Unit Height k-Position Map Labeling

F. Rostamabadi* M. Ghodsi†

Computer Engineering Department
Sharif University of Technology
Tehran, Iran

Abstract

We explore a new variation of k-Position Map Labeling problem, first introduced by Doddi *et. al.* [DMM00] which is: Given a set of points in the plain, a set of up to k allowable positions for each point and, a set of feasible labels (or labeling models) for each position, choose the maximum number of non-intersecting labels so that no point receive more than one label. We define and solve this problem with unit height rectangular labels, in 1D and 2D cases, and also in fixed and slider models.

We present two simple-to-implement 3-approximation algorithms for this problem in both fixed and slider models in 2D case and for unbounded k. We also show that even the problem in 1D is NP-Complete. We then present two 2-approximation algorithms for solving the problem in 1D case.

1 Introduction

Automated label placement is an important problem for map generation in geographical information systems (GIS). The problem is to attach one or more labels (regularly a text) to a point, a line, a curve, or a region in a given map. The point feature label placement has received good attention within computational geometers. Two basic requirements of such labeling are [MS91]: (1) The selected labels should be pair wise disjoint, and (2) Each label should be close enough to the corresponding feature to be identified as such. Other variations of this problem let the features receive more than one labels [KT98, ZQ01], or use specific shapes as labels [DMM+97, vKSW99, SW01]. There are also two labeling models, fixed and slider model. The former is when fixed positions are given as possible label positions [MS91] and the latter is for the cases where the labels can be placed at any position while touching the feature [SvK00, KM00].

Many different approaches have been proposed to solve this problem, including zero-one integer programming [Zor86], approximation algorithms [FW91, DMM+97, ZP00], expert systems [DF89], simulated annealing [ZP00] and force driven algorithms [Hir82].

Doddi *et. al.* [DMM00] introduced the problem of label size maximization k-Position Map Labeling (KPML). In this problem, a set of points in the plain is given and, for each point, we have a set of up to k allowable positions. The problem is to place uniform and non-intersecting labels of maximum size at each point in one of the allowable positions. They have focused on circular labels and have proposed a 3.6-approximation algorithm for it.

Our problem definition is different from that in [DMM00]; we are concerned with selecting the maximum number of labels from a set of feasible labels given for each position. We use nKPML to denote this

*`frostam@linux.sharif.edu`

†`ghodsi@sharif.edu`

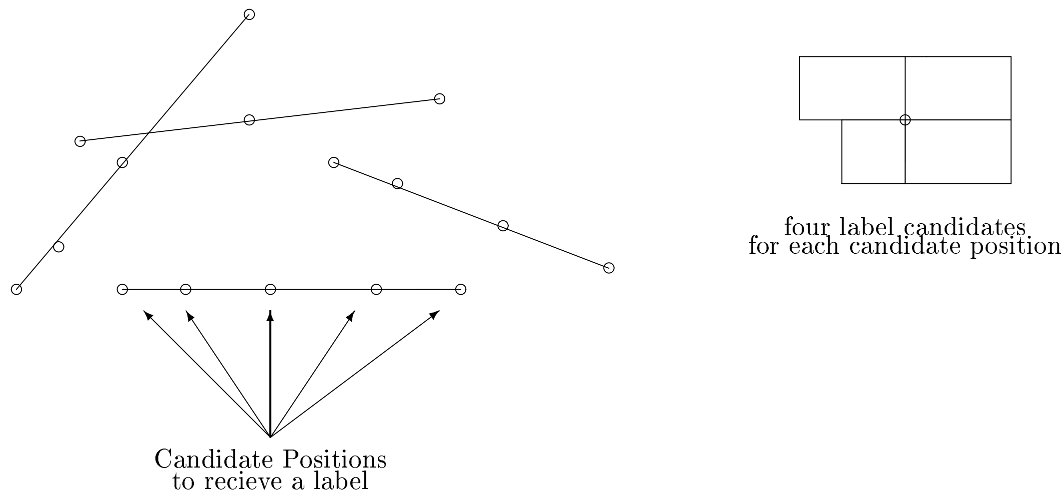

Figure 1: A simple usage of nKPML in line labeling

new problem. More precisely, nKPML is defined as follows: Given a set of points in the plain and a set of up to k allowable positions for each point, and a set of feasible label placements (or a labeling model) for each position, choose the maximum number of non-intersecting labels such that no point receive more than one label.

