

# 境界線を用いた動的ラベル配置問題

## Dynamic Boundary Label Placement Problem

情報工学専攻 柿沼 亘

KAKINUMA Wataru

### 概要

図やグラフの構成要素に対して、ラベルと呼ばれる文字情報のテキストを配置する問題をラベル配置問題という。本研究では、図を囲む長方形（境界線）の外側にラベルを配置し、図中の点（サイト）とラベルを線（leader）によって結びつける問題を扱う。この問題では、視認性の観点から leader の長さは短いほど良いとされている。本研究では、垂直線と水平線からなる leader を用いて、最も長い leader の長さの最小化を目的とした、境界線の 1 辺に接する任意の位置にラベルを配置する手法を提案する。本提案手法で提案するアルゴリズムは、最適なラベル配置が得られる保証のない発見的解法であるが、計算機実験の結果、多くの入力に対して最適なラベル配置が得られることを示した。

キーワード：ラベル配置, 境界線

## 1 序論

ラベル配置問題は、地図やグラフ等の図を構成する点や線に対して、ラベルと呼ばれる文字情報のテキストを配置する問題である。ラベル配置問題の 1 つとして、境界線を用いたラベル配置問題がある。この問題では、図を囲む長方形（境界線）の外側にラベルを配置し、図中の点（サイト）とラベルを線（leader）によって結び、サイトとラベルを対応付ける。また、サイトとラベルを結ぶ leader は、短いものほど良いとされている。この問題では、用いる leader の種類にもよるが、図の領域外にラベルを配置するだけの十分な領域が存在する限り、点が密集していても全ての点に対してラベルを配置する事が可能である。また、図中にラベルが配置されないため、図自体の視認性が向上する。

境界線を用いたラベル配置問題の既存の研究では、ラベルは予め境界線の外側に接する形で、固定された状態で配置されているため、図中の点とラベルを結ぶ leader を求めることに焦点を当てた手法が一般的であった [1, 2]。しかし、ラベルを境界線の外側に接する任意の位置に配置する事で、ラベルを予め固定した場合に比べ、leader の長さを短く出来る可能性がある。そこで近年研究されているのが、境界線の外側に接する任意の場所にラベルを配置する問題 [3] である。この問題を、境界線を用いた動的ラベル配置問題と呼ぶこととする。また、[1, 2, 3] の手法は、各 leader の長さの総和の最小化を目的としたラベル配置を求める手法である。これらの手法では leader の長さの平均値は最小化されるものの、個々の leader に対しては、とても長い leader が発生してしまう可能性がある。

そこで本研究では、境界線を用いた動的ラベル配置問題に対し、最も長い leader の長さの最小化を目的とした問題に対する発見的解法を提案する。なお、この問題を、Min-Max 型の境界線を用いた動的ラベル配置問題と呼ぶこととする。

## 2 Min-Max 型の境界線を用いた動的ラベル配置問題

ラベルを配置したい点を囲む境界線の外側に接する任意の場所にラベルを配置し、点とラベルを leader で結びつける問題を考える。

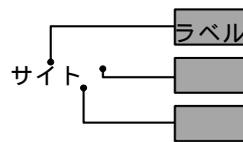


図 1. po-leader

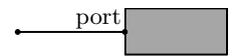


図 2. o-site な leader

### 2.1 用語の定義

本研究で扱う問題を述べるにあたり、いくつか用語を定義する。

- po-leader：垂直線分と水平線分からなり、折れ曲がる回数が高々 1 回である leader (図 1)
- o-site な leader：水平線分のみ leader (図 2)
- port：ラベルと leader が接する点

### 2.2 問題定義

以下の問題について考える。サイトとして  $n$  個の点と、それを囲む任意の境界線が与えられるものとする。ただし、同じ  $x$  座標、または  $y$  座標を持つサイトは存在しないものとする。このとき、ラベルを表す長方形を  $R$  の右側の垂直線分の外側に接する形で、任意の場所に配置し、全てのサイトとラベルを leader で一意に結びつける。なお、用いるラベルの高さ（垂直線分の長さ）は全てのラベルにおいて一定であるとする。leader は po-leader のみを用いる。また、本研究で扱うラベルの port は、図 2 のような、ラベルが境界線  $R$  と接するラベルの 1 辺の midpoint のみとする。ラベルの port の  $y$  座標のことを、ラベルの  $y$  座標と呼ぶ。この問題の目的は、最も長い leader の長さが最小となる legal なラベル配置を見つけることである。legal なラベル配置の条件は以下のとおりである。

