國立臺灣大學電機資訊學院電機工程學系
碩士論文
Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

同步邊界標籤之演算法設計與分析
Algorithm Design and Analysis for Simultaneous Boundary Labeling

鍾家涵
Chia-Han Chung

指導教授：顏嗣鈞博士
Advisor: Hsu-Chun Yen, Ph.D.

中華民國 105 年 3 月
March, 2016

doi:10.6342/NTU201600238

誌謝

很幸運這兩年半不只是課堂學術上的精進，博理 606 實驗室的老師學長同學、在台大課堂社團認識的各領域神人、以及在華盛頓大學交換學生時認識的朋友（特別感謝室友陳南蓁）改變了我很多想法和視野。我會懷念大家一起吃膩 118 的日子，從各種最新技術、新聞時事、美劇、運動到爛心情抒發，你們的陪伴都十分重要。這篇論文的完成感謝顏嗣鈞教授給予的的指導和幫助，柯有容 latex 顧問、顏玓德張以潤演算法顧問、蔡萱尹牢靠的心靈支柱和幫我加油打氣等我出關的朋友們。最後特別感謝一直以來最辛苦的媽媽和爸爸。

doi:10.6342/NTU201600238

摘要

在 *boundary labeling* 領域，圖上的每個點都會透過 *leader* 和一個相關的標籤相連，所有的標籤位在該矩形圖的邊界上。目前，所有關於 boundary labeling 的研究都在討論如何替單一的圖產生可讀性高的標籤排列。然而，有時可能會有一系列相關的圖需要作 boundary labeling，這些圖共享一部分或全部的點集合和標籤集合。當我們對每張圖分別作 boundary labeling，而沒有考慮它們共通的部分，產生的結果會很難觀察圖之間的關聯性。為了解決這個問題，在這份論文，我們提出了一個叫做 *simultaneous boundary labeling* 的新問題，透過限制每張圖相同的點和標籤在相同的位置上，來保持圖和圖之間相關的部分，然後去計算最少 leader crossing 數和最短 leader 總長的標籤擺法來提高可讀性。我們設計了一些 heuristic 演算法去解決同時計算兩張圖的 crossing minimization 問題，並證明當同時計算超過四張圖的時候，crossing minimization 問題是 NP-complete。另外，用 *weighted bipartite matching* 演算法解決了 leader length minimization 問題。

關鍵字*:* simultaneous graph drawing, boundary labeling, crossing minimization, bipartite matching, barycenter algorithm

doi:10.6342/NTU201600238

Abstract

In *boundary labeling*, each feature point is connected to a label placed on the boundary of a rectangular image by a *leader*, which may be a rectilinear or a straight line segment. Currently, all the research about boundary labeling focuses on how to generate label placements for one image with high readability. However, there may be a series of related images, which share all, or parts of the same feature and label set, need to be labeled. If we calculate label placements for each image separately, it is hard to keep track of the relationship between images. To overcome the above difficulty, in this thesis we propose a new problem called *simultaneous boundary labeling*. We keep the relationship between images by limiting common features and labels of a series of images in the same place, and find a common label placement for all images with minimal leader crossing number and minimal total leader length to increase the readability. We design some heuristic algorithms when there are two related images need to be labeled and show the problem to be NP-complete when there are more than four images in the series. The leader length minimization problem can be solved by a *weighted bipartite matching* algorithm.

Keywords: simultaneous graph drawing, boundary labeling, crossing minimization, bipartite matching, barycenter algorithm

doi:10.6342/NTU201600238

Contents

誌謝 i

摘要 ii

Abstract iii

1 Introduction 1

1.1 Motivation 1

1.2 Thesis Organization 5

2 Preliminaries 6

2.1 Problem Definition 6

2.2 Related Work 12

3 Simultaneous boundary labeling 13

3.1 type-s leader 13

3.2 type-po leader 23

3.3 type-opo leader 28

4 Extension of simultaneous boundary labeling for two images 32

5 Conclusion and Future Work 37

Bibliography 39

doi:10.6342/NTU201600238

List of Figures

1.1 Example of map labeling 2
1.2 Example of boundary labeling 2
1.3 Common leader types of boundary labeling 3
1.4 Example of simultaneous graph drawing 3
1.5 Example result of simultaneous boundary labeling 4
1.6 Example result of extension SCM-s-2 5

2.1 Single image model 7
2.2 Feature-label relationship 8
2.3 Example of simultaneous boundary labeling for two images 9
2.4 Example of simultaneous boundary labeling for three images 10

3.1 Example for illustrating the influence of the x-coordinate of features in SCM-s-2 problem 14
3.2 Three categories of label in type-s $k = 2$ simultaneous boundary labeling 15
3.3 The experiment result of algorithms for SCM-s-2 20
3.4 The execution time of algorithms for SCM-s-2 20
3.5 The example of how to build an image R_1 by a permutation π_1 21
3.6 The weighted bipartite graph of SML-s-2 22
3.7 The experiment result of algorithms for SCM-po-2 26
3.8 The execution time of algorithms for SCM-po-2 27
3.9 A special case of type-po leader boundary labeling 27
3.10 Example of type-opo label placement with minimal total leader length . . 30
3.11 Example of type-opo label placements with minimal crossing count . . . 31

doi:10.6342/NTU201600238

4.1 Model of the extended version of simultaneous boundary labeling 33
4.2 Common usages of extended version of simultaneous boundary labeling . 33
4.3 Example result of extension SCM-*opo*-2 34
4.4 Example result of Extension-SCM-*po*-2 35
4.5 Original output of Extension-SCM-*po*-2 and its leader crossing count . . . 35

doi:10.6342/NTU201600238

List of Tables

1.1 Summary of our contributions 5

3.1 The experiment result of algorithms for SCM-s-2 17

3.2 The possibility of crossing-free in SCM-s-2 when there must a crossing for PCM-2 18

3.3 The possibility of crossing in SCM-s-2 when there must be no crossing for PCM-2 19

3.4 The experiment result of algorithms for SCM-po-2 26

4.1 The input data of the example of extension-SCM-opo-2 36

doi:10.6342/NTU201600238

Chapter 1

Introduction

1.1 Motivation

Automatic label placement is one of the important tasks in information visualization. The task is annotating features of interest in images by textual labels to provide information about features. The features are typically divided into three types: point features, line features, and area features [13]. There are many algorithms that automate this task and many proofs of the computational complexity of labeling problems. Generally, automatic label placement is divided into two categories, map labeling [20] and boundary labeling [21, 9, 11]. Map labeling algorithms calculate positions for labels, and put them right beside their target features like Figure 1.1. Boundary labeling algorithms put labels on the boundary of the image like Figure 1.2. The model of boundary labeling is first proposed by Bekos et al [3]. Each label is connected to its corresponding feature by a line, called leader. Figure 1.3 shows three common leader types. After that, there are many transformation problem about boundary labeling, such as multi-sided boundary labeling [15, 14], and boundary labeling for dynamic focus region [18, 10]. However, most of the researches on automatic label placement focus only on labeling features of one image.

doi:10.6342/NTU201600238

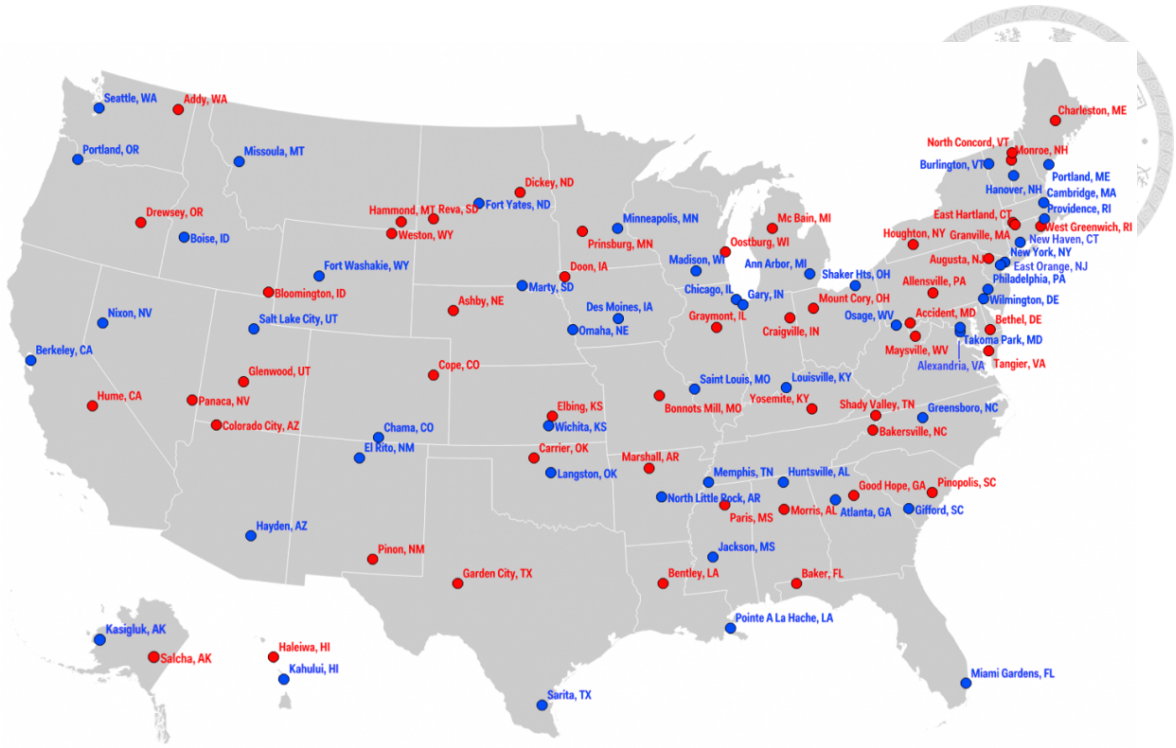


Figure 1.1: Example of map labeling

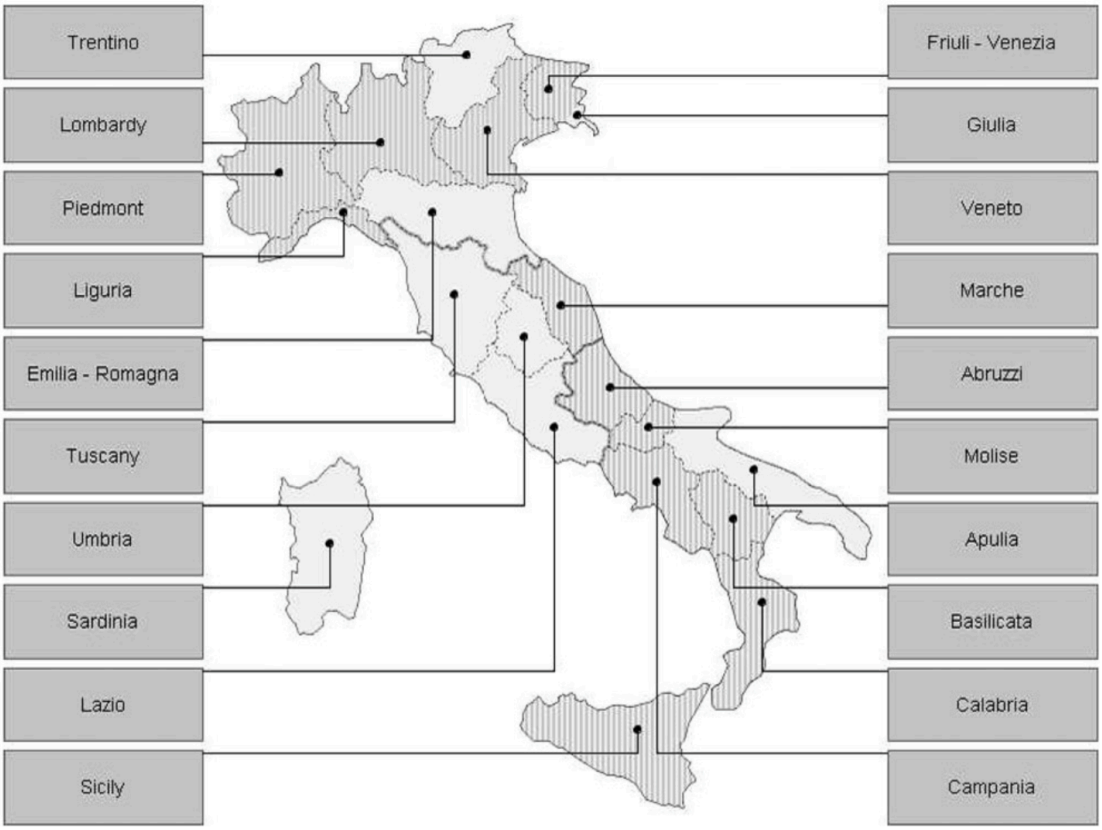


Figure 1.2: Example of boundary labeling

doi:10.6342/NTU201600238

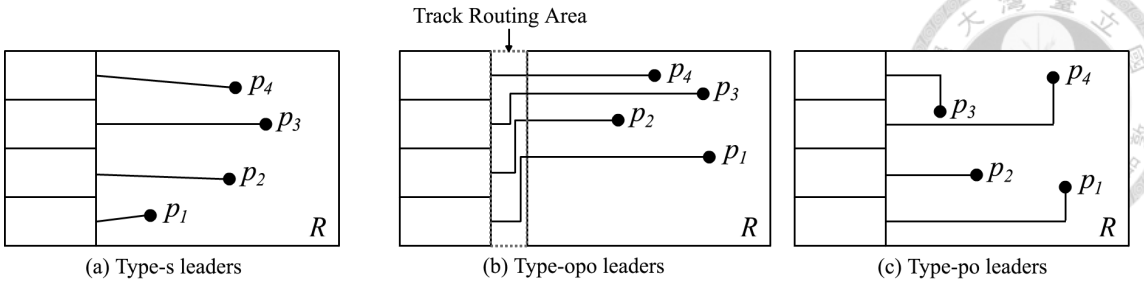


Figure 1.3: Common leader types of boundary labeling

In the graph drawing and infromation visualization area, there is a group of researches called *simultaneous graph drawing* which focus on how to generate the layout of multiple related graphs [6, 1, 8]. Consider the problem of drawing a series of graphs that share all, or parts of the same vertex set. The graphs may represent different relations between the same set of objects. For example, in social networks, graphs are often used to represent relations of people. Based on different relations, different graphs can be generated with the same set of people. Alternatively, the graphs may be the result of a single relation that changes through time. For example, the Facebook friendship relation of the same set of people may change through time. Based on different time point, different graphs can be generated. The problem of simultaneous graph drawing is how to generate a good layout (as shown in Figure 1.4) to visualize a series of related graphs. A good layout needs to consider two important criteria: the *readability* of the individual layouts and the *mental map preservation* in the series of layouts.

