be done in time proportional to the number of flipped labels. The same procedure can be used to check $\tau_{n+1}^{\downarrow}$.

Since there are two candidate labels for each new point, we need to check both of them that may need $O(n)$ total time in the worse case. So, we need to concurrently check both candidates with a BFS-like algorithm such that the number of processed labels for each candidate differ in at most one (i.e., process one label from the BFS queue of each candidate, alternatively). Hence, we can stop the checking procedure(s) when one of them finds a series of label flips. Therefore, we can generate a labeling of the same length in time $O(k)$ where k is the number of flipped labels.

If no such labeling exists, (i.e., $\sigma(\mathcal{L}^+) < \sigma(\mathcal{L})$), a new optimal labeling can be generated in $O(n)$ time with the 2PM label placement algorithm discussed above. Therefore:

Theorem 2 *Given an optimal labeling in closed-2PM model and a new unlabeled point, the algorithm generates an optimal labeling of the same length in $O(\lg n + k)$ time where k is the number of label flips, if such a labeling exists. Otherwise a new optimal labeling can be found in $O(n)$ time.*

4. CONCLUSION

In this paper, we considered an incremental optimal label placement in a closed-2PM model where labels are non-intersecting axis-parallel square-shaped of maximum length each attached to its corresponding point on one of its horizontal edges. Given an initial point set, we presented an algorithm that efficiently generates a new optimal labeling for all points which is capable of optimally label a series of new points, one at a time. Using $O(n)$ space, our algorithm generates each new optimal labeling in $O(\lg n + k)$ time where k is the number of required changes to the original labeling, if there is no need to shrink labels, or in $O(n)$ time otherwise.

References

[1] 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.

[2] Rob Duncan, Jianbo Qian, Antoine Vigneron, and Binhai Zhu. Polynomial time algorithms for three-label point labeling. *Theoretical Computer Science*, 296(1):75–87, 2003.

[3] Michael Formann. *Algorithms for Geometric Packing and Scaling Problems*. PhD thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, 1992.

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

[5] 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.

[6] Konstantinos G. Kakoulis and Ionnis G. Tollis. A unified approach to automatic label placement. *International Journal of Computational Geometry and Applications*, 13(1):23–59, 2003.

[7] Gunnar W. Klau and Petra Mutzel. Optimal labelling of point features in the slider model. In D.-Z. Du, P. Eades, V. Estivill-Castro, X. Lin, and A. Sharma, editors, *Proc. 6th Annual International Computing and Combinatorics Conf. (COCOON'00)*, volume 1858 of *Lecture Notes in Computer Science*, pages 340–350, Sydney, 26–28 July 2000. Springer-Verlag.

[8] Gunnar W. Klau and Petra Mutzel. Optimal labeling of point features in rectangular labeling models. *Mathematical Programming (Series B)*, pages 435–458, 2003.

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

[10] Zhongping Qin, Alexander Wolff, Yinfeng Xu, and Binhai Zhu. New algorithms for two-label point labeling. In Mike Paterson, editor, *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.

[11] Farshad Rostamabadi and Mohammad Ghodsi. A fast algorithm for updating a labeling to avoid a moving point. In *Proceeding of the 16th Canadian Conference on Computational Geometry (CCCG'04)*, pages 204–208, 2004.

[12] Farshad Rostamabadi and Mohammad Ghodsi. Label updating in 2pm to avoid a moving point. In *The 21st European Workshop on Computational Geometry (EWCG'05)*, 2005.

[13] Sasanka Roy, Partha P. Goswami, Sandip Das, and Subhas C. Nandy. Optimal algorithm for a special point-labeling problem. In *Proc. 8th Scandinavian Workshop on Algorithm Theory (SWAT'02)*, Lecture Notes in Computer Science, Turku, 3–5 July 2002. Springer-Verlag. To appear.

[14] Michael J. Spriggs and J. Mark Keil. A new bound for map labeling with uniform circle pairs. *Information Processing Letters*, 81(1):47–53, 2002.

[15] Tycho Strijk and Marc van Kreveld. Practical extensions of point labeling in the slider model. Technical Report UU-CS-2000-08, Department of Computer Science, Utrecht University, 2000.

[16] Tycho Strijk and Alexander Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 11(2):181–195, April 2001.

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

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

[19] Missae Yamamoto and Luiz Antonio Nogueira Lorena. A constructive genetic approach to point-feature cartographic label placement. In *Proc. Fifth Metaheuristics International Conference (MIC'03)*, pages 84–1–7, Kyoto, Japan, 25–28 August 2003.