- ラベル同士が重なってはならない。
- ラベルは  $R$  の右側の垂直線分の外側に接する形で配置されている。
- leader 同士が交差してはならない。
- leader とサイトが重なってはならない。

上記の legal なラベル配置の条件の 1 つに、leader とサイトの重なり禁止がある。本研究では、leader とサイトの垂直距離が  $\epsilon$  ( $\epsilon$  は定数) 以下になった場合、leader とサイトが重なったとみなす。このとき、 $\epsilon$  の値を誤差値と呼び、パラメータとして扱う。

### 3 提案手法

本研究で提案する, Min-Max 型の境界線を用いた動的ラベル配置問題に対する手法について述べる. まず, 本節で提案する手法における, いくつかの前提について述べる. この手法において,  $n$  個のサイトは,  $y$  座標の降順に予め添え字づけされているものとし,  $s_1, s_2, \dots, s_n$  と表す. 配置する  $n$  個のラベルも予め  $l_1, l_2, \dots, l_n$  と添え字づけされているものとする. これらのラベル  $l_1, l_2, \dots, l_n$  は境界線の 1 辺の任意の位置に配置される.  $l_i$  ( $i \in \{1, 2, \dots, n\}$ ) の port の  $y$  座標を  $l_i^y$  の  $y$  座標とし,  $l_i^y$  と表記する. このとき, ラベル配置において常に  $l_1^y > l_2^y > \dots > l_n^y$  となるようにラベルは配置されるものとする.  $s_{i'}$  ( $i' \in \{1, 2, \dots, n\}$ ) は, ある  $l_i$  ( $i \in \{1, 2, \dots, n\}$ ) と leader で結ぶことを考える. ただし, このサイトとラベルの対応関係は, アルゴリズムの操作の中で変わるものとする.

これらを踏まえた上で, 以下に手法の説明をする. 本節で提案する手法では, 全ての leader の長さの総和の最小化を目的とした動的ラベル配置問題に対する手法 [3] で用いられた, クラスタという概念を用いる. クラスタとは,  $l_i, l_{i+1}, \dots, l_j$  ( $i, j \in \{1, 2, \dots, n\}$ ) において隣り合うラベル同士の距離が 0 となるラベルの集合であり,  $C_{ij}$  と記述する (図 3). このとき,  $C_{ij}$  を構成するラベルの中で最も  $y$  座標の大きいラベル  $l_i$  の上底を  $C_{ij}$  の上底,  $C_{ij}$  を構成するラベルの中で最も  $y$  座標が小さいラベル  $l_j$  の下底を  $C_{ij}$  の下底と呼ぶこととする. そして  $C_{ij}$  の上底の  $y$  座標を  $U_{ij}$ ,  $C_{ij}$  の下底の  $y$  座標を  $L_{ij}$  とする. 以下で述べるアルゴリズム 3.1 では,

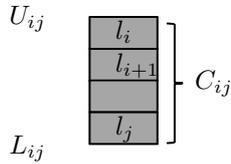


図 3. クラスタ  $C_{ij}$

このクラスタという概念を用いて, 予めどのラベル同士が 1 つのクラスタとなるのかを決定し, その後, そのクラスタの配置位置を決めることで, ラベル配置を行なう. なお, アルゴリズム 3.1 を述べる際に用いている  $\mathcal{C}$  は, クラスタの情報を格納するスタックとする.

#### 3.1 クラスタの決定

アルゴリズム 3.1 の Step 2 (b) では, どのラベル同士が 1 つのクラスタとなるのかを決定している. このとき, Step 2 (a) において, 各サイトに対し o-site な leader を引き配置されたラベルが, 既に配置されているラベルもしくはクラスタと重なったら, それらを 1 つのクラスタと考える. また Step 2 (c) において, クラスタの位置を決定するアルゴリズム 3.2 によって得られたクラスタが, 他のクラスタと重なった場合も, それらを 1 つのクラスタと見なしている. ただし, このような方法では, どのラベル同士が 1 つのクラスタとなるのかを厳密に求めることができない場合もある.