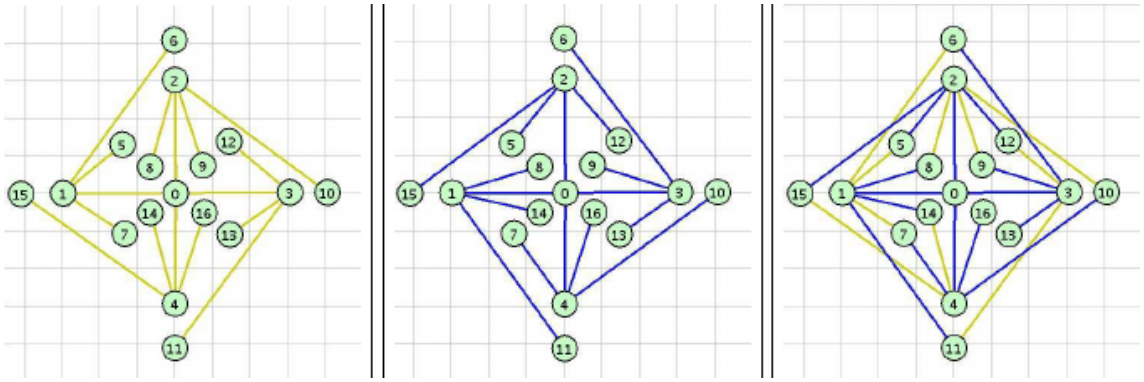


Figure 1.4: Example of simultaneous graph drawing

This work studies the *simultaneous automatic labeling problem*, which uses the concept of simultaneous graph drawing on automatic labeling. We are concerned with the

doi:10.6342/NTU201600238

problem of calculating label placements of a series of images, which share all, or parts of the same feature and label set. The label placement needs to fulfill two important criteria: the *readability* of the individual images and the *mental map preservation* in the series of images. The common criteria for readiblity are minimum crossings of leaders (the line connect corresponding feature and label), maximal size of labels, and minimum distance sum between features and labels. These help readers recognize the information on the image and connect features to their associated labels. The method for mental map preservation is to place the same labels on the same positions on different images. This helps readers see the relationship and observe differences among a series of images. If we individually label each image by traditional labeling algorithms, we may optimize the readibility. However, it may result in bad mental map preservation among the series of images, because label positions of the same labels may vary according to different topic of images. Conversely, if we calculate all the label placements of features from all the images by traditional labeling algorithms at the same time, we are optimizing the mental map preservation but the individual images may be far from readable.

In simultaneous automatic labeling (as shown in Figure 1.5), same labels of the same feature are placed at the exact same location in all images to preserve the mantal map. And we ease constraints to the readibility while calculating label placements for a series of images. For example, we only consider the crossings of leaders or the overlappings of labels in the same image. Therefore, we can trace the relationship or change between a series of images more easily than trace by the results from individually labeling each image. Also, we can read individual images to learn details more easily than read the results from labeling features of all images at the same time. Moreover, comparing to individually labeling each image, simultaneous labeling can save time by processing all the common feature-label pairs at once.

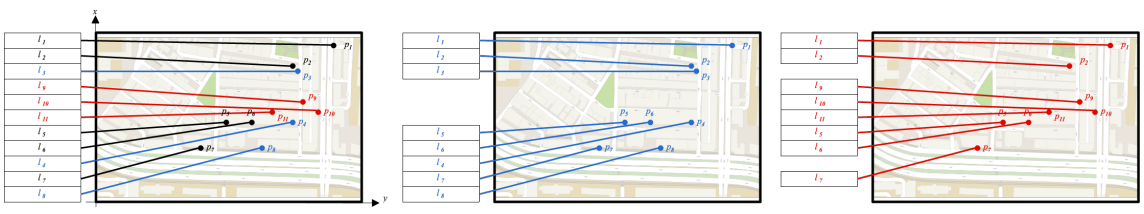


Figure 1.5: Example result of simultaneous boundary labeling

doi:10.6342/NTU201600238

1.2 Thesis Organization

In this chapter, we introduce the motivation and concept of simultaneous boundary labeling. The specific problem definition and related work are in Chapter 2. Our main contribution is in Chapter 3, problems are discussed under three kinds of leader, which are type-s, type-po, and type-opo. The parameter k is the number of images in a series of image set which is the input. For a series of two images ($k = 2$), we design many algorithms for the problem, and prove that when there are more than four images in the set ($k \geq 4$) the problem is NP-complete. In Chapter 4, we build an extended version of simultaneous boundary labeling problem (as example in Figure 1.6) and prove it can be the same problem as the $k = 2$ simultaneous boundary labeling problem. Then the conclusion of this thesis is in Chapter 5.

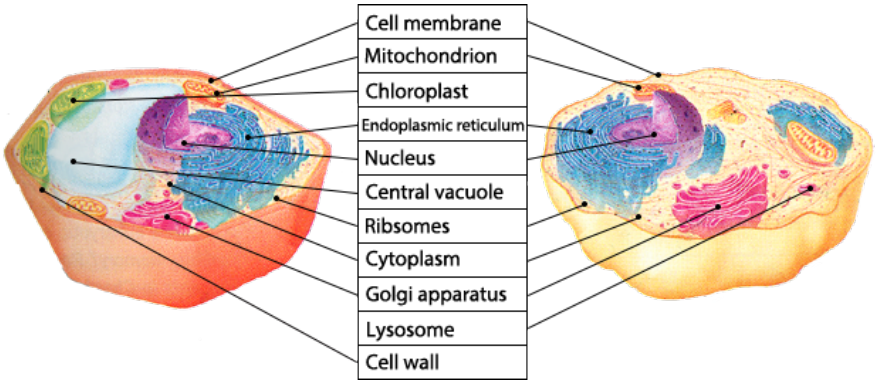


Figure 1.6: Example result of extension SCM-s-2

Table 1.1: Summary of our contributions

	type-s	type-po	type-opo
$k = 2$ minimum crossing number	heuristic	heuristic	$O(n \log n)$
$k = 3$ minimum crossing number	?	?	?
$k \geq 4$ minimum crossing number	NP-complete	NP-complete	NP-complete
any $k \in \mathbb{N}$ minimum total leader length	$O(n^3)$	$O(n^3)$	$O(n^3)$
extension (minimum crossing number)	the same as $k = 2$		

doi:10.6342/NTU201600238

Chapter 2

Preliminaries

2.1 Problem Definition

In this research we present efficient algorithms for computing optimal simultaneous boundary labelings of a series of images. Figure 2.1 shows a single image labeling in our model: we are given a rectangular image $R = (P, L)$ with width w and height h.

- $P = \{p_1, p_2, ..., p_n\}$ is a set of n *features* (or *sites*) where $p_i = (px_i, py_i)$ is the coordinate of the i-th feature point p_i in rectangle R.
- $L = (l_1, l_2, ..., l_n)$ is an ordering of a set of n *labels*, l_i is the label of feature p_i.

Each feature is associated with a different axis-parallel rectangular label l_i (think of the bounding box of the object name written as a single line of text). Each label and its associated feature are connected by a line segment, denoted as a *leader* [2]. The point where the leader is connected to it's label is called *port*. The set of n label ports is denoted as $Y = \{y_1, y_2, ..., y_n\}$, which is sorted by the value of y_i, the y-coordinate of port i in the increasing order. We assume the port is fixed in this research, and put it in the middle of label l_i's right edge. Our goal is to assign each label l_i to a port y_j to achieve some criteria.

doi:10.6342/NTU201600238

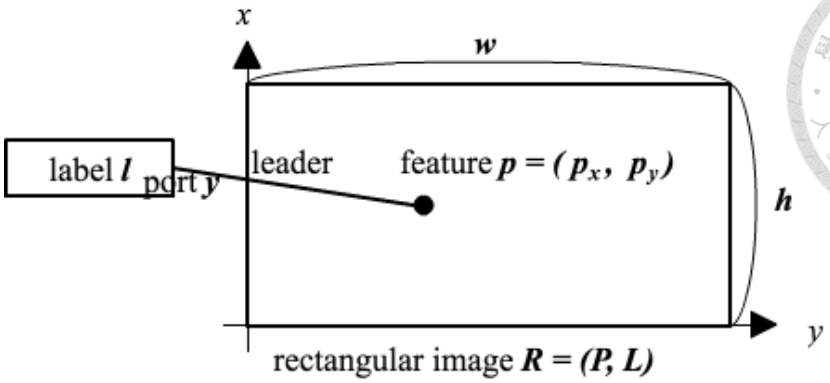


Figure 2.1: Single image model

When it comes to labeling a series of k rectangular images, the model of simultaneous boundary labeling problem is shown by the following example of $k = 2$. We are given two rectangular images of identical size $R_1 = (P_1, L_1)$ and $R_2 = (P_2, L_2)$.

- $n_1 = |P_1| = |L_1|$ is the number of feature-label pair of R_1
- $n_2 = |P_2| = |L_2|$ is the number of feature-label pair of R_2
- $n = |P| = |P_1 \cup P_2|$
- $m = |L| = |L_1 \cup L_2|$
- Y: a set of m label ports sorted by y-coordinate in increasing order
- y_i: the y-coordinate of the i-th port

In image 1 (e.g. Figure 2.3(a)), there is a set of n_1 features P_1 and each feature is associated with a distinct label in the set L_1. In image 2 (e.g. Figure 2.3(b)), there is a set of n_2 features P_2 and each feature is associated with a distinct label in the set L_2. These 2 images share some common features, that is feature at the same coordinates.

We assume that each feature can only have one associated label (one-to-one) in the same image, as the first row in Figure 2.2. The same feature in different images can have more than one associated labels and different features in different images can have the same asscoiated label, the second and third rows in Figure 2.2 respectively demonstrate these relationship.

doi:10.6342/NTU201600238

The layout like Figure 2.3(c) which is a good simultaneous boundary labeling result of a series of images according to the criterion of minimum crossing number of leaders, is the goal of our problem. All the labels are put on the left side of the image, all the features from both image 1 and image 2 are put in the rectangular image area (blue ones are from image 1, red ones are from image 2, and black ones are from both images).

