

G-109

地図ラベル配置問題に対する厳密解法の提案 An exact algorithm for the map labeling problem

城戸 直人[†]
Naoto Kido

増田 澄男[†]
Sumio Masuda

山口 一章[†]
Kazuaki Yamaguchi

1. まえがき

地図ラベル配置問題とは、地図上の各地点（サイト）に、地名などを表す文字列（ラベル）を適切に配置する問題である。この問題は、地図の作成、地理情報システムなどに応用をもち、グラフ描画へのラベル配置とも深く関連している。そのため、この問題に関してこれまでに多くの研究が行われてきている [1]。地図ラベル配置問題は、ラベル同士の重なりを禁止するかどうかなどにより、いくつかの種類に分かれる。本研究では、ラベル同士及びラベルとサイトとの重なりを禁止した上で、配置ラベルの個数を最大化する問題を扱う。この問題はラベル数最大化問題と呼ばれる。

ラベル数最大化問題に対して、ラベルを配置できる位置をいくつかの離散的な箇所に限定するという離散モデルを用いる方法（文献 [2] など）と、定められた範囲内であればサイトに接する任意の位置にラベルを配置できるとするスライダモデルを用いる方法（文献 [3] など）が存在する。明らかに、離散モデルを用いるよりもスライダモデルを用いたほうが、ラベルの配置位置の自由度が大きくなるため、結果としてより高いラベル配置率の達成につながる。スライダモデルにもいくつかの種類があるが、本研究ではその中でも最もラベル配置位置の自由度が大きい 4 スライダモデルを用いる。

ラベル数最大化問題は一般に NP 困難であることが知られている [1]。4 スライダモデルを用いた方法に限定した場合でも、いくつかの発見的手法（文献 [3],[4] など）と、厳密解法である Klau らの手法 [5] が存在する。

文献 [5] では、制約グラフと呼ばれる辺が重みをもつ有向グラフを作成し、それを用いて、問題を 0-1 整数計画問題に帰着している。この整数計画問題では、制約グラフ中に存在する重みが正の有向閉路ごとに制約式を設けるものとしているが、そのような閉路は非常に多く存在する可能性がある。そこで、文献 [5] では、制約式を最初は制限して作っておき、その後徐々に増やしていくという実装を行っている。しかしこのアプローチでは、一般に、整数計画問題を何度も繰り返して解く必要があ

る。また、Klau らの定式化では配置ラベルの位置を表す変数を設けていないため、0-1 整数計画問題の最適解を得た後に、最小コストフロー問題を解くことによって、各ラベルの配置位置を求めている。このように、Klau らの手法はやや複雑であり、実装が容易ではない。

本研究では、4 スライダモデルを用いた場合のラベル数最大化問題に対して、実装がより容易な厳密解法を提案する。提案手法は、問題を混合整数線形計画問題に帰着させるものであり、混合整数線形計画問題を 1 回解くだけで、各ラベルの配置位置まで求めることができる。

以下、2. では、本研究で扱うラベル数最大化問題についてより詳しく述べる。3. では提案手法について説明する。提案手法は指数時間アルゴリズムであるため、本研究では高速化の方法について検討中である。4. ではその方法の現状について簡単に説明する。

2. ラベル数最大化問題

本章では、4 スライダモデルを用いた場合のラベル数最大化問題について説明する。軸平行長方形領域 D と、サイトと呼ばれる有限個の点の集合 Φ が与えられる。サイトはすべて D の内部に含まれており、サイトごとにラベルの高さと幅が指定されるものとする。

各サイトには、ラベルを、その外周がサイトに接するようにして、たかだか 1 箇所に配置することができる。その際、各ラベルは、サイト、他のラベルや D の外周と重なってはならない。本研究で扱うラベル数最大化問題は、これらの条件の下で、できるだけ多くのラベルを配置する問題である。

3. 提案手法

本章では、提案手法について説明する。提案手法は、ラベル数最大化問題を混合整数線形計画問題 (MILP) に帰着して、数値計画ソルバーにより最適解を求めるものである。