#### 3.2 クラスタの位置の決定

ラベル  $l_i, l_{i+1}, \dots, l_j$  ( $i, j \in \{1, 2, \dots, n\}$ ) がクラスタ  $C_{ij}$  となっていると仮定した場合, クラスタ  $C_{ij}$  をラ

#### アルゴリズム 3.1 : クラスタを用いた手法

Step 1.  $i' = 1, \mathcal{C} = \phi$  とする.

Step 2.  $i' \leq n$  である限り, 以下の操作を繰り返す.

- (a)  $s_{i'}$  に対し,  $s_{i'}^y = l_i^y$  ( $i = i'$ ) となるような位置に  $l_i$  を配置し,  $s_{i'}$  と  $l_i$  を o-site な po-leader で結ぶ.  $\mathcal{C}$  に  $C_{ii}$  を push する.
- (b)  $\mathcal{C}$  に格納された先頭のクラスタを  $C_{ji}$  ( $j \leq i$ ), 先頭から 2 番目のクラスタを  $C_{kj-1}$  ( $k \leq j$ ) とする. ただし,  $C_{kj-1}$  が存在しないならば,  $i' = i' + 1$  とし, Step 2 の先頭へ戻る. もしも  $U_{ji} > L_{kj-1}$  ならば,  $C_{kj-1}$  と  $C_{ji}$  を 1 つのクラスタとし,  $\mathcal{C}$  の  $C_{kj-1}$  と  $C_{ji}$  を  $C_{ki}$  に置き換え, Step 2 (c) へ. もしも  $U_{ji} \leq L_{kj-1}$  ならば,  $i' = i' + 1$  とし, Step 2 の先頭へ戻る.
- (c) アルゴリズム 3.2 を用い  $C_{ki}$  のクラスタの位置を決定し, Step 2 (b) へ.

ベルを配置する  $R$  の 1 辺のどの位置に配置するかによって,  $C_{ij}$  に接続する最も長い leader の長さに変化する.  $C_{ij}$  に接続する最も長い leader の長さを  $M$  とする. 図 4 (a) は, あるクラスタ  $C_{14}$  の配置位置を,  $s_2^y = l_2^y$  となる位置に固定した場合の, サイトとラベルを結ぶ最適な leader を表している (このときの最適な leader とは, クラスタに接続する leader の中で, 最も長い leader の長さが最小となるような, サイトとラベルの対応関係によって得られた leader である). なお, 赤色の leader は最も長い leader を表している. クラスタの配置位置を固定した場合における, サイトとラベルの最適な対応関係は, ラベルが固定されている場合において, 任意に定義された最適化基準に対して, po-leader を用いて最適なラベル配置を求める手法 [2] を用いることで求められる. なお, その際は任意に定義する最適化基準として, 最も長い leader の長さの最小化を目的とした最適化基準を定義する.

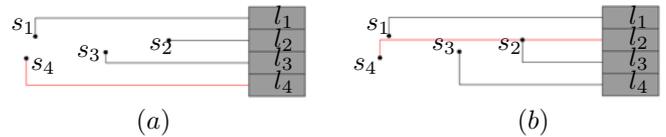


図 4. (a) :  $l_2^y = s_2^y$  となるような位置に配置された  $C_{14}$   
(b) :  $l_2^y = s_2^y + \epsilon$  となるような位置に配置された  $C_{14}$

クラスタの位置を連続的に変化させたときの, クラスタに接続する最も長い leader の長さの値の変化を考える. 図 5 のグラフは, 図 4 (a) の  $C_{14}$  の下底の  $y$  座標  $L_{14}$  を, 徐々に大きくした場合の,  $L_{14}$  の値に対する  $C_{14}$  に接続する最大長の leader の長さ  $M$  のグラフを表しており,  $x$  軸が  $L_{14}$  の値,  $y$  軸が  $M$  の値を表している. 図 4 (a) は, 図 5 のグラフの  $p_1$  における  $C_{14}$  である. また, この  $C_{14}$  に接続する最も長い leader は,  $s_4$  と  $l_4$  を結ぶ leader である. 図 4 (b) は, 図 5 のグラフの  $p_2$  におけるクラスタ  $C_{14}$  であり, このクラ