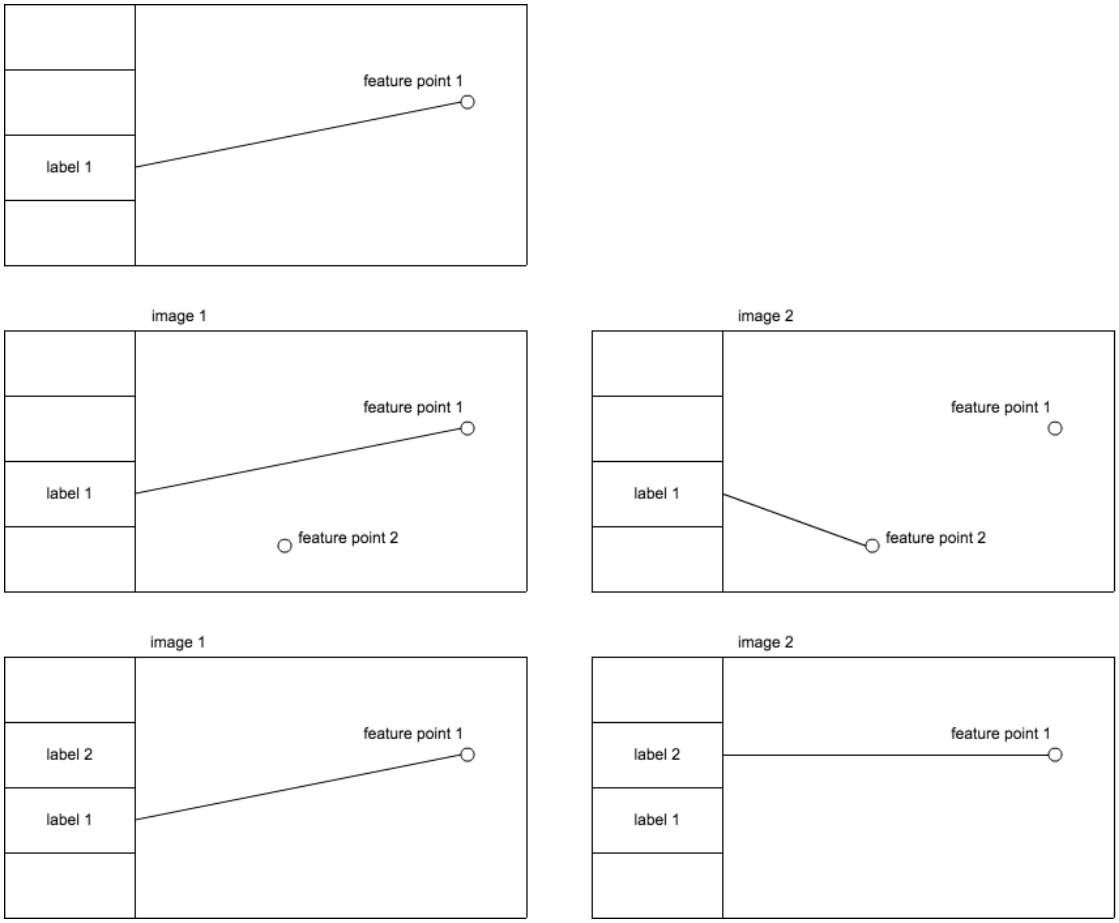
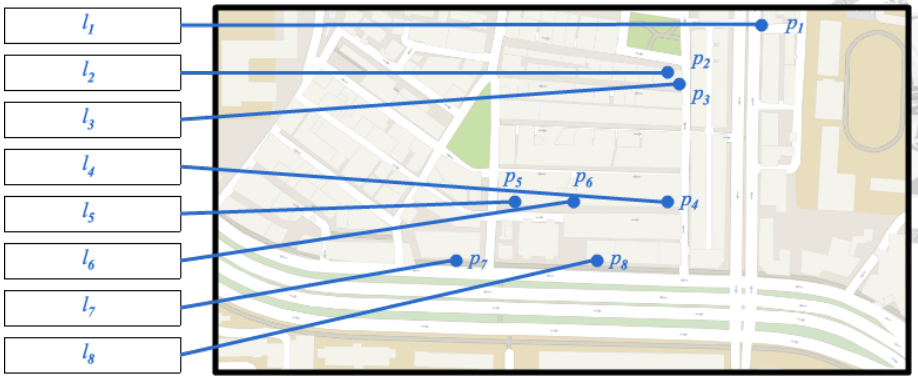
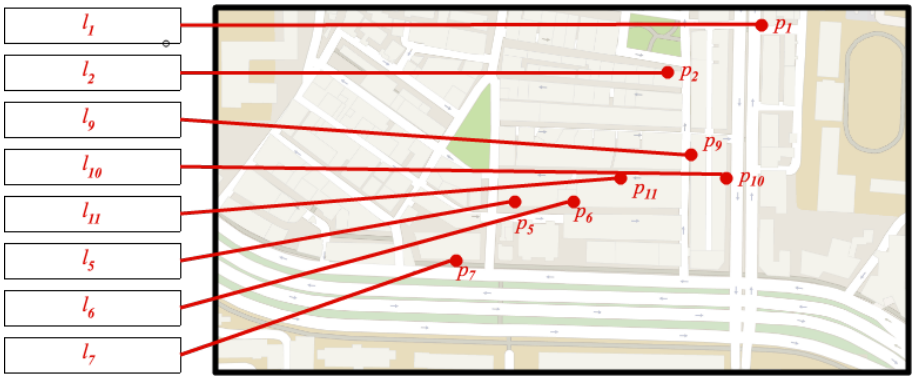


Figure 2.2: Feature-label relationship

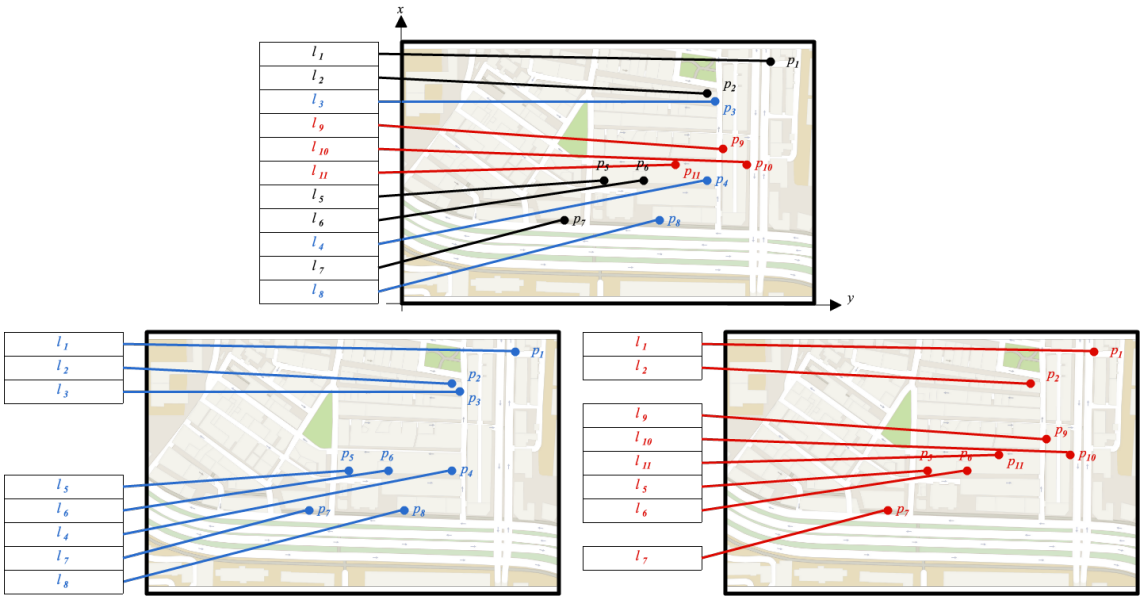
doi:10.6342/NTU201600238



(a) Image 1



(b) Image 2



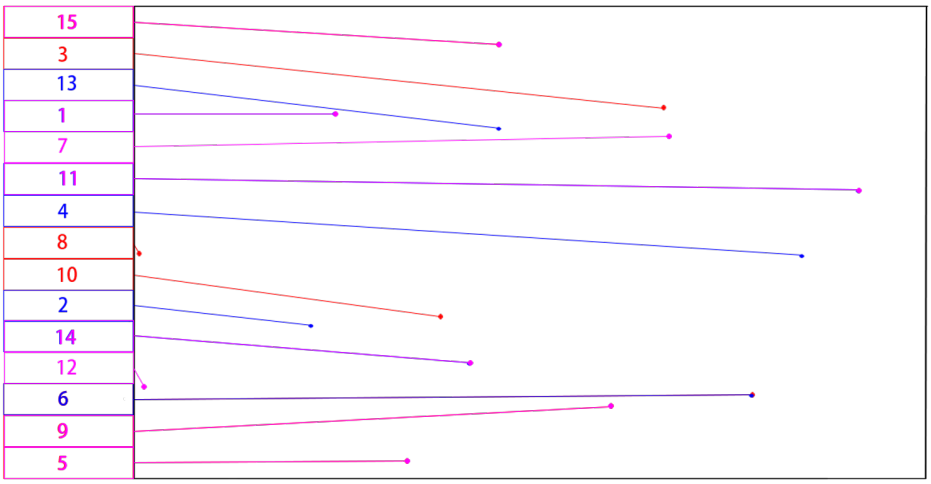
(c) Result of $k = 2$ simultaneous boundary labeling

Figure 2.3: Example of simultaneous boundary labeling for two images

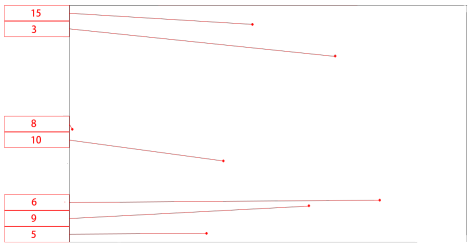
Figure 2.4 is the example result of simultaneous boundary labeling for three images ($k = 3$). These three images share some common feature points and labels. The common

doi:10.6342/NTU201600238

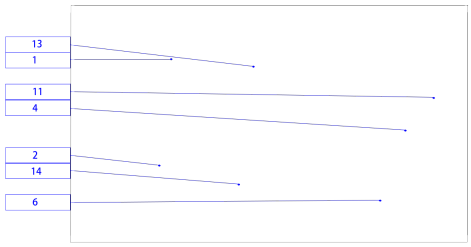
labels and feature points are put on the same position to keep the relationship between images and when we look three images separately like Figure 2.4(b), 2.4(c), and 2.4(d), we can find that the readability of these three images is good (no leader crossing).



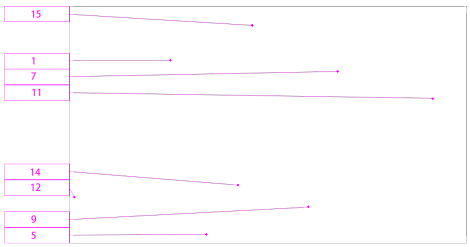
(a) Result of $k = 3$ simultaneous boundary labeling



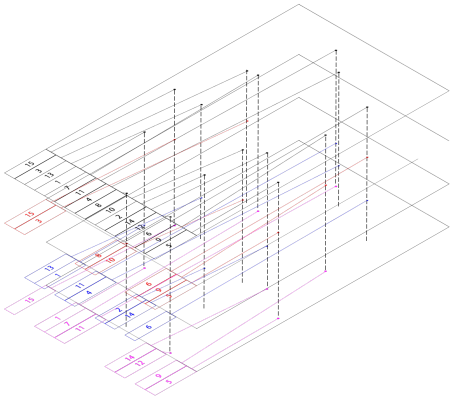
(b) Image 1



(c) Image 2



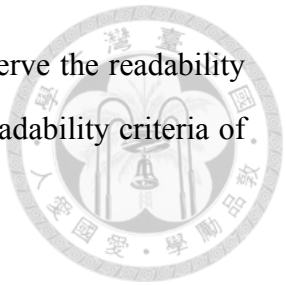
(d) Image 3



(e) The relationship of three images

Figure 2.4: Example of simultaneous boundary labeling for three images

doi:10.6342/NTU201600238



The result helps observing relationship between images and preserve the readability of individual image if we want to read it separately. The common readability criteria of map labeling quality [4] are:

- minimal crossing numbers of leaders.
- minimal length sum of leaders.
- minimal bend numbers of leaders.
- maximal label size.
- minimal overlap numbers of labels.

These criteria help shortening the distance between features and their related labels, reducing distraction from leader crossings and increasing the recognizability of label content. Therefore, the associated feature and label pairs can be found and located more easily.

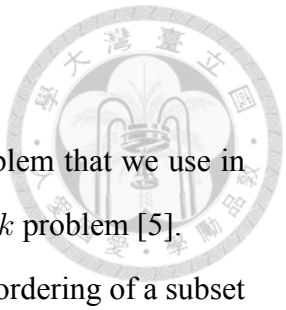
Each feature and its associated label are connected by a *leader*. A leader consists of a sequence of segments. These segments are parallel (p) or orthogonal (o) to the side of the bounding rectangle R to which the label is attached. The segments can also be a 45 degree diagonal-line (d) or a straight-line (s) which connects a label and a feature directly. In this paper we focus on leaders of the types s, po and opo, see Figures 1.3. For each type-opo leader we further insist that the parallel p-segment is immediately outside the bounding rectangle R and is routing in the so-called *track routing area*.

The *Simultaneous boundary labeling problem* is given a series of k rectangular images, determine a boundary labeling for $R_1, ..., R_k$, i.e. compute a label placement for each distinct label such that the sum of number of leader crossings in each image is minimum.

In this thesis, we denoted the Crossing Minimization problem for k images for simultaneous boundary labeling with type s leaders as SCM-s-k. There are also SCM-s-1 and SCM-s-2 which represent the same problem of 1 image simultaneous boundary labeling and 2 images simultaneous boundary labeling. The total leader Length Minimization problem for k images simultaneous boundary labeling with type s leaders is denoted as SLM-s-k. There are also SLM-po-k and SLM-opo-k represent the same problem with type po and type opo leaders respectively.

doi:10.6342/NTU201600238

2.2 Related Work



Here is a review of related work on a crossing minimization problem that we use in this research. The following are definition and the result of the PCM-k problem [5].