3.1 定数と変数の定義

まず、MILP で用いる定数の定義を行う。各サイト $s \in \Phi$ に対して、その座標を (X_s, Y_s) とし、 s に配置するラベルの幅と高さをそれぞれ W_s, H_s とする。また L, R, B, T

[†] 神戸大学, Kobe University

を定数として、領域 D は $L \leq x \leq R, B \leq y \leq T$ を満たす領域とする。

次に、変数の定義を行う。各サイト $s \in \Phi$ にラベルを配置するかどうかを表す 0-1 変数として、 b_s を定義する。 $b_s = 1$ のときかつそのときに限り、 s にラベルが配置される。サイト s のラベルの左下の頂点を ll_s とし、 ll_s の x 座標を示す変数 x_s と y 座標を示す変数 y_s を作る。図 1 参照。

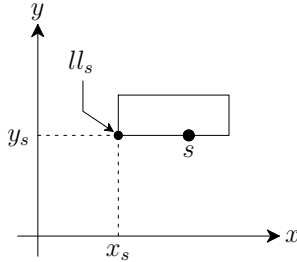


図 1 サイト s のラベルの左下の頂点 ll_s

以上に述べたもの以外にも、次節で定義するいくつかの定数及び変数を用いる。

3.2 目的関数と制約式

ラベル数最大化問題を MILP に帰着する方法について述べる。まず、目的関数は次式 (1) で表される。

$$\text{maximize } \sum_{s \in \Phi} b_s \quad (1)$$

以下、制約式について説明する。各ラベルは、その外周が対応するサイトに接するように、かつ D からはみ出さないように配置する必要がある。そのために、まず以下の制約式を設ける。

$$X_s - W_s \leq x_s \leq X_s \quad (2)$$

$$Y_s - H_s \leq y_s \leq Y_s \quad (3)$$

これらの式は、各ラベルの ll_s の位置を、図 2 に示すような長方形領域 $rect_s$ の内部もしくは外周上に制限している。

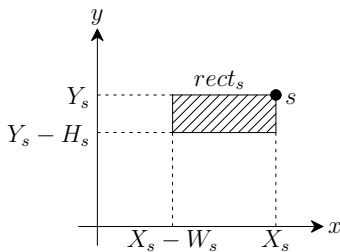


図 2 領域 $rect_s$

次に、 D からはみ出さないようにラベルを配置するために、以下の制約式を設ける。

$$X_s + (L - X_s)b_s \leq x_s \leq X_s + (R - W_s - X_s)b_s \quad (4)$$

$$Y_s + (B - Y_s)b_s \leq y_s \leq Y_s + (T - H_s - Y_s)b_s \quad (5)$$

式 (4), (5) は、 $b_s = 1$ のとき

$$L \leq x_s \leq R - W_s \quad (6)$$

$$B \leq y_s \leq T - H_s \quad (7)$$

となる。これらの式によって、各ラベルの ll_s の位置は、図 3 に 2 重線で示した領域の内部もしくは外周上に制限される。これにより、ラベルが D からはみ出して配置されることを防いでいる。 $b_s = 0$ のときについては後述する。

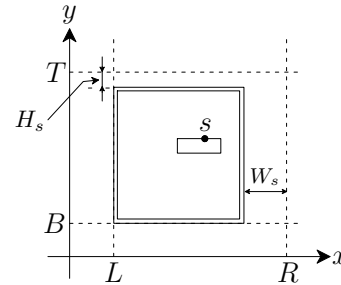


図 3 D に対する ll_s の位置の制限

各ラベルが対応するサイトに接するように配置するために、 ll_s を $rect_s$ の外周上に置く必要がある。これを表現する制約式が以下の 5 つである。

$$x_s - M\alpha_s^1 \leq X_s - W_s \quad (8)$$

$$x_s + M\alpha_s^2 \geq X_s \quad (9)$$

$$y_s - M'\alpha_s^3 \leq Y_s - H_s \quad (10)$$

$$y_s + M'\alpha_s^4 \geq Y_s \quad (11)$$

$$\alpha_s^1 + \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \leq 4 - b_s \quad (12)$$