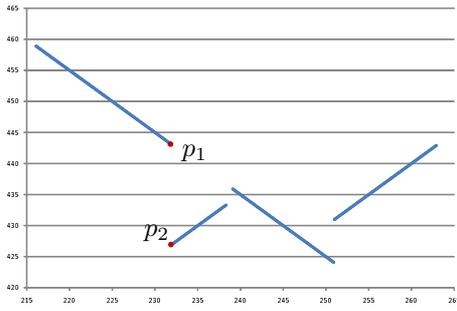


図 5. 図 4 (a) の  $C_{14}$  の下底の  $y$  座標  $L_{14}$  に対する,  $M$  の値を表すグラフ

スタは図 4 (a) における  $C_{14}$  の  $L_{14}$  を  $\epsilon$  だけ大きくした位置に配置されている。このときの  $C_{14}$  に接続する最も長い leader は  $s_4$  と  $l_2$  を結ぶ leader であり、図 4 (a) における最も長い leader よりも短い。このように、サイトとラベルの対応関係が変わるのは、クラスター  $C_{14}$  を構成する各ラベルの  $y$  座標が  $\epsilon$  だけ大きくなり、 $l_2^y = s_2^y + \epsilon$  となったため、 $s_4$  と  $l_2$  を結ぶ場合の leader が  $s_2$  と重なることなく配置できるためである。クラスター  $C_{ij}$  の位置を動かす過程で、そのクラスターに含まれる任意のラベル  $l_k \in \{l_i, l_{i+1}, \dots, l_j\}$  の  $y$  座標  $l_k^y$  が、サイト  $s_m \in \{s_i, s_{i+1}, \dots, s_j\}$  の  $y$  座標  $s_m^y$  に対し、 $l_k^y = s_m^y + \epsilon$  ( $l_k^y = s_m^y - \epsilon$ ) を満たすような位置に  $C_{ij}$  が配置されることによって、 $C_{ij}$  に接続する最適な leader によるサイトとラベルの対応関係が変わる場合がある。その際に、最も長い leader の長さが離散的に小さくなった場合、そのときのクラスターの配置位置は、図 5 のグラフの  $p_1, p_2$  のように、局所的に最適な配置位置であると考えられる。

このような考えを基に、任意の  $C_{ij}$  の位置を決定する方法をアルゴリズム 3.2 に記す。アルゴリズム 3.2 では、まず Step 1 (a) で、 $l_k^y = s_m^y + \epsilon$  ( $l_k^y = s_m^y - \epsilon$ ) となるような位置に  $C_{ij}$  の位置を固定する。そして、Step 1 (b) では、Step 1 (a) で配置位置を固定した  $C_{ij}$  に対し、任意の最適化基準に対する最適なラベル配置を得る手法 [2] を用いて、サイトとラベルの対応関係を決定する。Step 1 (b) で得られたクラスター  $C_{ij}$  において、最も長い leader の長さを  $M$ 、2 番目に長い leader の長さを  $M'$  する (図 6 (a))。Step 1 (c) で、Step 1 (b) で得られた  $C_{ij}$  の位置を  $(M - M')/2$  だけ動かした場合、長さが  $M, M'$  であった leader の長さを、ともに  $(M/2 - M'/2)$  にできる場合 (図 6 (b)) は、Step 1 (b) で得られた  $C_{ij}$  の位置を  $(M - M')/2$  だけ動かす。このような操作を全ての  $k, m \in \{i, i+1, \dots, j\}$  の組み合わせに対して行ない、 $C_{ij}$  の配置位置を決定する。

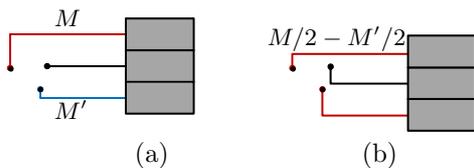


図 6. クラスターの位置を  $(M - M')/2$  だけ動かす場合 (青色の leader の長さは  $M'$ )

### アルゴリズム 3.2 : クラスター $C_{ij}$ の位置の決定

Step 1. 全ての  $s_m$  と  $l_k$  ( $m, k \in \{i, i+1, \dots, j\}$ ) の組合せに対して以下を実行する。