Given a set of n labels U, a permutation π with respect to U is an ordering of a subset S of U. For example, $U = \{l_1, l_2, ..., l_5\}$, a permutation can be $\pi = (l_2, l_1, l_4, l_3, l_5)$. We call π a full permutation, if $S = U$, and call π a partial permutation, if $S \subset U$.

Definition 1. Permutation Crossing Minimization problem with k permutations (PCM-k) Question: $P = \{\pi_1, ..., \pi_k\}$ is a given set of (full or partial) k permutations on a set of n labels $U = \{l_1, l_2, ..., l_n\}$. The crossing minimization problem is finding a best permutation π^* such that the crossing number of P is minimal. The crossing number definition is as following,

$cross(P) = min_{\pi^*} cross(P, \pi^*)$

$cross\,(P, \pi^*) = \sum_{i=1}^{k} cross\,(\pi_i, \pi^*)$

$cross\,(\pi_i, \pi^*) = |\{(l_u, l_v)\,|\pi_i\,(l_u) < \pi_i\,(l_v) \quad and \quad \pi^*\,(l_u) > \pi^*\,(l_v)\}|$

Result: This problem is proved to be NP-hard for any $k \geq 4$ permutations. However it can be solved in $O(n)$ time when $k = 2$ by barycenter algorithm.

Definition 2. Barycenter algorithm

The Barycenter algorithm is also a common heuristic algorithm for one-sided bipartite crossing minimization problem [12]. The algorithm sorts vertexes from free side according to the barycenter value. The barycenter value of each vertex u, vertex from free side, is the average x-coordinates of all neighbors (vertices, which are at fixed side, connected to u) of vertex u.

As for the PCM-k problem, order of each vertex u in π^* is chosen as the barycenter(average) of its order in permutation π_1 to π_k. If two vertices have the same barycenter, we order them arbitrarily.

$\pi^*(u) = bary(u) = \frac{1}{deg(u)} \sum_k \pi_k(u)$

For more about algorithms of crossing minimization and their performance see references [12, 19, 7, 17].

doi:10.6342/NTU201600238

Chapter 3

Simultaneous boundary labeling

In this chapter, we only discuss one-sided (left side) boundary labeling. All the results could be used in right-sided boundary labeling by minor changes.

3.1 type-s leader

Lemma 1. *SCM-s-1 can be solved in $O(n \log n)$ time.*

Proof. Given $k = 1$, an image $R = (P, L)$, and Y, a crossing-free type-s leader label placement can be constructed in $O(n \log n)$ time with the following algorithm.

For $i = 1, ..., n$, a ray emitted from y_i vertically downwards, then rotate in counter-clockwise. The first unlabeled feature $p \in P$ that is hit by the ray is assigned to y_i [3].

The correctness can be proved by contradiction. If there would be a crossing of two leaders (y_1, p_1) and (y_2, p_2), the rotating ray emitted from y_1 would have found p_2 first, and connected p_2 to y_1 not y_2. The time complexity of this algorithm is $O(n^2)$. The first touched feature point p_i is being searched from total n feature points for each port y_j in $O(n)$ time and there is n ports. The time complexity can be improved to $O(n \log n)$ by deleting p_i from the feature point set P when p_i is assigned to a port y_j.

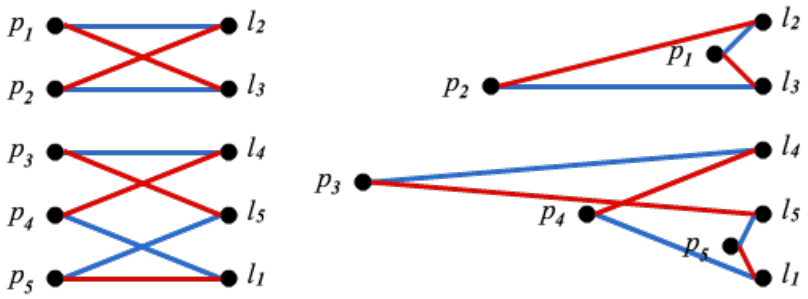
For the condition that all the labels are being put on the right side of the image, we only need to change the direction of the ray emitted from each port y.

□

doi:10.6342/NTU201600238

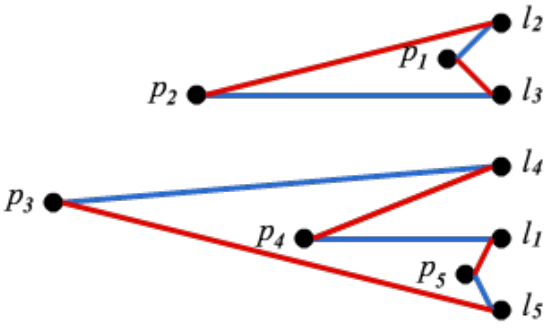
For SCM-s-2, we are not able to give a polynomial-time algorithm or show NP-hardness for the problem at this point.

Given $k = 2$, images R_1, R_2, and Y, simply assigning each feature to a port by the order of their y-coordinates might lead to crossings. Unlike one-sided bipartite crossing minimization problem or PCM-k, x-coordinates of feature points influence crossing numbers in the type-s boundary labeling problem. For example, in Figure 3.1, given images $R_1 = (P_1, L_1)$ and $R_2 = (P_2, L_2)$. $P_1 = P_2 = \{p_1, p_2, p_3, p_4, p_5\}$. $L_1 = (l_2, l_3, l_4, l_1, l_5)$. $L_2 = (l_3, l_2, l_5, l_4, l_1)$. The solution π^* to PCM-2 problem is $(l_2, l_3, l_4, l_5, l_1)$ as in Figure 3.1(a). When this solution is applied to SCM-s-2 problem, the result is shown in Figure 3.1(b). However, there exists a non-crossing label placement $(l_2, l_3, l_4, l_1, l_5)$ of SCM-s-2 problem as in Figure 3.1(c).



(a) The graph of best permutation π^* for PCM-2 problem.

(b) The graph of π^* applied in SCM-s-2 problem.



(c) The best solution to SCM-s-2 problem.

Figure 3.1: Example for illustrating the influence of the x-coordinate of features in SCM-s-2 problem

Therefore, we design some polynomial time heuristic algorithms to generate a type-s

doi:10.6342/NTU201600238

leader label placement of minimum number of crossings and compare their results with the optimal result.

3.1.1. Greedy-SCM-s-2

The first algorithm for SCM-s-2 is a greedy algorithm. When $k = 2$, labels can be divided into the following three categories by the relationship between labels and their associated features.

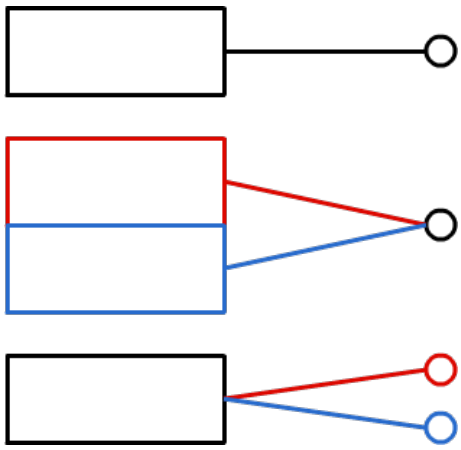


Figure 3.2: Three categories of label in type-s $k = 2$ simultaneous boundary labeling

Labels in the first category are the ones which have the same associated feature point in image R_1 and R_2. Labels in the second category are the ones which only have one associated feature point in either image R_1 or R_2. Labels in the third category are the ones which have two different associated feature points in image R_1 and R_2 respectively. Figure 3.2 illustrates these three categories of label, the blue feature point and label are from image R_1. The red feature point and label are from image R_2. The black ones are from both images R_1 and R_2.

If all the labels are from the first 2 categories, a non-crossing label placement can be constructed by the algorithm in lemma 1 in $O(n \log n)$ time. So the idea is that we use the algorithm in lemma 1 to calculate a label placement for labels in the first 2 categories then deal with labels in the third category. Assume there are total m labels in the 3rd category.

First, a label ordering is built for labels from 1st and 2nd categories by the algorithm from lemma 1. Second, we sort the remaining m labels (labels in the 3rd category) by the y-coordinate of their associated points in increasing order. At last, each of the sorted label are assigned to the port one by one at where it cause the minimal leader crossings. The

doi:10.6342/NTU201600238

time complexity of the greedy algorithm is $O((n-m)\log(n-m)+mn)$. The fewer the labels in the 3rd category, the faster the algorithm.

3.1.2. BaryY-SCM-s-2

The second algorithm for SCM-s-2 is a barycenter algorithm according to the y-coordinate of associated feature points. Labels are sorted by their average y-coordinates of associated feature points in increasing order. Then each label is assigned to a port from bottom to top according to this order. The time complexity is $O(n \log n)$ due to sorting algorithm.

3.1.3. BaryRay-SCM-s-2

The third algorithm for SCM-s-2 is a barycenter algorithm according to the label placement of each image. To consider x-coordinates of features as well. First, the label placements of image R_1 and image R_2 are generated by the algorithm in lemma 1. Second, labels are sorted by the average of their order in image R_1 and image R_2 in increasing order. Then each label is assigned to a port from bottom to top according to this order. The time complexity is $O(n \log n)$ for both the initial label placement part and the sorting algorithm.

3.1.4. BaryRayEnhanced-SCM-s-2

The fourth algorithm for SCM-s-2 is the enhanced version of the preceding algorithm. When the average ranks of labels are the same, we compare the value of their average y-coordinates of associated feature points. The label with smaller average y-coordinate is assigned to the lower port. The time complexity is still $O(n \log n)$.

3.1.5. LenMin-SCM-s-2

The fifth algorithm for SCM-s-2 is a bipartite matching algorithm which finds the label placement with minimal total leader length. This label placement also gives a good but not optimal result to the minimal leader crossing problem. The time complexity is $O(n^3)$.

The experiment result is shown in Table 3.1. Table 3.1(a) is the average total crossing count of $k = 2$ type-s full simultaneous boundary labeling problem solved by differnt algorithms according to different number of given features. Table 3.1(b) is the total execution time of $k = 2$ type-s full simultaneous boundary labeling problem solved by differnt algorithms according to different number of given features. For each set of test param-

doi:10.6342/NTU201600238

eters, we randomly generated 1000 test instances. All feature points have un-repetitive integer x-coordinates between 0 and 800 as well as y-coordinates between 0 and 600. The feature sets and label sets of image R_1 and image R_2 are the same. There are n distinct features in the feature sets and n distinct labels in the label sets. The associated labels of a feature point in different images are different, i.e., the feature-label relationships are different between these two images.

Table 3.1: The experiment result of algorithms for SCM-s-2

(a) average leader crossing counts

run = 1000, w = 800, h = 600

Algorithm \ label count	5	6	7	8	9	10	15	20	25	50
Initial	8.07	12.22	16.97	22.56	29.53	36.94	86.65	156.62	248.27	1016.13
PCM-2	5.03	7.41	10.39	14.02	18.17	22.59	52.47	95.38	150.85	616.58
Greedy-SCM-s-2	3.32	4.99	6.94	9.39	12.30	15.27	36.46	66.80	106.44	447.65
BaryY-SCM-s-2	3.84	5.70	8.08	11.02	14.29	17.72	42.20	76.81	122.49	505.91
BaryRay-SCM-s-2	4.02	5.88	8.38	11.37	14.74	18.30	43.00	77.95	124.03	508.82
BaryRayEnhanced-SCM-s-2	3.84	5.70	8.08	11.02	14.29	17.72	42.20	76.81	122.49	505.91
LenMin-SCM-s-2	3.45	5.11	7.22	9.71	12.63	15.75	37.19	67.45	106.92	441.78
Optimal	3.32	4.60	6.36	8.52	11.14					

(b) total execution time (s)

Algorithm \ label count	5	6	7	8	9	10	15	20	25	50
PCM-2	0.09	0.08	0.07	0.19	0.17	1.09	1.04	0.90	1.16	2.02
Greedy-SCM-s-2	1.32	1.19	1.46	6.47	7.69	37.64	48.50	74.65	120.26	703.26
BaryY-SCM-s-2	0.10	0.07	0.07	0.24	0.23	1.13	1.05	1.11	1.45	2.35
BaryRay-SCM-s-2	0.14	0.22	0.10	0.27	0.26	1.47	1.48	1.49	1.85	2.69
BaryRayEnhanced-SCM-s-2	0.17	0.14	0.13	0.29	0.32	1.81	1.95	1.86	1.91	2.58
LenMin-SCM-s-2	0.68	0.42	0.42	1.82	2.49	14.74	22.99	31.35	52.23	140.28
Optimal	6.56	16.31	104.3	6143	53948					

Besides the five algorithms mentioned above, we use the result of the PCM-2 problem and optimal case to compare. To generate the optimal result for the PCM-2 problem, we set all the x-coordinates of feature points to one, and solve it by the barycenter algorithm. In PCM-2 case, only y-coordinates matter, so the result generated by barycenter algorithm is optimal, that has been proved by [5]. To generate the optimal result of simultaneous boundary labeling problem, which is the label placement with minimal leader crossing, we

doi:10.6342/NTU201600238

calculate the crossing number for all possible label placements. It's really time-consuming because there are total $n!$ label placements for each test case.