M, M' は十分大きい定数で、 $\alpha_s^1, \alpha_s^2, \alpha_s^3, \alpha_s^4$ はすべて 0-1 変数である。 $b_s = 0$ のときは、 $\alpha_s^1, \alpha_s^2, \alpha_s^3, \alpha_s^4$ の値に関わらず式 (12) は常に成り立つ。また、 $\alpha_s^1 = \alpha_s^2 = \alpha_s^3 = \alpha_s^4 = 1$ とすれば、式 (8) ~ (11) は常に成り立つ。よって、 $b_s = 0$ のときはこれらの制約式は何の影響も及ぼさない。

一方、 $b_s = 1$ のときは、式 (12) より、 $\alpha_s^1, \alpha_s^2, \alpha_s^3, \alpha_s^4$ のうちの少なくとも一つが 0 にならなければならない。例えば、 $\alpha_s^1 = 0$ のとき、式 (8) は $x_s \leq X_s - W_s$ となる。この式と式 (2) より $x_s = X_s - W_s$ となる。また式 (3) より $Y_s - H_s \leq y_s \leq Y_s$ であるから、 $\alpha_s^1 = 0$ のとき、 ll_s の位置が領域 $rect_s$ の左辺上に制限されることが分かる。同様に、 $\alpha_s^2 = 0$ のときは $rect_s$ の右辺上に、 $\alpha_s^3 = 0$ のときは下辺上に、 $\alpha_s^4 = 0$ のときは上辺上に、 ll_s の位置が制限される。よって $b_s = 1$ のときは、式 (8) ~ (12) により、 ll_s の位置を $rect_s$ の外周上に制限することができている。

ラベル同士の重なり、ラベルとサイトの重なりはどちらも避けなければならない。相異なる二つのサイト s, t

について考える．一般性を失うことなく $X_s \geq X_t$ と仮定し，まず $Y_s > Y_t$ であるものとする．このとき，二つのラベル同士が重なる必要十分条件は， $x_s < x_t + W_t$ かつ $y_s < y_t + H_t$ となることである．これをもとに，サイトの組 (s, t) に対して，ラベル同士，及びラベルとサイトの重なりを避けるための制約式を以下のように立式する．

$$x_s - x_t + N\beta_{st} \geq W_t b_t \quad (13)$$

$$y_s - y_t + N'\gamma_{st} \geq H_t b_t \quad (14)$$

$$\beta_{st} + \gamma_{st} \leq 1 \quad (15)$$

N, N' は十分大きな定数で， β_{st}, γ_{st} は 0-1 変数である． $b_t = 1$ のときは，この式は上記のラベル同士が重ならない条件を表している．

$b_t = 0$ のときは，式 (4), (5) より以下ようになる．

$$X_t \leq x_t \leq X_t \quad (16)$$

$$Y_t \leq y_t \leq Y_t \quad (17)$$

また，式 (13), (14) の t のラベルの横幅，高さを表す右辺は 0 になる． l_t がサイト上にあり，横幅，高さがともに 0 のラベルは，サイトそのものである．よって $b_t = 0$ のときは，式 (13)~(15) は t のラベルとの重なりではなく， t との重なりを回避させる． $b_s = 0$ のときも同様のことがいえる．以上より，式 (13)~(15) によってラベル同士の重なりだけでなく，ラベルとサイトの重なりも同時に回避できている．

$X_s \geq X_t$ かつ $Y_s \leq Y_t$ のときには，式 (14) の代わりに次式を作ればよい．

$$y_t - y_s + N'\gamma_{st} \geq H_s b_s \quad (18)$$

式 (13)，式 (14) あるいは (18)，及び式 (15) は， $X_s \geq X_t$ を満たすすべてのサイトの組み合わせ (s, t) に対して作る必要はなく，対応するラベル同士が重なる可能性のある場合のみ設ければよい．