- $C_{ij}$  に含まれるラベル  $l_k$  の  $y$  座標  $l_k^y$  が、 $l_k^y = s_m^y + \epsilon$  ( $l_k^y = s_m^y - \epsilon$ ) となるような位置にクラスターを固定する。
- 固定されたラベルに対して最適なラベル配置を求める手法 [2] を用い、(a) で位置を固定した  $C_{ij}$  に対する最適な leader を求める。
- (b) の実行後、 $C_{ij}$  に接続する最も長い leader の長さを  $M$ 、その次に長い leader の長さを  $M'$  とする。このとき、クラスター  $C_{ij}$  を上(下)に  $(M - M')/2$  だけ動かして固定する。この固定された  $C_{ij}$  に対して、[2] の手法を用い、 $C_{ij}$  に対する最適な leader を求める。
- (b), (c) で得られた 2 つの  $C_{ij}$  の中で、 $C_{ij}$  に接続する最も長い leader の長さが短い方の  $C_{ij}$  を  $\mathcal{L}_{mk}$  ( $m, k \in \{i, i+1, \dots, j\}$ ) とする。

Step 2. 全ての  $\mathcal{L}_{mk}$  の中で、最も長い leader の長さが 1 番短い  $\mathcal{L}_{mk}$  を出力する。

## 4 計算機実験

Min-Max 型の境界線を用いた動的ラベル配置問題に対して、混合整数計画問題 (以降 MIP) による定式化を用いて計算機実験を行なった結果と、提案手法を用いて計算機実験を行なった結果を比較する。実験環境は以下のとおりである。サイトとなる点の入力は、

### 実験環境 (MIP)

- OS : Windows Vista Business
- CPU : Intel Xeon X5550 (2.66 GHz  $\times$  2)
- Memory : 6.00 GB
- 使用ソフト : NUOPT 13.0.0

### 実験環境 (提案手法)

- OS : Windows 7 Professional
- CPU : Intel Core i5 (2.67 GHz)
- Memory : 4.00 GB

横 400  $\times$  縦 600 の範囲内に乱数を用いて生成した。ただし、同じ  $x$  座標もしくは  $y$  座標値を持つサイトは存在しないように調整を行なった。

## 実験結果

MIP で得られたラベル配置を最適なラベル配置と考え、最適なラベル配置における最も長い leader の長さの値を OPT と表す。また、提案手法によって得られたラベル配置において最も長い leader の長さの値を Val と表す。このとき、 $(OPT \div Val) \times 100$  で表される値を定義し Gap と呼ぶこととする。この値が 0% に近いほど、提案手法で得られたラベル配置が、最適なラベル配置に近いことを示している。

表 1. 100 個の入力に対する実行結果の比較

$n, h, \text{回数}$	MIP		提案手法	
	OPT	time [s]	Gap [%]	time [s]
10, 40, 100	377.32	0.66 (23.34)	0.02 (1.33)	0.01 (0.09)
15, 30, 100	397.23	71.89 (1622)	0 (0)	0.34 (1.38)
20, 20, 82	368.57	349.11 (3260)	0 (0)	0.54 (2.04)
25, 15, 49	386.44	388.80 (3132)	0 (0)	1.29 (11.72)

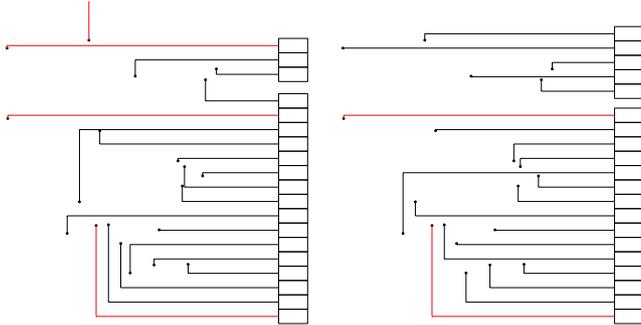


図 7. ラベル配置 (MIP) 図 8. ラベル配置 (提案手法)

表 1 は、各サイト数の入力に対する、それぞれの手法の実行結果を表している。この表の 1 列目における  $n$  はサイト数、 $h$  はラベルの高さである。また“回数”の項目は、各  $n, h$  の値に対し、MIP を用いて 100 回の計算機実験を行なった結果、3600 秒以内に最適なラベル配置が得られた回数を表している。この複数回の MIP を用いた計算機実験で得られたラベル配置における OPT の値と計算時間の平均を記したのが表 1 の 2, 3 列目である。また、MIP を用いて 3600 秒以内に最適なラベル配置を得られたインスタンスに対して提案手法を実行する。その際の、Gap と計算時間の平均を示したのが表 1 の 4, 5 列目である。また、() で記した値は、各項目における最大値を表している。図 7, 8 は、サイト数が 20 の入力に対する実行結果の例である。この入力では、提案手法によって最適なラベル配置が得られている。