In the case of boundary labeling with type-s leader, the x-coordinates of features matter as well. For example, given 2 labels l_1 and l_2 as well as 2 features p_1 and p_2. Every label-feature pair is connected by a leader, l_1 is the label of p_1 and l_2 is the label of p_2. If the y-coordinate of p_1 is bigger than y-coordinate of p_2 and the y-coordinate of l_1 is smaller than y-coordinate of l_2, there must exist a crossing of leaders in PCM-k, 2-layer 1-sided bipartite graph crossing minimization, and type-*opo* boundary labeling problems. However, when it comes to type-s boundary labeling problem, intersection between leaders (l_1,p_1) and (l_2,p_2) is not inevitable in this situation.

Based on the relative relationship between feature p and coordinates of l_1 and l_2, we can divided p into three categories. The first category is that p is above both l_1 and l_2. The second category is that p is in the middle of l_1 and l_2. The third category is that p is under both l_1 and l_2.

Table 3.2: The possibility of crossing-free in SCM-s-2 when there must a crossing for PCM-2

category of p_1	category of p_2	intersection is avoidable
1	1	⋁
2	1	⋁
3	1	
2	2	⋁
3	2	
3	3	⋁

doi:10.6342/NTU201600238

Table 3.3: The possibility of crossing in SCM-s-2 when there must be no crossing for PCM-2

category of p_1	category of p_2	intersection may occur
1	1	⋁
1	2	
2	2	
1	3	
2	3	
3	3	⋁

When $l_1y > l_2y$ and $p_2y > p_1y$, there must be a crossing in PCM-2, however there are four out of six cases that the intersection of 2 leaders can be avoided in SCM-s-2 according to the different x-coordinates of p_1 and p_2. When $l_1y > l_2y$ and $p_1y > p_2y$, there must be crossing-free in PCM-2, however there are only two out of six cases that the intersection of 2 leaders may occur in SCM-s-2 according to the different x-coordinates of p_1 and p_2. Therefore, the average crossing number of optimal result is fewer than the result of PCM-2 in our experimental test cases.

Greedy-SCM-s-2 algorithm has the best performance among all parameters of feature point number and it costs more time than the other four algorithms (as shown in Figure 3.3). LenMin-SCM-s-2 algorithm performs next to it, but it has higher complexity than the complexity of Greedy-SCM-s-2 algorithm. Because the given number of feature points is not big enough, it seems Greedy-SCM-s-2 algorithm takes more time in the experiment owing to its complicated steps. BaryY-SCM-s-2 algorithm, which only considers the y-coordinates of features, is the fastest one (as shown in Figure 3.4). BaryRay-SCM-s-2 algorithm, which also considers the x-coordinates of features, is slower and does not outperform the others.

doi:10.6342/NTU201600238

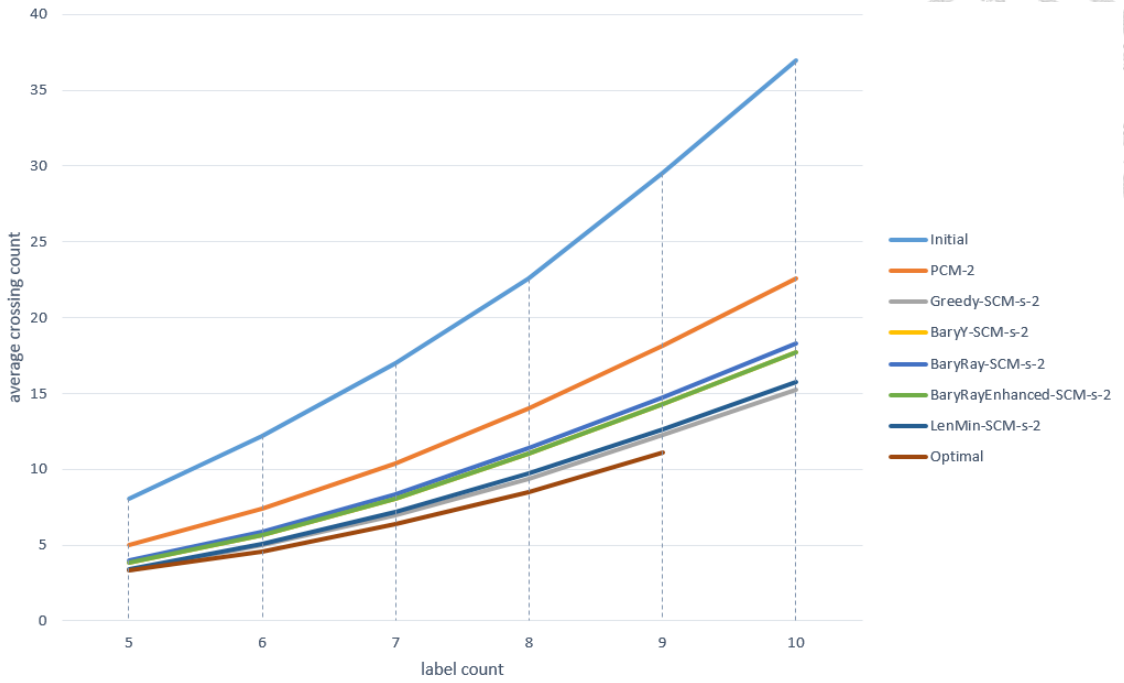


Figure 3.3: The experiment result of algorithms for SCM-s-2

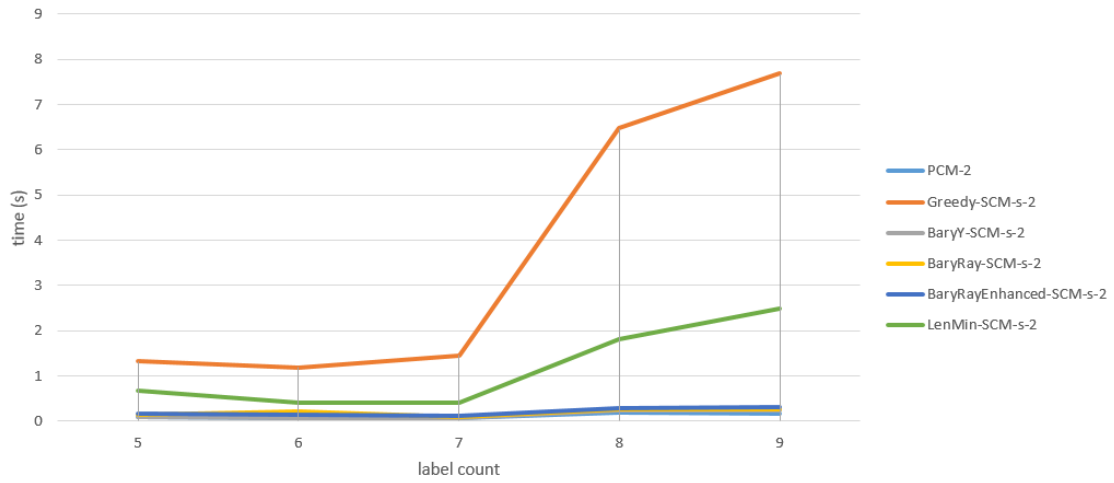


Figure 3.4: The execution time of algorithms for SCM-s-2

Theorem 1. *SCM-s-k is a NP-complete problem when $k \geq 4$.*

Proof. The problem is obviously in NP. Now we show the lower bound. The PCM-k problem can be reduced to type-s leader simultaneous boundary labeling. Since the PCM-k problem is NP-hard when $k \geq 4$ [5], so as this problem.

We show how to reduce PCM-k to our problem as following. Given a set of (full or partial) permutations $\pi = \{\pi_1, ..., \pi_k\}$ on a set of labels $U = \{l_1, l_2, ..., l_{n_l}\}$. First, we

doi:10.6342/NTU201600238

construct a series of k images $R_1, R_2, ..., R_k$ of which the height h is n_l and the width w is 2. The number n_l is equals to $|U|$, which means there are n_l distinct labels in the series of images. The number n_p of distinct features points in the series of images is $\max_{i\in\{1,...,k\}} |\pi_i|$. The x-coordinates of these distinct n_p points $p_1, p_2, ..., p_{n_p}$ are the same, i.e., they are on the same column. The y-coordinates of these distinct n_p points are 1 to n_p. Then for i = 1 to k, we build an image R_i for each permutation π_i. If there are $|n_i|$ elements in π_i, the feature point sets of R_i is $P_i = \{p_1, p_2, \ldots, p_{n_i}\}$, and the associated label of each point in R_i is the same as π_i.

For example in Figure 3.5, if the permutation $\pi_1 = (l_1, l_3, l_2, l_4, l_5)$, we can build an image R_1 of which the height h is 5 and the width w is 2. The point set of image R_1 is $P_1 = \{p_1, p_2, p_3, p_4, p_5\}$ and the coordinates of feature p_i is (1, i). The label ordering of R_1 is $L_1 = (l_1, l_3, l_2, l_4, l_5)$, which means l_1 is the associated label of p_1 and l_3 is the associated label of p_2, and so on.

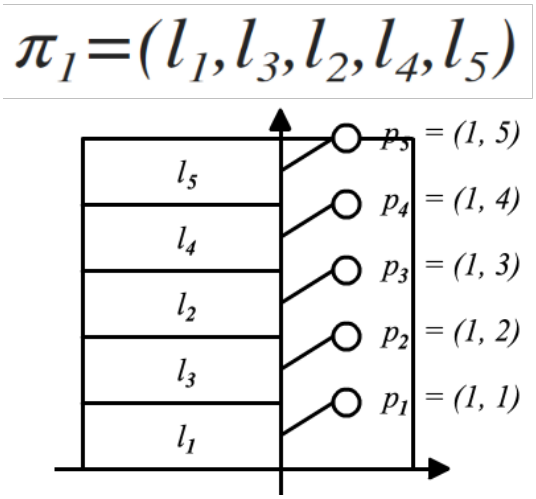


Figure 3.5: The example of how to build an image R_1 by a permutation π_1

If we can find a label placement, i.e., assign each label in L to a port in Y, which guarantee the sum of crossing number from image R_1 to R_k is minimal, the order of the label placement according to y-coordinate is also the best permutation π^* such that the crossing number of π is minimal.

If the order of l_a and l_b in permutation π_i and the order of which in best permutation π^* is different, means the y-coordinate order of l_a and l_b is different with their associated

doi:10.6342/NTU201600238

feature points p_a and p_b. This condition must cause a crossing in both PCM-k and the type-s simultaneous boundary labeling problem.

□

Theorem 2. *SLM-s-2 can be solved in $O(n^3)$ time.*

Proof. We transform the problem to Min-Weighted Bipartite Matching problem and solve it by the Hungarian Algorithm which can solve the Min-Weighted Bipartite Matching problem in $O(n^3)$ time. In SLM-s-2 problem, images R_1, R_2, and Y are given. The vertexs on one side of the bipartite graph are labels in $L_1 \cup L_2$, and the vertexs on the other side are coordinates of label ports. The weight of edge (l_i, y_i) is the sum of the Euclidean distance between the associated points of l_i and the port y_i. Figure 3.6 is the graph of the weighted bipartite graph of this problem. The label placement of minimal total leader length criterion for a series of two images does not guarantee the minimal leader crossing number according to our experiment result of SCM-s-2.

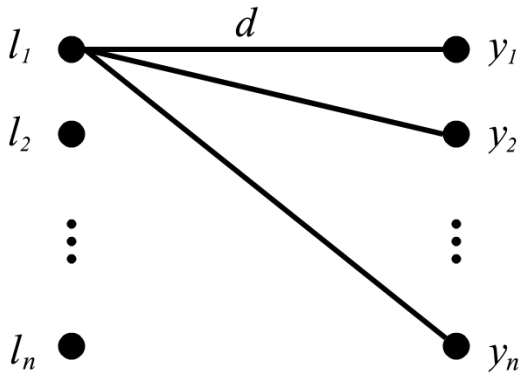


Figure 3.6: The weighted bipartite graph of SML-s-2

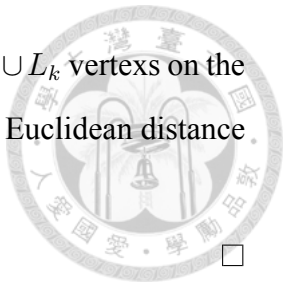
□

Theorem 3. *SLM-s-k ($k \in \mathbb{N}$) can be solved in $O(n^3)$ time.*

Proof. We transform the problem to Min-Weighted Bipartite Matching problem and solve it by the Hungarian Algorithm which can solve the Min-Weighted Bipartite Matching problem in $O(n^3)$ time. In SLM-s-k problem, images $R_1, R_2, ..., R_k$, and Y are given.

doi:10.6342/NTU201600238

The vertexs on one side of the bipartite graph are labels in $L_1 \cup L_2 \cup ... \cup L_k$ vertexs on the other side are label ports. The weight of edge (l_i, y_i) is the sum of the Euclidean distance between the associated points of l_i and the port y_i.



