Computers and Electrical Engineering xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Incremental labeling in closed-2PM model ☆

Farshad Rostamabadi *, Mohammad Ghodsi

Department of Computer Engineering, Sharif University of Technology, P.O. Box 11365-9415, Tehran, Iran
School of Computer Science, Institute for Studies in Fundamental Sciences (IPM), P.O. Box: 19395-5746, Tehran, Iran

ARTICLE INFO

Keywords:
Computational geometry
Map labeling
Label updating

ABSTRACT

We consider an incremental optimal label placement in a closed-2PM map containing points each attached with a label. Labels are assumed to be axis-parallel square-shaped and have to be pairwise disjoint with maximum possible length each attached to its corresponding point on one of its horizontal edges. Such a labeling is denoted as optimal labeling. Our goal is to efficiently generate a new optimal labeling for all points after each new point being inserted in the map. 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 the labels and flip some of them. The algorithm uses $O(n)$ space in both cases. This is a new result on this problem.

1. Introduction

Label placement is a well-known problem in the field of map generation, where, in its general form, is to attach some textual or graphical information (referred to as labels) to certain features (like points or lines) of a given map. Examples of a map are geographical maps, electrical circuit maps, power-line maps, and so on. Automated point-label placement has received good attention, where features are points and labels are either squares or rectangles. In a *valid* labeling, labels should be pairwise disjoint, and each label should be attached to its feature point [10]. Various models for labeling points have been discussed, e.g. in terms of the label shape. There are algorithms for labeling points with squares [3,13], unit-height rectangles [1,12], arbitrary rectangles [1], or circles [20,18,23].

Most versions of map labeling problem are NP-Hard and many researchers have provided approximation algorithms for different variations of the problem. However, some polynomial time solvable versions of the map labeling problem have been considered [17]. In the "Elastic labeling" model, each point receives a rectangular label where its area is fixed globally, and the goal is to find the maximum value of area size [7], where all points can be labeled. In this problem, if all points are positioned on the positive part of x and y axis and labels are placed in the first quarter of Euclidean 2D-space, then the optimal labeling can be found in polynomial time using a dynamic programming algorithm [8,6].

Another interesting labeling model is the line labeling considered in [11,19], where rectilinear lines should receive rectangular labels parallel to their corresponding line. Using the fact that each line has only three label candidates in the proposed model, the solution can be found with a reduction to 2SAT problem and can efficiently be solved in polynomial time.

In this paper, we consider labeling of a set of points in the *closed-2PM model*. In this model, 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

☆ This research was in part supported by a Grant from IPM (N. CS2386-2-01.) A preliminary version of this paper was presented in the 11th CSI Computer Conference (Iran, January 2006).

* Corresponding author.
E-mail addresses: farshad@mehr.sharif.edu (F. Rostamabadi), ghodsi@sharif.edu (M. Ghodsi).

0045-7906/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2008.04.012

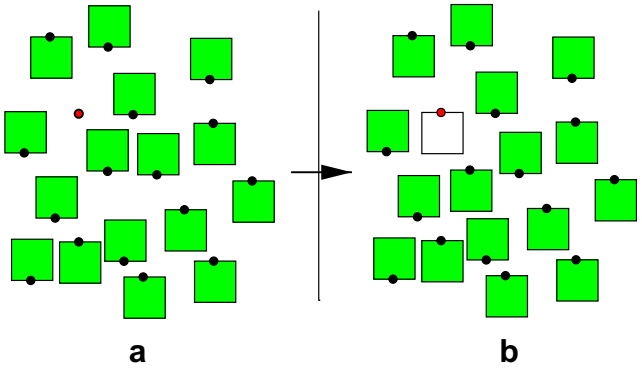

Fig. 1. (a) Initial given labeling and a newly inserted point (shown in red) and; (b) the updated labeling. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2PM stands for this). 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 a similar model, called 4PM, each point is attached to the middle of either its horizontal or vertical edges of its label. 4PM labeling is NP-Hard and is first considered in [9,5] to model and solve a problem in meta-font. To convert 4PM problem to a non NP-Hard problem, a labeling direction for each point, which is known in advance, is introduced [16]. Label direction of each point restricts its label to be attached to it, via just either horizontal or vertical edges. This yield another polynomial time solvable labeling called as r4PM (letter r stands for restricted). In r4PM, there are two label candidates per each point, hence, n points can be labeled optimally, in r4PM model, in $O(n \lg n)$ time using a reduction to 2SAT problem (like the reduction given in [11]). In [14,16] we introduced the r4PM model and provided a new approach to the this relabeling problem: How a labeling can be maintained when an obstacle appears in the map? We considered the case of a moving obstacle in a labeled map where at each given time, an optimal obstacle-avoiding labeling is constructed. With an $O(n^2)$ preprocessing time, we presented an algorithm that obtains such a labeling in $O(\lg n + k)$ response-time, where k is the number of changes needed to avoid the obstacle. Later in [15,16],we considered the same problem in 2PM model and reduced the preprocessing time from $O(n^2)$ to $O(n \lg n)$. The latter algorithm was much easier to implement and required simpler data structures than the former one.