We focus on nKPML problem with unit height rectangular labels and consider both fixed and slider models. In the fixed model, at each position a finite set of feasible label placements is allowed (see Figure 1).

We will show that this problem is NP-Complete even in 1D case. We propose two different 2-approximation algorithms for both fixed and slider model in 1D case. We then will generalize this solution to obtain 3-approximation algorithms for both models in 2D case.

2 Problem Definition of 2D-nKMPL in Fixed Model

Let $S = \{S_1, S_2, \ldots, S_n\}$ be a set of points where each $S_i = \{s_{i_1}, s_{i_2}, \ldots, s_{i_{n_i}}\}$ shows all possible positions of point S_i, and let L_{ij} be all unit-height labels for object s_{i_j} of S_i. It is also assumed that

$$\forall_{1 \leq i \leq n} \cap_{1 \leq j \leq n_i} L_{i_j} \neq \emptyset$$

(i.e., each pair of labels of a given site must have non-empty intersection.)

The problem is to find the maximum number of labels such that following conditions hold:

- At most one label is chosen from the set $\cup_{1 \leq j \leq n_i} L_{i,j}$ (for each i).
- All chosen labels should be pair wise disjoint.

This problem is a generalization of the standard point label placement problem, since any instance of label placement problem is in fact an instance of this problem by setting $S_i = \{p_i\}$ for each i, where

$P = \{p_1, p_2, \ldots, p_n\}$ is the set of point locations. Therefore, this problem is also NP-Complete [MS91] and can not be approximated with a factor better than 2 in time $\Omega(n \log n)$ [FW91, Wag94].

2.1 1D-nKPML in Fixed Model

We define 1D-nKPML in fixed model as follows. A set $G = g_1, g_2, \ldots, g_m$ of interval group $g_i = I_{i1}, I_{i2}, \ldots, I_{in_i}$ is given, where each I_{ij} is an interval on x-axis. The problem is to choose at most one interval from every interval group such that all chosen intervals are pair wise disjoint.

The general idea of defining the problem in 1D is to solve the problem in a very special case where there exists a single horizontal line intersecting all possible label positions.

2.2 1D-nKPML Problem is NP-Complete

It can be shown that there is a polynomial time reduction from standard unit-length square labeling in fixed model to 1D-nKPML problem. The standard unit-length labeling problem is:

> Suppose that the point set P is given and we are asked to find the maximum number of disjoint unit-length squares such that each point is adjacent to exactly one square in just one of its vertices and each square is adjacent to exactly one point.

Given an instance of unit-length labeling problem, we define a set of horizontal lines $H_l : y = l$, for all l that there is a label position intersecting H_l. Now, define intersection of label candidates of point p_i with H_j as intervals I_{ij1} and I_{ij2}, and grouping these two intervals into one group G_{i1} of p_i. Since each point in standard unit-height labeling has four label candidates, it is obvious that exactly two horizontal lines H_j and H_{j+1} (for some j) will intersect with all these four label candidates, and two groups G_{i1} and G_{i2} will be generated per point. We merge the horizontal lines H_l into a single horizontal line (e.x. x-axis) with serializing the intervals from left to right i.e. Put intervals of H_{l_1} somewhere on x-axis, then put intervals of H_{l_2} after them, and so on. Now, we have an instance of 1D-nKPML.

It is known that if we restrict the point coordinates to be integers, the problem still remains in NP-Complete. Using this restriction, it is obvious that a solution to unit-length labeling problem can be constructed from a solution of 1D-nKPML which is known to be NP-Complete and a 2-approximation can be achieved in time $\Omega(n \log n)$ [Wag93].