□

3.2 type-po leader

For SCM-*po*-2, The exact complexity of the problem or optimal algorithm are not known at this point. Therefore, we design some polynomial time heuristic algorithms to generate a type-*po* leader label placement of minimum number of crossings and compare their results with the optimal result.

In SCM-*po*-2, images R_1, R_2, and Y are given as input.

3.2.1. Sort-SCM-po-2

The first algorithm uses the concept of bubble sort. From i=1 to n, we swap the i-th label with i+1-th label if the crossing number is smaller after swapping in every iteration. The iteration ends until the number of crossing is not reduced. The time complexity is $O(n^2)$.

Algorithm Sort-SCM-po-2 algorithm

Input $order$:= the initial label placement

Output $result$:= the final label placement

$order$ = the initial label placement

repeat

  for i = 1 to n-1 **do**

    $tempOrder$ = $order$ after swapping the i-th and i+1-th label

    if crossing($tempOrder$) > crossing($order$) **then**

      $order$ = $tempOrder$

until the number of crossing is not reduced;

3.2.2. BaryY-SCM-po-2

doi:10.6342/NTU201600238

The second algorithm for SCM-po-2 is a barycenter algorithm according to the y-coordinate of associated feature points. Labels are sorted by their average y-coordinates of associated feature points in increasing order. Then each label is assigned to a port from bottom to top according to this order. The time complexity is $O(n \log n)$ due to sorting algorithm.

3.2.3. SortEnhanced-SCM-po-2

The third algorithm is the enhanced sorting algorithm. To enhanced the sorting algorithm, we set the initial label order from the result of barycenter algorithm. Then we improve the result (reduce crossing number) by the sorting algorithm. It is intuitional that the difference between label placement from barycenter algorithm and optimal label placement is very small, that is, every label is not far from its best label position. Therefore, through swapping the label with their neighbors and record the best order among them, the result of enhanced sorting algorithm can be very closed to the optimal result.

The time complexity is $O(n^2)$.

Algorithm SortEnhanced-SCM-po-2

Input $order$:= the label placement generated by barycenter algorithm

Output $result$:= the final label placement

$order$ = the label placement generated by barycenter algorithm

repeat

  for i = 1 to n-1 **do**

    $tempOrder$ = $order$ after swapping the i-th and i+1-th label

    if crossing($tempOrder$) > crossing($order$) **then**

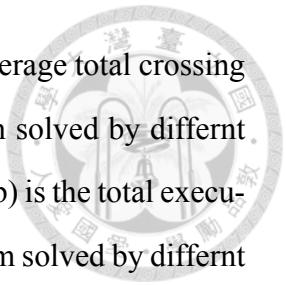
      $order$ = $tempOrder$

until the number of crossing is not reduced;

3.2.4. LenMin-SCM-po-2

The fourth algorithm for SCM-po-2 is a bipartite matching algorithm which finds the label placement with minimal total leader length. This label placement also gives a good but not optimal result to the minimal leader crossing problem. The time complexity of this algorithm is $O(n^3)$.

doi:10.6342/NTU201600238



The experiment result is shown in Table 3.4. Table 3.4(a) is the average total crossing count of $k = 2$ type-po full simultaneous boundary labeling problem solved by differnt algorithms according to different number of given features. Table 3.4(b) is the total execution time of $k = 2$ type-po full simultaneous boundary labeling problem solved by differnt algorithms according to different number of given features. For each set of test parameters, we randomly generated 1000 test instances. All feature points have un-repetitive integer x-coordinates between 0 and 800 as well as y-coordinates between 0 and 600. The feature sets and label sets of image R_1 and image R_2 are the same. There are n distinct features in the feature sets and n distinct labels in the label sets. The associated labels of a feature point in different images are different, i.e., the feature-label relationships are different between these two images.

Besides the four algorithms mentioned above, we generate the optimal result of simultaneous boundary labeling problem, which is the label placement with minimal leader crossing, by calculating the crossing number for all possible label placements. It's really time-consuming to calculate the crossing number for each label placement because there are total $n!$ label placements for each test case.

SortEnhanced-SCM-po-2 algorithm has the best performance among all parameters of feature point number (as shown in Figure 3.7) and is more than two times faster than the Sort-SCM-po-2 algorithm (as shown in Figure 3.8). By using the result of BaryY-SCM-po-2, the SortEnhanced-SCM-po-2 algorithm converges faster. LenMin-SCM-po-2 algorithm has the worst performance for type-po simultaneous boundary labeling problem of two images.

doi:10.6342/NTU201600238

Table 3.4: The experiment result of algorithms for SCM-*po*-2

(a) average leader crossing counts

run = 1000, w = 800, h = 600

Algorithm \ label count	5	6	7	8	9	10	15	20	25	50
Initial	6.02	9.10	12.90	17.40	22.41	28.47	67.00	123.21	196.13	802.46
Sort-SCM-po-2	3.02	4.75	7.14	10.27	13.42	17.39	45.78	89.28	149.95	681.600
SortEnhanced-SCM-po-2	2.26	3.38	5.06	6.89	8.89	11.34	28.18	53.23	87.56	388.58
BaryY-SCM-po-2	3.07	4.72	6.74	9.08	11.82	14.79	35.32	64.56	104.06	434.85
LenMin-SCM-po-2	3.43	5.02	7.15	9.72	12.64	15.71	36.98	67.47	106.56	440.36
Optimal	1.66	2.35	3.46	4.75	5.96					

(b) total execution time (s)

run = 1000, w = 800, h = 600

Algorithm \ label count	5	6	7	8	9
Sort-SCM-po-2	15.36	17.96	25.92	32.71	40.38
SortEnhanced-SCM-po-2	6.78	7.89	10.77	14.98	20.40
BaryY-SCM-po-2	0.74	0.87	1.20	1.15	0.78
LenMin-SCM-po-2	1.99	1.98	2.58	3.17	3.78
Optimal	31.62	153	1610	14756	134369

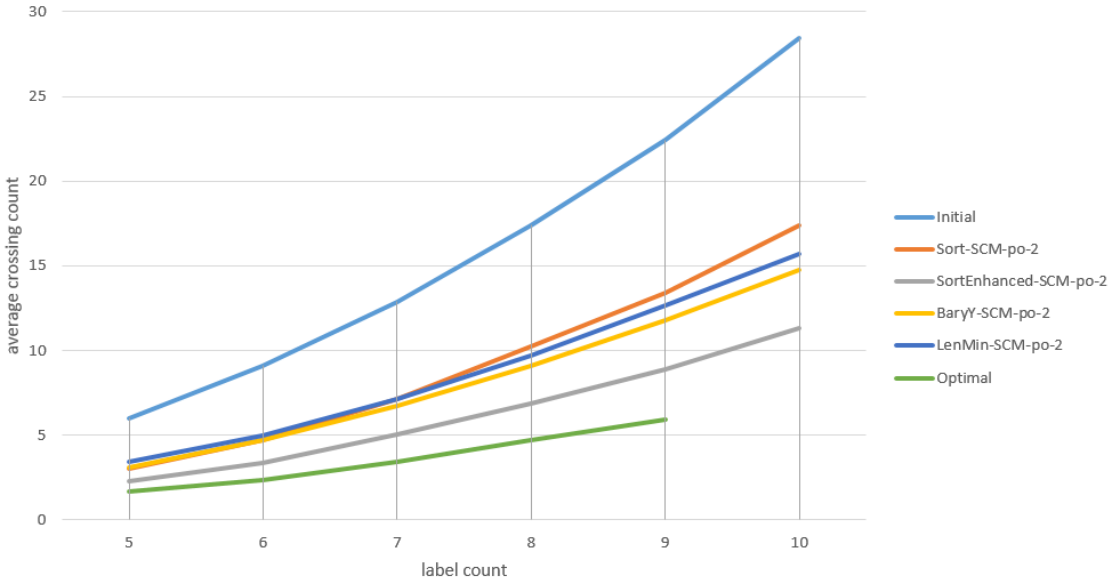


Figure 3.7: The experiment result of algorithms for SCM-*po*-2

doi:10.6342/NTU201600238

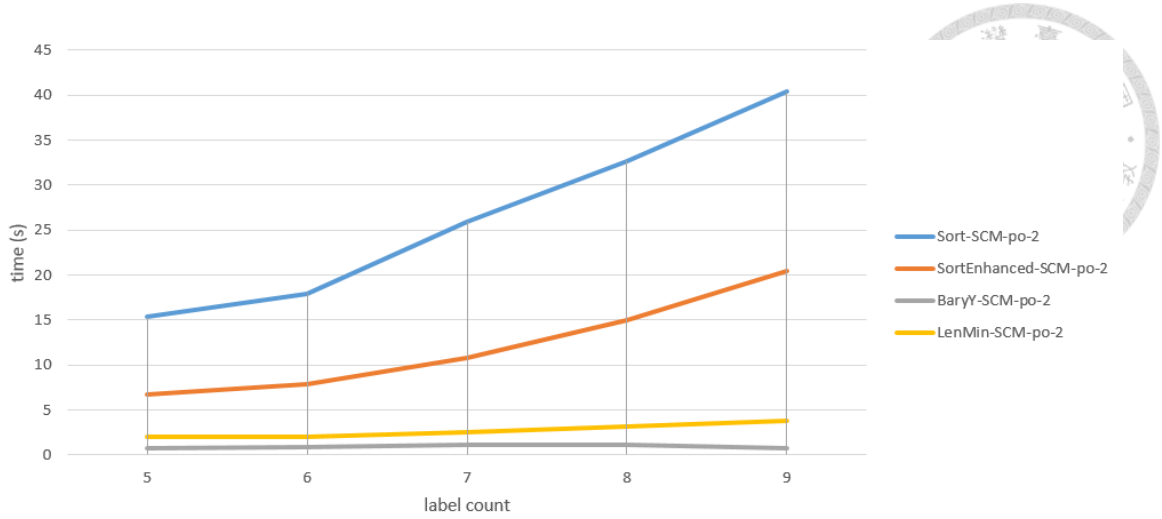


Figure 3.8: The execution time of algorithms for SCM-*po*-2

Theorem 4. *SCM-po-k is a NP-complete problem when* $k \geq 4$.

Proof. The problem is obviously in NP. Now we show the lower bound. We can reduce the PCM-k problem to a special case of type-*po* leader simultaneous boundary labeling problem where the y-coordinate of any feature is smaller than that of the lowest port of the lowest label, as Figure 3.9. (Note that some of edges are omitted in the figure.) Since the PCM-k problem is NP-hard when $k \geq 4$ [5], so as this problem. In SCM-*po*-k k ($k \geq 4$) problem, images $R_1, R_2, ..., R_k$, and Y are given. It is obvious that in order not to induce the crossing of leaders, a label should be assigned to a port from bottom to up according to the order of features sorted by their x-coordinates in increasing order.

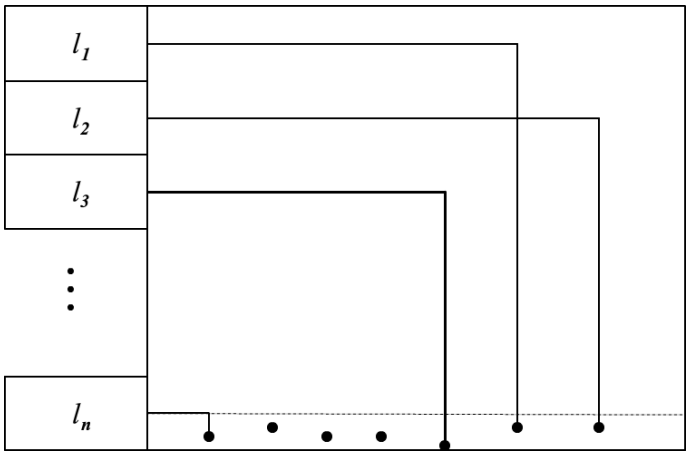


Figure 3.9: A special case of type-*po* leader boundary labeling

In this case, two leaders cross only when the x-coordinate increasing order of any two

doi:10.6342/NTU201600238

features is differnt with the y-coordinate increasing order of their corresponding labels. Recall that in PCM-k, two edges cross only when the order of any two items in π_i is different with the the order of the same two items in π^*. Obviously, the special case of type-po leader simultaneous boundary labeling and PCM-k are equivalent as counting the crossing number. Therefore this problem is NP-complete.

□

Theorem 5. *SLM-po-k ($k \in \mathbb{N}$) can be solved in $O(n^3)$ time.*

Proof. We can use the same method from theorem 3 to solve the problem. In SLM-po-k problem, images $R_1, R_2, ..., R_k$, and Y are given. Note that the total leader length of the horizontal part is fixed for type-po leaders, so the weight of edge (l_i, y_i) is the y-coordinate difference between the associated points of l_i and the port y_i.

□

3.3 type-opo leader

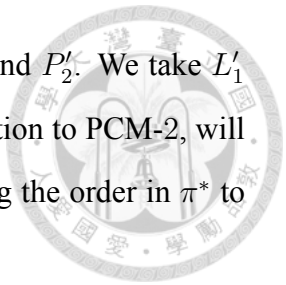
In a boundary labeling for type-opo leaders, we only need to consider the crossings of leaders in tracking route area because all the type-opo leaders go from a feature point through the left borderline of image R orthogonally so that there are no crossings inside image R. The crossing number of type-opo leaders only affected by the y-coordinate of features points. In [16], they provide a leader routing algorithm and prove that two type-opo leaders p_1l_1 and p_2l_2 cross if and only if the y-coordinate of feature points p_1 and p_2 in increasing (or decreasing) order and the y-coordinate of the port of labels l_1 and l_2 in decreasing (or increasing) order.