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 after each new point insertion. 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 the label lengths [22,4], 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 [2]. So, insertion of n points in an empty point set and generation of an optimal labeling after each insertion needs $O(n^2 \lg n)$ time with this strategy.

We present two incremental labeling algorithms, where, one of them is used inside another. Given a set of labeled points, sometimes, it is possible to label a new point optimally by flipping current labels without shrinking other labels, and sometimes it is not. An example of the closed-2PM labeling problem is shown in Fig. 1 where a point is inserted into a given labeling. In the latter case, we proposed another incremental sweep line based algorithm to generate an optimal labeling for all points including the new one.

Given a set of n points $\mathscr{P}$, 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. 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 Section 2.2), 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, introduced in Section 2.3. 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.

2. Closed-2PM label placement problem

Given a set of n points $\mathscr{P} = \{p_1, p_2, \ldots, p_n\}$, we define a *valid labeling* of $\mathscr{P}$ as a placement of equal-length axis-parallel square-shaped labels $\mathscr{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ 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 $\mathscr{L}$, denoted by $\sigma(\mathscr{L})$, is the (same) length for all labels in $\mathscr{L}$. A valid labeling with the max-

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

Please cite this article in press as: Rostamabadi F, Ghodsi M, Incremental labeling in closed-2PM model, Comput Electr Eng (2008), doi:10.1016/j.compeleceng.2008.04.012

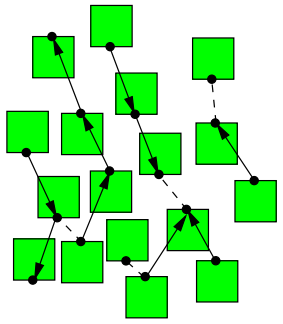

Fig. 2. A given labeled map and the conflict graph: $\mathscr{B}$ edges (dashed), directed edges (solid).

imum value of $\sigma(\mathscr{L})$ is referred to as an optimal labeling. Given $\mathscr{P}$, the problem of finding an optimal labeling is referred to as the *closed-2PM label placement problem*.

Lemma 1. *The time needed to decide for existence of a labeling of a given length in Closed-2PM (and also 2PM) model for $\mathscr{P}$ is $\Omega(n \lg n)$.*

Proof. Proof is by a reduction from the ϵ-closeness problem, which is known to be to $\Omega(n \lg n)$ [21], to the point labeling in Closed-2PM model 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 $\mid \phi_i - \phi_j \mid < \epsilon$ exist if and only if no valid labeling of length ϵ for a set of n points $\mathscr{P} = \{(\phi_i, 0) \mid 1 \leqslant i \leqslant n\}$ exists. □

2.1. Preliminaries and data structures

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 $\tau_i^{\uparrow}(\sigma(\mathscr{L}))$ (resp. $\tau_i^{\downarrow}(\sigma(\mathscr{L}))$).

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 $\Delta(p_i, p_j) = \max(|x_i - x_j|, |y_i - y_j|)$, which is the L_∞ norm. 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 $+\infty$. More formally, $\eta(p_i) = \min(\{\Delta(p_i, p_j) | y_i \geqslant y_j\} \cup \{+\infty\})$. 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* $\mathscr{G}$ to look for possible label intersections when assigning a label to a given point. Precisely, $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $v_i \in \mathscr{V}$ corresponds to $p_i \in \mathscr{P}$ $(1 \leqslant i \leqslant n)$ and a directed edge $(v_i, v_j) \in \mathscr{E}$ exists if we have, (a) p_i lies above p_j (i.e., $y_i > y_j$), and (b) $\Delta(p_i, p_j) \leqslant 2\eta(p_i)$. 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 $\mathscr{E}$ that corresponds to four nearest neighbors of p_i. Since each vertex of $\mathscr{G}$ has no more than 4 outgoing edges, there are $O(n)$ vertices. Moreover, there are $O(n)$ edges in $\mathscr{G}$ that can be constructed in $O(n \lg n)$ time [16].

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 [14–16]. For clarity, we define the conflict graph precisely and briefly in the following.