表 1 で示した、 $n = 10, h = 40$  の場合における 100 回の計算機実験では、提案手法を用いた場合において、97 回の実験で最適なラベル配置を得ることができた。このとき提案手法によって最適なラベル配置を得られなかった 3 つ実行結果では、提案手法によって得られるラベル配置において各ラベルが構成するクラスタが、最適なラベル配置における各ラベルが構成するクラスタと一致していなかった。図 9 の最適なラベル配置に対して、図 10 のように全てのラベルを 1 つのクラスタとして考えた場合は、最適なラベル配置を得ることはできない。これは、アルゴリズム 3.1 の Step 2 において、どのラベル同士を 1 つのクラスタとするかを厳密に求められない場合があるためだと考えられる。一方、計算機実験の結果、最適なラベル配置において、配置する全てのラベル  $l_1, l_2, \dots, l_n$  が 1 つのクラスタと

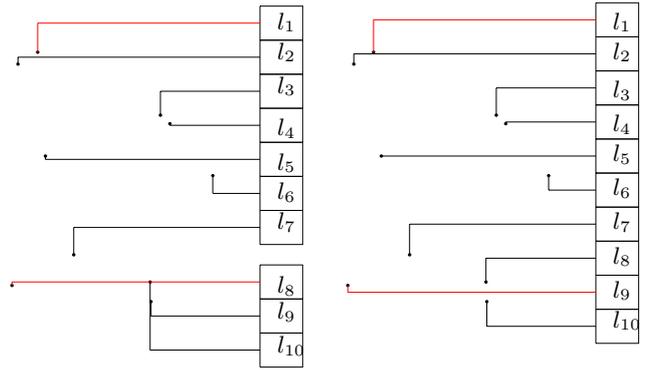


図 9. 最適なラベル配置 (C<sub>17</sub>, C<sub>810</sub>) 図 10. 案手法の実行結果 (C<sub>110</sub>)

なっている場合は、提案手法によって最適なラベル配置が得られていた。

## 5 結論

本研究では、po-leader と呼ばれる垂直線と水平線からなる leader を用いて、最も長い leader の長さの最小化を目的とした、境界線の 1 辺の外側の任意の位置にラベルを配置する問題に対する手法を提案した。本提案手法を実装し、ランダムに発生させたサイトの入力に対して計算機実験を行なった結果、殆どの入力に対して最適なラベル配置を得ることが出来た。特に、最適なラベル配置において各ラベルが構成するクラスタの数が高々 1 (配置する全てのラベル  $l_1, l_2, \dots, l_n$  が 1 つのクラスタとなる) となるような入力において、最適なラベル配置を得ることができた。

今後の課題は、Min-Max 型の境界線を用いた動的ラベル配置問題の計算複雑性を明らかにすることである。最適なラベル配置において、クラスタの数が 1 つとなるようなサイトの入力に対しては、提案手法を用いることで、最適なラベル配置を得られる可能性があると考えられる。まず、このような特殊な入力に対して最適なラベル配置が得られるかどうかを議論することも必要であると考えられる。

## 謝辞

本研究を進めるにあたり、適切な御指導、御指摘をいただきました中央大学理工学部情報工学科の今井桂子教授に心から感謝致します。また、多くの助言をいただいた今井研究室の学生各位に感謝致します。

## 参考文献

- [1] Michael. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff, “Boundary labeling: Models and efficient algorithms for rectangular maps,” *Computational Geometry Theory and Applications*, vol. 36, pp. 215–236, 2007.
- [2] M. Benkert, H. Haverkort, M. Kroll, and M. Nöllenburg, “Algorithms for multi-criteria bound-ary labeling,” *Journal of Graph Algorithms and Applications*, vol. 13(3), pp. 289–317, 2009.
- [3] Martin. Nöllenburg, V. Polishchuk, and M. Sysikaski, “Dynamic one-sided boundary labeling,” *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 310–319, 2010.