2.3 2-Approximation Algorithm of 1D-nKPML

A simple greedy algorithm can approximate the 1D-nKPML problem within factor two. The main idea to solve the 2D case, is to repeatedly choose the interval with minimum right end point such that a consistent solution is found.

Algorithm Sort all intervals according to their right end point in ascending order and save them in a sorted list I. As long as the sorted list I is not empty, pick the interval I_{ij} with minimum right end point. If the interval I_{ij} has no intersection with previously selected intervals, and if no interval of the group G_i is not selected before, then select interval I_{ij}, otherwise just remove interval I_{ij} from I. □

It is obvious that the running time of the above algorithm is $O(l \log l)$ where l is total number of intervals. The initial sorting of intervals need no more than $O(l \log l)$ and the while loop runs once per interval. Each iteration of while loop requires $O(1)$ to check if the interval I_{ij} is free and another constant time is required to check if the group G_i is already labelled or not. So, the overall running time of the algorithm is $O(l \log l)$.

Lemma 1 *Given an instance of fixed model 1D-nKPML, a solution of size at least $\lambda/2$ can be computed, where λ is the maximum number of intervals in an optimal solution.*

Proof Let S^*, S be an optimal solution and output of our algorithm accordingly. We show that for every selected interval $I_{ij} \in S$ at most two intervals from the optimal solution, S^*, will be missed.

Let $I_{ij} \in S$ be the interval with the minimum right end point, then the first missing label might be $I_{ij'} \in S^*$ which can not be selected since the I_{ij} is selected (Note that $I_{ij'}$ is the optimum interval selected for group G_i in $S*$). The second missing label might be the label with minimum end point in $I_{xy} \in S^*$ (that may have an intersection with I_{ij} and can not be selected anymore).

Now it is obvious that by selecting the interval I_{ij} the algorithm will miss at most two intervals of a given optimal solution. So, the overall approximation factor of this algorithm is two. □

2.4 3-Approximation Algorithm for Unit Height 2D-nKPML in Fixed Model

A simple greedy algorithm can approximate the 2D-nKPML problem within factor three. The main idea is to repeatedly choose the left most label such that a consistent solution is found.

Algorithm Sort all labels according to their right edge in ascending order and save them in a sorted list L. As long as the sorted list L is not empty, pick the first label L_{ij} from the list. If the label L_{ij} has no intersection with previously selected labels, and if no label is assigned to feature S_i, then select label L_{ij}. □

It is obvious that the running time of the above algorithm is $O(l \log l)$ where l is total number of label candidates. The initial sorting of labels needs no more than $O(l \log l)$ and the while loop runs once per label. Each iteration of while loop requires $O(\log l)$ to check if the Label L_{ij} has an intersection with previous selected labels and a constant time is required to check if the feature S_i is already labelled or not. So, the overall running time of the algorithm is $O(l \log l)$.

Lemma 2 *Given an instance of fixed model 2D-nKPML, a solution of size at least $\lambda/3$ can be computed, where λ is the maximum number of labels in an optimal solution.*

Proof Let S^*, S be an optimal solution and output of our algorithm accordingly. We show that for every selected label $L_{ij} \in S$ at most three labels from the optimal solution, S^*, will be missed.

Let $L_{ij} \in S$ be the label with leftmost right edge, then the first missing label might be $L_{ij'} \in S^*$ which can not be selected since the L_{ij} is selected (Note that $L_{ij'}$ is the optimum label selected for feature S_i in $S*$). The next two missing labels might be labels with minimum right edge in $L_{xy} \in S^*$ (that may have an intersection with I_{ij} and can not be selected anymore). Also note that since all labels has unit height, at most two such labels from the may exist that has an intersection with the first leftmost label.