For a given $\mathscr{L}$, the conflict graph $\mathscr{H} = (\mathscr{P}, \mathscr{F} \cup \mathscr{B})$ is a weighted and directed graph. There is a directed edge $(p_i, p_j) \in \mathscr{F}$ if $f(\ell_i)$ intersects ℓ_j. Moreover, $(p_i, p_j) \in \mathscr{B}$ if $f(\ell_i)$ also intersects $f(\ell_j)$ and $(p_i, p_j) \in \mathscr{F}$ if $f(\ell_i)$ is disjoint from $f(\ell_j)$. A directed path of edges in $\mathscr{F}$ represents a series of label flips and an edge in $\mathscr{B}$ represents when no more flipping is reasonable. Since all edges in $\mathscr{B}$ are reflective (i.e., if $(p_i, p_j) \in \mathscr{B}$ then $(p_j, p_i) \in \mathscr{B}$), we show and treat edges in $\mathscr{B}$ as undirected edges. An example of a conflict graph is shown in Fig. 2.

The weight of $(p_i, p_j) \in \mathscr{F} \cup \mathscr{B}$, which is the optimal label length resolving the intersection of $f(\ell_i)$ and ℓ_j is denoted by $w_e(p_i, p_j)$ 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. $^e(p_i, p_j)$ is the maximum value of ρ where $r(f(\ell_i), \rho)$ does not intersect $r(\ell_j, \rho)$.

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

Please cite this article in press as: Rostamabadi F, Ghodsi M, Incremental labeling in closed-2PM model, Comput Electr Eng (2008), doi:10.1016/j.compeleceng.2008.04.012

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

Obviously, a simple bottom-up algorithm can calculate all weight values of all vertices of $\mathscr{H}$ in linear time.

2.2. Properties of 2PM label placement

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

Lemma 2. *There exists an optimal labeling for $\mathscr{P}^+$ in which $\tau^{\downarrow}_{n+1}(\gamma^+)$ is attached to p_{n+1}.*

Proof. The proof is by contradiction. Assume that there is no optimal labeling where ℓ_{n+1} placed on top of p_{n+1}. Let $\mathscr{L}^+$ be an optimal labeling of P^+ and flip ℓ_{n+1}. Since p_{n+1} is the topmost point in $\mathscr{P}^+$ and ℓ_{n+1} has no intersection with labels in $\mathscr{L}^+$ then $f(\ell_{n+1})$ cannot have intersection with labels $\ell_1 \dots \ell_n$. So, there is a valid labeling where ℓ_{n+1} is on top of p_{n+1}. □

Lemma 3. *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}.*

Proof. The proof is by construction. By definition, vertex weight is the optimal label length when the corresponding label flips. So, if all intersecting labels with $\tau^{\downarrow}_{n+1}(\gamma)$ are flipped, a labeling of length γ is generated with no intersecting label with $\tau^{\downarrow}_{n+1}(\gamma)$. Hence p_{n+1} can be labeled with $\tau^{\downarrow}_{n+1}(\gamma)$ and an optimal labeling of length γ is available. □

Obviously, if $\tau^{\downarrow}_{n+1}(\gamma)$ intersects ℓ_i where $w_v(p_i) < \gamma$, 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 $w_v(p_i)$. So, the length of the resulting is $h'(p_{n+1}, p_i) = \max(\Delta(p_{n+1}, p_i), w_v(p_i))$. Therefore,

Lemma 4. *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. Lemmas 2 and 3 give the clue to attach each new label on its bottom edge to its corresponding points and, Lemma 4 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 5. *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. Lemmas 6 and 7 show that these invalidated vertex weights are not required to be recalculated.

Lemma 6. *In the bottom-edge labeling scheme, the vertex weights of flipped labels are not required to be updated.*

Proof. According to Lemma 5, 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 4. □

Lemma 7. *Given an optimal labeling $\mathscr{L}$, if there is a directed downward path from p_i to p_j with $\mathscr{F}$ edges, then after flipping ℓ_j, $w_v(p_i)$ requires no update.*

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

Please cite this article in press as: Rostamabadi F, Ghodsi M, Incremental labeling in closed-2PM model, Comput Electr Eng (2008), doi:10.1016/j.compeleceng.2008.04.012

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 (Lemmas 2 and 3). This may cause other intersecting labels to flip down which may occur at most once per label (Lemma 5), or all labels to shrink (Lemma 4) without the need to have other vertex weights updated (Lemmas 6 and 7).