Hence, we can reduce our problem to PCM-k problem. When $k = 2$, it can be solved by barycenter algorithm in $O(n)$ time. When $k \geq 4$, we can reduce PCM-k to our problem and prove that it is a NP-complete problem.

Theorem 6. *SCM-opo-2 can be solved in $O(n \log n)$ time.*

Proof. First, we show how to reduce our problem to PCM-2. P_1' and P_2' are sorted feature point sets of P_1 and P_2 respectively by y-coordinates of feature points in increasing order.

doi:10.6342/NTU201600238



L'_1 and L'_2 are the ordered sequential of corresponding labels to P'_1 and P'_2. We take L'_1 and L'_2 as inputs π_1 and π_2 of the PCM-2 problem. Then π^*, the solution to PCM-2, will also be the best solution to our problem by assigning labels according the order in π^* to left side from bottom to top.

Second, the barycenter algorithm which is used to solve PCM-2, can also solve our problem in $O(n \log n)$ time.

As a result, the overall algorithm takes $O(n \log n)$ time due to the sorting process at the first step.

□

Theorem 7. *SCM-opo-k is a NP-complete problem when $k \geq 4$.*

Proof. The problem is obviously in NP. Now we show the lower bound. The reduce process of transforming PCM-k to type-*opo* leader simultaneous boundary labeling problem is the same as the process for type-s leader in the theorem 1.

If the order of l_a and l_b in permutation π_i and the order of which in best permutation π^* is different, means the y-coordinate order of l_a and l_b is different with the order of their associated feature points p_a and p_b. This condition must cause a leader crossing in both PCM-k and the type-*opo* simultaneous boundary labeling problem.

□

Theorem 8. *SLM-opo-k ($k \in \mathbb{N}$) can be solved in $O(n^3)$ time.*

Proof. We can use the same method from theorem 3 to solve the problem. In SLM-*opo*-k problem, images $R_1, R_2, ..., R_k$, and Y are given. Note that the total leader length of the horizontal part is fixed for type-*opo* leaders, so the weight of edge (l_i, y_i) is the y-coordinate difference between the associated points of l_i and the port y_i.

Plus, we can prove that the label placement of minimum crossing number does not guarantee the label placement of minimum total leader length and vice versa by contradiction examples. The first example is trivial. It's easy to find an instance that exists multiple label placements which fulfill the requirement of minimum total crossing number, with different total leader length. (As example in Figure 3.11, given $P = \{p_1, p_2, p_3\}$.

doi:10.6342/NTU201600238

$L_1 = (l_1, l_2, l_3)$. $L_2 = (l_3, l_2, l_1)$. The label placement (l_2, l_1, l_3), as shown in the left of the figure, has the minimal crossing number which is three. But its total leader length is bigger than that of the label placement (l_1, l_2, l_3), as shown in the right of the figure, which also has the minimal crossing number.) Therefore, the label placement of minimum total crossing number is not necessarily the one of minimum total leader length. The second example is that assume all feature points have the identical y-coordinate like Figure 3.10. Whatever the order of labels is, the total leader length is the same. Therefore, a label placement which fits the criterion of minimum total leader length does not guarantee the minimum crossing number.

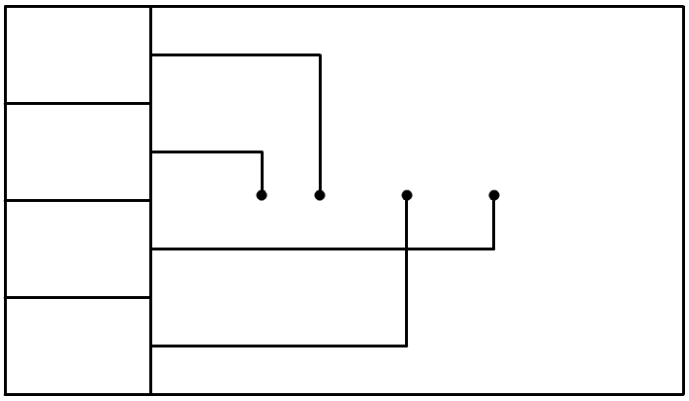


Figure 3.10: Example of type-*opo* label placement with minimal total leader length

□

doi:10.6342/NTU201600238

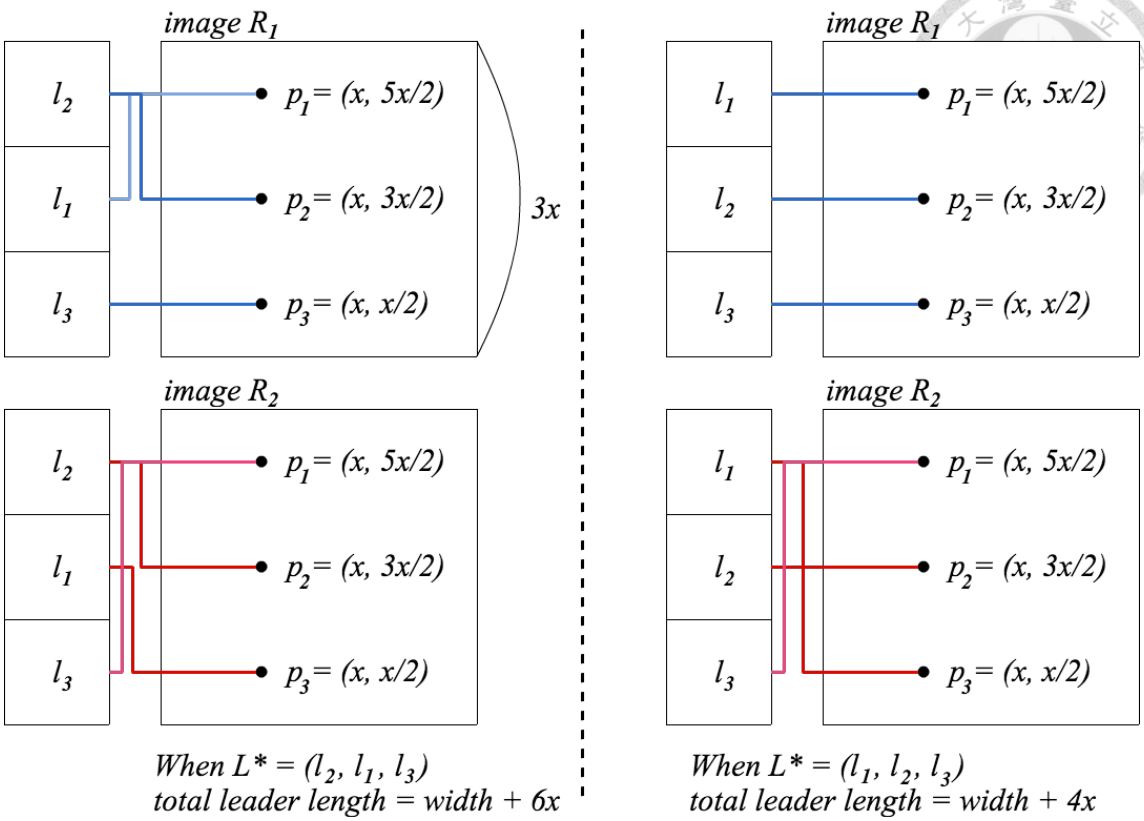


Figure 3.11: Example of type-*opo* label placements with minimal crossing count

doi:10.6342/NTU201600238

Chapter 4

Extension of simultaneous boundary labeling for two images

This is another transformation of $k = 2$ simultaneous boundary labeling, which can be useful in many applications. Given two images R_l and R_r of which the height and width could be different. They share all or parts of the same label set L, but have their own feature point sets P_l and P_r. The labels are placed between images such as Figure 4.1 and Y is a set of legal label positions. This kind of images are widely used in atlases of anatomy and biological structure maps for comparison, such as examples in Figure 4.2.

Our research prove that the extended version of simultaneous boundary labeling problem can be transformed to standard simultaneous boundary labeling problem of two images, and solved by algorithms from previous chapter. Then we show some real-world application results.

doi:10.6342/NTU201600238

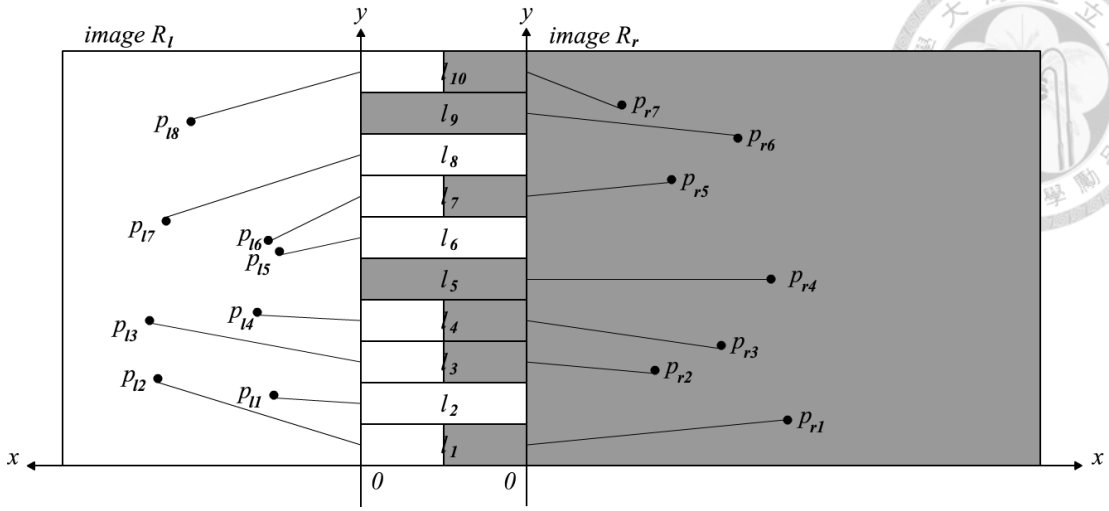
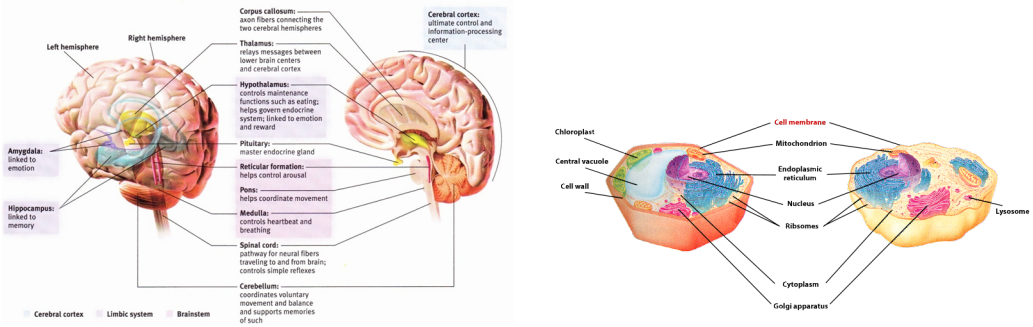


Figure 4.1: Model of the extended version of simultaneous boundary labeling



(a) brain structure (b) cell structure

Figure 4.2: Common usages of extended version of simultaneous boundary labeling

Theorem 9. *Extension-SCM-opo-2 can be solved in* $O(n \log n)$ *time.*

Proof. This problem can be reduced to SCM-*opo*-2 problem, then be solved in $O(n \log n)$ time. In Extension-SCM-*opo*-2 problem, two images R_l, R_r, and Y are given. We transform the image $R_l(R_r)$ to image $R_1(R_2)$ with the coordinate axes illustrated in Figure 4.1. It is obvious that when there is a crossing in image $R_1(R_2)$, there must be a crossing in image $R_l(R_r)$, and vice versa.