3.3 MILP の全体の記述

MILP の目的関数及び制約式を以下にまとめておく．定数 M, M', N, N' の値は，浮動小数点の誤差を考えなければそれぞれ

$$M \geq \max_{s \in \Phi} \{W_s\} \quad (19)$$

$$M' \geq \max_{s \in \Phi} \{H_s\} \quad (20)$$

$$N \geq \max_{s, t \in \Phi} \{W_s + W_t\} \quad (21)$$

$$N' \geq \max_{s, t \in \Phi} \{H_s + H_t\} \quad (22)$$

とすればよい．

maximize

$$\sum_{s \in \Phi} b_s$$

subject to

$$X_s - W_s \leq x_s \leq X_s$$

$$Y_s - H_s \leq y_s \leq Y_s$$

$$(\forall s \in \Phi)$$

$$X_s + (L - X_s)b_s \leq x_s \leq X_s + (R - W_s - X_s)b_s$$

$$Y_s + (B - Y_s)b_s \leq y_s \leq Y_s + (T - H_s - Y_s)b_s$$

$$(\forall s \in \Phi)$$

$$x_s - M\alpha_s^1 \leq X_s - W_s$$

$$x_s + M\alpha_s^2 \geq X_s$$

$$y_s - M'\alpha_s^3 \leq Y_s - H_s$$

$$y_s + M'\alpha_s^4 \geq Y_s$$

$$\alpha_s^1 + \alpha_s^2 + \alpha_s^3 + \alpha_s^4 \leq 4 - b_s$$

$$(\forall s \in \Phi)$$

$$x_s - x_t + N\beta_{st} \geq W_t b_t$$

$$y_s - y_t + N'\gamma_{st} \geq H_t b_t \text{ (if } Y_s > Y_t)$$

$$y_t - y_s + N'\gamma_{st} \geq H_s b_s \text{ (if } Y_s \leq Y_t)$$

$$\beta_{st} + \gamma_{st} \leq 1$$

$$(\forall s, t \in \Phi \text{ such that } s \neq t \text{ and } X_s \geq X_t)$$

4. アルゴリズムの高速化

前章で示した提案手法は指数時間アルゴリズムであるため，本研究では，高速化の方法についても検討している．以下 4.1 において，その方法について説明する．4.2 では，計算機実験により，現時点で実装ができていない部分だけでも実行時間をかなり短縮できることを示す．

4.1 高速化の方法

4.1.1 概要

ラベル数最大化問題の入力が与えられたとき，前章で説明した提案手法を直ちに実行するのではなく，前処理として以下の Step1, 2 を実行する．

Step 1: 以下の条件 (*) を満たすようなサイトの集合 S_1, S_2 と， S_1 に属する各サイト s に対するラベル位置 $p(s)$ を求める．

(*) S_1 の各サイト s に対しては $p(s)$ にラベルが配置されており， S_2 に属するサイトに対しては全くラベルが配置されていないような最適解が存在する．

Step 2: S_1, S_2 以外のサイトに対するラベル配置問題を、いくつかの部分問題に分割する。

Step 1, 2 の実行終了後、Step 2 で生成した各部分問題に対して、数理計画ソルバーを用いて最適解を求める。このとき、それらの最適解における配置ラベルと、集合 S_1 に属する各サイトのラベルを合わせたものが、元の問題の最適解を構成する。

以下、各ステップについて説明する。

4.1.2 Step 1

本研究では Step 1 を実行するために、寺脇 [4] の発見的手法（以下、寺脇法と呼ぶ）の一部を利用する。寺脇法は、フェーズ 1、フェーズ 2、後処理という三つの部分よりなるアルゴリズムである。これらのうち本研究で利用するのはフェーズ 1 のみであるので、以下ではこのフェーズについてのみ説明する。