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 $\mathscr{G}$ of $\mathscr{P}$. Without loss of generality, we assume $y_1 \leqslant y_2 \leqslant \cdots \leqslant y_n$. Initially, an optimal labeling of the first three points (i.e., p_1, p_2 and p_3) is calculated. Assuming that $\mathscr{L}_{i-1} = \{\ell_1, \ell_2, \ldots, \ell_{i-1}\}$ is an optimal labeling for $\mathscr{P}_{i-1} = \{p_1, p_2, \ldots, p_{i-1}\}$, a sweep line stops at each y_i $(i > 3)$, and generates an optimal labeling $\mathscr{L}_i$ for $\mathscr{P}_i$ as follows:

(1) **Let** $\gamma_i = \min\{\{\gamma_{i-1}\} \cup \{h'(p_i, p_j) \mid \tau_i^{\downarrow}(\gamma_{i-1}) \cap \ell_j \neq \emptyset\}\}$ (Lemma 2 and 4).
(2) **If** $\gamma_i = \gamma_{i-1}$ **then** p_i can be labeled by $\tau_i^{\downarrow}(\gamma_{i-1})$ (Lemma 3).
(3) **Else** generate a labeling of length γ_i [16].
(4) Add p_i to $\mathscr{H}_{i-1}$ and build $\mathscr{H}_i$.

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 $\mathscr{G}$ attached to v_i. Other steps are done in $O(1)$ amortized time since no label flips more than once (Lemma 5). Hence, the algorithm needs $O(n)$ time to generate an optimal labeling. Fig. 3 shows the algorithm in action.

The following theorem states the main result of this section.

Theorem 8. *For a given sequence of points sorted by their y-coordinates and the adjacency graph $\mathscr{G}$ of $\mathscr{P}$, an optimal labeling $\mathscr{L}$ can be found in $O(n)$ time.*

3. Incremental labeling

Given $\mathscr{P}$, we build the adjacency graph $\mathscr{G}$ of $\mathscr{P}$ in $O(n \lg n)$ and build an optimal labeling of $\mathscr{P}$ 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 $\mathscr{L}$ should also be updated to $\mathscr{L}^+$. It is easy to see that a new vertex can be inserted in both $\mathscr{G}$ and $\mathscr{H}$ 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., $\sigma(\mathscr{L}^+) < \sigma(\mathscr{L})$), a new optimal labeling can be generated in $O(n)$ time with the 2PM label placement algorithm discussed above. Therefore:

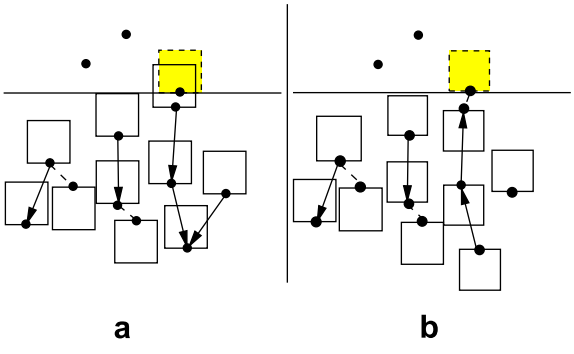

Fig. 3. (a) Inital placement of the new label; and (b) labeling after resolving the intersection.

Please cite this article in press as: Rostamabadi F, Ghodsi M, Incremental labeling in closed-2PM model, Comput Electr Eng (2008), doi:10.1016/j.compeleceng.2008.04.012

Theorem 9. *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 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. We showed that finding the optimal label length in 2PM model is in $\Omega(n \lg n)$. Moreover, 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 flips to the original labeling, if there is no need to shrink labels, or in $O(n)$ time otherwise.