Now it is obvious that by selecting the Label L_{ij}, above algorithm will miss at most three labels from a given optimal solution. So, the overall approximation factor of this algorithm is three. □

3 Problem Definition of 2D-nKMPL in Slider Model

Let $S = \{(S_1, S_2, \ldots, S_n\}$ be a set of points where each $S_i = \{s_{i_1}, s_{i_2}, \ldots, s_{i_{n_i}}\}$ shows all possible positions of point S_i, and a slider labeling model (1S, 2S or 4S) is given. The parameter l_{ij} specifying the length label at s_{i_j} is also given. It is also assumed that all labels have unit height and

$$\forall_{1 \le i \le n} \cap_{1 \le j \le n_i} L_{i_j} \neq \emptyset$$

(i.e., each pair of labels of a given site must have non-empty intersection.)

The 2D-nKMPL problem is to find the maximum number of non-intersecting labels such no point receives more than one label.

3.1 3-Approximation Algorithm of 2D-nKPML in Slider Model

A simple greedy algorithm can approximate the 2D-nKPML problem in slider model within factor three. The general idea is the same as the one in slider model algorithm, but with some modifications.

Algorithm Use the slide labeling algorithm [vKSW99] and whenever a label is assigned to a feature, remove other positions of the feature. □

Lemma 3 *Given an instance of slider model 2D-nKPML, a solution of size at least* $\lambda/3$ *can be computed, where* λ *is the maximum number of intervals in an optimal solution.*

Proof Since the algorithm of van Kreveld, Strijk and Wolff chooses the leftmost labels first, so the proof will be the same as fixed model. □

4 Conclusion

We considered a new variation of KPML problem introduced in [DMM00]. We show that the problem with unit height rectangular labels is in NP-Complete even in one dimensional case. We then propose two different, yet simple to implement, 2-approximation algorithms for solving the 1D-nKPML in fixed. Generalizing the idea, we obtain two 3-approximation algorithms for 2D-nKPML in both models.

We do not yet know the lower bounds of approximation algorithms and it can be interesting problem to work on.

References

[DF89] Jeffrey S. Doerschler and Herbert Freeman. An expert system for dense-map name placement. In *Proc. Auto-Carto 9*, pages 215–224, 1989.

[DMM+97] 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.

[DMM00] Srinivas Doddi, Madhav V. Marathe, and Bernard M.E. Moret. Point set labeling with specified positions. In *Proc. 16th Annu. ACM Sympos. Comput. Geom. (SoCG'00)*, pages 182–190, Hongkong, 12–14 June 2000.

[FW91] 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.

[Hir82] Stephen A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

[KM00] 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. Sixth Annual International Computing and Combinatorics Conference (COCOON'00)*, volume 1858 of *Lecture Notes in Computer Science*, pages 340–350, Sydney, 26–28 July 2000. Springer-Verlag.

[KT98] 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.

[MS91] Joe Marks and Stuart Shieber. The computational complexity of cartographic label placement. Technical Report TR-05-91, Harvard CS, 1991.

[SvK00] 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.

[SW01] Tycho Strijk and Alexander Wolff. Labeling points with circles. *International Journal of Computational Geometry and Applications*, 11(2):181–195, April 2001.

[vKSW99] Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13:21–47, 1999.

[Wag93] Frank Wagner. Approximate map labeling is in $\Omega(n \log n)$. Technical Report B 93-18, Fachbereich Mathematik und Informatik, Freie Universität Berlin, December 1993.

[Wag94] Frank Wagner. Approximate map labeling is in $\Omega(n \log n)$. *Information Processing Letters*, 52(3):161–165, 1994.

[Zor86] Steven Zoraster. Integer programming applied to the map label placement problem. *Cartographica*, 23(3):16–27, 1986.

[ZP00] Binhai Zhu and Chung Keung Poon. Efficient approximation algorithms for two-label point labeling. *International Journal of Computational Geometry and Applications*, 2000. To appear.

[ZQ01] Binhai Zhu and Zhongping Qin. New approximation algorithms for map labeling with sliding labels. *Journal of Combinatorial Optimization*, 2001. To appear.