寺脇法のフェーズ 1 では、まず、各サイト s の上下左右に、ラベルを配置できる可能性のある領域（スリットと呼ぶ）を一つずつ作成する。上下スリットの幅は $2W_s$ 、高さは H_s であり、その内部でラベル位置を左右に移動することができる（図 4(a) 参照）。一方、左右スリットの幅は W_s 、高さは $2H_s$ であり、その内部でラベル位置を上下に移動することができる（図 4(b) 参照）。

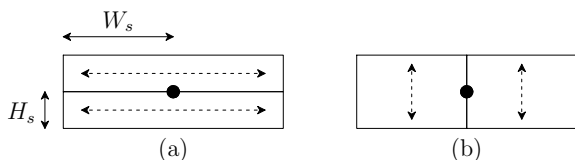


図 4 スリットの例

作成した各スリットに対し、他のサイトや領域 D の外周に重なっていて、ラベルを置くことができない部分があれば、そのような部分を縮小する。その結果、ラベルサイズよりも小さいスリットができれば除去する。

この処理の後、6 種類のルール R1~R6 を繰り返し適用していく。これらのルールは、それぞれの適用条件を満たすスリットが存在する場合に、その内部でのラベル位置の決定やそのスリットの縮小あるいは除去を行うものである。あるサイト s にラベルを配置したときには、 s のすべてのスリットを除去する。また、配置したラベルに重なるスリットは、その部分を縮小するものとし、その結果、ラベルサイズよりも小さいスリットができれば除去する。寺脇法のフェーズ 1 は、どのルールも適用できなくなった時点で終了する。

Step 1 では、最初に S_1, S_2 を空集合とした後、寺脇法のフェーズ 1 を実行する。そして、あるサイト s のラベル位置を決定したときには、その位置を $p(s)$ として記録し、 s を集合 S_1 に加える。フェーズ 1 終了時点でスリッ

トが残っていない各スリット s については、それが集合 S_1 の要素でなければ、集合 S_2 に加える。このようにして集合 S_1, S_2 及び S_1 の各サイト s のラベル位置 $p(s)$ を定めるとき、前述の条件 (*) が成立する [4]。

4.1.3 Step 2

ここでは、Step 2 における部分問題の生成方法について説明する。まず、各サイト $s \in \Phi - (S_1 \cup S_2)$ に対して、Step 1 のように s の上下左右にスリットを一つずつ作成し、領域 D の外周に重なっている部分を縮小する。その結果、ラベルサイズよりも小さいスリットができれば除去する。このとき、Step 1 とは異なり、他のサイトとの重なりは考慮しないことに注意する。次に、各サイト $s \in \Phi$ に対一に対応して頂点を作り、グラフ G を作成する。最初は、 G の辺集合 $E(G)$ を空集合としておく。

任意のサイト $s \in \Phi - (S_1 \cup S_2), t \in S_1 \cup S_2$ に対して、 s のスリットと、 t に配置されるラベルまたは t が重なっていれば、その部分だけ s のスリットを縮小する。そして、 s と t に対応する頂点間を結び辺を $E(G)$ に加える。スリットを縮小した結果、ラベルサイズよりも小さいスリットができれば除去する。次に、任意のサイト $s, t \in \Phi - (S_1 \cup S_2) (s \neq t)$ に対して、 s のスリットと t のスリットが重なっていれば、 s と t に対応する頂点間を結び辺を $E(G)$ に加える。 G に辺を加える処理は以上で終了である。

$\Phi - (S_1 \cup S_2)$ のサイトに対応する頂点の集合から誘導される G の部分グラフを考え、その連結成分を C_1, C_2, \dots, C_m とする。このとき、一つの連結成分が一つの部分問題に対応する。