We use the picture of cell structural map in Figure 4.2 as an example. Figure 4.3 is the result of extended version of type-*opo* simultaneous boundary labeling. Table 4.1 is the input data of this figure.

doi:10.6342/NTU201600238

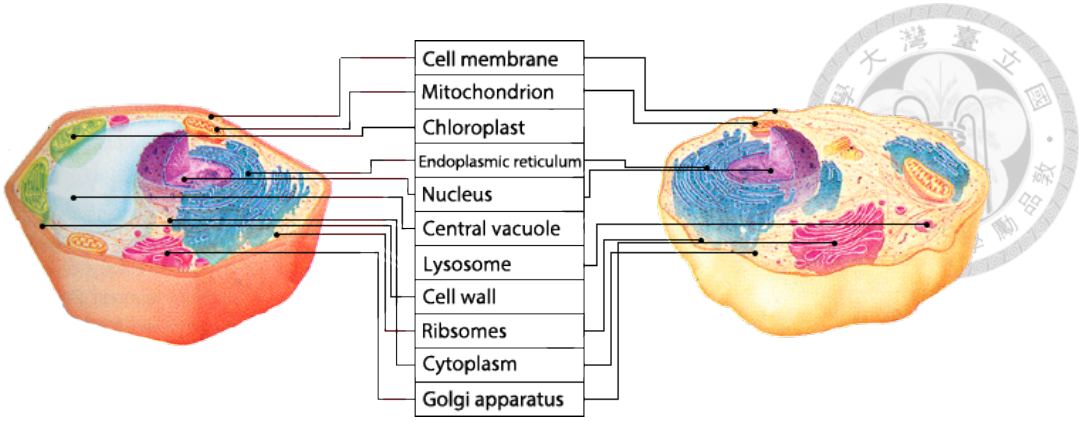


Figure 4.3: Example result of extension SCM-*opo*-2

□

Theorem 10. *Extension-SCM-s-2 and SCM-s-2 are equivalent.*

Proof. This problem can be reduced to SCM-s-2 problem, then the heuristic algorithms for SCM-s-2 problem can be applied on it. In Extension-SCM-s-2 problem, two images R_l, R_r, and Y are given. We transform the image $R_l(R_r)$ to image $R_1(R_2)$ with the coordinate axes illustrated in Figure 4.1. It is obvious that when there is a crossing in image $R_1(R_2)$, there must be a crossing in image $R_l(R_r)$, and vice versa.

We use the picture of cell structural map in Figure 4.2 as an example. Figure 1.6 is the result of extended version of type-s simultaneous boundary labeling. We use the Greedy algorithm and get a non-crossing label placement. □

Theorem 11. *Extension-SCM-po-2 and SCM-po-2 are equivalent.*

Proof. This problem can be reduced to SCM-po-2 problem, then the heuristic algorithms for SCM-po-2 problem can be applied on it. In Extension-SCM-po-2 problem, two images R_l, R_r, and Y are given. We transform the image $R_l(R_r)$ to image $R_1(R_2)$ with the coordinate axes illustrated in Figure 4.1. It is obvious that when there is a crossing in image $R_1(R_2)$, there must be a crossing in image $R_l(R_r)$, and vice versa.

We use the picture of cell structural map in Figure 4.2 as an example. Figure 4.4 and 4.5 are the result of extended version of type-po simultaneous boundary labeling. We use the Enhanced sorting algorithm and get a non-crossing label placement.

doi:10.6342/NTU201600238

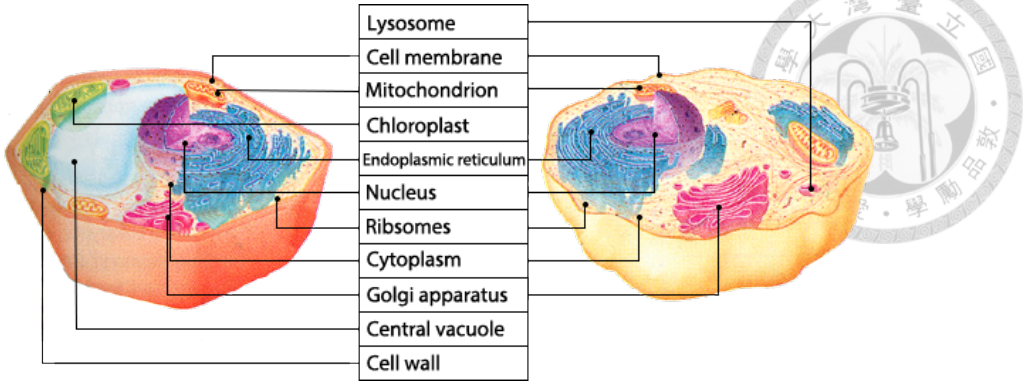


Figure 4.4: Example result of Extension-SCM-*po*-2

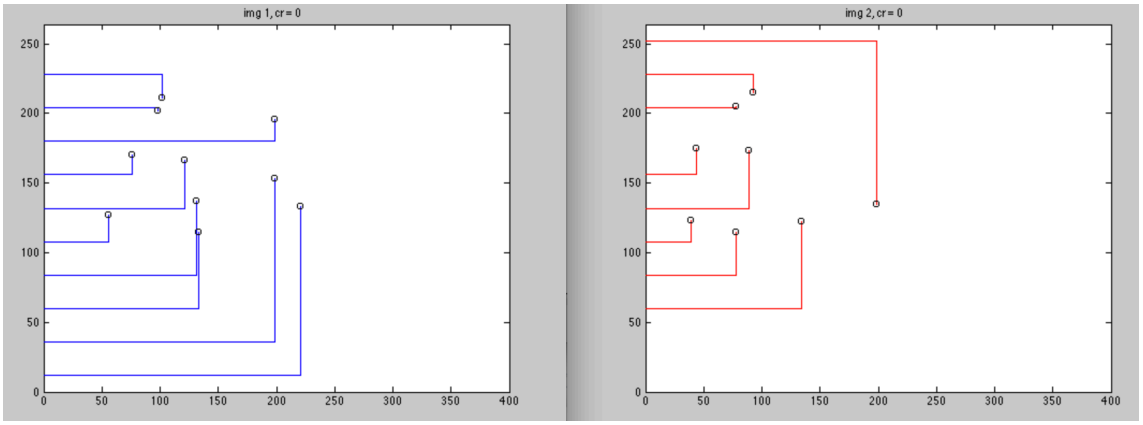


Figure 4.5: Original output of Extension-SCM-*po*-2 and its leader crossing count

□

doi:10.6342/NTU201600238

Table 4.1: The input data of the example of extension-SCM-*opo*-2

(a) Input of R_l

$R_l = (P_l, L_l)$				
P_l	x	y	L_l	label content
p_1	133	115	l_7	golgi apparatus
p_2	56	127	l_5	ribosomes
p_3	221	133	l_{11}	cell wall
p_4	199	153	l_{10}	central vacuole
p_5	131	137	l_6	cytoplasm
p_6	121	166	l_4	nucleus
p_7	76	170	l_3	endoplasmic reticulum
p_8	98	202	l_2	mitochondrion
p_9	102	211	l_1	cell membrane
p_{10}	199	196	l_9	chloroplast

(b) Input of R_r

$R_r = (P_r, L_r)$				
P_r	x	y	L_r	label content
p_{11}	78	115	l_6	cytoplasm
p_{12}	39	123	l_5	ribosomes
p_{13}	134	122	l_7	golgi apparatus
p_{14}	199	135	l_8	lysosome
p_{15}	93	215	l_1	cell membrane
p_{16}	89	173	l_4	nucleus
p_{17}	44	175	l_3	endoplasmic reticulum
p_{18}	78	205	l_2	mitochondrion

doi:10.6342/NTU201600238

Chapter 5

Conclusion and Future Work

In this thesis, we proposed the model of simultaneous boundary labeling for calculating label placements for a series of related images. As listed in Table 1.1, the main contibution of this thesis includes:

1. In Section 3.1, we proved that for type-s simultaneous boundary labeling problem with minimal leader crossing number criterion, given a series of four or more images, the problem is NP-complete. When given a series of two images, we designed five polynomial time heuristic algorithms in which the performance of Greedy-SCM-s-2 algorithm is the best. Also, for the minimal total leader length criterion, we designed an $O(n^3)$ time algorithm to solve the problem.

2. In Section 3.2, we proved that for type-*po* simultaneous boundary labeling problem with minimal leader crossing number criterion, given a series of four or more images, the problem is NP-complete. When given a series of two images, we designed four polynomial time heuristic algorithms in which the performance of SortEnhanced-SCM-*po*-2 algorithm is the best. Also, for the minimal total leader length criterion, we designed an $O(n^3)$ time algorithm to solve the problem.

3. In Section 3.3, we proved that for type-*opo* simultaneous boundary labeling problem with minimal leader crossing number criterion, given a series of four or more images, the problem is NP-complete. When given a series of two images, the Barycenter algorithm can solve the problem in $O(n \log n)$ time. Also, for the minimal total

doi:10.6342/NTU201600238

leader length criterion, we designed an $O(n^3)$ time algorithm to solve the problem.

4. In Chapter 4, we proved that the extended version of simultaneous boundary labeling problem (as show in Figure 1.6) is the same as the standard version of simultaneous boundary labeling problem of two images, whatever the leader type is.

The following are open questions of simultaneous boundary labeling. First, in this thesis, we only proved $k \geq 4$ crossing minimization problem for simultaneous boundary labeling with type-s/po to be NP-complete. The NP-hard proofs or approximation algorithns of $k = 2$ and $k = 3$ are remained to be research. Second, our heuristitc algorithms only been tested under the condition of $k = 2$, a general algorithm for all k crossing minimization problem is still lacking. Third, in boundary labeling problem, the *port* can be divided into two categories, *fixed* and *sliding*. We assumed port to be fixed in this thesis, that is leaders can only be attached to the middle point of the side of labels. The simultaneous boundary labeling problem with sliding ports, that is leaders can be attached to any point of the side of labels are remained to be discussed. Lastly, the simultaneous labeling concept can also be used on map labeling, which is the problem of calculating the label placement to put each label right beside its corresponding feature point in map.

doi:10.6342/NTU201600238

Bibliography

[1] On simultaneous planar graph embeddings. *Computational Geometry*, 36(2):117 – 130, 2007.

[2] M. A. Bekos, M. Kaufmann, M. Nöllenburg, and A. Symvonis. Boundary labeling with octilinear leaders. *Algorithmica*, 57(3):436–461, 2009.

[3] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry*, 36(3):215 – 236, 2007.

[4] M. Benkert, H. J. Haverkort, M. Kroll, and M. Nöllenburg. Algorithms for multi-criteria one-sided boundary labeling. In *Graph Drawing, 15th International Symposium*, GD 2007, pages 243–254, 2007.

[5] T. Biedl, F. J. Brandenburg, and X. Deng. On the complexity of crossings in permutations. *Discrete Mathematics*, 309(7):1813–1823, 2009. 13th International Symposium on Graph Drawing, 2005.

[6] T. Bläsius, S. G. Kobourov, and I. Rutter. Simultaneous embedding of planar graphs. In *Handbook of Graph Drawing and Visualization*, chapter 11. CRC Press, 2013.

[7] C. Demetrescu and I. Finocchi. Breaking cycles for minimizing crossings. *Journal of Experimental Algorithmics*, 6, 2001.

[8] C. Erten, S. G. Kobourov, V. Le, and A. Navabi. *Graph Drawing: 11th International Symposium, GD 2003 Perugia, Italy, September 21-24, 2003 Revised Papers*, chap-

doi:10.6342/NTU201600238

ter Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes, pages 437–449. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[9] A. Gemsa, J.-H. Haunert, and M. Nöllenburg. Boundary-labeling algorithms for panorama images. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 289–298. ACM, 2011.

[10] N. Heinsohn, A. Gerasch, and M. Kaufmann. Boundary labeling methods for dynamic focus regions. In *Visualization Symposium (PacificVis)*, pages 243–247. IEEE, 2014.

[11] Z.-D. Huang, S.-H. Poon, and C.-C. Lin. *Algorithms and Computation: 8th International Workshop, WALCOM 2014, Chennai, India, February 13-15, 2014, Proceedings*, chapter Boundary Labeling with Flexible Label Positions, pages 44–55. Springer International Publishing, Cham, 2014.

[12] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, pages 1–25, 1997.

[13] K. G. Kakoulis and I. G. Tollis. Labeling algorithms. In *Handbook of Graph Drawing and Visualization*, chapter 15. CRC Press, 2013.

[14] P. Kindermann, B. Niedermann, I. Rutter, M. Schaefer, A. Schulz, and A. Wolff. *Algorithms and Data Structures: 13th International Symposium, WADS 2013, London, ON, Canada, August 12-14, 2013. Proceedings*, chapter Two-Sided Boundary Labeling with Adjacent Sides, pages 463–474. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[15] P. Kindermann, B. Niedermann, I. Rutter, M. Schaefer, A. Schulz, and A. Wolff. Multi-sided boundary labeling. *Algorithmica*, pages 1–34, 2015.

[16] K.-C. Lin, H.-J. Kao, and H.-C. Yen. Many-to-one boundary labeling. *Journal of Graph Algorithms and Applications*, 12, 2008.

doi:10.6342/NTU201600238

[17] X. Muñoz, W. Unger, and I. Vrt'o. One sided crossing minimization is np-hard for sparse graphs. In *Graph Drawing*, volume 2265 of *Lecture Notes in Computer Science*, pages 115–123. Springer Berlin Heidelberg, 2002.

[18] M. Nöllenburg, V. Polishchuk, and M. Sysikaski. Dynamic one-sided boundary labeling. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 310–319. ACM, 2010.

[19] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man & Cybernetics*, 11(2):109–125, 1981.

[20] F. Wagner and A. Wolff. A practical map labeling algorithm. *Computational Geometry*, 7(5–6):387 – 404, 1997. 11th ACM Symposium on Computational Geometry.

[21] A. Wolff. Graph drawing and cartography. In *Handbook of Graph Drawing and Visualization*, chapter 23. CRC Press, 2013.

doi:10.6342/NTU201600238