References

[1] Agarwal Pankaj K, Kreveld Marc van, Suri Subhash. Label placement by maximum independent set in rectangles. In: Proceedings of the 9th Canadian conference on computational geometry (CCCG'97); 11–14 August 1997. p. 233–8.
[2] Doddi Srinivas, Marathe Madhav V, Mirzaian Andy, Moret Bernard ME, Zhu Binhai. Map labeling and its generalizations. In: Proceedings of the 8th ACM-SIAM symposium on discrete algorithms (SODA'97); 4–7 January 1997. p. 148–57.
[3] Duncan Rob, Qian Jianbo, Vigneron Antoine, Zhu Binhai. Polynomial time algorithms for three-label point-labeling. Theor Comput Sci 2003;296(1):75–87.
[4] Michael Formann. Algorithms for geometric packing and scaling problems. PhD thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin; 1992.
[5] Michael Formann, Wagner Frank. An efficient solution to Knuth's METAFONT labeling problem. Fachbereich Informatik, Freie Universität Berlin; 1993.
[6] Iturriaga Claudia. Map labeling problems. PhD thesis, University of Waterloo; 1999.
[7] Iturriaga Claudia, Lubiw Anna. Elastic labels on the perimeter of a rectangle. In: Whitesides Sue H, editor. Proceedings of the symposium on graph drawing (GD'98). Lecture notes in computer science, vol. 1547. Springer-Verlag; 1998. p. 452–3.
[8] Iturriaga Claudia, Lubiw Anna. Elastic labels around the perimeter of a map. In: Proceedings of the 8th international workshop on algorithms and data structures (WADS'99). Lecture notes in computer science, vol. 1663. Springer-Verlag; 1999. p. 306–17.
[9] Knuth Donald E, Raghunathan Arvind. The problem of compatible representatives. SIAM J Discr Math 1992;5(3):422–7.
[10] Marks Joe, Shieber Stuart. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard CS; 1991.
[11] Poon Chung Keung, Zhu Binhai, Chin Francis. A polynomial time solution for labeling a rectilinear map. Informat Process Lett 1998;65(4):201–7.
[12] Poon Sheung-Hung, Shin Chan-Su, Strijk Tycho, Uno Takeaki, Wolff Alexander. Labeling points with weights. Algorithmica 2003;38(2):341–62.
[13] Qin Zhongping, Wolff Alexander, Xu Yinfeng, Zhu Binhai. New algorithms for two-label point-labeling. In: Paterson Mike, editor. Proceeings of the 8th annual European symposium on algorithms (ESA'00). Lecture notes in computer science, vol. 1879. Springer-Verlag; 2000. p. 368–79.
[14] Rostamabadi Farshad, Ghodsi Mohammad. A fast algorithm for updating a labeling to avoid a moving point. In: Proceedings of the 16th Canadian conference on computational geometry (CCCG'04); 2004. p. 204–8.
[15] Rostamabadi Farshad, Ghodsi Mohammad. An efficient algorithm for label updating in 2PM model to avoid a moving object. In: Proceedings of the 21st European workshop on computational geometry (EWCG'05); 9–11 March 2005. p. 131–4.
[16] Rostamabadi Farshad, Ghodsi Mohammad. Label updating to avoid point-shaped obstacles in fixed model. Theor Comput Sci 2006;369(1–3): 197–210.
[17] Roy Sasanka, Goswami Partha P, Das Sandip, Nandy Subhas C. Optimal algorithm for a special point-labeling problem. In: Penttonen M, Meineche Schmidt E, editors. Proceedings of the 8th Scandinavian workshop on algorithm theory (SWAT'02). Lecture notes in computer science, vol. 2368. Springer-Verlag; 2002. p. 110–20.
[18] Spriggs Michael J, Mark Keil J. A new bound for map labeling with uniform circle pairs. Informat Process Lett 2002;81(1):47–53.
[19] Strijk Tycho, van Kreveld Marc. Labeling a rectilinear map more efficiently. Informat Process Lett 1999;69(1):25–30.
[20] Strijk Tycho, Wolff Alexander. Labeling points with circles. Int J Comput Geomet Appl 2001;11(2):181–95.
[21] Wagner Frank. Approximate map labeling is in $\Omega(n \log n)$. Technical report B 93-18, Fachbereich Mathematik und Informatik, Freie Universität Berlin; December 1993.
[22] Wagner Frank, Wolff Alexander. A practical map labeling algorithm. Comput Geomet: Theor Appl 1997;7:387–404.
[23] Wolff Alexander, Thon Michael, Xu Yinfeng. A simple factor-2/3 approximation algorithm for two-circle point-labeling. Int J Comput Geomet Appl 2002;12(4):269–81.

Farshad Rostamabadi received his BS, MS, and Ph.D degrees all from Computer Enginnering Department of Sharif University of Technology. He defended his Ph.D thesis on 2-PM 4-PM map labeling in March 2007, He is now the CEO of ParsEBiz Company in Tehran, Iran. His main research interests are computational geometry, digital map problems and geometric algorithms.

Please cite this article in press as: Rostamabadi F, Ghodsi M, Incremental labeling in closed-2PM model, Comput Electr Eng (2008), doi:10.1016/j.compeleceng.2008.04.012

Mohammad Ghodsi received his BS in EE from Sharif University of Technology (SUT) in Iran in 1975, MS in EECS from UC Berkeley in 1978, and Ph.D in computer science from PSU, in 1989. He has been a faculty member of SUT since 1979. Presently, he is a full Professor in Computer Engineering Department of SUT, Tehran, Iran. His main research interests include computational geometry, and design of efficient algorithms.

Please cite this article in press as: Rostamabadi F, Ghodsi M, Incremental labeling in closed-2PM model, Comput Electr Eng (2008), doi:10.1016/j.compeleceng.2008.04.012