$i = 1, 2, \dots, m$ に対して、 C_i に対応する部分問題は以下のように構成する。 C_i に含まれる各頂点に対応するサイトの集合を Φ とする。そして、3. で述べたように MILP を構成する。ただし、ラベル同士、ラベルとサイトの重なりを禁止するための制約式に関しては、特別に考慮する必要がある。まず、 C_i において辺が存在する頂点間に対応するサイト同士に対してのみ立式する。次に、 G において、 C_i の頂点（対応するサイトを s とする）と、 $t \in S_1 \cup S_2$ に対応する頂点との間に辺が存在すれば、そのような s, t に対して立式する。 $t \in S_1$ であれば t のラベルの配置位置 $p(t)$ が、 $t \in S_2$ であれば t のラベルを配置しないことが、それぞれ確定している。よって、制約式中の t に関する変数を定数に置き換えることができる。具体的には、 $t \in S_1$ のときは、 ll_t の座標を (X, Y) とおくと、 $x_t = X, y_t = Y, b_t = 1$ と置き換え、 $t \in S_2$ のときは、 $x_t = X_t, y_t = Y_t, b_t = 0$ と置き換える。

$i = 1, 2, \dots, m$ に対して、以上の手順で生成した MILP の最適解を数理計画ソルバーを用いて求める。このとき、前述のとおり、それらの最適解における配置ラベルと、

集合 S_1 に属する各サイトのラベルを合わせたものが、元の問題の最適解を構成する。

4.2 計算機実験

4.1 で述べた前処理が、実行時間の削減に有効であることを確かめるために、計算機実験を行った。サイト数 800、領域 D の幅 792、高さ 612 として、 D の内部に各サイトを一様ランダムに配置した。各ラベルの幅と高さは、それぞれ [27, 33], [6, 8] の範囲内の一様ランダムな整数値とした。このようなデータを 20 個用意して、それぞれに対して前処理を行わない提案法と、前処理を行う提案法を実行し、その実行時間の比較を行った。

前述のとおり、前処理の Step 1 では、寺脇法の一部を利用した。寺脇法は 6 種類のルール R1~R6 を用いているが、本研究ではこれまでに R1, R2 及び R5 の特別な場合である R5' を実装しており、計算機実験でもこれらの 3 種類のルールのみを利用している。これらのルールについては付録で説明する。

実験に使用した計算機の CPU は Intel Core i5-4460, 3.20GHz、メモリは 16GB、OS は Windows10、プログラミング言語は Java8.0 である。また、提案法における MILP のソルバーとして、CPLEX12.8.0 [6] を用いた。

実験結果を表 1 に示す。表中の非高速化、高速化は、それぞれ前処理を行わない提案法、前処理を行う提案法の実行時間を表す。比率はそれらの比であり、この値が小さいほど前処理の効果が大きい。表 1 から、どのデー

表 1 実行時間の比較

No.	非高速化 [s]	高速化 [s]	比率 [%]
1	34.93	0.36	1.03
2	13.21	0.30	2.28
3	8.09	0.29	3.59
4	3.48	0.24	6.96
5	11.22	0.65	5.82
6	9.71	0.21	2.13
7	3.05	0.29	9.45
8	8.71	0.63	7.20
9	7.44	0.27	3.61
10	9.39	0.24	2.51
11	5.66	0.26	4.54
12	15.53	0.40	2.57
13	7.77	0.33	4.20
14	5.34	0.33	6.16
15	20.27	0.30	1.47
16	5.05	0.25	4.98
17	8.75	0.20	2.27
18	8.02	0.59	7.39
19	6.16	0.27	4.40
20	6.42	0.36	5.60

タに対しても実行時間を 1/10 以下に短縮できており、前処理が実行時間削減に有効であることが分かる。

提案法（前処理あり）により得られたラベル配置の例を図 5 に示す。この例では、800 個のサイトのうちの 790 個にラベル配置をすることができている。前処理を実行しなかったときの実行時間は 34.93s であったが、前処理により実行時間は 0.36s まで短縮された。

今後、R1, R2, R5' 以外のルールも実装していきたいと考えている。これにより、前処理の Step 1 終了時での $|S_1 \cup S_2|$ の値が増加することが期待できるため、実行時間のさらなる削減につながる可能性がある。

参考文献

- [1] G. Neyer, “Map labeling with applications to graph drawing,” Drawing Graphs, Lecture Notes in Computer Science, vol.2025, pp.247-273, Springer, Berlin, 2001.
- [2] F. Wagner, A. Wolff, V. Kapoor and T. Strijk, “Three rules suffice for good label placement,” Algorithmica, vol.30, no.2, pp.334-349, 2001.
- [3] M. van Kreveld, T. Strijk, and A. Wolff, “Point set labeling with sliding labels,” Proc. 14th Annual Symp. on Computational Geometry, pp.337-346, Minneapolis, 1998.
- [4] 寺脇宏高, 4 スライダーモデルとルール処理を用いたラベル配置アルゴリズムの改良, 神戸大学大学院工学研究科修士論文, 2016.
- [5] G.W. Klau and P. Mutzel, “Optimal labeling of point features in rectangular labeling models,” Mathematical Programming, Ser. B, vol.94, pp.435-458, 2003.
- [6] IBM: IBM ILOG CPLEX, <https://www-03.ibm.com/software/products/ja/ibmilogcplex>.

付 録

ここでは、3 種類のルール R1, R2 及び R5' について説明する。

s を任意のサイトとし、その任意のスリットを sl とする。 sl の安全領域とは、以下の 3 条件を満たす最大の軸平行長方形領域 SR を意味する。

- sl が s の上下スリットであれば、 SR の高さは H_s である。一方、 sl が s の左右スリットであれば、 SR の幅は W_s である。
- SR は s 以外のサイトのスリットと重なっていない。
- SR はサイト s を外周上にもつ。

(ルール R1)

サイト s のスリット sl がラベルサイズ以上の大きさの安全領域をもつとき、 s のラベルをその領域の中央に配



図5 ラベル配置例

置する．図 A1 に実行例を示す．この図において，斜線を引いた長方形が sl の安全領域を示し，2 重線の長方形が配置したラベルを示している．

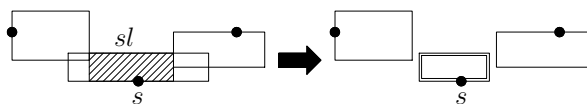


図 A1 ルール R1 の実行例

このとき，(iii) で述べた位置に s, t のラベルを配置する．図 A2 参照．条件を満たさない場合，スリット sl_1, sl_2 は縮小前の状態に戻る．

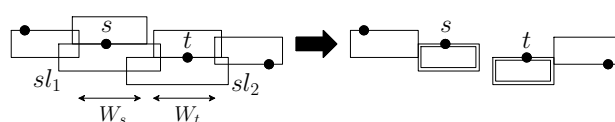


図 A2 ルール R2 の実行例

(ルール R2)

あるサイト s とそのスリット sl_1 に対し，別のサイト t とそのスリット sl_2 が存在して，以下の 3 条件が成立しているものとする．

- (i) sl_1 を， t 以外のサイトのスリットと重ならないように縮小したとき，ラベルサイズ以上の大きさのスリット sl'_1 ができる．
- (ii) sl_2 を， s 以外のサイトのスリットと重ならないように縮小したとき，ラベルサイズ以上の大きさのスリット sl'_2 ができる．
- (iii) sl'_1 内の t から最も遠い位置， sl'_2 内の s から最も遠い位置にそれぞれ仮にラベルを配置したとき，それらが互いに重ならない．

(ルール R5')

サイト s の上下スリット sl が安全領域をもつとき， sl の右辺と安全領域の左辺の距離，及び sl の左辺と安全領域の右辺の距離がそれぞれ W_s となるように， sl の左右両端を縮小する．図 A3 参照． s の左右スリット sl が安全領域をもつときも同様に， sl の上辺と安全領域の下辺の距離，及び sl の下辺と安全領域の上辺の距離がそれぞれ H_s となるように， sl の上下両端を縮小する．

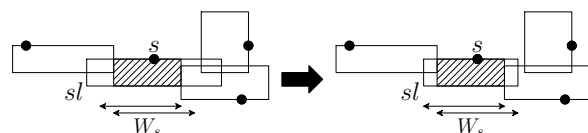


図 A3 ルール R5' の実行例